БИБЛИОТЕКА ПРОГРАММИСТА

Х. Вильямсон

Универсальный Dynamic HTML

- технологии, используемые при создании интерактивных сайтов
- сходство и различие функций Netscape Navigator и Internet Explorer
- элементы языка сценария и HTML, одинаково воспринимаемые обоими браузерами
- параметры HTML и CSS, обеспечивающие общедоступность сайта

и многое другое...

Apress™

ПИТЕР®

СЕРИЯ

БИБЛИОТЕКА ПРОГРАММИСТА

ПИТЕР®

Heather Williamson

Writing Cross-Browser Dynamic HTML

Apress™

Х. Вильямсон

БИБЛИОТЕКА ПРОГРАММИСТА

Универсальный Dynamic HTML

Санкт-Петербург
Москва • Харьков • Минск
2001

ПИТЕР®

Хэйзер Вильямсон

Универсальный Dynamic HTML. Библиотека программиста

Перевел с английского А. Марков

Главный редактор	*Е. Строганова*
Заведующий редакцией	*И. Корнеев*
Руководитель проекта	*А. Васильев*
Научный редактор	*А. Юрин*
Художник	*Н. Биржаков*
Корректоры	*А. Ерашенкова, Н. Лукина*
Верстка	*М. Миклин*

ББК 32.988.02
УДК 681.324

Вильямсон Х.

В46 Универсальный Dynamic HTML. Библиотека программиста. — СПб.: Питер, 2001. — 304 с.: ил.

ISBN 5-318-00368-0

Эта книга адресована всем, кто разрабатывает web-страницы назло препятствиям, заложенным в существующее программное обеспечение. На ее страницах вы найдете всесторонний анализ программной поддержки, необходимой для размещения динамического сайта в Интернете. Все web-разработчики знают, насколько сложно сделать страницу, одинаково выглядящую при четырех разрешениях монитора и во всех существующих браузерах. С помощью автора вы освоите тонкости DHTML, и разработанный вами web-сайт будет радовать вас, его хозяина и посетителей.

ISBN 5-318-00368-0

ЗАО «Питер Бук», 196105, Санкт-Петербург, Благодатная ул., д. 67.
Лицензия ИД №01940 от 05.06.00.
Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2; 95 3000 — книги и брошюры.
Подписано к печати 17.09.01. Формат 70x100/16. Усл. п. л. 24,51. Тираж 5000. Заказ 538

Отпечатано с готовых диапозитивов
в ФГУП ордена Трудового Красного Знамени «Техническая книга»
Министерства Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций
198005, Санкт-Петербург, Измайловский пр., 29

Краткое содержание

Содержание

Всем, кто пытается разрабатывать web-страницы назло существующему программному обеспечению...

Благодарности

Преклоняюсь перед великолепной командой «Apress» за их веру в меня в ходе реализации трудновыполнимой задачи подготовки этой книги к изданию. Без их помощи у меня ничего бы не получилось. Спасибо Гари Корнелу (Gary Cornell), Валери Перри (Valerie Perry), Кирстен Бурк (Kiersten Burke), Энн Наварро (Ann Navarro), Гарри Возняку (Harry Wozniak), Грэйс Вонг (Grace Wong) и тем, кого я никогда не видела, но с чьей помощью удалось достичь столь изумительного результата. Не хотелось бы также забыть переплетчиков, производителей бумаги и всех, кто принимал участие в издании этой книги. Я не знаю ваших имен, но я всем благодарна.

Я бы хотела поблагодарить Энн, которая заставляла меня отвлечься от компьютера для того, чтобы поиграть с ней. Ты постоянно будешь в моей жизни неиссякающим источником шуток и любви, только не расти слишком быстро! Луис, спасибо тебе за то, что ты был так терпелив к тому, кто, по-видимому, получал удовольствие от работы по четырнадцать часов в день. Обещаю, что больше не буду тебя игнорировать, по крайней мере до тех пор, пока не придется писать следующую книгу.

Предисловие

У каждой книги есть свой стиль. Несмотря на то что названия многих разделов похожи на темы классных занятий, которые вызывают у вас ощущение, что вы снова оказались в зале, в котором читаются лекции по информатике, и считаете мух на потолке, я стремилась, чтобы эта книга стала просто дружеским обменом идей. На ее страницах вы найдете всесторонний анализ поддержки, требующейся для размещения динамического сайта в Интернете. Все web-разработчики знают насколько сложно сделать так, чтобы страница выглядела именно так, как надо, при четырех разрешениях экрана и при использовании дюжины различных браузеров. Эта книга поможет вам найти способ избежать непредсказуемости результата при работе с программным обеспечением или обойти ее таким образом, чтобы разработанный вами web-сайт радовал и вас, и его хозяина.

Как менеджер компании Cat's Back Consulting, находящейся в Wallowa Valley (шт. Орегон), я заметила, что клиенты, которые хотят «оживить» сайты, также хотят, чтобы все, включая бабушек, могли в них разобраться, спокойно перемещаться по ним и получать от этого удовольствие. Поскольку лишь около 80 % используемых в настоящее время web-браузеров «умеют» правильно воспроизводить DHTML-сайты, вы должны быть уверены, что вашему клиенту достаточно, что лишь эти 80 % получат удовольствие от его сайта. А также вы должны сделать так, чтобы эти 80 % действительно получили истинное наслаждение, убедившись, что каждый разработанный вами web-сайт можно просматривать любым браузером, начиная с версии 4.0.

Основная цель моей книги: помочь HTML-разработчикам (также называемых web-разработчиками, web-мастерами, web-продюсерами... список можно продолжить) разрабатывать интерактивные страницы и не биться при этом головой об стену (мне приходилось это делать много раз, для того чтобы в конечном результате клиент остался доволен). В этой книге приведены все уловки и тонкости, открытые мной в ходе работы.

Когда вы работаете на заказчика, вы должны сделать его самого и *его клиентов* счастливыми. А для этого требуется полная совместимость с различными браузерами. Ваши заказчики вряд ли останутся довольны, если их сайты смогут просматривать лишь пользователи Internet Explorer. Ваши клиенты вряд ли станут счастливее, если их сайты будут корректно воспроизводиться только Netscape Navigator. Я потратила очень много времени на изучение поддержки HTML, CSS и сценариев обоими браузерами. За счет нахождения «наименьшего общего знаменателя» HTML и CSS, а также разработкой для сценариев вашего сайта API можно создать полноценный интерактивный сайт, который может доставить удовольствие всем пользователям. Таким образом, вы можете гарантировать своим заказчикам, что все посетители, или по крайней мере, пользующиеся новейшими браузерами, смогут испытать удовольствие при работе с web-сайтом их компании.

Структура книги

Я использовала ряд приемов, позволяющих в наибольшей степени облегчить восприятие информации. Как и в большинстве справочников по компьютерной тематике, для выделения важных терминов используется *курсив,* а для представления листингов программ используется соответствующий стиль. Это позволяет легко определять наиболее важные сведения и отличать их от основного текста. Периодически в тексте встречаются примечания и советы, которые представляют собой «информацию для служебного пользования» или разъяснения способов устранения проблем, которые можно «обойти» предлагаемым образом.

При чтении книги вы увидите, что она разделена на три части. В первой рассматриваются технологии, используемые при создании интерактивных сайтов. Особое внимание уделено сходству (и различиям) функций Netscape Navigator и Internet Explorer, связанных с рассматриваемыми вопросами. В ней также содержится предисловие к последующим частям книги. Здесь содержится объяснение, почему в конкретных случаях при написании сценариев используется некоторое «упрощенчество» или обходные методы, а также приводятся примеры того, как можно быстро распознать тин браузера для реализации различных уровней поддержки HTML, CSS и JavaScript/JScript.

Вторая часть в большей степени и является обучающей. Однако вы здесь не увидите пошаговых инструкций типа:

1. Наберите: <script>.
2. Нажмите Enter.

Здесь отсутствует излишне подробное описание, поскольку подразумевается, что вы и так знакомы с JavaScript/JScript. Вместо этого во второй части подробно рассматривается, какие элементы языка сценария и HTML одинаково воспринимаются обоими браузерами и как необходимо использовать эти команды. И хотя вторая часть перегружена примерами, вы найдете множество комментариев, объясняющих использование каждой команды. В конце второй части представлен полный листинг рассмотренного примера. Изобилие комментариев не освобождает вас от необходимости вникнуть в исходный текст и разобраться с работой каждой используемой функции. Представленный в главе 10 исходный код был использован при создании сайта `http://www.webravin.com`, описание которого приведено в книге. Ознакомьтесь с ним и можете спокойно использовать его отдельные части в собственных разработках. Единственное, о чем я попрошу вас, — не забудьте похвалить мой сайт WebRavin.

В третьей части рассматриваются вопросы, относящиеся к web-дизайну. В ней обсуждены вопросы обеспечения общедоступности сайтов (обеспечение возможности просмотра сайта людьми с ограниченной функциональностью), планирование, а также приведены некоторые подсказки и тонкости, которые были неуместны во второй части. В третьей части представлен обзор этапов планирования и подготовки, предшествующих созданию web-сайта, определяющие его тин как динамический или статический. Рассмотрены различные параметры HTML и CSS, обеспечивающие общедоступность сети, благодаря чему ее могут использовать люди с различными ограничениями, а также с различным уровнем знаний. Пос-

ледняя глава третьей части, являющаяся последней в книге, посвящена уловкам и хитростям, которые помогут вам наилучшим образом удовлетворить требования заказчика, а также содержит идеи и основные направления web-разработки.

В книге также имеются четыре приложения, в которых представлены команды, относящиеся к различным технологиям, используемым при создании динамических сайтов. Каждое из приложений содержит данные, представленные в виде таблиц, позволяющие определить наличие поддержки каждого элемента различными браузерами. Эти сведения очень важны при разработке web-сайта. Дело в том, что в большинстве случаев разработка на языке JavaScript функций, необходимых для создания или изменения требуемых объектов, возможна, даже если эти эффекты не поддерживаются обоими основными браузерами.

Чего вы не найдете в этой книге

Вы могли допустить, что я не стану афишировать, чего *нет* в этой книге. Это не так. Как уже говорилось, вы не найдете в этой книге пошагового обучения составлению программ на JavaScript и JScript. Если вам необходимо изучить JavaScript «с нуля», то вы можете посетить любой из сотен сайтов, посвященных этому языку, и найти обучающие курсы для новичков. На страницах этой книги вы не найдете подробных описаний каждой команды HTML или CSS. Если вам необходимы эти сведения, то их можно найти в другой моей книге: *HTML Master Reference* (IDG Books Worldwide, Inc.).

Глядя в будущее

Для того чтобы уследить за постоянным развитием HTML и CSS, заглядывайте на сайт Интернет-консорциума: http://www.w3.org. Здесь вы всегда сможете найти информацию о внесенных предложениях, а также о том, к чему приведет использование этих предложений. Вы можете прочитать разговоры между членами Интернет-консорциума и их собеседниками, если вы хотели бы сделать свое собственное предложение или спросить их, почему выбраны именно данные методы.

Куда обратиться?

Как и во всех книгах, независимо от объема проделанной работы в книге возможно наличие ошибок. А при чтении у читателей могут возникнуть некоторые вопросы. Множество ответов и другой полезной информации можно найти на сайте издательства «Apress»: http://www.apress.com.

Кроме того, сайт-пример, созданный при работе над этой книгой, размещен по адресу http://www.webravin.com/acropolls/. А пример, представленный в главе 10, — по адресу http://www.webravin.com.

Если у вас все-таки остались вопросы или имеются какие-то замечания, обращайтесь непосредственно ко мне по адресу: heather@webravin.com.

Краткая биография автора

Хитэр Вильямсон (Heather Williamson) последние пять лет занимается разработкой и дизайном HTML-документов как для корпоративных, так и для Интернет-сайтов. После слияния корпорации ей пришлось искать другую работу, и она начала с маленькой компании, занимающейся консультированием и разработкой web-дизайна, которая обеспечивает целый ряд услуг по программированию и разработке для компаний северо-западного побережья Тихого океана, Техаса, Нью-Йорка и Аризоны. В число услуг входит разработка web-страниц, реализация проектов электронной коммерции, графическое оформление, разработка компьютерных учебных курсов и разработка различных приложений. В общей сложности она написала 1400 страниц HTML Master Reference (Справочник web-дизайнера) для издательства IDG, сделав его совершенным справочником.

В свободное время Хитэр занимается разведением лошадей, перегоном крупного рогатого скота и превращается из заумного компьютерщика в сорванца, выросшего в деревне.

От издательства

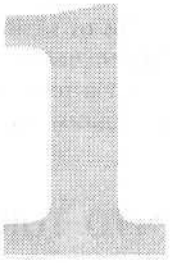
Ваши замечания, предложения, вопросы отправляйте по адресу электронной почты comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

Все исходные тексты, приведенные в книге, вы можете найти по адресу http://www.piter.com/download.

На web-сайте издательства http://www.piter.com вы найдете подробную информацию о наших книгах.

DHTML: Обзор



Пользователи постоянно жалуются на «разрастающееся программное обеспечение» различных продуктов, которое не имеет практического применения. В компаниях зачастую используется программное обеспечение одного направления, например текстовой процессор, и осуществляется его комбинирование с крупными таблицами, базами данных и графическими программами. Хорошо, если все эти программы действительно используются, в противном случае они просто занимают место на диске. Сейчас программисты пытаются сделать то же самое с web-приложениями и web-страницами за счет разработки динамических документов. Они не вызывают перегрузки дисковой подсистемы и не приводят к активному использованию оперативной памяти. Пока это удается. Язык DHTML наряду с базовым набором команд языка HTML основывается на создании сценариев, использовании объектной модели документа и таблиц стилей. Помимо этого имеется возможность использовать на страницах огромное разнообразие аудиофайлов и изображений при помощи оперативных, быстро загружаемых, легко управляемых интерактивных интерфейсов, которые без труда могут создаваться любым web-программистом.

Квалифицированные web-разработчики активно применяют преимущества динамического HTML (Dynamic HTML). DHTML-сайты используют расширенный набор HTML-элементов и атрибутов, а также элементы таблиц стилей. Кроме того, при помощи сценариев можно осуществлять динамическое изменение сайта. Все вместе это позволяет создавать странички, которые «оживают» благодаря перемещающемуся тексту, подвижной графике и наличию интерактивных элементов. В настоящее время глобальная сеть предоставляет пользователям давно ожидаемые ими функциональность и интерактивность без использования подключаемых модулей (plug-ins), добавок (add-ons), серверных программ, не вызывая увеличения времени загрузки и нагрузки процессора.

По мере того как Интернет становится основным источником информации для компаний и частных лиц, возникает потребность в том, чтобы HTML-авторы сознательно обеспечивали достоверность представления документа незави-

симо от используемого пользователем Интернета оборудования и программного обеспечения. И хотя HTML-документы никогда не смогут представить документ в том же виде, в котором он был создан, HTML в настоящее время является одним из самых популярных форматов представления документов.

Что такое DHTML?

Динамический HTML (Dynamic HTML, DHTML), видимо, недооценен разработчиками-новичками. Это не отдельный язык HTML и не расширенный набор команд, собранных из разных языков. DHTML соответствует требованиям HTML 4.0 и 4.1, поддерживает каскадные таблицы стилей (Cascading Style Sheets, CSS), объектную модель документа (Document Object Model, DOM), а также языки сценариев JavaScript, JScript, ECMAScript и VBScript, что позволяет создавать действительно интерактивные web-сайты.

Если вы являетесь web-программистом, то DHTML может кардинальным образом изменить ваш мир. Однако от вас потребуется больше, чем просто знание элементов HTML и их атрибутов. Для изменения внешнего вида и форматирования каждого объекта и элемента документа вы должны уметь использовать каскадные таблицы стилей. Вы должны четко представлять, как работает объектная модель документа и как она связана со структурой документа, а также как при использовании этих знаний можно изменять содержание и внешний вид вашего документа при взаимодействии читателя с вашими страницами. Вы также должны уметь создавать сценарии на языках JavaScript, JScript, VBScript или ECMAScript. Всестороннее знание любого из этих языков позволит вам определять, а затем изменять любой объект, входящий в структуру документа, которая, в свою очередь, определяется объектной моделью документа.

Объектная модель документа

Объектная модель документа (Document Object Model, DOM) является независимой от операционной системы и используемого языка системой взаимодействия, которая предоставляет программам и сценариям динамический доступ и возможность обновления содержания, структуры и стиля документа. DOM определяет последовательную модель, используемую при создании HTML-документов. Кроме того, она обеспечивает стандартный интерфейс, используемый для доступа к HTML-объектам, управлению ими и организации их взаимодействия. DOM отвечает требованиям web-разработчиков, которые при разработке документов основываются на единых универсальных требованиях, а не на требованиях программы определенного производителя. В общем, эта модель расширяет возможности взаимодействия в сети. Не имея четкого представления об объектной модели документа, очень сложно создавать странички, которые бы использовали все достоинства сети и web-браузеров. Более подробно объектная модель документа рассмотрена в главе 2.

Каскадные таблицы стилей

Каскадные таблицы стиляей (Cascading Style Sheets, CSS) являются довольно простой технологией, позволяющей подключать к HTML-документам набор свойств, определяющих шрифты, цвета и интервалы, относящиеся к текстовой информации, размещаемой в HTML-документах. Для реализации всех преимуществ HTML 4.1 и возможностей динамических интерактивных эффектов CSS требуется использование браузеров версии 4.0 и выше. Более подробно каскадные таблицы стилей рассмотрены в главе 3.

Почему необходимо использовать DHTML?

Почему из всех возможных решений, обеспечивающих персонализацию и интерактивное содержание вашего web-сайта, вы должны использовать совокупность HTML, CSS, DOM и сценария? Почему бы не навязать клиенту использование таких вставок, как Flash и Real Player? Почему бы не продолжить использование таких серверных решений, как Active Server Pages (ASP) и Cold Fusion? На вопрос, какое из решений будет наилучшим, не существует готового ответа. Все зависит от конкретной ситуации. Для сравнения в табл. 1.1 приведено несколько «за» и «против» использования различных решений.

Таблица 1.1. Сравнение решений, обеспечивающих содержание вашего сайта

Свойство	DHTML	Вставки	Серверные решения
Клиент независим от операционной системы	Частично	Да	Да
Сервер независим от операционной системы	Да	Да	Нет
Стоимость для клиента	Бесплатно	Бесплатно	Бесплатно
Стоимость разработки программного обеспечения (для заказчика)	Определяется только временем работы	В среднем $300	В среднем $1000
Производительность	В зависимости от мощности оборудования клиента	Высокая	В зависимости от скорости соединения с Интернетом
Доступ к базам данных	Нет	Нет	Есть

На выбор решения частично влияют ваше финансовое положение, операционная система, в которой ведется разработка, операционная система сервера и желаемый уровень быстродействия.

DHTML-решение

DHTML-решения обеспечивают относительно универсальные средства создания интерактивных web-сайтов. Они требуют, чтобы посетитель использовал браузер версии 4.0 и выше, поддерживающий JavaScript или все языки сценариев. Возможно, вам будет интересно, что в середине 2000 года, согласно данным трех статистических организаций сети (web-Snapshot, Internet.com и web-Statistics.com), приблизительно 70 % рынка браузеров принадлежит Internet Explorer корпорации Microsoft, а оставшиеся 30 % — Netscape Navigator. Из них только 1 % пользователей браузера корпорации Microsoft и менее 2 % пользователей браузера компании Netscape используют браузеры версий более старых, чем 4.0. При использовании данного решения основные усилия разработчика должны быть направлены на изучение различных технологий и языков, необходимых для создания динамических сайтов. После овладения этими знаниями они могут быть применены для «доработки» статических сайтов. Недостатком DHTML-сайта является то, что его производительность определяется аппаратными и программными настройками компьютера посетителя. Это в известной мере верно для всех методов создания интерактивности, но наиболее остро снижение производительности может быть связано с использованием DHTML-страниц.

Plug-in-решения

Plug-in-решения (использование подключаемого модуля) позволяют быстро создавать высококачественные динамические страницы. Основной недостаток заключается в том, что посетитель должен *установить* подключаемый модуль на своем компьютере. Большинству людей не нравится дополнительная загрузка и установка, когда им лишь надо просмотреть страничку. Кроме того, данное решение требует наличие у вас, разработчика, программы, с помощью которой создаются файлы, воспроизводимые подключаемым модулем. Эта программа может стоить от $200 до $1000 и, кроме того, для работы с ней требуются соответствующие знания.

Серверные решения

Серверные решения, такие как Active Server Pages (ASP) и Cold Fusion, предназначены для работы на специальных серверах или требуют серверной поддержки. Изначально технология ASP предназначалась для работы с информационным сервером Интернета (Internet Information Server, IIS) корпорации Microsoft, но впоследствии была адаптирована под другие платформы. Теперь она, так же как и Cold Fusion, поддерживается Windows- и Unix-серверами. При использовании большинства серверных решений ваши возможности зависят от используемой вами операционной системы. Эти решения дорогостоящи и требуют наличия у вас довольно больших полномочий в отношении вашего сервера.

DHTML и Microsoft

Компании, занимающиеся производством программного обеспечения, особенно «программные гиганты», стремятся использовать все новейшие технологии, связанные с компьютерной индустрией. Корпорация Microsoft имеет три различные среды web-программирования — JScript, ActiveX (COM) и VBScript. Корпорация занимается обеспечением возможности их взаимодействия помимо поддержки стандартных команд HTML, каскадных таблиц стилей и реализации DOM.

Использование инструментальных средств корпорации Microsoft

В данной книге мы кратко рассмотрим использование программы FrontPage. Пакет разработки FrontPage корпорации Microsoft является мощнейшим средством, полезным не только для разработчика, но и для пользователей и администраторов сервера. FrontPage является программой, позволяющей создавать динамические сайты за счет множества различных встроенных средств и при использовании специальных расширений, устанавливаемых на web-сервер. Проблема заключается в том, что эти расширения доступны не на всех типах web-серверов, а некоторые серверы вообще не способны поддерживать расширения FrontPage. Основная цель создания этого программного гиганта — обеспечить разработчикам, которые хотят создавать страницы, корректно воспроизводимые различными браузерами и всеми серверами, непрерывный конвейер. Если вы являетесь преданным сторонником Microsoft, то вы просто получите удовольствие от работы с программой FrontPage. Должна заметить, что все примеры, приведенные в данной книге, могут быть составлены при помощи FrontPage.

С практической точки зрения FrontPage является основным средством, с помощью которого обеспечивается доступ ко всем возможностям DHTML, реализованным Microsoft, несмотря на то что разработчик может воспользоваться информационным сервером Интернета (IIS). Предлагаемый подход включает множество серверных сценариев, требующих использования расширений FrontPage на сервере, и поддерживает JScript и те элементы JavaScript, которые совпадают с JScript. Я бы назвала код, сгенерированный программой FrontPage, «господин HTML». Должна признать, что текст HTML-документа, создаваемый программой FrontPage 2000, получается более высокого качества, чем у его предшественников. Данный вариант HTML великолепно подходит для корпоративных сетей, построенных на базе продуктов Microsoft или на базе web-сервера NT, если вы знаете, что в ближайшее будущее не собираетесь переходить на другой сервер.

Команды Microsoft

А теперь давайте рассмотрим элементы, которые были созданы корпорацией Microsoft для использования в документах HTML:

BGSOUND LISTING MARQUEE

Список выглядит очень маленьким, но подождите, мы перейдем к рассмотрению параметров, созданных для работы с этими элементами, в главах 2 (DOM) и 4 (JavaScript/JScript).

Только что упомянутые элементы BGSOUND, LISTING и MARQUEE не могут использоваться при просмотре документов продуктами компании Netscape, хотя при помощи элемента EMBED можно реализовать поддержку элемента BGSOUND браузером Netscape Navigator. Использование элемента LISTING можно обойти при помощи элементов LI, OL, UL и MENU, содержащихся в HTML 4.O. Ваш список может выглядеть иначе, чем при использовании стандартных элементов HTML 4.0, но, по крайней мере, он будет доступен всем посетителям вашей странички. Эффект, реализованный с помощью элемента MARQUEE, можно создать за счет последовательного перемещения слоя, поэтому данная команда не является необходимой, хотя для реализации данной задачи потребуется поддержка сценариев. (Список всех элементов приведен в приложении 1.)

Microsoft и DOM

В вопросах поддержки функций и свойств DHTML браузер Internet Explorer 5 значительно обогнал Netscape Navigator 4.x. Да, компания Netscape сейчас разработала новую версию браузера (Netscape Navigator G), но если сравнить Internet Explorer 4 и Navigator 4, то в поддержке DHTML Explorer занимает лидирующую позицию.

В настоящее время Microsoft поддерживает большинство свойств DOM, а также свойства CSS. Язык JScript позволяет осуществить взаимодействие как DOM-объектов, так и свойств CSS. (Internet Explorer поддерживает каждый из этих стандартов — см. приложения 2, 3 и 4.)

Язык JScript будет подробно рассмотрен в главе 4, а сейчас вам достаточно представлять, что JScript является версией языка JavaScript, реализованной корпорацией Microsoft. По сути он похож на JavaScript компании Netscape, но имеет более детальную поддержку DOM, а также дополнительные функции и объекты, не поддерживаемые JavaScript. Каждая из этих функций внедрена для обеспечения взаимной поддержки JScript и VBScript, а также для того, чтобы дать больше преимуществ пользователям Internet Explorer.

И хотя реализация DOM в Internet Explorer 5 осуществлена не полностью, но то, что уже сделано, очень полезно для программистов, создающих сценарии, или «сценаристов», как я их называю. Эта поддержка дает возможность обращаться к любым параметрам любого объекта, используемого в вашем документе, и изменять их. Например, в документах, использующих объектную модель, при помощи сценария можно изменить стиль шрифта, используемый в конкретном заголовке H2. Вы можете осуществить анимацию слоев текста, изменять изображения «на лету» и менять реальность мира сети, если захотите это сделать.

Microsoft и каскадные таблицы стилей

Конечно же, для того чтобы использовать преимущества DOM-структуры, требуется применение каскадных таблиц стилей. Это позволит манипулировать объектами документа. Наилучшую поддержку CSS обеспечивают браузеры Internet Explorer 5.x, которые были выпущены в середине 2000 года, но появление Netscape Navigator 6 заставит Microsoft предпринять дополнительные шаги. При отсутствии поддержки таблиц стилей невозможно изменить стиль шрифта, как это можно было сделать в упоминавшемся выше примере.

DHTML и Netscape

Компания Netscape всегда стремилась «персонализировать» свое программное обеспечение, сделать его несовместимым с программным обеспечением других компаний. По реализации некоторых параметров продукты Netscape хуже, чем аналогичные продукты корпорации Microsoft. (Вспомните, например, элемент BLINK — одну из наиболее неудачных попыток проявить оригинальность. При его активном использовании в документе создается впечатление, что вы сидите перед стробоскопом.) А в некоторых вопросах программные продукты Netscape лучше (например, в них отсутствует избыток событий, которые работают только в пределах своего браузера.)

Использование инструментальных средств компании Netscape

В отличие от Microsoft компания Netscape до сих пор не выпустила программного обеспечения, обеспечивающего всестороннюю работу с DHTML. Composer — это наилучшее средство работы со страницами, позволяющее разработчику немножко приукрасить свои странички, но оно не идет ни в какое сравнение с прекрасно продуманным FrontPage, предлагаемым корпорацией Microsoft. Netscape допускает, чтобы при разработке страниц реализация DHTML осуществлялась программным обеспечением сторонних производителей, хотя эта возможность поддерживается и собственным браузером.

Команды Netscape

Netscape была первой компанией, реализовавшей идею использования перемещаемых и изменяемых слоев на web-сайтах. Это была замечательная идея, но Netscape выбрала неверный путь, создав свой элемент <1 ayer>. Интернет-консорциум (World Wide Web Consortium, W3C) и Microsoft предпочли осуществлять те же самые эффекты перемещения при помощи новых свойств позиционирования, которые используются в таблицах CSS, работающих с уже существующими элементами HTML. Единственным положительным результатом разработки

компанией Netscape тега <layer> стало стимулирование Интернет-консорциума и Microsoft на разработку интерактивного содержания HTML-страниц.

Существует всего два элемента, созданных компанией Netscape и поддерживаемых только их браузером:

BLINK LAYER

Элементы BLINK и LAYER не реализуются браузером Internet Explorer, можете не пытаться! Эффект, создаваемый тегом <blink> можно реализовать при помощи свойства таблицы стилей, назначив значение blink свойству text-decoration. Не имеет значения, как вы будете создавать этот эффект, но поверьте, он очень раздражающий! Вместо тега <layer> можно осуществлять позиционирование объекта , задав его при необходимости.

Netscape и DOM

Осенью 2000 года компания Netscape выпустила браузер Navigator 6. Он полностью поддерживает предложенную Интернет-консорциумом реализацию объектной модели документов, без всяких новых дополнений от компании Netscape. Новая версия Netscape Navigator все еще поддерживает такие собственные старые элементы, как <layer>, но в дальнейшем планируется отказаться от поддержки этих элементов после, того как текущая разработка будет широко распространена. Netscape наконец-то делает достойный шаг к полной поддержке стандарта и обеспечению совместимости. Это будет большим событием для разработчиков, ориентирующихся на различные браузеры, поскольку им теперь не придется сдерживать творческую фантазию и идеи, приводя их к «наименьшему общему знаменателю» различных реализаций объектной модели документа.

Netscape и CSS

Существующий браузер осуществляет расширенную поддержку CSS, охватывающую поддержку большинства свойств CSS 2 (Cascading Style Sheets Level 2). Поскольку новый браузер пока еще официально не выпущен, о чем мы уже упоминали, сложно сказать, что действительно будет поддерживаться. Однако планировалась полная поддержка требований Cascading Style Sheet 2.0, представленных Интернет-консорциумом.

В чем схожи оба браузера?

Этот вопрос, возможно, звучит вызывающе, как будто между двумя самыми популярными браузерами существует слишком много различий в поддержке DHTML. Нет, сходства значительно больше, чем различий. Это видно из табл. 1.2, в которой представлены функциональные возможности обоих браузеров.

Таблица 1.2. Характеристики, поддерживаемые Netscape Navigator и Internet Explorer

Свойство	Наличие в Navigator	Наличие в Internet Explorer
GIF-анимация	Да	Да
<BGSOUND>	Нет	Да
<DIV>	Да	Да
<EMBED>	Да	Да
Фреймы	Да	Да
Плавающие фреймы	Нет	Да
Цвет шрифта	Да	Да
Формы	Да	Да
Java	Да	Да
JavaScript	Да	Да
<LAYER>	Да	Нет
<MARQUEE>	Нет	Да
<OBJECT>	Нет	Да
<SPAN>	Нет	Да
Основные свойства CCS 1	Да	Да
Основные свойства CCS 2	Да	Да
Свойства позиционирования CSS	Да (хотя полная поддержка всех пар «свойство-значение» все еще разрабатывается)	Да
Поддерживаемые языки сценариев	JavaScript ECMAScript	JScript VBScript ECMAScript
Таблицы	Да	Да
Цвет фона ячейки таблицы	Да	Да
Фоновое изображение в ячейке таблицы	Нет	Да
Встроенные таблицы	Нет	Нет
Автоматический перенос текста на следующую строку	Да	Да

Вы можете отметить, что иногда Netscape Navigator и Internet Explorer имеют явные противоречия при воспроизведении одного и того же документа DHTML. Однако по большинству вопросов они имеют одинаковый подход. Оба браузера имеют ряд собственных, несовместимых с другими элементов, и оба браузера реализуют большинство (но не все) характеристик стандарта HTML 4.0. Понятно, что различия добавляют головной боли разработчикам, однако если вы сосредо-

точитесь на вопросе, что у них общего, то вы просто сможете определить области, которых пока необходимо избегать.

Несколько лет назад, когда использовались браузеры версий 2.0 и 3.0, основными темами обсуждения были таблицы, фреймы и формы. Именно по поддержке этих элементов пользователь и делал выбор подходящего ему браузера. В настоящее время все изменилось, теперь приходится думать об изменении свойств CSS и DOM с помощью сценария. В Интернете проблемы для разработчиков никогда не исчезают, а просто переходят на более высокий технический уровень.

Обзор объектной модели документа

2

Интернет-консорциум (World Wide Web Consortium), известный также под сокращением W3C, издал документ, описывающий объектную модель документа (Document Object Model, DOM). В практических целях ее понимание требуется всякому, кто хочет разобраться в сложности блочной структуры HTML-документов, особенно при переходе к XHTML.

Вам необходимо уяснить, что объектная модель документа — это не программа или пакет. Это описание всех частей документа, к которым может обращаться разработчик. W3C рекомендует, чтобы DOM воспринималась как стандарт для определения согласованности объектных моделей web-страниц и работы разработчиков, использующих языки сценариев, независимо от используемого программного обеспечения.

Что же такое DOM?

DOM характеризует содержание web-документа как набора объектов. Обращаться к этим объектам и манипулировать ими можно различными способами, в том числе с помощью сценариев, исполняемых как на сервере, так и на стороне клиента. Под *объектом* понимается стандартный блок, множество которых и образует документ. Например, в масштабе этой книги данный абзац — это объект. Абзац является индивидуальным элементом, обладающим определенными свойствами (размером, типом шрифта и т. д.). Слова составляют содержание абзаца. Тип шрифта является свойством слов. Например, для того чтобы текст абзаца изменялся в зависимости от положения указателя мыши пользователя, необходимо иметь возможность выбрать и изменить этот абзац, не затронув другие части web-страницы.

Все объекты в web-документе взаимосвязаны. Для описания их взаимоотношений используется понятие *дерево* (*tree*), поскольку, подобно реальному дереву, структура данных «разветвляется» от корня. В HTML-документе корнем

документа является элемент <HTML>. Элементы <HEAD> и <BODY> происходят от корня и продолжают разветвляться далее. Например, элемент <HEAD> может иметь ветки, представленные элементами <TITLE> и <META>, а элемент <BODY> — другими действительными тегами HTML, такими как <P> и <FONT>. Документ обычно представляется перевернутым вверх корнями деревом, отдельные ветки которого «растут» вниз (см. рис. 2.1).

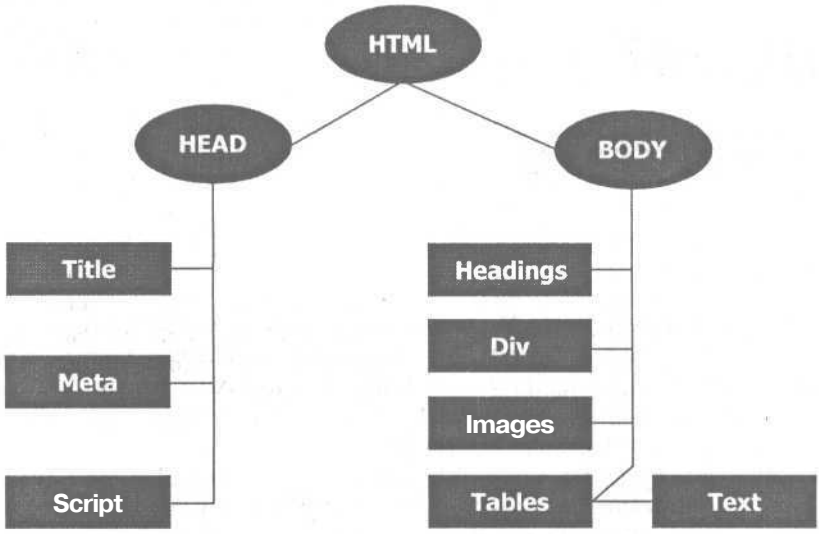


Рис. 2.1. Дерево HTML-документа расширяется с добавлением каждого нового элемента

Хотя существование дерева документа всегда предполагалось языком HTML, до появления объектной модели документа понятие «дерево» было не особенно полезным. Недостатком этого понятия является то, что многие разработчики неправильно понимают те преимущества, которые оно дает. С помощью языка сценариев значительно легче управлять документом, представленным в виде дерева, чем простым, «плоским» документом, как документ MS Word. Когда структура представлена деревом, то можно четко выделить такие его составляющие, как корень, ствол, ветви и листья. На языке программирования каждый из этих компонентов называется *узлом (node)*. Любое дерево имеет лишь один корень и множество ветвей, также и любой HTML-документ имеет один корень и множество узлов. За исключением корня, все узлы имеют «родителя». Родителем листа является ветка, а в HTML-документе родителем элемента <BODY> является элемент <HTML>.

Дерево может иметь множество ветвей, а ветка — множество листьев. Аналогично и каждый узел может содержать множество элементов одного уровня. Например, в HTML-документе одноранговым элементом для <BODY> является элемент <HEAD>. Любой узел может иметь дочерние элементы (наследников). Дочерним элементом ветви является лист, а в HTML-документе для элемента <HTML> дочерними являются элементы <BODY> и <HEAD>, а также их наследники. Если вы составите полное дерево различных документов, то вы, заметите, что

размер схемы по ширине получится значительно больше размера по высоте. Если же вы составите дерево HTML-документа, в котором используются слои, то дерево такого документа будет «выше» дерева обычного web-документа.

Предоставляемое объектной моделью описание типов объектов HTML-документа и их взаимосвязей позволят разработчикам обращаться к ним стандартным образом. Рекомендации W3C являются платформонезависимыми, что позволяет руководствоваться ими как в Windows, так и в Macintosh, Linux или Unix. А поскольку DOM также не зависит от языка, то web-браузер может быть разработан на языке Си, Си++, Паскаль или любом другом языке, но с сохранением интерфейса прикладного программирования (API-интерфейса).

Наша задача — создавать web-страницы и web-приложения, которые бы корректно представлялись как браузером корпорации Microsoft, так и браузером компании Netscape. Для повышения возможности взаимодействия в сети, все производители (не только Microsoft и Netscape) должны придерживаться единой стратегии, заключающейся в поддержке объектной модели документа, а не разработкой собственных идей на тему: что из себя представляет web-документ и как им лучше манипулировать. Как и всегда, корпорация Microsoft в web-браузер Internet Explorer 5 включила свои собственные свойства сценариев и описания объектов. То же самое обычно делает и Netscape, но после выпуска Netscape Navigator 6 этому разногласию будет положен конец.

Структура документа с точки зрения объектной модели

Мы уже определили, что объектная модель является API-системой, независимой от языка и платформы, позволяющей программам и сценариям осуществлять динамический доступ и изменение содержания, структуры и стиля web-документа. Стандарт объектной модели (который можно найти по адресу: http://www.w3c.org/TR/PR-DOM-Level-1/) представляет разработчикам последовательную модель, используемую для определения и манипулирования объектами в XML- и HTML-документах.

ПРИМЕЧАНИЕ

В настоящее время W3C ведет разработку объектной модели документа второго уровня (Document Object Model Level 2, DOM2). Дополнительную информацию об этой версии можно узнать на страничке http://www.w3c.org/POM/.

Объектная модель предоставляет разработчику стандартные средства обращения к объектам web-страниц и их связям. Эта модель предлагает производителю использовать готовые интерфейсы программирования вместо разработки собственных. Она также предлагает авторам содержания (web-программистам) создавать документы, лишь придерживаясь объектной модели, вместо использования специализированных программ производителя. В целом данная модель просто способствует расширению возможностей взаимодействия в сети. Использование принципов DOM позволяет разработчикам максимально использовать все преимущества, предоставляемые сетью и существующими web-браузерами.

ПРИМЕЧАНИЕ

В настоящее время DOM не применяется в отношении стандартного обобщенного языка разметки (Standard Generalized Markup Language, SGML). SGML является стандартным языком, используемым в качестве базиса или для обучения другим языкам разметки, таким как HTML или XML. Самую подробную информацию вы можете получить, изучив стандарт ISO 8879:1986, который можно найти на http://www.iso.ch/cate/dl6387.html или на сайтах, посвященных языку SGML, например http://www.arbortext.com/Think_Tank/SGML_Resources/sgml_resources.html или http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/WebSGML.html

Согласно DOM-стандарту каждый документ представляется в виде дерева, состоящего из модулей, которые определяют каждый объект документа и обеспечивают возможность настройки любого из объектов отдельно от остального документа. Пример подобного документа представлен ниже, а его структура — на рис. 2.2.

```
<html>
  <head>
    <title> This is a small document </title>
  </head>
  <body>
    <h1> Heading 1 </h1>
    <p> Add some introductory paragraph material here. </p>
    <h2> Heading 2 </h2>
    <p> <font face="arial">This is the body of heading 2, with the font
changed.</font></p>
  </body>
</html>
```

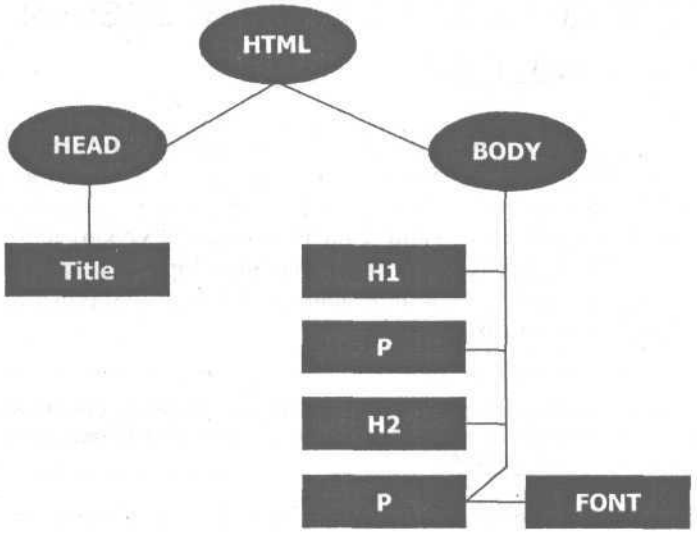


Рис. 2.2. Узлы документа, поддерживающего DOM, к которым можно обращаться раздельно

Как видно из рис. 2.2, DOM представляет логическую структуру в виде перевернутого дерева. Более сложные документы можно представить в виде леса, который имеет множество корневых элементов, хотя отдельный документ может иметь только один корневой элемент.

Помните, что объектная модель не определяет, как должен быть реализован документ, она лишь предлагает логическую модель, подходящую для интерпретации структуры документа. Одним из наиболее важных свойств объектной модели является *структурная стабильность (structural isomorphism)*, то есть DOM не зависит от конкретной программной реализации. Она реализуется одинаково, вне зависимости от того, как было создано данное программное обеспечение. Вследствие этого не имеет никакого значения, какая реализация DOM будет использоваться при воссоздании документа на компьютере посетителя, — любая из них использует ту же структурную модель с теми же объектами и их взаимосвязями.

Если изучать DOM с точки зрения традиционного объектно-ориентированного программирования, то она действительно является объектной моделью, поскольку документы в ней состоят из объектов, а сама модель отражает структуру документа, сам документ и поведение объектов документа. Подобная модель отличается от традиционного абстрактного представления данных, используемого в SGML-документах. *Абстрактная модель данных (abstract data model)* сосредоточена на данных, в то время как объектная модель инкапсулирует данные, защищая их от изменения. Если рассматривать DOM с точки зрения объектной модели, то она определяет:

- объекты, с помощью которых представляется и видоизменяется документ,
- интерфейсы, используемые для представления и видоизменения документа,
- поведение и атрибуты интерфейсов и объектов, используемых в документе,
- взаимоотношения между всеми интерфейсами и объектами.

DOM можно подразделить на две части: ядро и собственно HTML. Спецификация ядра определяет фундаментальные объекты, используемые для определения любого структурированного документа. Оно включает структуры, необходимые для определения и работы XML-документов. Документы различных типов не требуют использования некоторых конструкций XML, определенных спецификацией ядра, хотя они должны реализовывать общие структуры документов, определяемых в основе ядра. Например, если вы создали программное обеспечение, которое не должно поддерживать XML, оно должно лишь поддерживать основные структуры ядра, имеющиеся в вашем приложении и нет необходимости реализовать структуры XML.

Раздел HTML определяет все объекты и структуры, характерные для данного документа. Если вы создаете программное обеспечение, которое должно поддерживать HTML, то ваше приложение должно поддерживать как основу ядра, так и HTML-часть объектной модели документа. Если планируется, что ваше приложение также должно поддерживать и XML, то вам необходимо обеспечить поддержку всей спецификации ядра, включая XML-конструкции, а также спецификацию HTML.

Реализация поддержки DOM теоретически выглядит довольно просто, но при практической реализации существуют некоторые сложности, с которыми вы познакомитесь во второй части данной книги.

Как потребитель и/пли разработчик вы зависите от других разработчиков, которые осуществляют реализацию поддержки тех функций, которые бы вы хотели использовать. Зачастую различные компании предоставляют поддержку одних функций, но совершенно не поддерживают другие. В этом нет ничего удивительного: даже продукты Microsoft и Netscape не полностью удовлетворяют всем требованиям DOM-стандарта.

Реализация DOM корпорацией Microsoft

В Internet Explorer 5 корпорация Microsoft обеспечила довольно полную реализацию объектной модели документа. Она значительно улучшена по сравнению с Internet Explorer 4, хотя он и обеспечивал поддержку большинства объектов документа. Microsoft обеспечила доступ к объектам двумя способами. Тот способ, которому посвящена данная книга, заключается в использовании языков сценариев. Кроме того, к объектам можно обращаться при помощи набора интерфейсов модели многокомпонентных объектов (Component Object Model, COM).

ПРИМЕЧАНИЕ

COM является моделью, которая идентифицирует последовательности объектов, используемых программистами, для создания таких новых технологий, как Active X и технология «связывания и внедрения объектов» (object linking and embedding, OLE). Microsoft разработала данную модель как средство, допускающее распространение программных решений, которые могли бы быть легко развернуты и реализованы в системах Windows.

Поддержка DOM корпорацией Microsoft выполнена практически в соответствии со всеми рекомендациями Интернет-консорциума. Осуществлена поддержка большинства отдельных идентификаторов, связанных с объектами HTML, используемых в web-документах на основе HTML 4.x. Задача DOM — представить каждый HTML-элемент как объект, а каждый атрибут HTML или CSS — как свойство. Microsoft вплотную приблизилась к такому подходу. И если еще в Internet Explorer 4.5 большинство свойств не поддерживалось, то в Internet Explorer 5 поддерживаются почти все свойства.

При использовании объектной модели разработчики, использующие продукты корпорации Microsoft, могут обращаться ко всем объектам на странице или оперативно изменять документ, в зависимости от введенных пользователем данных, его поведения или настроек его оборудования. Одним из наиболее полезных свойств DHTML и DOM является поддержка в Internet Explorer модели всех событий. К сожалению, ее поддерживают далеко не все web-браузеры.

Модель событий Microsoft

Основным недостатком предшествующих объектных моделей Microsoft была несовершенная модель событий. Только незначительная часть событий была доступна для ограниченного набора тегов. А в IE 5 все элементы связаны с полным набором специальных событий, событий мыши, клавиатуры и выделения. IE 5 поддерживает практически весь набор событий, включая даже те события, которые не поддерживаются Netscape Navigator 4.5:

onafterupdate	ondragstart	onrowen'ter
onbeforeunload	onerrorupdate	onrowexit
onbeforeupdate	onfilterchange	onscroll
onbounce	onfinish	onselect
onchange	onhelp	onselectstart
ondataavailable	onreadystatechange	onstart
ondatachanged	onreset	onsubmit
ondatasetcomplete		

Изменение текста в реальном масштабе времени

Следующей областью DOM, в которой Internet Explorer превосходит предыдущие версии продуктов Netscape, включая Navigator 4.7, является реализация замены текста и HTML-кода. Применение Internet Explorer позволяет использовать свойства innerHTML, outerHTML, innerText и outerText. Эти свойства позволяют обращаться к основному HTML-коду и текстовому содержанию объектов документа. Свойства innerHTML и innerText работают только с объектами-контейнерами, такими как DIV, SPAN и заголовки. Свойства outerHTML и outerText могут применяться ко всем HTML-тегам в теле документа, поэтому они могут использоваться для замены всего элемента и его содержимого, а не только его текстового содержимого.

Свойства innerText и outerText возвращают строку HTML-кода лишь как текстовую информацию, найденную в той области, к которой осуществляется обращение, в то время как свойства innerHTML и outerHTML считывают всю HTML-информацию, включая настройку атрибутов. При использовании свойства HTML Internet Explorer обращается к строке, представляемой как HTML-код, и соответствующим образом ее обрабатывает, обновляя формат текста на основе представленных HTML-элементов. Если осуществляется манипулирование строкой при помощи свойства Text, то она будет буквально вставлена в документ, без обработки HTML-команд, поэтому в вашем документе может появиться строка типа «My Father <I> Fred </I> is a cool guy» вместо строки «My Father *Fred* is a cool guy».

Реализация DOM компанией Netscape

Хотя Netscape поддерживает DOM, но ее реализация, как и ожидалось, отличается от реализации DOM корпорации Microsoft. Netscape занимается реализацией DOM несколько медленнее, чем ее главный соперник. Я не собираюсь определять, что является причиной этого, но с выпуском Netscape Navigator 4.7, а особенно давно ожидаемого Netscape Navigator 6, наконец появилась необходимая поддержка.

Netscape Navigator 3 был первым браузером, позволяющим изменять значения свойств объектов «на лету». Первым таким объектом был атрибут scr тега img. Конечно же, этого эффекта можно было достичь, поместив объект в элемент a и использовать свойство onmouseover, которое доступно для этого элемента.

Рост компании Netscape

Компании Netscape не удалось полностью реализовать свойства каскадных таблиц стиля и свойства HTML 4, ограничивающие возможности реализации DOM, которые не претерпели значительных изменений с момента выхода Navigator 2. Тем не менее Navigator 4 поддерживает позиционирование CSS, общие свойства CSS и язык сценариев JavaScript. Однако с помощью JavaScript определяются не все объекты, и реализуется этот процесс несколько иначе, чем определение таких же или подобных объектов в Internet Explorer.

Поддерживаемые объекты

В соответствии с синтаксисом, принятым в Netscape Navigator 4.x, необходимо, чтобы объект (tag) был размещен перед идентификатором элемента. Данный объект позволяет при помощи JavaScript обращаться к HTML-элементам определенного типа. Ниже приведены объекты и свойства, поддерживаемые только Netscape Navigator 4.x:

above	home	previous
anchor	innerHeight	print
anchors	innerWidth	releaseEvents
applet	layer	reset
area	locationbar	right
below	menubar	routeEvents
bottom	next	scrollbar
captureEvents	outerHeight	siblingAbove
classes	outerWidth	siblingBelow
current	pageX	statusbar
disableExternalCapture	pageXoffset	submit
elements	pageY	tags
enableExternalCapture	pageYoffset	textarea
eval	parentLayer	toolbar
fileUpload	password	toString
find	personalBar	whitespace
handleEvent	pixelDepth	valueOf

В чем схожи оба браузера?

Продукты как Microsoft, так и Netscape поддерживают языки сценария, которые обеспечивают возможность взаимодействия с объектами и элементами HTML-документа. Два языка очень похожи и имеют общее основание, но при реализации некоторых элементов они используют различные структуры. (В главе 7 рассмотрен простой способ обойти эти разногласия.)

Совместимая модель событий

Областью, в которой можно найти почти полное соответствие между двумя браузерами, является поддержка модели событий. Они поддерживают следующие события:

onabort	onkeydown	onmouseout
onblur	onkeypress	onmouseover
onclick	onkeyup	onmouseup
ondblclick	onload	onmove
ondragdrop	onmousedown	onresize
onerror	onmousemove	onunload
onfocus		

Эти события позволяют манипулировать элементами при стандартном позиционировании и перемещении элементов при помощи массивов и ключевых кадров, а также при общей корректировке документа или объектов. Помимо указанных событий существует множество схожих, но не совпадающих событий. Конечно же, вы сами можете создать свой пользовательский интерфейс или объектную модель, которая будет использовать сценарии сервера, что позволит поддерживать множество платформ, но привязка к сценариям сервера не всегда будет наилучшим решением для многих web-разработчиков, создающих динамические web-сайты. Использование сценариев сервера, скорее всего, потребует от вас основательного изучения нового языка, на изучение которого у вас просто может не хватить времени. Вы также можете остановиться на разработке web-сайтов, основанной на использовании программного обеспечения определенного типа web-сервера, что приведет к усложнению переноса сайта или дальнейшего обновления сервера.

Совместимые объекты и свойства

Ниже представлены объекты и свойства, поддерживаемые обоими браузерами, несмотря на различия в синтаксисе.

A
ACRONYM
ADDRESS
APPLET
applets
anchors
areas
B
BASE
BASEFONT
BIG
BLOCKQUOTE
BODY
BR
BUTTON
CAPTION
CENTER
CITE
CODE
COL
COLGROUP
DD
DEL
DFN
DIR
DIV
DL
document
DT
EM
EMBED
embeds
event
FIELDSET
FONT
FORM
forms
FRAME
frames
FRAMESET
H1
H2
H3
H4
H5
H6
HEAD
history
HR
HTML
I
IFRAME
images
IMG
INPUT
Input Types:
button
checkbox
file
hidden
image
password
radio
reset
submit
text
INS
KBD
LABEL
LAYER
LI
LINK
links
location
MAP
MENU
META
navigator
NEXTID
OBJECT
OL
OPTION
P
PLAINTEXT
plugins
PRE
Q

S	style	TITLE
SAMP	SUB	TR
screen	SUP	TT
SCRIPT	TABLE	U
SELECT	TBODY	UL
SMALL	TD	VAR
SPAN	TEXTAREA	window
STRIKE	TFOOT	XMP
STRONG	TH	
STYLE	THEAD	

Используя эти объекты и немного поколдовав над сценарием, вы можете создавать интерактивные сайты, которые будут работать как с Internet Explorer 4 и Netscape Navigator 4.x, так и с их новыми «собратьями» — Internet, Explorer 5.x и Netscape Navigator 6.x.

Реализация каскадных таблиц стилей

HTML-программистам, желающим максимально использовать все интерактивные эффекты и возможности HTML 4.0, которые могут быть воспроизведены самыми последними браузерами, просто не обойтись без использования каскадных таблиц стилей. *Таблицы стилей* — это списки свойств, определяющие, как будут выглядеть объекты при просмотре web-страниц. Например, таблица стиля позволяет создавать текст или изображения с помощью определенного тега HTML или указав определенное имя, выполняющее те же действия для всего документа. При использовании таблицы стиля можно управлять внешним видом таблиц, заголовков, маркеров списков и стилем нумерованных списков, а также определять отступы страниц, лишь указывая наименование стиля.

Предложенная Интернет-консорциумом спецификация Cascading Style Sheets Level 1 (CSS 1) является простым механизмом, позволяющим авторам HTML-документов применять к тексту определенные шрифты, цвета и интервалы, связанные с определенным указателем стиля (style guide). Сначала эта спецификация была ограничена лишь управлением внешнего вида текста и некоторых общих свойств объектов, поддерживаемых HTML. В 1998 году, сразу после выхода браузеров четвертой версии, Интернет-консорциум завершил работу над спецификацией CCS 2. Среди прочих нововведений в данную спецификацию было включено управление печатью документа и звуковым сопровождением документов. CSS 2 включает также множество дополнительных свойств для работы с контейнерами (containment boxes), блоками (blocks), таблицами (tables), новыми форматами дисплея (new display formats) и контурами (outlines).

По мере того как web-проекты становятся жизненно важными источниками информации как для предприятий, так и для отдельных людей, HTML-авторы все в большей степени заинтересованы в наличии возможности управления взаимодействием своих документов с оборудованием абонента. Чем точнее HTML-документ будет соответствовать требованиям настольного издания, тем большее распространение получит язык HTML. HTML-документ сможет из документа локальной сети или Интернета превратиться в тип документа, отвечающий последним требованиям глобального распространения документов.

«Лучшая десятка» самых полезных возможностей, которые привнесли CSS в web-дизайн, выглядит следующим образом:

1. Соответствие шрифтов (подробности можно узнать на сайте корпорации Microsoft: http://msdn.microsoft.com/workshop/author/css/feature_2.htm).
2. Управление цветами и фоном (будет рассмотрено в главе 8).
3. Добавление заранее установленных элементов оформления текста.
4. Управление полями страниц (page margins), заполнением объекта (object padding) и контурами (outlines).
5. Выделение текста отступами.
6. Управление межстрочными и межбуквенными интервалами.
7. Создание больших заглавных букв (буквиц).
8. «Вырезание» объектов (см. главу 8).
9. Позиционирование и упорядочивание объектов (см. главу 6).
10. Добавление фильтров.

Еще одной важной возможностью, которой обладают CSS, является автоматическое создание «на лету» эффектов при помощи псевдоэлементов :before и :after, а также свойства counter. Псевдоэлементы задают способ просмотра структуры документа собственно в самом языке HTML. Например, этот язык не обеспечивает возможности обращения к первой букве или первой строке содержания элемента. Псевдоэлементы позволяют разработчикам обращаться к данным элементам, недоступным никаким иным способом. Псевдоэлементы также обеспечивают возможность создания содержания web-страниц «на лету». Проблема заключается лишь в том, что эффекты, создаваемые :before и :after, не реализуются браузерами четвертой и более «старых» версий. Internet Explorer 5 в некоторой степени поддерживает свойство counter, что является лишь минимальным требованием поддержки эффектов :before и :after. Netscape заявила, что браузер Navigator 6 обеспечивает полную поддержку этих эффектов.

Если вы считаете, что каскадные таблицы стилей не решают всех проблем дизайна и форматирования, с которыми приходится сталкиваться разработчикам HTML, то вы правы. Тем не менее CSS в тысячу раз лучше того, чем приходилось оперировать разработчикам, когда им предлагался лишь простой язык HTML плюс несколько объектов и свойств языка сценария.

ВНИМАНИЕ

Ни один из браузеров третьей версии (1995) не будет правильно считывать или импортировать файл CSS, а также не будет воспринимать элемент STYLE. Эти возможности появились только в браузерах четвертой версии (1998).

Реализация CSS корпорацией Microsoft

Корпорация Microsoft приступила к реализации CSS на этапе разработки Internet Explorer 4.0. Получилась довольно удачная реализация, и сейчас ведется работа по ее дальнейшему совершенствованию.

Реализация CSS корпорацией Microsoft отличается по некоторым моментам от реализации CSS в Netscape Navigator 4.0. Полный список команд, поддерживаемых одним, но не поддерживаемых другим браузером, вы можете найти в приложении 2 «Совместимость каскадных таблиц стилей». Наиболее очевидные отличия касаются реализации соответствия шрифтов OpenType, которое позволяет посетителям увидеть в Интернете документ с теми шрифтами, которые выбрал автор. Помимо этой технологии в браузере Internet Explorer реализованы и другие рекомендации CSS:

- дополнительные элементы оформления текста и прочие эффекты;
- отступы, контуры и другие свойства контейнеров;
- слои и позиционирование;
- фильтры.

Соответствие шрифтов OpenType

Microsoft использует метод *OpenFont* — это средство, обеспечивающее взаимодействие шрифтов, находящихся на компьютере посетителя, со шрифтом, установленным автором в HTML-документе (соответствие указанного автором шрифта максимально схожему шрифту из набора имеющихся на компьютере пользователя иногда называют *внедрением шрифта OpenType (OpenType Font Embedding)*). Компания Netscape использует систему соответствия шрифтов под названием TrueDoc (будет рассмотрена далее в данной главе) и не поддерживает механизм OpenType, используемый в продуктах Microsoft.

ПРИМЕЧАНИЕ

OpenType — то же самое, что и Microsoft TrueType Open второй версии, — является расширением исходного формата шрифтов TrueType. Шрифтовая система OpenType — это действующий расширенный набор существующих форматов TrueType и Type 1. Он был разработан в ходе совершенствования поддержки как экранных, так и неэкранных шрифтов. Технология сжатия, применяемая в формате OpenType, позволяет организовать очень эффективную схему шрифтов, когда необходимый для соответствия шрифт может быть загружен из Интернета.

Использование шрифтов OpenType в Интернете позволяет разработчикам создавать более полноценные документы без увеличения размеров файла, что обеспечивает их быструю загрузку и воспроизведение. Формат шрифтов OpenType разработан для удовлетворения следующих требований:

- поддержка в различных ОС;
- поддержка национальных шрифтов;
- защита данных шрифта;
- сокращенный размер файла, позволяющий более эффективно осуществлять распространение шрифта;
- возможность дополнительного управления типографской разметкой шрифта.

Для создания требуемых шрифтов или преобразования существующих шрифтов в формат OpenType разработчикам, конечно же, придется обратиться к средствам Microsoft или Adobe. Однако любые шрифты TrueType, имеющиеся у разработчика, могут использоваться без всякого преобразования. Все шрифты

OpenType могут иметь расширение файла либо OTF (OpenType Font) либо TTF (TrueType Font).

ПРИМЕЧАНИЕ

OTF-файлы не «читаются» такими 16-разрядными версиями Windows, как Windows 3.1 или Windows для рабочих групп. Они работают только в 32-разрядных операционных системах типа Windows 95/98 или Windows NT/2000.

Дополнительные элементы оформления текста и другие эффекты

В соответствии со спецификацией CSS 2 возможно использование заранее определенных элементов оформления текста. К этим элементам относятся мигание (blinking), подчеркивание (underlines), зачеркивание (strikethroughs) и надчеркивание[1] (overlines) (см. рис. 3.1).

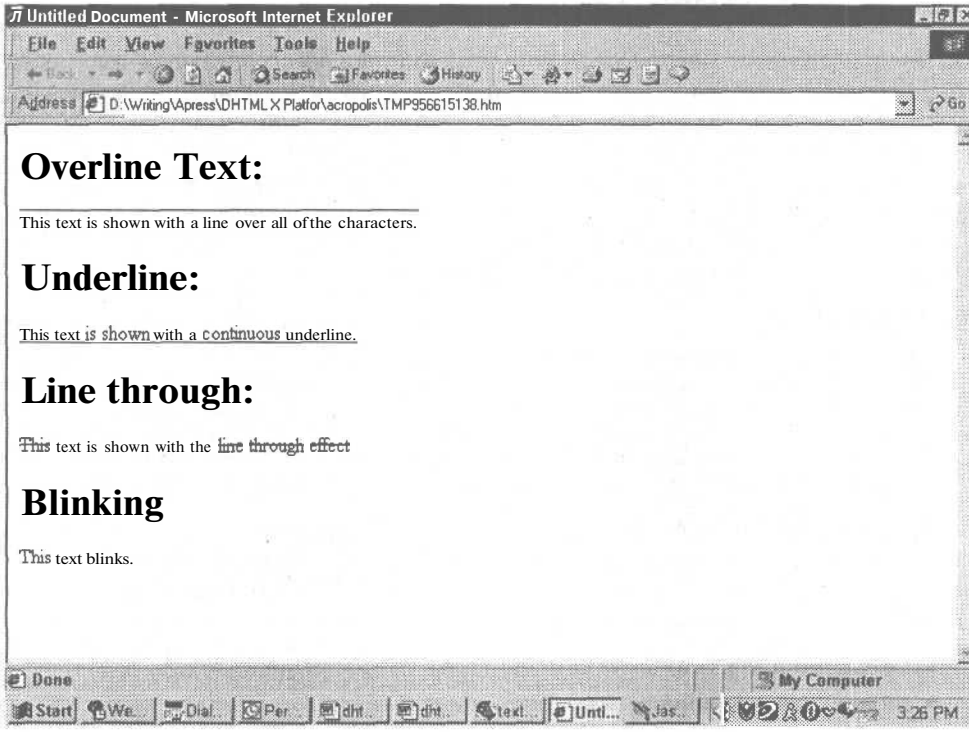


Рис. 3.1. Каждый из этих эффектов в CSS 2 определяется свойством text-decoration

За счет изменения цвета фона, межстрочного интервала, межсимвольного интервала, высоты символов и использования тени возможно создание и других текстовых эффектов. Теперь все эти эффекты можно создать без использования графики. Используя псевдоэлементы: `first-line` и `:first`, можно создавать такие

[1] Сопровождение чертой над строкой.

эффекты, как оформление только первых строк или буквицы (большая первая буква в начале главы или статьи). Свойство font-stretch позволяет растягивать или сжимать символы в строке текста, изменяя межсимвольный интервал, а свойство font-variant используется для отображения всех символов капителью.

Кроме того, Internet Explorer осуществляет поддержку таких свойств, как text-shadow, text-transform, vertical-align и word-spacing. Каждое из этих свойств поможет сделать ваши web-странички красивыми и уникальными, но увидеть все эти эффекты можно только с помощью Internet Explorer 5. И даже в этом случае возможны ошибки в реализации, которые могут привести к тому, что эффект просто не будет представлен.

При использовании этих свойств текста вы можете полностью изменить внешний вид документа, лишь внеся изменения в файл *.css. Эта возможность позволяет разработчику сэкономить время, необходимое для изменения множества документов.

Отступы, контуры и другие свойства блоков

Internet Explorer поддерживает большинство свойств блоков (контейнеров), добавленных в CSS 2. И хотя большинство из них не поддерживаются браузером Netscape Navigator 4, реализация этих свойств добавлена в Netscape Navigator 6. Internet Explorer позволяет использовать сочетания или сокращения свойств (таких как border, margin, outline и padding) в качестве отдельных команд. Например, Internet Explorer поддерживает команды, которые устанавливают верхнюю (top), нижнюю (bottom), левую (left) или правую (right) ширину рамки (border), отступа (margin), заполнения (padding) или контура (outline), с помощью которых можно установить индивидуальную ширину для каждой рамки или края, одновременно указав цвет и стиль.

В табл. 3.1 представлены все свойства блоков. Различия в поддержке всех свойств смотрите в приложении 2 «Совместимость каскадных таблиц стилей». (Более подробные сведения об этих командах представлены в главе 6 «Подготовка к войне браузеров».)

Таблица 3.1. Свойства блоков

CSS-свойство	Синтаксис	Описание
border	border: <border-color>, <border-style>, <border-width>	Устанавливает параметры рамки для всех сторон контейнера
border-color	border-color: colorname	Устанавливает цвет рамки
border-style	border-style: none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset	Управляет внешним видом рамки
border-top	border-top: <border-color>, <border-style>, <border-width>	Устанавливает параметры рамки верхней части объекта
border-right	border-right: <border-color>, <border-style>, <border-width>	Устанавливает параметры рамки правой части объекта

Таблица 3.1 *(продолжение)*

CSS-свойство	Синтаксис	Описание
border-left	border-left: <border-color>, <border-style>, <border-width>	Устанавливает параметры рамки левой части объекта
border-bottom	border-bottom: <border-color>, <border-style>, <border-width>	Устанавливает параметры рамки нижней части объекта
border-top-color	border-top-color: colorname	Устанавливает цвет верхней рамки объекта
border-right-color	border-right-color: colorname	Устанавливает цвет правой рамки объекта
border-left-color	border-left-color: colorname	Устанавливает цвет левой рамки объекта
border-bottom-color	border-bottom-color: colorname	Устанавливает цвет нижней рамки объекта
border-top-style	border-top-style: none | hidden | dotted | dashed | solid | double | groove | ridge| inset | outset	Устанавливает внешний вид рамки верхней части объекта
border-right-style	border-right-style: none | hidden | dotted | dashed | solid | double | groove | ridge| inset | outset	Устанавливает внешний вид рамки правой части объекта
border-left-style	border-left-style: none | hidden | dotted | dashed | solid | double | groove | ridge| inset | outset	Устанавливает внешний вид рамки левой части объекта
border-bottom-style	border-bottom-style: none | hidden | dotted | dashed | solid | double | groove | ridge| inset | outset	Устанавливает внешний вид рамки нижней части объекта
border-top-width	border-top-width: medium | thin | thick | length	Устанавливает высоту и ширину рамки
border-right-width	border-right-width: medium | thin | thick | length	Устанавливает ширину правой рамки
border-left-width	border-left-width: medium | thin | thick | length	Устанавливает ширину левой рамки
border-bottom-width	border-bottom-width: medium | thin | thick | length	Устанавливает ширину нижней рамки
border-width	border-width: medium | thin | thick | length	Устанавливает ширину рамки объекта
bottom	bottom: length | percentage | auto	Устанавливает интервал между нижним краем объекта и содержащим его контейнером (глава 6)
clear	clear: none | left | right | both	Определяет, допускает ли данный элемент обтекание объекта с обеих сторон
clip	clip: shape | auto	Определяет, какая часть объекта будет видима (глава 8)
float	float: left | none | right	Определяет положение плавающего блока

CSS-свойство	Синтаксис	Описание
height	height: length | percentage	Задает высоту объекта (глава 6)
left	left: length | percentage | auto	Задает интервал между левым краем объекта и содержащим его блоком (глава 6)
margin	margin: margin width	Задает отступы на странице
margin-bottom	margin-bottom: margin width	Задает нижний отступ на странице
margin-left	margin-left: margin width	Задает левый отступ на странице
margin-right	margin-right: margin width	Задает правый отступ на странице
margin-top	margin-top: margin width	Задает верхний отступ на странице
marks	marks: crop | cross | none	Заставляет принтеры отмечать вывод HTML-страницы
max-height	max-height: none | length | percentage	Принудительно задает высоту объекта
max-width	max-width: none | length | percentage	Принудительно задает ширину объекта
min-height	min-height: none | length | percentage	Задает минимальную высоту объекта
min-width	min-width: none | length | percentage	Задает минимальную ширину объекта
orphans	orphans: integer	Назначает минимальное количество строк абзаца, которым разрешено «плавать» в конце страницы
outline	outline: <outline-width> | <outline-style> | <outline-color>	Управляет внешним видом контура объекта
outline-color	outline-color: colorname	Задает цвет контура рамки
outline-style	outline-style: none | hidden | dotted | ' dashed | solid | double | groove | ridge | inset | outset	Управляет стилем контура рамки
outline-width	outline-width: medium | thin | thick | length	Задает ширину линии контура
overflow	overflow: visible | scroll | hidden | auto	Управляет тем, как информация будет появляться на экране
padding	padding: padding-width	Управляет заполнением или пространством между объектом и содержащим его контейнером
padding-bottom	padding-bottom: padding-width	Управляет заполнением или пространством между нижней границей объекта и содержащим его контейнером
padding-left	padding-left: padding-width	Управляет заполнением или пространством между левой границей объекта и содержащим его контейнером
padding-right	padding-right: padding-width	Управляет заполнением или пространством между правой границей объекта и содержащим его контейнером

продолжение

Таблица 3.1 *(продолжение)*

CSS-свойство	Синтаксис	Описание
padding-top	padding-top: padding-width	Управляет заполнением или пространством между верхней границей объекта и содержащим его контейнером
page	page: identifier	Управляет размером и расположением страницы, а также расположением объектов на странице. В качестве идентификатора (identifier) может использоваться любое имя, но оно будет связано с указателем @раде. Например, @раде narrow (size: landscape)
page-break-after	page-break-after: auto | always | avoid | left | right	Вставляет разрыв страницы после указанного объекта
page-break-before	page-break-before: auto | always | avoid | left | right	Вставляет разрыв страницы перед указанным объектом
position	position: static | relative | absolute | fixed	Управляет способом позиционирования объекта в документе (глава 6)
quotes	quotes [<char> <char>] | none	Устанавливает символы кавычек, которые будут вставляться в начале и конце блока текста
right	right: length | percentage | auto	Устанавливает интервал между правым краем объекта и содержащим его контейнером (глава 6)
size	size: <length> [1,2] | auto | portrait | landscape	Устанавливает размер и ориентацию страницы
top	top: length | percentage | auto	Устанавливает интервал между верхним краем объекта и содержащим его контейнером (глава 6)
visibility	visibility: visible | hidden | collapse	Определяет, как будет формироваться изображение объекта в документе (глава 8)
widows	widows: integer	Определяет минимальное количество строк абзаца, которые должны остаться в верхней части страницы, когда абзац оказывается расположенным на двух страницах
width	width: length | percentage | auto	Управляет шириной блоков, которые создаются группой элементов (глава б)
z-index	z-index: auto | integer	Регулирует последовательность слоев документа (глава 6)

Слои и позиционирование

Хотя Netscape и была первой компанией, которая реализовала систему слоев при представлении страниц, корпорация Microsoft поддержала и развила эту идею в соответствии со спецификацией Интернет-консорциума. В табл. 3.2 представлены стандартные элементы, касающиеся позиционирования и использования слоев, а также свойства CSS.

Таблица 3.2. Элементы и свойства позиционирования и управления слоями

Элемент/ Свойство	Синтаксис	Описание
DIV (элемент)	<div attributes...>	Обеспечивает общую блочную структуру документа, благодаря которой возможно позиционирование и разделение на слои (глава 6)
SPAN (элемент)	<span attributes...>	Обеспечивает общую встроенную в документ структуру, которая может позиционироваться как слой (глава 6)
bottom (свойство)	bottom: length | percentage | auto	Устанавливает интервал между нижним краем объекта и содержащим его контейнером (глава 6)
height (свойство)	height: length | percentage	Устанавливает высоту объекта (глава 6)
left (свойство)	left: length | percentage | auto	Устанавливает интервал между левым краем объекта и содержащим его контейнером (глава 6)
overflow (свойство)	overflow: visible | scroll | hidden | auto	Определяет, как информация будет появляться на экране (глава 8)
position (свойство)	position: static | relative | absolute | fixed	Управляет способом позиционирования объекта документа (глава 6)
right (свойство)	right: length | percentage | auto	Устанавливает интервал между правым краем объекта и содержащим его контейнером (глава 6)
top (свойство)	top: length | percentage | auto	Устанавливает интервал между верхним краем объекта и содержащим его контейнером (глава 6)
visibility (свойство)	visibility: visible | hidden | collapse	Определяет, как будет формироваться в документе содержание объекта (глава 8)
width (свойство)	width: length | percentage | auto	Управляет шириной блоков, создаваемых группой элементов (глава 6)
z-index (свойство)	z-index: auto | integer	Регулирует последовательность слоев документа (глава 6)

Элементы, определяющие слой

Элементы и могут использоваться для создания перемещаемых интерактивных слоев. Элемент — это указатель блока или объекта, содержащего другие блоки, такие как, например, изображения и таблицы. Блочный объект создает специальную область, в которой содержится объект. Например, если представить блок в качестве дома, то вокруг него всегда имеется определенное пространство, которое и определяет его как отдельный дом. Элемент

является объектом, который можно вставить прямо в строку (встраиваемый элемент) и который может содержать любой текст. Встраиваемые объекты (Inline objects) — это объекты, которые не требуют вокруг себя дополнительного пространства. Например, слово в абзаце является встраиваемым объектом.

ПРИМЕЧАНИЕ

Различные блочные элементы, такие как изображения и таблицы, также могут использоваться в качестве слоев. Однако все-таки их лучше размещать внутри элемента . Это облегчит осуществление управления, анимации и изменения слоев благодаря большему набору свойств элемента , позволяющих осуществлять позиционирование и настройку.

Позиционирование

Для указания расположения информации в документе используются свойства position, height и width. Позиционирование элементов может задаваться абсолютными или относительными значениями.

Элемент, для которого задано *абсолютное позиционирование* (position: absolute), помещается строго в указанном месте блока. При использовании данного типа позиционирования этот элемент установит новую модель блока для всех блоков-потомков или равноправных блоков. Поскольку содержание абсолютно позиционированных блоков не обтекает другие блочные объекты, они могут наложиться и перекрыть видимое содержание других блоков. При абсолютном позиционировании объект не влияет на положение любых последующих равнозначных блоков. Следующие элементы размещаются так, как будто предшествующего объекта не существует. Если вы «привязали» элемент, используя абсолютное позиционирование, то последующие элементы будут позиционироваться в зависимости от предыдущего «плавающего» или относительно позиционированного контейнера. Более подробно абсолютное позиционирование рассмотрено в главе 8 «Использование слоев».

Фиксированное позиционирование (position: fixed) — это вариант абсолютного позиционирования, используемый браузером в качестве точки привязки для позиционирования контейнера объектов. При создании документа, имеющего большой объем, фиксированные контейнеры при прокрутке экрана перемещаться не будут. Если вы установили фиксированный слой ближе к нижней части вашего документа, то вся информация документа, кроме слоев, использующих фиксированное позиционирование, будет прокручиваться по экрану, включая все абсолютно позиционированные слои. Если вы поместите фиксированный контейнер в материале, состоящем из множества страниц, то этот контейнер всегда будет появляться в конце каждой страницы. Это позволяет сделать в документе заголовок (верхний колонтитул) и нижний колонтитул или подпись в конце одностраничных сообщений. Этот эффект было довольно сложно реализовать на web-страницах, так как они рассматриваются как непрерывный объем информации. Более подробно фиксированное позиционирование рассмотрено в главе 8.

По умолчанию используется *относительное позиционирование* (position: relative). Объекты размещаются в соответствии текущим порядком, в котором они находятся в документе, т. е. при помещении «отцентрированного» объекта вслед за объектом, в отношении которого применялось выравнивание по левому краю, он

будет размещен по центру следующей строки. После того как документ представит все объекты, из которых он состоит, будет произведен сдвиг этих объектов относительно их текущей позиции в соответствии с раскладкой, задаваемой размещением их «братьев», «родителей» и «потомков». При относительном позиционировании объектов невозможно даже частичное перекрытие, поскольку каждое перемещение любого контейнера ведет к изменению местоположения остальных контейнеров. Более подробно относительное позиционирование рассмотрено в главе 8.

Управление смещениями

Когда для управления местоположением объектов используется позиционирование, вы можете задавать их смещениями (offsets) с помощью свойств left, right, top и bottom. При написании сценария с использованием этих свойств можно задавать перемещение объектов и слоев в документах. Можно осуществить появление, исчезновение и изменение изображений и текста, их движение в зависимости от шагов пользователя, предпринимаемых им при выборе пунктов меню, заполнении форм и прочих элементов, обеспечивающих взаимодействие с web-сайтом.

Наложение объектов

Когда требуется осуществить наложение (перекрытие) объектов с помощью абсолютного позиционирования, то для управления последовательностью вертикального «складирования» слоев необходимо использовать свойство z-index. Указываемый этим свойством порядок слоев при совместном использовании со свойством visibility позволяет управлять видимостью отдельных слоев, что может использоваться как прозрачность. Совокупность этих свойств позволяет создавать интересные анимационные визуальные эффекты. Пример использования «стопки» прозрачных слоев вы можете увидеть в главе 6 «Подготовка к войне браузеров».

ПРИМЕЧАНИЕ

По умолчанию слои прозрачны, непрозрачны лишь объекты, размещаемые на них. Тем не менее при помощи свойства background вы можете задать сплошной цвет фона для любого слоя.

Управление размером слоя

Для управления размерами слоя применяются свойства height и width, а для управления порядком представления информации на данном слое используется свойство overflow.

При совместном использовании этих свойств со свойствами позиционирования и видимости имеется возможность создавать документы, использующие слои изящно и эффективно. При их использовании пользователю становится легче перемещаться по сайту и получать информацию. Несмотря на очень красивый внешний вид, который может приобрести ваша страничка, время ее загрузки остается в пределах разумного.

Фильтры

Для создания мультимедийных эффектов корпорация Microsoft использует три типа фильтров:

- визуальные фильтры;
- динамические фильтры открывания;
- динамические фильтры перехода.

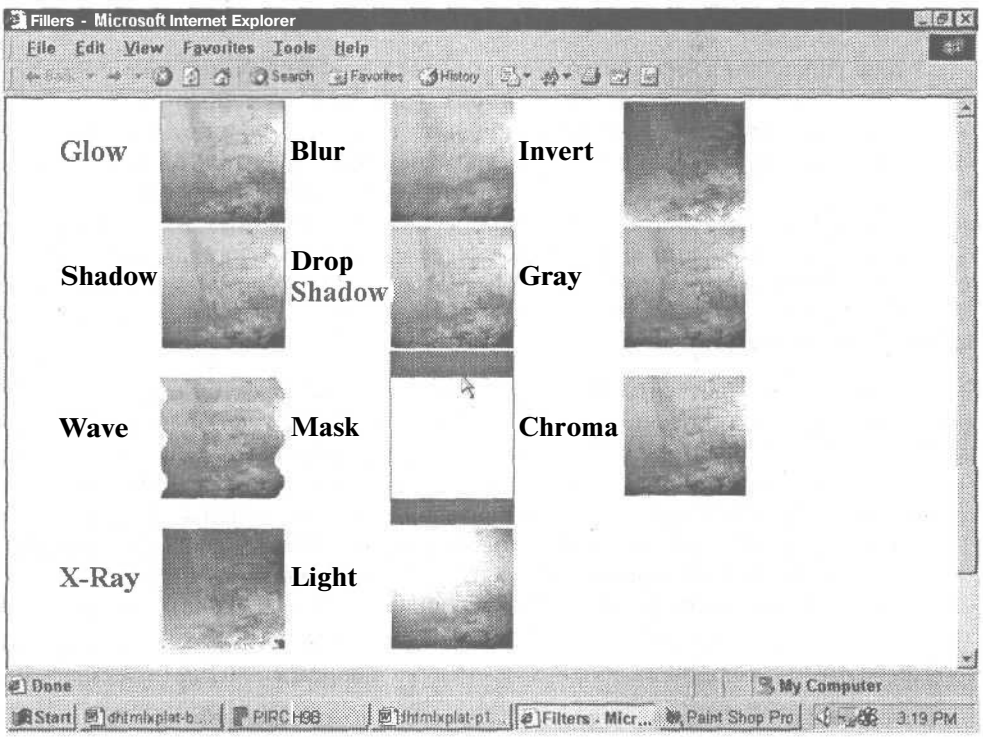


Рис. 3.2. Microsoft обеспечивает поддержку всех фильтров, доступных при использовании CSS

Каждый фильтр имеет множество настроек и может использоваться для создания дополнительных мультимедийных эффектов, представленных на рис. 3.2. Эти эффекты включают:

- blur (размытие);
- glow (свечение);
- invert (инверсия);
- chroma (одноцветность);
- light (освещение);
- shadow (тень);
- dropshadow (капельная тень);
- gray (черно-белое изображение);

- wave (волнистость);
- mask (маска-транспарант);
- x-ray («рентген»).

ПРИМЕЧАНИЕ

Поскольку издание книги черно-белое, некоторые эффекты, представленные на рисунке, вероятно, не проявятся. Попробуйте реализовать описанные фильтры самостоятельно с использованием цветного монитора.

Для использования этих фильтров совместно со свойством style элемента <img> применяется команда `filters:`*`filtername`*`(`*`parameters`*`)`. Например:

```
<img id="wavyboat" src="shipwreck.jpg"
   style="filter:wave (
    strength=8,
    freq=3,
    lightstrength=20,
    add=0,
    phase=90);">
```

Более подробную информацию об использовании фильтров можно почерпнуть из «мастерской» MSDN Web Workshop, расположенной по адресу http://msdn.microsoft.com/workshop/.

Реализация CSS компанией Netscape

Вы уже знаете, что до тех пор, пока на рынке программных продуктов существует конкуренция, вам постоянно придется сталкиваться с тем, что Netscape Navigator поддерживает CSS совершенно иным образом, нежели они реализуются корпорацией Microsoft. И это не зависит от стандартов, устанавливаемых такими органами, как Интернет-консорциум. Способы реализации рекомендованных стандартов полностью определяются компаниями. Рассмотрим, чем реализация Netscape отличается от Microsoft:

- соответствие шрифтов TrueDoc;
- дополнительное оформление текста;
- отступы, контуры и другие свойства контейнеров;
- слои и позиционирование;
- фильтры.

Соответствие шрифтов TrueDoc

Netscape использует технологию TrueDoc Font Embedding, которая «снимает» образ символов и преобразует их в побитное изображение. TrueDoc является платформонезависимой технологией и может использоваться любым приложением, при любом разрешении и на любом оборудовании. Microsoft не поддерживает TrueDoc (см. рис. 3.3), в отличие от Netscape (см. рис. 3.4).

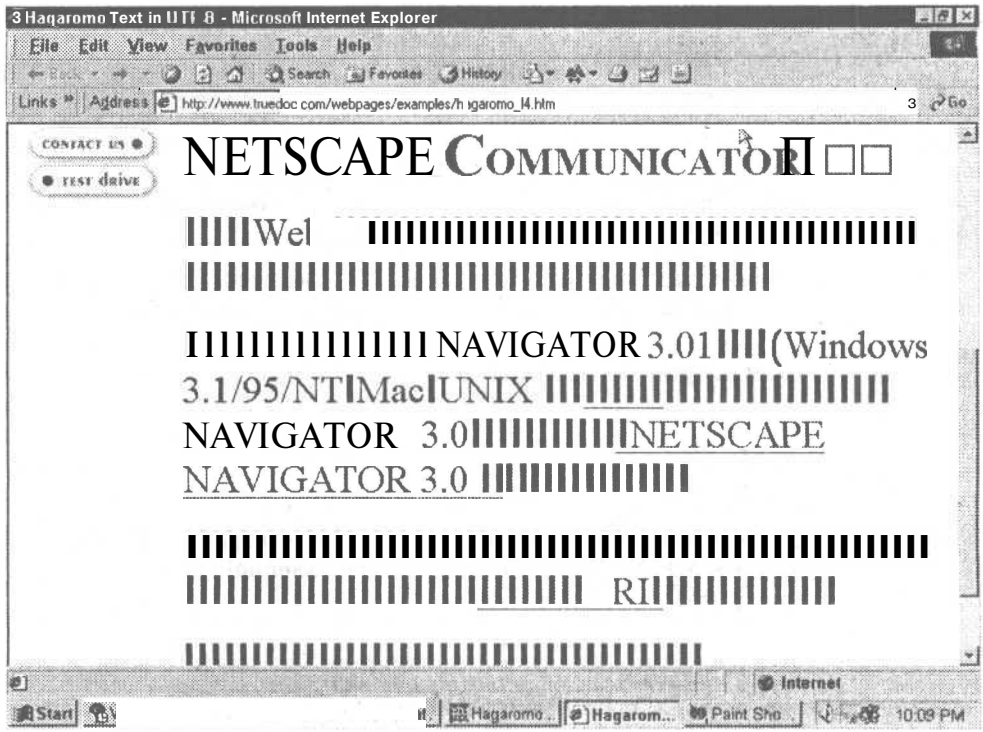


Рис. 3.3. Internet Explorer корпорации Microsoft не удается воспроизвести сайты, использующие технологию TrueDoc

Технология TrueDoc предоставляет возможность использования любого шрифта в сети и может использоваться на любом компьютере:

- шрифты хранятся в небольших, компактных, перемещаемых файлах;
- TrueDoc запоминает и быстро воспроизводит символы;
- документ может быть воспроизведен при любом разрешении с сохранением типографского качества оригинальною документа;
- TrueDoc использует сглаживание контурных неровностей шрифтов и фильтрацию краев, обеспечивая наивысшее качество вывода на любом устройстве.

Для добавления шрифтов на web-странице при помощи технологии TrueDoc сейчас невозможно воспользоваться конструкцией `@font-face`, которая определяла правила, используемые для подстановки шрифтов. При использовании шрифтов TrueDoc браузером компании Netscape для их подключения к документу необходимо использовать тег `<link>`. Выражение `@font-face` совместимо с технологией TrueDoc, но компания Netscape отказалась от его использования в Navigator 4.5. Его реализация была запланирована в Netscape Navigator 6.

Выражение `@font-face` имеет следующий синтаксис:

```
@font-face { свойство: значение; }
```

К свойствам CSS, поддерживаемым правилом @font-face, относятся:

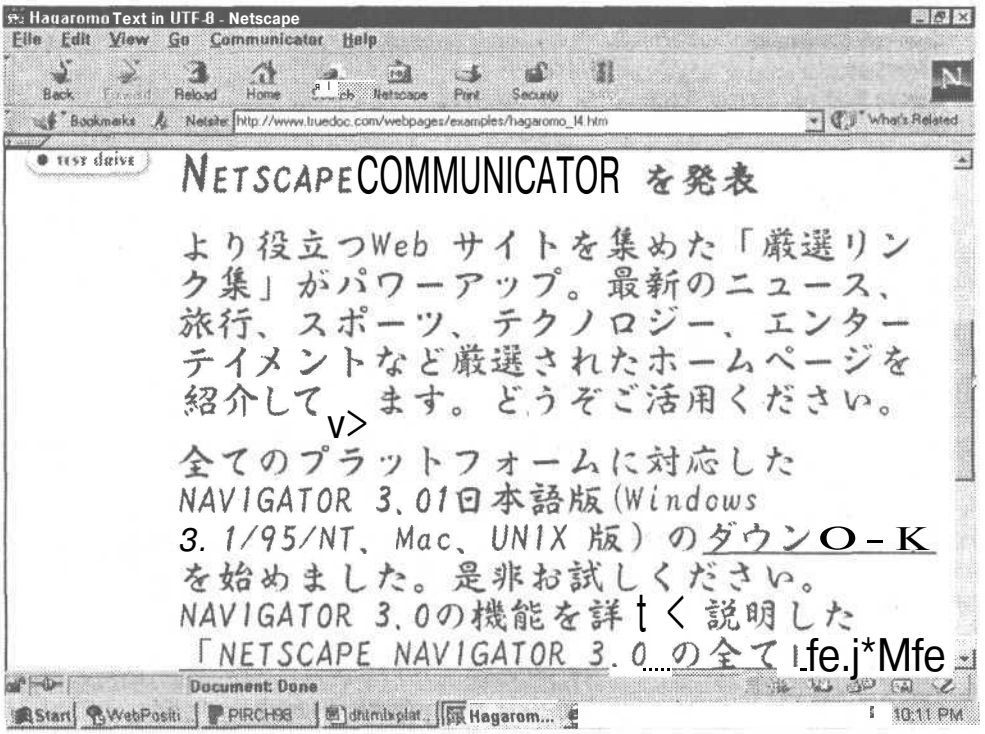


Рис. 3.4. Netscape Navigator правильно воспроизводит символы шрифта Kanji, не требуя от пользователя загрузки на его компьютер этого шрифта

font-family
font-size
font-weights
font-variant
font-style
font-face
font-set
src

Эти свойства могут быть реализованы при помощи элемента <STYLE> следующим образом:

```
<5TYLE>
  @font-face { font-family: "Jester";
          src: url(http://website.com/fonts/jester.ttf"}
  H1       { font-family: "Jester" }
</STYLE>
```

Одним из недостатков использования этой технологии является необходимость иметь формирователь шрифтов TrueDoc для создания перемещаемых ресурсов шрифтов (portable font resources, PFR). Различные шрифты, которые уже преобразованы в формат PFR, можно найти по адресу: http://www.truedoc.com. Среди них:

Amelia	Exotic
American Garamond	Geometric SlabSerif
Baker Signet	OCR-A
Breman Bold and Breman Black	Oz Handicraft
Brush	Poster Bodoni
Calligraphic	SnowCap
Engravers Gothic	Czurich

Более подробную информацию о системе шрифтов TrueDoc вы сможете найти в главе 6 в разделе «Управление шрифтами». Огромное количество информации о шрифтах TrueDoc можно найти на сайтах http://www.bitstream.com и http://www.truedoc.com.

Дополнительное оформление текста

Netscape, так же как и Microsoft, поддерживает свойство `text-decoration`, но не поддерживает все его допустимые значения. Netscape Navigator 4.0 поддерживает только эффекты подчеркивания, зачеркивания и мигания, эффект надчеркивания не реализован. Кроме того, Netscape для создания эффекта мигающего текста ввела свой элемент `<blink>`, который не поддерживается браузерами других производителей. Netscape заявила, что Navigator 6 поддерживает все свойства, изложенные в спецификации CSS 2.

Отступы, контуры и другие свойства контейнеров

Netscape Navigator 4.5, так же как и Internet Explorer, поддерживает не все свойства контейнеров. Netscape Navigator 4.5 совсем не поддерживает значения свойства `outline` и поддерживает только основные свойства margin, padding и border. Netscape заявила, что Navigator 6 поддерживает все свойства.

Свойства, поддерживаемые Netscape:

Border	border-bottom-style	padding
border-color	border-width	padding-bottom
border-style	clear	padding-left
border-top-color	float	padding-right
border-right-color	left	padding-top
border-left-color	margin	position
border-bottom-color	margin-left	top
border-top-style	margin-right	visibility
border-right-style	margin-top	width
border-left-style	overflow	z-index

ПРИМЕЧАНИЕ

Более подробно эти свойства представлены в табл. 3.1.

Слои и позиционирование

Так же как и Microsoft, компания Netscape обеспечивает стандартные CSS-свойства позиционирования и слоев. Кроме того, предшествующие версии Netscape Navigator поддерживают собственные элементы `<layer>` и `<nolayer>`.

Элемент <1 ayer> предоставлял возможность пользователям Netscape Navigator (и только им) возможность просмотра страниц с точным перекрытием слоев с прозрачным или непрозрачным содержанием. Эту возможность Microsoft реализовала при помощи элемента , поддерживаемого Netscape Navigator 4.6 и последующими браузерами. Для перемещения, переупорядочивания и/или изменения определенного содержания слоя разработчики могут использовать язык JavaScript. Сочетание JavaScript и HTML позволяет создавать интерактивную анимацию. Приложение, объединяющее две эти технологии, представляет собой объединение нескольких перекрывающихся слоев, которые могут последовательно «отслаиваться», открывая нижележащие слои. Подобная система особенно удобна для показа диаграмм и последовательности работы инженерных схем. Она также может использоваться в качестве дополнительного способа повествования. Например, в ходе воспроизведения аудиофайла на экране можно перемещать изображения или вставлять соответствующий текст, как в телепередаче «Пойте вместе с нами».

В отличие от элементов и , использующих идентификатор слоя и координаты x, y для указания последовательности и расположения отдельных блоков, элемент <1 ayer> задает строгий порядок и расположение всей последовательности блоков. Эта система z-порядка называется *порядком слоев (stacking order)*. Согласно этой системе слой А служит основой, над ним располагается слой В, над слоем В — слой С и т. д. Тег <layer> также позволяет для каждого слоя определить дополнительные подслои.

Тег <layer> помещается на странице в пределах тега <body>. При использовании этого тега не надо при определении содержания предварительно определять слой. Тег <1ayer> — нестандартный, и его «понимают» только Netscape Navigator 4.0 и последующие версии.

Фильтры

Netscape не поддерживает фильтры. Вы можете создать эффекты, подобные различным фильтрам, с помощью графики, апплетов Java или JavaScript, но не за счет реализации фильтров CSS.

В чем схожи два браузера?

Как Internet Explorer 4.0, так и Netscape Navigator 4.5 поддерживают большинство свойств таблиц стилей, представленных в спецификации CSS 1. Пока эти реализации нестабильны, имеют ошибки и несовместимы, но их уже можно использовать. CSS 2 содержит определение множества новых элементов помимо тех, что существовали в спецификации CSS 1. Можно предположить, что поддержка CSS 2 в браузерах 5-й и 6-й версий пока будет с ошибками, но поддержка CSS 1 станет более стабильной.

Как Internet Explorer, так и Netscape Navigator обеспечивают соответствие шрифтов. И хотя каждая сторона делает это по-своему, такая поддержка существует. Это означает, что для того, чтобы каждый ваш документ был представлен именно в том виде, в котором вы хотите, придется использовать оба метода.

Текст, шрифты, цвета и фон можно устанавливать при помощи свойств таблиц стиля. Представленные в следующем списке свойства CSS поддерживаются обоими браузерами:

- background-color
- background-image
- background-repeat
- color
- font-family
- font-size
- font-style
- font-weight
- line-height
- text-align
- text-decoration
- text-indent
- text-transform

Более подробную информацию о поддержке этих свойств вы найдете в приложении 2 «Совместимость каскадных таблиц стиля».

Помимо того что Netscape Navigator поддерживает некоторые дополнительные элементы оформления текста, он имеет свой **собственный** элемент <blink>. С другой стороны, корпорация Microsoft приняла решения строго соответствовать рекомендациям CSS 2. Netscape Navigator и Internet Explorer поддерживаются цвета и изображения фона для различных объектов, по Netscape Navigator *4.5* не обеспечивает поддержку таких свойств фоновых изображений, как repeat и background-position, что делают Internet Explorer 4.0 и 5.0.

С появлением контейнеров и их настроек **большинство** свойств отступов, рамок и расположения ваших объектов стали поддерживать браузеры Netscape Navigator 4.0 и 6.0, но настройки контуров поддерживает только Internet Explorer. Существуют и другие свойства, относящиеся к контейнерам, такие, как bottom и right, не поддерживаемые Netscape Navigator.

Сведения о поддержке элементов и свойств CSS различными браузерами представлены в приложении 2 «Совместимость каскадных таблиц стиля». А список свойств, которые поддерживаются обоими браузерами, представлен ниже:

- border
- border-color
- border-style
- border-right-width
- border-top-width
- border-bottom-width
- border-left-width
- border-width
- clear
- float
- left
- margin
- margin-left
- margin-right
- margin-top
- overflow
- padding
- padding-bottom
- padding-left
- padding-right
- padding-top
- Position
- Top
- Visibility
- Width
- z-index

При использовании сценариев существует возможность создавать одинаковые или подобные эффекты в обоих браузерах и на всех платформах. На это и нацелены усилия всех HTML-разрабогчиков. Ваша заслуга будет заключаться в том, чтобы любой созданный вамп сайт одинаково воспроизводился любым браузером. Объектная модель документа, рассмотренная в главе 2, обеспечивает идентификацию объектов, которая может использоваться в сценариях для динамического изменения настроек свойств CSS.

Обзор языков сценария

4

В начале 90-х небольшая группа программистов компании Sun Microsystems разработала язык программирования Oak (дуб) (назвав его в честь деревьев, растущих вокруг здания, в котором они работали). Согласно изначальному плану компании Sun, этот язык должен был использоваться такими бытовыми электронными устройствами, как интерактивный телевизор, видеомагнитофон. После расширения рынка Интернета идея интерактивного телевизора была забыта, но инженеры компании Sun не забыли Oak и стали использовать его в качестве языка разработки в web-браузере WebRunner. Таким образом, Oak является предком языка Java. В конце 1994 года Oak был усовершенствован и стал использоваться для разработки апплетов, которые можно было использовать на любой системе, поскольку в основе реализации языка была использована виртуальная машина. На этот раз язык мог использоваться для создания анимации на web-страницах.

Поскольку Oak и WebRunner уже являлись зарегистрированными торговыми марками, комитет Sun назвал этот *язык Java.* Независимо от названия, Java является весьма значимым элементом в истории компьютерных технологий.

Краткий обзор языка Java

Структура и синтаксис Java позаимствованы из языка Си++. Java стал очень мощным языком программирования с одним уникальным требованием, на которое пользователю необходимо было обращать внимание: наличие виртуальной машины Java (JVM).

В отличие от большинства языков программирования Java *не компилируется*! JVM работает как посредник, интерпретируя Java-программы для компьютера, на котором она запущена. Воспринимайте JVM как кольца в скоросшивателе. Если ваша Java-программа — это страницы, а ваш компьютер — скоросшиватель, то, независимо от типа и цвета скоросшивателя вы сможете скрепить ваши

страницы этим скоросшивателем. Другими словами, виртуальная машина Java позволяет выполнять Java-программу на компьютере с любой операционной системой, следовательно, Java может использоваться на любом компьютере. Это означает, что разработчику необходимо написать лишь одну версию программы, не задумываясь о ее совместимости с различным аппаратным обеспечением. Аналогично и пользователям не приходится беспокоиться, что приложения не будут работать на их компьютерах.

В 1996 году с выходом Netscape Navigator 2 язык Java начал активное наступление на Интернет. В браузере Netscape предусмотрена возможность использования Java-апплетов, что позволило проверить эту технологию и завоевать признание у пользователей сети. Улучшение функциональности Интернета, которое стало возможным при использовании Java, привело к широкому распространению этого языка. Принцип языка — «написал один раз, применяй везде» - заставил разработчиков влюбиться в новый язык программирования. Один из основных недостатков Java заключается в том, что для приемлемой скорости выполнения требуются довольно мощные настольные компьютеры. Хотя программы Java могут выполняться и на 16-разрядной системе, это будет происходить весьма медленно из-за ограниченной скорости, с которой эта система способна обрабатывать команды. Для достижения приемлемой производительности требуется 32-разрядная операционная система, установленная на компьютере, использующем процессор Pentium. Это обеспечит достаточную скорость работы даже с очень сложными Java-сценариями. В языках-наследниках Java (рассматриваемых далее) этот недостаток устранен: «последователи» могут исполняться на любых аппаратных средствах с приемлемым уровнем производительности.

Семейство JavaScript

По мере возрастания интереса к языку Java в сети компания Netscape разработала еще один язык, который был назван LiveScript. Теперь безразлично, было ли это просто моментом вдохновения или большим маркетинговым трюком, но компания Netscape переименовала свой язык LiveScript в JavaScript, подтолкнув тем самым остальные компании к разработке языков сценария для всех основных браузеров. С использованием JavaScript, а также языка JScript, являющегося аналогией языка JavaScript, разработанной корпорацией Microsoft, и недавно появившегося языка ECMAScript, являющегося новейшим стандартным языком, пытающимся объединить все языки сценария в один пакет, значительно легче обеспечить автоматизацию и анимацию на web-страницах. Тем не менее, в отличие от своего родителя, эти языки не поддерживают основного принципа языка Java: функционирования на любой системе.

JavaScript является интерпретируемым языком сценариев, используемым для обеспечения интерактивности web-страниц при выполнении *на стороне клиента*. Он также может использоваться *на стороне сервера* для страниц Active Server Pages (ASP), распространяемых информационным сервером Интернета (Internet Information Server, IIS) и другими серверными пакетами, включая некоторые серверы компании ChiliSoft. Элементы управления со стороны клиента (кнопки форм и различные навигационные системы) с помощью языка JavaScript осуще-

ствляют предписанные им операции. После включения JavaScript в документ и использования браузера, который может выполнять сценарии, все элементы управления считаются «клиентскими» элементами. Иначе говоря, они осуществляют управление со стороны клиента. Серверные сценарии осуществляют управление со стороны web-сервера и никогда не «перекачиваются» на компьютер посетителя. При использовании JavaScript в документе автор может, например, проверить перед отправкой формы со стороны клиента, что в нее корректно внесены данные, переслать информацию Java-апплету или встраиваемому модулю браузера, изменить содержание документа или просто определить программное обеспечение, используемое пользователем, и загрузить ту копию документа, которая наилучшим образом будет представлена данным браузером.

JavaScript разрабатывался как наиболее простой для восприятия язык, чтобы любой заинтересованный пользователь смог легко научиться использовать его для составления программ. Одним из недостатков JavaScript является невозможность трассировки написанных на нем программ с целью выявления ошибок. JavaScript по-разному реализуется браузерами Netscape и Microsoft. Корпорация Microsoft назвала собственную реализацию этого языка JScript, включив в нее методы, объекты и свойства сценария, не поддерживаемые Netscape. Таким образом, JScript не поддерживает некоторые методы, объекты и свойства сценария, имеющиеся в языке JavaScript компании Netscape. Большинство из этих различий могут существенно повлиять на результат выполнения сценариев, поэтому вопрос использования тех или иных элементов языка является очень важным. Из-за этого разработка динамических сайтов, способных работать с обоими браузерами, является довольно сложной задачей, особенно если необходимо использовать специфичные возможности одного из этих языков.

В конце 1995 года компания Netscape активно продвигала язык JavaScript (впоследствии назвав его LiveWire), представив его перед Европейской ассоциацией производителей компьютеров (European Computer Manufacturers Association, ECMA) как «открытый стандарт языка сценария Интернета». Одна из текущих задач Netscape заключается в синхронизации версии JavaScript с принятым ECMA новым стандартом — ECMAScript. Если мечты и желания web-дизайнеров сбудутся, то Netscape и Microsoft полностью перейдут на использование стандарта ECMAScript и не станут добавлять в него своих собственных «улучшений».

ECMAScript

ECMA (ее сайт в сети находится по адресу: http://www.ecma.ch) — это организация стандартизации информационных и коммуникационных систем. Она официально утвердила ECMAScript в качестве открытого стандарта языка сценария в июне 1997 года, после многомесячной работой над ним представителей Netscape, Microsoft, Borland, Sun, Internet Engineering Task Force (IETF) и W3C. Стандарт, которому присвоено наименование ECMA-262, изначально основан на языке JavaScript компании Netscape и языке JScript корпорации Microsoft, а также включает расширения и конструкции из других языков сценариев, промышленного использования.

Стандарт ECMA определяет все типы, значения, объекты, свойства и функции языка ECMAScript. Он также определяет синтаксис программ, необходи-

мый для надлежащей интерпретации сценария. В ходе разработки спецификации этого языка была сформирована группа «слова, зарезервированные для предстоящего использования», позволяющая легко развивать язык в пределах ранее установленных границ. Поскольку ECMA старалась отклонить понятия, которые планируется использовать в будущем, вы сможете обращаться к «штучкам» как к «штучкам», а не к «непонятным объектам»... Это, по сравнению с другими языками, в значительной степени облегчает развитие ECMAScript, поскольку его дальнейшая разработка уже частично намечена и конкретизирована, чего нет в других языках.

Как работает ECMAScript

ECMAScript — объектно-ориентированный язык сценария, так же как и JScript, и JavaScript. Подобно другим языкам сценария, он зависит от своего окружения — HTML-документа и web-браузера. Поскольку ECMAScript изначально разрабатывался как язык web-сценария, он может использоваться на целом спектре систем, аналогично использованию языков Java, JavaScript и JScript. Нет необходимости иметь Unix- или NT-машину, чтобы воспользоваться преимуществами применения ECMAScript.

Любой web-браузер полностью обеспечивает функционирование ECMAScript, включая такие объекты, как окна, меню, диалоговые окна, текстовые области, якоря, фреймы, cookies и ресурсы ввода/вывода. Аналогично JavaScript и JScript код ECMAScript располагается внутри HTML-документов и обрабатывается модулем выполнения сценария, содержащимся в web-браузере. Поскольку web-браузер выполняет огромное количество действий при обработке HTML-документа и имеется возможность непосредственной привязки ECMAScript к документу, то ECMAScript может взаимодействовать с браузером всякий раз при открытии или закрытии документа. Обращение к сценарию может происходить при изменении области просмотра документа, при отправке формы, при перемещении мыши или в случае возникновения ошибки. Поскольку язык сценария может ориентироваться на все действия пользователя (идентифицируемые языком как *события (event)*), нет необходимости иметь одну программу, содержащую все сценарии, разделенные по функциям. Каждую функцию можно реализовать непосредственно в той части программы, которой требуется эта функция.

Web-сервер предлагает иную возможность работы с запросами дополнительной информации с http-сервера, от клиента или из файлов. Существует разнообразие серверных языков сценария, которые зачастую используются совместно с функциями ECMAScript. Серверные сценарии предоставляют вам механизм совместного использования данных вашими сценариями. При использовании серверных сценариев или программ можно формировать содержание web-страниц, производить обновление этого содержания, а также полностью формировать всю страницу «па лету». Вы можете внедрить ECMAScript (или любую другую версию языка JavaScript) в размещенную на сервере страницу. При этом получается многофункциональная страница, в которой каждая задача выполняется программой, браузером клиента или сервером, который наиболее подходит для обработки данных соответствующим образом. Совокупность сценариев, выполняемых на стороне клиента, и серверных сценариев позволяет распределять работу но формированию страницы web-сайта между двумя компьютерами и обес-

печивать строго **индивидуальный интерфейс** для ваших web-приложений. Это позволяет также ускорить весь процесс.

Свойства ECMAScript

Каждый объект ECMAScript обладает серией таких свойств, как `eval(x)`, `parseInt`, parseFloat, parseString, `length`, `value` и т. д. Эти свойства обеспечивают возможность обращения к отдельным аспектам объекта. Например, объект image имеет свойства width (ширина) и height (высота). Каждое свойство, в свою очередь, имеет ряд атрибутов, определяющих, как данное свойство может использоваться. Например, если один из этих атрибутов имеет значение «только для чтения», вы не сможете изменить значение этого свойства при помощи ECMAScript. Свойства представлены в приложении 4 «Совместимость JavaScript и JScript» н практически полностью отвечают спецификации ECMAScript по форме и выполняемым функциям. Хотя языки все еще называются JavaScript и JScript, их создатели переходят на реализацию ECMAScript, просто не желая изменять названия.

Свойства могут содержать другие объекты, значения или методы. Если свойство содержит значение, то оно должно принадлежать к одному из встроенных типов, например boolean, null, number, string или undefined. *Метод* — это просто функция, которая связана с объектом при помощи свойства. Если свойство содержит другой объект, то он может быть одним из встроенных объектов, например array, boolean, date, global, math, number, object или string.

В ECMAScript используется набор *операторов,* позволяющих сравнивать или определять взаимоотношения данных или объектов в соответствии со сценарием. Для вычислений в ECMAScript используются стандартные операторы, аналогичные операторам языков JavaScript н JScript. Ниже приведена часть этих операторов.

Сложение (+)	Поразрядное исключающее ИЛИ (^)	Меньше (<)
Увеличение значения на единицу (++)	Поразрядное исключающее ИЛИ с присвоением результата левому операнду (^=)	Меньше или равно (<=)
Назначение (=)	Вычитание (--)	Логическое И (&&)
Поразрядное И (&)	Деление (/)	Логическое отрицание (!)
Поразрядное И с присвоением результата левому операнду (&=)	Деление с присвоением результата левому операнду (/=)	Логическое ИЛИ (||)
Поразрядный сдвиг влево (<<)	Равенство (==)	Умножение (*)
Поразрядное отрицание (NOT) (~)	Больше (>)	Умножение с присвоением результата левому операнду (*=)
Поразрядное ИЛИ (OR) (|)	Больше или равно (>=)	Отрицание (-val)
Поразрядное ИЛИ с присвоением результата левому операнду (|=)	Оператор инкрементации (++)	Вычитание (-)
Поразрядный сдвиг вправо (>>)	Не равно (!=)	Вычитание с присвоением результата левому операнду (-=)

Упрощенный формат синтаксиса ECMAScript делает его схожим с JavaScript и JScript, что позволяет его также отнести к категории простых языков сценария. Примером является работа с переменными. В ECMAScript перед использованием переменных не требуется их описание и определение типа данных. Большинство языков сценария, да и обычных языков программирования, требует перед использованием переменных произвести их декларирование и точно указать типы таких значений, как string, integer, array или Boolean.

Microsoft, Netscape, Sun, Borland и IBM заявили о принятии ECMAScript и приступили к реализации его поддержки в своих продуктах. В ближайшем будущем вы, вероятно, все реже будете видеть JavaScript и JScript, а ECMAScript — чаще.

JScript: реализация Microsoft

Самый простой способ описать язык JScript — это заявить, что он — просто переделка JavaScript. JScript — это собственная версия корпорации Microsoft языков ECMAScript и JavaScript. JScript 3 — это полная реализация спецификации ECMAScript с дополнительными улучшениями, которые могут реализоваться только с помощью Internet Explorer. JScript, так же как JavaScript, использует некоторые возможности языка Java, но является не просто упрощенной версией этого языка. JScript — это полноценный объектно-ориентированный язык сценария. (Объекты, свойства и методы, поддерживаемые JScript, представлены в приложении 4 «Совместимость JavaScript и JScript».)

JavaScript поддерживает ряд коллекций и функций, не поддерживаемых в оригинальной версии JavaScript компании Netscape. С принятием языка ECMAScript обеими организациями общие команды станут более универсальными, но JScript все еще будет содержать собственные свойства, представленные в табл. 4.1.

Таблица 4.1. Коллекции, функции, объекты и операторы, реализованные лишь Microsoft

Команда	Часть языка
Boolean	Объект
Comparison s	Оператор
Compound Assignments	Оператор
Conditional (trinary)	Оператор
Dictionary	Объект
Drive	Объект
Drives	Коллекция
Enumerator	Объект
File	Объект
Files	Коллекция
Folder	Объект
Folders	Коллекция
Identity (===)	Оператор
NonIdentity (!==)	Оператор
ScriptEngine()	Функция
ScriptEngineBuildVersion()	Функция
ScriptEngineMajorVersion	Функция
ScriptEngineMinorVersion	Функция
TextStream	Объект
VBArray	Объект

Помимо этих элементов существует множество методов, свойств и выражений, реализуемых JScript и недоступных для JavaScript. Полный список различий можно найти в приложении 4 «Совместимость JavaScript и JScript».

JavaScript: реализация Netscape

Если вы изучили JScript, то после него сложно вернуться к изучению JavaScript, держа в уме все их нюансы. С помощью JScript можно выполнить те действия, которые вообще невыполнимы при помощи JavaScript. Кроме того, существует множество объектов, методов, свойств и выражений, поддерживаемых JavaScript и не поддерживаемых JScript. Вам необходимо помнить что даже если объекты и методы имеют одинаковое наименование, они могут иметь совершенно различное предназначение. Существует единственная языковая конструкция, представленная только в JavaScript: FileSystemObject.

В чем схожи два браузера?

Обработка сценариев в Internet Explorer и Netscape Navigator в основном практически одинакова. Они оба поддерживают JavaScript, что позволяет осуществлять модификацию объектов и откликаться на действия посетителя сайта или на события, происходящие с самим документом. Реализация основывается на объектной модели документа, которая также позволяет создавать элементы взаимодействия и анимацию. Поддержка соответствующей совокупности событий позволяет создавать документы, которые осуществляют аналогичное взаимодействие в обоих браузерах. Поскольку в качестве события может выступать как нажатие клавиши пользователем, так и факт завершения загрузки изображения на страничку, существует возможность назначить наступление события или реагирование практически на любое изменение.

Оба браузера поддерживают следующие события:

onMouseMove	onScroll
onMouseOut	onChange
onMouseOver	onKeyDown
onBlur	onKeyPress
onClick	onKeyUp
onDblClick	onReset
onFocus	onSubmit
onMouseDown	onLoad
onMouseUp	unLoad
onResize	

Помимо этих Internet Explorer обрабатывает множество дополнительных событий, которые не должны использоваться на сайтах, рассчитанных на посещение их обоими браузерами. Полный список событий приведен в приложении 4.

Помимо объектов и событий, поддерживающих объектную модель документа, JScript и JavaScript одинаково используют специальные языковые команды для работы с объектами, методами, операторами, свойствами и выражениями. Ниже представлены команды сценария, обрабатываемые обоими браузерами. Но вам, вероятно, все-таки придется проверить каждую из этих команд. Некоторые из них могут работать с обоими браузерами не совсем так, как вы ожидаете.

Команды и объекты

Array	Math
Date	Number
FileSystemObject	Object
Global	RegExp
Function	RegularExpression

Команды и методы

abs(number)	getDate()	match(regexp)	setUTCSeconds()
acos(number)	getDay()	max(numl, num2)	setYear()
anchor(string)	getFullYear	min(num1, num2)	sin(number)
asin(number)	getHours()	parse(datestring)	slice(i,j)
atan(number)	getMinutes()	parseFloat(string)	small()
atan2(x,y)	getMonth	parseInt(string)	sort(comp.function)
big()	getSeconds()	pow(num, power)	split(char)
blink()	getTime()	random()	sqrt(number)
bold()	getTimezoneOffset()	replace(regexp, string)	strike()
ceil(number)	getUTCDate()	reverse()	sub()
charAt(index)	getUTCDay()	round(number)	substr(start, length)
charCodeAt(index)	getUTCFullYear()	search(string, string)	substring(intA, intB)
compile(pattern)	getUTCHours()	setDate()	sup()
concat(array2)	getUTCMinutes()	setFullYear()	tan(number)
CreateTextFile (filename[, overwrite [, unicode]])	getUTCMonth()	setHours()	test(string)
escape(charstring)	getUTCSeconds()	setMinutes()	toGMTString()
eval(codestring)	getYear()	setMonth()	toLocaleString()
exec(string)	indexOf(string, [i,j])	setSeconds()	toLowerCase()
exp(number)	isNaN(expression)	setTime()	toString()
fixed()	italics()	setUTCDate()	toUpperCase()
floor(number)	join(char)	setUTCFullYear()	unescape(string)
fontcolor(color)	lastIndexOf(string, [i,j])	setUTCHours()	UTC()
fontsize(intSize)	link(url)	setUTCMinutes()	
fromCharCode(code1, code2, ..., coden)	log(number)	setUTCMonth()	

Команды для операторов

Добавление (+)	Увеличение на единицу (++)
Добавление значения (++)	Не равно (!=)
Назначение (=)	Меньше (<)
Поразрядное И (&)	Меньше или равно (<=)
Поразрядное И с присвоением результата левому операнду(&=)	Конъюнкция (&&)
Поразрядный сдвиг влево (<<)	Логическое отрицание (!)
Поразрядное инвертирование внутреннего двоичного кода аргумента (~)	Дизъюнкция (||)
Поразрядное ИЛИ (|)	Получение остатка от деления операндов (%)
Поразрядное ИЛИ с присвоением результата левому операнду (|=)	Присвоение левому операнду остатка от деления операндов (%=)
Поразрядный сдвиг вправо (>>)	Умножение (*)
Поразрядное исключающее ИЛИ (^)	Умножение с присвоением результата левому операнду (*=)
Поразрядное исключающее ИЛИ с присвоением результата левому операнду (^=)	Отрицание (-val)
Запятая (,)	Оператор new
Удалить (delete)	Вычитание (-)
Уменьшение на единицу (--)	Вычитание с присвоением результата левому операнду (-=)
Деление (/)	Оператор this
Деление с присвоением результата левому операнду (/=)	Оператор typeof
Равно (==)	Оператор унарного отрицания (-)
Больше (>)	оператор сдвига вправо без учета знака (>>>)
Больше или равно (>=)	Оператор void

Команды для свойств

$1...$9 Properties	LN2
Arguments	LN10
caller	LOG2E
E	LOG10E
global	MAX_VALUE
ignoreCase	MIN_VALUE
index	multiline
Infinity	NaN
input	NEGATIVE_INFINITY
lastIndex	POSITIVE_INFINITY
lastMatch	prototype
lastParen	rightContext
leftContext	SQRT1_2
length	SQRT2

Команды для выражений

Break	if...else
@cc_on	Labeled
Comment s	return
continue	@set
do...while	switch
for	this
for... in	var
function	with
@if	while

Определение настроек среды пользователя

Наиболее значимой частью DHTML-сайта является определение среды пользователя. Для организации взаимодействия web-сайта с любой машиной прежде всего необходимо определить операционную систему пользователя, используемый им браузер, разрешение дисплея и настройки Windows. Без этих сведений вы не сможете адаптировать свой сайт под конкретного пользователя, что предполагается при реализации динамического сайта.

Как известно, каждая система имеет свои особенности и работает по-разному. Кроме того, любой компьютер имеет свои собственные варианты настройки: цвета, размеры и типы шрифтов, разрешение дисплея, количество цветов дисплея и установленное программное обеспечение. Эти настройки, а также и другие особенности установленного программного обеспечения приводят к расширению возможных вариантов взаимодействия с web-браузером. Есть десятки программ, которые подключаются непосредственно к браузеру пользователя, занимают экранное пространство и по-разному взаимодействуют со сценариями, запускаемыми на web-сайте. За счет установления потенциальных причин вы можете заранее определить возможность конфликта еще на этапе написания сценария, что избавит посетителей от раздражения от появляющихся сообщениях об ошибках и дополнительных окон на их экранах.

ПРИМЕЧАНИЕ

Все примеры сценариев в данной главе основаны на использовании языков JavaScript/JScript. Для простоты при одновременной ссылке на оба языка используется JavaScript.

Определение среды

Интерактивный сайт *должен* определять браузер каждого посетителя сайта. Это можно сделать множеством способов. Один из способов определить web-браузер заключается в использовании двух простых выражений IF..., представленных ниже. Первое позволяет определяет использование Netscape Navigator 4 или более нового браузера, а второе — использование Internet Explorer 4.

```
if (document.layer) ns=1;
if (document.all) ie=1
```

Пара элементов document.layer и object.property поддерживается только браузером Netscape Navigator (ns), в то время как пара document.all и object.property — только Internet Explorer (ie). В данном примере выражение IF проверяет действительность каждого их этих свойств. Если код программы в выражении, проверяющем свойство document.layer, установит значение переменной ns в 1, то есть True (Истинно), это означает, что программа говорит: «Да, это действительно браузер компании Netscape». Если второе выражение окажется истинно, то программа установит значение переменной ie, сообщая: «Да, это — Internet Explorer».

Самой большой проблемой данного метода является то, что он не проверяет версию браузера, возможность отключения поддержки сценариев браузером, а также то, что прочие браузеры могут поддерживать данные элементы сценария, абсолютно не поддерживая остальные.

Более полная проверка браузера может выглядеть следующим образом.

Листинг 5.1. Улучшенная проверка браузера[1]

```
/* Назначить переменной browser значение «пустая строка» */
    var browser = "";
            /* bwr - это переменная, содержащая название web-браузера */
    var bwr = navigator.appName;
           /* ver - это версия, содержащая номер версии web-браузера  */
    var ver = parseInt(navigator.appVersion, 10);

/* Логика данного выражения говорит: если brw - Netscape и Version - 6 */
/* тогда это — Netscape 6.0. Этот тест повторяется для всех номеров    */
/* версий Netscape и Internet Explorer.                                */
if ( bwr == "Netscape" && ver == 6 ) browser = "Netscape 6.0";
else if ( bwr == "Netwcape" && ver == 4 ) browser = "Netscape 4.0";
else if ( bwr == "Netscape" && ver == 3 ) browser = "Netscape 3.0";
else if ( bwr == "Netscape" && ver == 2 ) browser = "Netscape 2.0";
else if ( bwr == "Microsoft Internet Explorer" && ver == 3 )
          browser = "MSIE 3.0";
else if ( bwr == "Microsoft Internet Explorer" && ver == 4 )
          browser = "MSIE 4.0";
else if ( bwr == "Microsoft Internet Explorer" && ver == 5 )
          browser = "MSIE 5.0";
else browser = "Other";
```

Этот тест проверяет имена и номера версий обоих браузеров и помещает их в отдельную переменную, которая может быть проверена, до *разветвления кода*. Разветвление кода — это процесс, с помощью которого вы определяете, какой метод использовать для выполнения задачи, основываясь на значении другой переменной. Применительно к нашему случаю это оператор условного перехода, облегчающий дальнейшее тестирование, поскольку приходится проверять лишь одну переменную, но в данном случае приходится тестировать строковое значение, а не логическое. При проверке строкового значения необходимо выполнить строчное сравнение во всех оставшихся модулях программы. Строчное сравнение требует проверки каждого символа строки текста с символами другого текста на их точное совпадение.

А теперь рассмотрим еще одну систему проверки типа браузера, представленную в листинге 5.2, которую мы будем использовать в ознакомительном сценарии, который помещен в конец данной книги.

[1] Все исходные тексты можно найти на сайте издательства «Питер» www.piter.com.

Листинг 5.2. Окончательная система определения браузера

```
/* bwr - это переменная, содержащая наименование web-браузера */
    var bwr = navigator.appName;
/* ver — это переменная, содержащая номер версии браузера */
    var ver = parseInt(navigator.appVersion, 10);
/* Эта программа проверяет значение переменных bwr и ver. Если они равны
значениям в круглых скобках, то тест выдаст значение true */
/* и будет содержать такое значение, как NS4. Если оба значения не подой-
дут, то переменная будет содержать значение 0 (false). */
NS4 = ( bwr == "Netscape" && ver == 4 )  ? 1 : 0;
NS6 = ( bwr == "Netscape" && ver == 6 )  ? 1 : 0;
IE4 = ( bwr == "Microsoft Internet Explorer" && ver == 4 ) ? 1 : 0;
IE5 = ( bwr == "Microsoft Internet Explorer" && ver == 5 ) ? 1 : 0;
/* Эта часть программа проверяет, что мы используем браузер,
поддерживающий динамический HTML, за счет сравнения */
/* со значениями NS4, NS6, IE4 и IE5. Если по крайней мере одно из них
совпадет, */
/* то переменная ver4 будет установлена в true. */
ver4 = (NS4 || IE4 || NS6 || IE5) ? 1 : 0;
/* Проверяет версию браузера Macintosh */
isMac = (navigator.appVersion.indexOf("Mac") != -1) ? 1 : 0;
/* если браузер окажется версии NS4, NS6, IE5, IE4 и не Mac версии IE4,
то этот браузер способен отображать  динамические эффекты, создаваемые
данными сценариями*/
isDynamic = (NS4 || (IE4 && !isMac) || NS 6 || IE5) ? 1 : 0;
```

Первое выражение просто проверяет браузер: является ли он Netscape Navigator 4 или 6 либо Internet Explorer 4 или 5. В этом случае значение функции, равное 1, свидетельствует об успешном чтении, а 0 — о неудачном прочтении значения. Если посетитель использует браузер компании Netscape, то NS4 будет иметь значение 1 (или True). Необходимо проверить лишь номера версий браузеров, поддерживающих динамический код, к которым относятся браузеры 4-й и более новых версий. Известно, что эти браузеры должны быть не менее 4-й версии, поскольку более ранние версии не поддерживают JavaScript на требуемом уровне и не поддерживают объектную модель документа, поддержка которой требуется для построения динамического документа.

Переменная ver4 получает значение Истинно только в том случае, если значения переменных NS4, NS6, IE4 и IE5 — истинно. Таким образом, вы можете определить ранние версии браузеров или браузеры, не поддерживающие объектную модель Microsoft или Netscape. В том случае, если браузер более старый, чем 4-й версии, то переменная ver4 будет установлена в 0 (или False).

Переменная isMac будет установлена в true, если приложение запущено на операционной системе Macintosh. Последнее выражение данной программы устанавливает переменную isDynamic. Это выражение проверяет все браузеры и устанавливает эту переменную в true, если браузер — версии 4.x или более новой, но не Internet Explorer для Macintosh.

При использовании данной комбинации проверок вы можете точно определить, сможет ли браузер пользователя поддерживать и корректно реализовать сценарии, имеющиеся на страницах сайта. На данном этапе у вас имеется два варианта общения с браузером пользователя. Если это браузер версии 4.x и выше, вы просто продолжаете отображать страницы, как это было задумано. Если вы имеете дело с более старыми браузерами, то вы должны представить пользователю для просмотра другой документ, содержащий указание на причину, по которой ему представлен статический, «плоский» документ вместо интерактивной

страницы, которую он ожидал увидеть. Я обычно указываю ссылки на сайты, с которых пользователь может загрузить соответствующие версии Netscape Navigator и Internet Explorer и продолжить знакомство с динамическими сайтами сети.

ПРИМЕЧАНИЕ

Приведенная программа осуществляет проверку только двух основных браузеров — Netscape Navigator и Microsoft Internet Explorer. Существует множество других браузеров, которые поддерживают некоторые или все команды сценариев и объекты DOM, поддерживаемые основными браузерами. Вы можете добавить проверку и этих браузеров таким же образом, как и проверку основных браузеров. Если вы собираетесь написать сценарии для всех браузеров, которые поддерживают JavaScript и DOM, то вам необходимо знать язык сценария, поддерживаемый каждым из этих браузеров.

Когда вы предоставляете пользователю, имеющему недостаточно мощное обеспечение, «плоский» документ, то в нем должно быть представлено как минимум содержание сайта. Вы можете снабдить его таким же визуальным интерфейсом, как и у диалогового сайта, только без систем автоматической навигации и изменения изображений или автоматического создания содержания. Это позволит указанным пользователям получить ту же информацию, что и пользователям браузеров четвертых версий, такое же визуальное изображение и желание обновить программное обеспечение.

Web-сайт Acropolis может быть представлен двумя интерфейсами, которые представлены на рис. 5.1 и рис. 5.2.

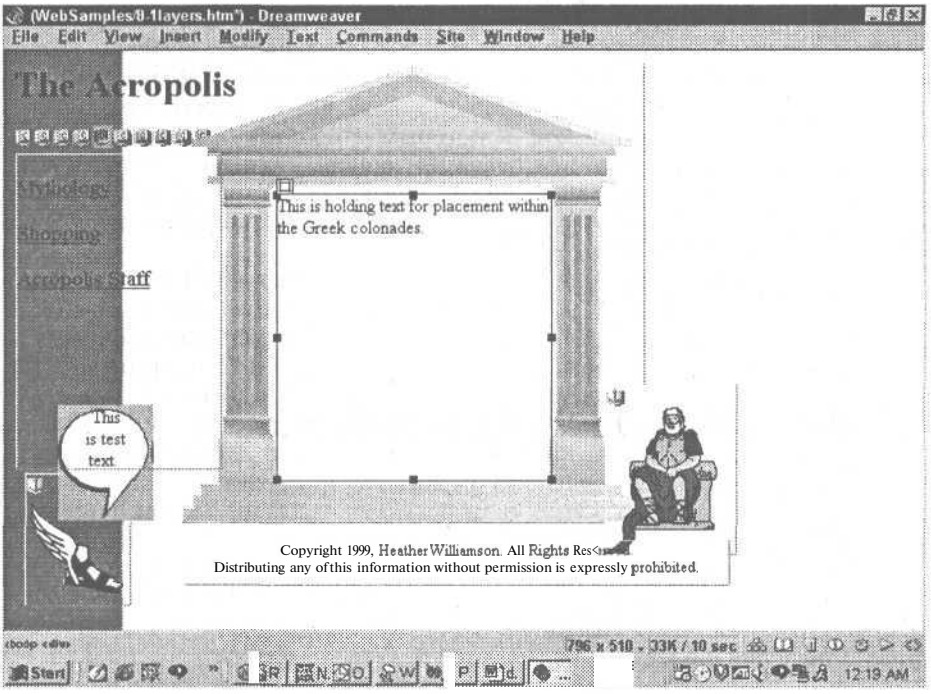


Рис. 5.1. Web-сайт Acropolis, на котором полностью интерактивный web-интерфейс имеет простой внешний вид, который легко воспроизводится более ранними браузерами. В то же время он обеспечивает приятный интерфейс для пользователей самых свежих браузеров

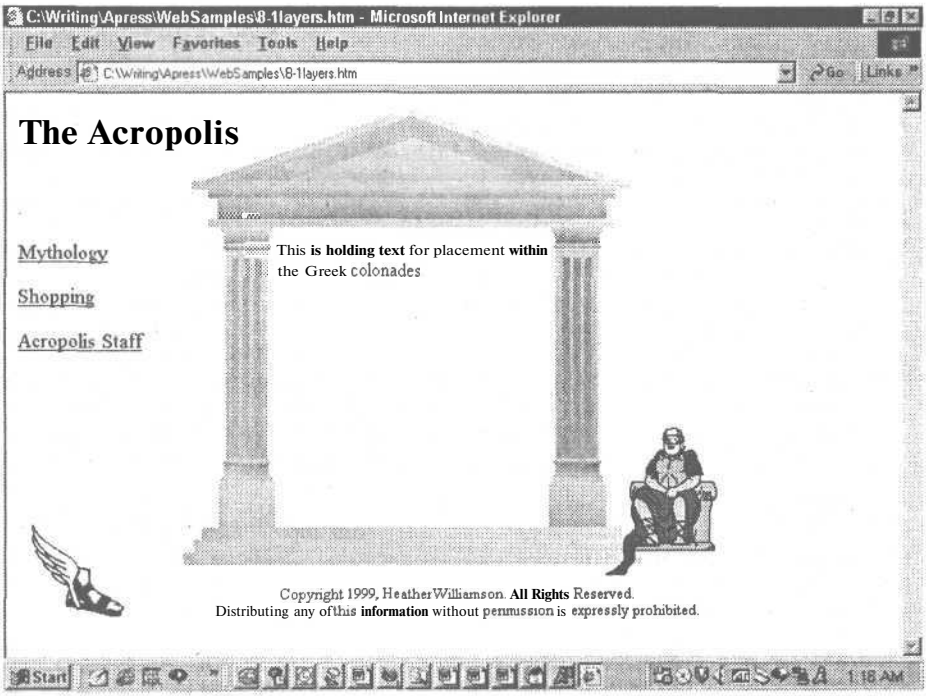


Рис. 5.2. Интерфейс Acropolis может быть представлен как «плоский» документ, воспроизводящий интерактивные графические элементы как изображения, размещенные в статическом документе

Для загрузки нединамической версии сайта вы можете использовать нижеприведенный пример. В нем производится загрузка в окно браузера файла, указанного справа от оператора присвоения.

```
if !ver4 {
   window.location.href = "indexplain.htm"
   }
```

Определение размеров экрана

Следующим шагом в подстройке сайта под определенного пользователя является определение доступного экранного пространства, предоставляемого web-браузером. Выполнять его следует клиент-ориентированной программой. При первоначальной загрузке документа необходимо установить полезную ширину экрана web-браузера. Наиболее удачным приемом является использование события onLoad элемента BODY (см. листинг 5.3).

Листинг 5.3. Событие onLoad определяет размер видимой области браузера посетителя сайта

```
<BODY … onLoad = "
if (NS4 || NS6) {
```

продолжение

Листинг 5.3 *(продолжение)*

```
      avail_width = innerWidth;
      avail_height = innerHeight;
   /* Определение доступной ширины и высоты экрана посетителя*/
   /* при помощи значений innerWidth и innerHeight, поддерживаемых Netscape.*/
   /* Эти параметры не поддерживаются Internet Explorer 3, 4, и 5.        */
      ZeusLyr = document [ 'ZeusLayer'];   //Определить слой Zeus
      ZeusLyr.left = avail_width - 100;   //Разместить слой Zeus в 00 пикс.
   от левого края
      ZeusLyr.top = avail_height - 100;   //Разместить слой Zeus на 100 пикс.
                                           выше нижнего края
      GreekLyr = document [ 'GreekHallLayer'];   //Определить слой Greek
      GreekLyr.left = 100;                       //Разместить слой в 100 пикс.
                                                   от левого края
      GreekLyr.top = 20;                         //Разместить слой на 20 пикс.
                                                   ниже верхнего края
      HermesLyr = document ['HermesLayer']; //Определить слой Hermes
      HermesLyr.left = 25;                  //Разместить слой Hermes в 25 пикс.
                                              от левого края
      HermesLyr.top = avail_height - 100;   //Разместить слой Hermes на
                                              100 пикс. выше нижнего края
   /* Каждый слой документа необходимо установить таким же образом. */
   }
   else (IE4 || IE5) {
      avail_width = document.body.clientWidth;
      avail_height = document.body.clientHeight;
   /* Определение доступной ширины и высоты экрана пользователя */
   /* при помощи поддерживаемых Microsoft значений clientWidth и clientHeight. */
   /* Эти значения не поддерживаются Netscape Navigator 3, 4 и 6.*/
      ZeusLyr = document.all['ZeusLayer'].style; //Определить слой Zeus
      ZeusLyr.left = avail_width - 100;   //Разместить слой Zeus в
                                            100 пикс. от левого края
      ZeusLyr.top = avail_height - 100;   //Разместить слой Zeus на
                                            100 пикс. выше нижнего края
      GreekLyr = document.all['GreekHallLayer'].style;
            /* Определить контейнер Greek.*/
      GreekLyr.left = 100;                //Поместить контейнер в 100 пикс.
                                            от левого края
      GreekLyr.top = 20;                  //Поместить контейнер на 20 пикс.
                                            ниже верхнего края
      HermesLyr = document.all['HermesLayer'].style; //Определить слой
   Hermes
      HermesLyr.left = 25;                //Поместить слой Hermes в 25 пикс.
                                            от левого края
      HermesLyr.top = avail_height - 100; //Поместить слой Hermes на 100 пикс.
                                            выше нижнего края
   /* Каждый слой в документе необходимо установить таким же образом. */

   }" ... >
```

ПРИМЕЧАНИЕ

Определение отдельных слоев требуется проводить только один раз. Далее эти переменные могут использоваться для идентификации слоев в остальном сценарии. В главе 7 мы рассмотрим более простой способ определения каждого слоя, не определяя его отдельно для каждого браузера.

В листинге 5.3 значения переменных avail_width и avail_height используются в математическом равенстве, которое позволяет задать расположение изображения относительно размеров экрана. Это позволяет настроить сайт в соответствии с размерами экрана посетителя, не устанавливая для него никаких ограничений.

ПРИМЕЧАНИЕ

Вам нет необходимости беспокоиться о написании сценариев воспроизведения изображений для различных значений ширины экрана. Когда разрешение экрана составляет 640x480 или меньше, в соответствии со значением атрибута lowsrc, можно загрузить уменьшенные версии изображений. Для автоматического изменения загружаемого изображения для мониторов с низким разрешением атрибут изображение lowsrc может использоваться следующим образом: <IMG src="bkg.jpg" lowsrc="bkg_sm.jpg">.

Что произойдет, если пользователь во время просмотра сайта изменит размер экрана? Если использовались строго позиционированные элементы, то при уменьшении экрана пользователь потеряет часть видимой информации или интерактивных элементов. Это можно предотвратить, осуществляя автоматическое обновление содержание экрана при изменении размера окна.

Самым простым способом является использование события onResize элемента BODY (см. листинг 5.4).

Листинг 5.4. Событие onResize позволяет управлять взаимодействием слоев при изменении окна браузера

```
<BODY ... onResize = "
if (NS4 || NS6) {
   avail_width =  innerWidth;
   avail_height = innerHeight;
/* определить новые ширину и высоту экрана посетителя */
/* при помощи поддерживаемых Netscape значений innerWidth и innerHeight.*/
/* Они не поддерживаются Internet Explorer 3, 4 и 5. */
   ZeusLyr.left = avail_width - 100;   //Разместить слой Zeus в 100 пикс.
                                         от края
   ZeusLyr.top = avail_height - 100;   //Разместить слой Zeus на 100 пикс.
                                         ниже верхнего края
 GreekLyr.left = 100;                 //Размесить слой в 100 пикс. от края
   GreekLyr.top = 20;                  //Разместить слой на 20 пикс. ниже
                                         верхнего края
   HermesLyr.left = 25;                //Разместить слой Hermes на
                                         25 пикс. от края
   HermesLyr.top = avail_height - 100; //Разместить слой Hermes на
                                         100 пикс. вверх от нижнего края
/* каждый слой в этом документе необходимо установить аналогичным образом.*/
}
else (IE4 || IE5) {
   avail_width = document.body.clientWidth;
   avail_height = document.body.clientHeight;
/* Определить новые размеры width и height экрана пользователя */
/* при помощи поддерживаемых Microsoft значений clientWidth и clientHeight. */
/* Эти значения не поддерживаются Netscape Navigator 3, 4 и 5. */
   ZeusLyr.left = avail_width - 100;   //Разместить слой Zeus в 100 пикс.
                                         от края
   ZeusLyr.top = avail_height - 100;   //Разместить слой Zeus в 100 пикс.
                                         вверх от нижнего края
   GreekLyr.left = 100;                //Разместить контейнер в 100 пикс.
                                         от левого края
```

продолжение

Листинг 5.4 *(продолжение)*

```
    GreekLyr.top = 20;                    //Разместить контейнер на 20 пикс.
                                            ниже верхнего края
    HermesLyr.left = 25;                  //Разместить слой Hermes в 25 пикс.
                                            от левого края
    HermesLyr.top = avail_height - 100; //Разместить слой Hermes на
                                            100 пикс. выше нижнего края
/* Каждый слой этого документа необходимо настроить аналогичным образом. */

}" … >
```

По существу, программа, приведенная в листинге 5.4, просто осуществляет изменение размещения объектов относительно новых размеров экрана. Программу можно разбить на несколько отдельных функций, которые могут вызываться из любого места документа, а не повторять ее периодически в документе.

В каждом из этих сегментов программы пропущен один важный элемент - функция проверки того, что экран не превратился в нагромождение перекрывающихся изображений, что возможно при сокращении размеров. Эту проверку можно осуществить различными способами. Один из них заключается в принудительном установлении размеров окна пользователя, а другой — в загрузке уменьшенных версий изображений при сокращении размеров экрана менее определенного предела.

Управлять размером окна пользователя можно следующим образом:

```
window.open("acropolis.htm",null,
"height=390,width=550,status=yes,toolbar=no,menubar=no,location=no");
```

Посмотрим, как можно определить и перезагрузить уменьшенные версии изображений в том случае, когда размер экрана меньше, чем требуемый для вывода основных изображений. Первая из этих функций — checksizeNS() — осуществляет выбор загружаемых изображений для браузеров компании Netscape, а вторая — checksizeIE() — для Internet Explorer.

```
function checksizeNS ()   {
   if (avail_Width < 640) && (avail_Height < 480)   {
/* Если экран имеет разрешение 640x480 или меньше, то загрузить
уменьшенные изображения, представленные ниже */
      document['ZeusImg'].src="images/zeussmall.gif";
      document['GreekImg'].src="images/greeksmall.gif";
      document['HermesImg'].src="images/hermessmall.gif";
/* Так необходимо поступить со всеми изображениями страницы. */
   }
}
function checksizeIE ()   {
   if (avail_Width < 460) && (avail_Height < 600)   {
      document.all['ZeusImg'].src="images/zeussmall.gif";
      document.all['GreekImg'].src="images/greeksmall.gif";
      document.all['HermesImg'].src="images/hermessmall.gif";
   }
}
```

После изменения размеров пользователем вам необходимо или самому написать дополнительные функции, или, при совместном использовании двух функций проверки изображений, изменить изображения и их размещение на экране. Программа, использованная на сайте Acropolis, выполняет все эти функции одновременно (см. листинг 5.5).

Листинг 5.5. Проверка, изменение и размещение изображений

```
if (NS4 | | N56) {
   avail_width = innerWidth;
   avail_height = innerHeight;
/* Определение новых доступных размеров экрана посетителя*/
/* при помощи значений innerWidth и innerHeight, поддерживаемых Netscape.*/
/* Они не поддерживаются Internet Explorer 3, 4 и 5.      */
   if (avail_Width < 640) && (avail_Height < 480)   {
/* Если размер экрана 640x480 или меньше, то загрузить уменьшенные
изображения, перечисленные ниже*/
      document['ZeusImg'].src = "images/zeussmall.gif";
      ZeusLyr.left = avail_width - 80;        //Разместить слой Zeus в
                                                80 пикс. от левого края
      ZeusLyr.top = avail_height - 80;        //Разместить слой Zeus на
                                                80 пикс. выше нижнего края
      document['GreekImg'].src = "images/greeksmall.gif";
      GreekLyr.left = 80;                     //Разместить контейнер в
                                                80 пикс. от левого края
      GreekLyr.top = 20;                      //Разместить контейнер на
                                               20 пикс. ниже верхнего края
      document['HermesImg'].src = "images/hermessmall.gif";
      HermesLyr.left = 25;                    //Разместить слой Hermes
                                                в 25 пикс. от левого края
      HermesLyr.top = avail_height - 80;      //Разместить слой Hermes на
                                                80 пикс. выше нижнего края
/* Это необходимо выполнить со всеми изображениями данной страницы. */
   }
   else {
      ZeusLyr.left = avail_width - 100;       //Разместить слой Zeus
                                                в 100 пикс. от левого края
      ZeusLyr.top = avail_height - 100;       //Разместить слой Zeus на
                                                100 пикс. выше нижнего края
      GreekLyr.left = 100;                    //Разместить контейнер
                                                в 100 пикс. от левого края
      GreekLyr.top = 20;                      //Разместить контейнер на
                                                20 пикс. ниже верхнего края
      HermesLyr.left = 25;                    //Разместить слой Hermes
                                                в 25 пикс. от левого края
      HermesLyr.top = avail_height - 100;     //Разместить слой Hermes на
                                                100 пикс. вверх от нижнего
                                                края
/* Каждый слой документа необходимо настроить аналогичным образом. */

   }
}

else (IE4 || IE5) {
   avail_width = document.body.clientWidth;
   avail_height = document.body.clientHeight;
/* Определить новые доступные размеры width и height экрана пользовате-
ля*/
/* при помощи поддерживаемых Microsoft значений clientWidth и clientHeight. */
/* Эти значения не поддерживаются Netscape Navigator 3, 4 и 5. */

   if (avail_Width < 640) && (avail_Height < 480)   {
/* Если экран имеет разрешение 640x480 или меньше, то загрузить уменьшен-
ные изображения, представленные ниже */
      document['ZeusImg'].src = "images/zeussmall.gif";
```

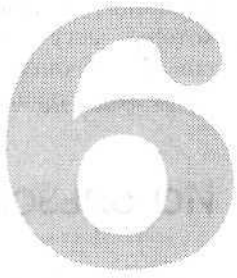
продолжение

Листинг 5.5 (*продолжение*)

```
        ZeusLyr.left = avail_width - 80;        //Разместить слой Zeus на
                                                  80 пикс. от левого края
        ZeusLyr.top = avail_height - 80;        //Разместить слой Zeus на
                                                  80 пикс. выше нижнего края
        document['GreekImg'].src = "images/greeksmall.gif";
        GreekLyr.left = 80;                     //Разместить контейнер на
                                                  80 пикс. от левого края
        GreekLyr.top = 20;                      //Разместить контейнер на
                                                  20 пикс. ниже верхнего края
        document['HermesImg'].src = "images/hermessmall.gif";
        HermesLyr.left = 25;                    //Разместить слой Hermes
                                                  25 пикс. от левого края
        HermesLyr.top = avail_height - 80;      //Разместить слой Hermes на
                                                  80 пикс. вверх от нижнего
                                                  края
        /* Так необходимо поступить со всеми изображениями страницы. */
    }

    else {
    ZeusLyr.left = avail_width - 100;           //Разместить слой Zeus
                                                  в 100 пикс. от левого края
    ZeusLyr.top = avail_height - 100;           //Разместить слой Zeus
                                                  в 100 пикс. выше нижнего края
    GreekLyr.left = 100;                        //Разместить контейнер
                                                  в 100 пикс. от левого края
    GreekLyr.top = 20;                          //Разместить контейнер на
                                                  20 пикс. ниже верхнего края
    HermesLyr.left = 25;                        //Разместить слой Hermes
                                                  в 25 пикс. от края
    HermesLyr.top = avail_height - 100;         //Разместить слой Hermes на
                                                  100 пикс. вверх от нижнего
                                                  края
/* Каждый слой этого документа необходимо настроить аналогичным образом. */

    }
}
```

После того как будут расставлены эти «сети безопасности», вы можете быть уверены, что при изменении пользователем программного обеспечения внешний вид вашего сайта не пострадает. (Конечно же, ничто не вечно под луной.)

Подготовка к войне браузеров



Теперь, после того как мы можем с помощью сценария определять тип браузера пользователя, необходимо «научить» сценарий взаимодействовать со специальными требованиями определенного браузера. Общий язык с различными браузерами можно найти несколькими способами. Первым решением, к которому прибегают все разработчики, является создание отдельных документов для каждого браузера. Очевидно, что данное решение не является наилучшим, поскольку требует огромных затрат от тех, кто поддерживает работу сайта. Необходимо создание копии сайта для пользователей браузера Netscape Navigator, для пользователей Internet Explorer, а также копии для пользователей других, не упомянутых здесь браузеров. Такой способ в значительной степени повышает затраты на содержание и обслуживание сайта.

Другим решением является нахождение «наименьшего общего знаменателя». Такой подход ограничивает возможности разработчика набором выражений JavaScript, элементов HTML и их атрибутов, а также свойств таблиц стилей, которые поддерживаются обоими основными браузерами и, может быть, такими другими браузерами, как Opera, и может существенно сократить творческий потенциал. К сожалению, именно этот метод и применяется в большинстве документов, существующих в Интернете. Он позволяет сократить затраты разработчиков и людей, обслуживающих сайты, но не позволяют web-разработчикам в полной мере реализовать свои знания и воображение. Однако если вы прибегнете к этому методу, то вам не надо будет обслуживать копии документа. Вам не придется заботиться о совершенстве сценариев. Вам не придется заботиться о сложных схемах дизайна. Вас будет интересовать только то, чтобы все ваши клиенты остались довольны сайтом.

Понятно, что ни один из этих вариантов не является оптимальным. Но существует и третий вариант. Фактически, я предлагаю подход, который заключается в следующем: необходимо создать простой API, позволяющий оперировать с большинством событий и процессов, встречающихся при разработке диалогового сайта, при необходимости осуществлять разветвление документа на дополнительные страницы и создавать страницы «на лету». Ключевым моментом применения этих механизмов является определение в начале каждого документа переменной, которая бы определяла используемый браузер и его версию. После

того как переменная определена, вы можете создавать дополнительные ветви страницы, создавая части документа «па лету», а также определяя переменные, которые могут использоваться в выражении eval() для создания частей сценария, обеспечивающих интерактивность. Давайте рассмотрим наиболее сложную часть этого процесса — создание API.

Использование DOM при создании API

У вас как у разработчика имеется множество способов для создания программного интерфейса (API), используемого всеми браузерами. Интерфейс может быть сложным, но для большинства документов достаточно простого интерфейса. Для того чтобы он был доступен всем документам сайта, файлы со сценариями JavaScript необходимо поместить так, чтобы они были доступны всем пользователям.

ПРИМЕЧАНИЕ

Единственной возможностью выполнить это требование является размещение всех файлов сценария в каталоге Scripts, который вы легко можете поместить в том же каталоге, в котором находятся документы вашего сайта. Когда у меня получается более двух сценариев, я обычно создаю каталог, позволяющий хранить их отдельно от документов и изображений.

Нижеприведенная часть сценария создает переменные, которые будут использованы в качестве основы программного интерфейса.

```
If (NS4 || NS6) {
   docs =  "document";
   styles = " ";
   html = ".document";
   xposition = "e.pageX";
   yposition = "e.pageY";
} else if (IE4 || IE6) {
   docs =  "document.all";
   styles = ".style";
   html = " ";
   xposition = "event.x";
   yposition = "event.y";
}
```

Определение этих переменных обеспечивает межплатформенный программный интерфейс, позволяющий использовать их всеми сценариями сайта. Этот сценарий необходимо разместить в элементе HEAD каждого документа вашего сайта при помощи тегов <script></script>.

```
<head>
<script type = "text/JavaScript">
...
</script>
</head>
```

После того как сценарии будут вставлены в отдельный документ или в ряд внешних файлов сценария, к которым обращается основная HTML-страница, вы можете использовать их для создания интерактивного сайта с буферизованными сценариями. Если вы сохраните все сценарии во внешних файлах, а затем будете загружать их с помощью отдельных команд <script>, то они будут сохра-

няться в кэш-памяти компьютера пользователя, сокращая общее время загрузки сайта. К внешним файлам сценария можно обращаться следующим образом:

```
<script lang = "JavaScript" src = "scriptfile.js">
</script>
```

После загрузки первой страницы, за счет использования связанных и *кэшированных страниц,* сайт будет загружаться довольно быстро. Кэшированные - это те страницы, которые могут храниться в памяти компьютера, просматривающего сайт. Они могут загружаться непосредственно с жесткого диска компьютера посетителя без необходимости повторной загрузки их из сети. Но первая страница, к сожалению, будет загружаться медленнее, так как она является документом, осуществляющим предварительную загрузку изображений и сценариев, используемых сайтом. Один из способов запуска предварительной загрузки изображений заключается в использовании «страницы-ловушки» («catch page»), представляющей небольшую аннотацию сайта, сведений о том, что здесь может найти пользователь, и указание предпочтительного браузера для просмотра данного сайта. Эта страница выполняет множество функций. Основная задача заключается в предварительной загрузке изображений и сценариев для сокращения общего времени загрузки остальных страниц сайта. Помимо этого, для пользователей браузеров версии старше 4.0 она выполняет роль информативной страницы, а также позволяет быстро заинтересовать посетителя и вызвать у него желание посетить остальную часть вашего сайта. А вы должны использовать эту страницу-заставку для автоматической загрузки сценариев и изображений, используемых на основной странице сайта (в главе 8 предварительная загрузка изображений рассмотрена более подробно).

ПРИМЕЧАНИЕ

Если страница-заставка будет загружаться очень долго, это может создать негативное впечатление у посетителей вашего сайта. Люди хотят немедленно получать информацию из сети. Они не должны долго ждать.

Настройка внутренних разветвлений страницы

Внутреннее разветвление страницы (Internalpage branching) — это понятие, связанное с созданием «на лету» содержания документа, которое зависит от браузера, используемого для просмотра документа. Этот вид разработки страниц объединяет документ и его сценарии в одно общее целое, которое делает неуместным разговор о том, что обслуживание сайта — это трудоемкая работа. Нижеприведенный сценарий является примером того, как содержание документа может изменяться в зависимости от типа браузера, используемого для его просмотра. В данном сценарии используются элемент <layer>, «понимаемый» только Netscape-браузерами, и элемент , используемый для создания слоев Microsoft-браузерами.

```
<SCRIPT type ="text/JavaScript">
<!--
var docOutput =" ";
if (IE4 ||IE5) {
```

```
    docOutput += "<DIV id='bodytext' style='top:75; width:300; left: 110; ";
    docOutput += "border:none; background-color: #999999; font_face: Arial;";
    docOutput += "font_color:blue;'>";
    docOutput += "<P align='center'><H1> Myth's For Sale .. Maybe </H1>";
  docOutput += "... Add more text here...";
} else if (NS4 || NS6){
    docOutput += "<LAYER id='bodytext' top=75 width=300 left=110";
    docOutput += " src='store.htm'>";
}
document.write (docOutput);
//-->
</SCRIPT>
```

В этом сценарии в переменной docOutput формируется содержание, которое будет размещено на странице. Команда `document.write` помещает это содержание на экран, то есть содержание будет сформировано и корректно воспроизведено любым браузером Netscape или Microsoft. Если содержание страницы формируется сценарием, то любая информация, содержащаяся в сценарии, будет загружена так же, как будто браузер прочитал обычный документ.

Создание таблиц стилей

После того как написан сценарий, представляющий программный интерфейс (API), необходимо создать таблицу стилей, определяющую внешний вид всего web-сайта. Таблица стилей может быть подключена к любой странице сайта, что приведет к ее кэшированию на стороне клиента и, следовательно, к ускорению загрузки. Приведенное ниже выражение `<link>` должно быть размещено в начале каждого документа. Это выражение осуществляет подключение документа к определенной таблице стиля, и при таком способе определения стилей посетителю придется загружать таблицу стилей лишь один раз.

```
<HEAD>
   <LINK rel="stylesheet" src="acropolis.css">
</HEAD>
```

После того как осуществлено внедрение таблицы стилей в основной HTML-документ, необходимо убедиться, что в ней используются лишь те свойства, которые поддерживаются как Netscape Navigator, так и Internet Explorer. Вот свойства и их допустимые значения, которые «работают» в обоих браузерах:

background: <background-image url>
clear: none
color: <color>
font: <font-family> | <font-style>
font-family: <family-name> <generic-family> | serif | monospace
font-size: <percentage>
font-weight: bold
line-height: normal
text-align: left right | center
text-decoration: line-through
text-indent: <length> | <percentage>

Таблица стилей (CSS-документ), представленная в листинге 6.1, была создана для web-сайта Acropolis (Акрополь). В ней используются большинство элементов стиля, используемых при оформлении сайта.

Листинг 6.1. Пример. Таблица стиля для web-сайта Acropolis

```
//Каскадная таблица стиля web-сайта Acropolis//
H1          { color: "red";
              font: "arial italic"; .
              font-size: "largest";
              text-align: "center";}
H2          { color: "red";
              font: "italic";
              font-family:"monospace";
              font-size: "largest";
              text-align: "left";
              text-indent: "30px";}
p:zeustxt { background: "red";
              border: "black";
              color: "black";
              font: "Verdana, normal";
              font-size: "larger";
              text-align: "left";}
p:hermestxt { background: "yellow";
                border: "blue";
                color: "black";
                font: "Times new Roman, normal";
                font-size: "smaller";
                text-align: "left";}
p:bldgtxt { background: "green";
              border: "blue";
              color: "black";
              font: "Arial, normal";
              font-size: "normal";
              text-align: "left";}
p:menutxt  { background: "black";
               border: "red";
               color: "white";
               font: "Times New Roman, normal";
               font-size: "larger";
               text-align: "left";}
```

Эта таблица стилей использует лишь свойства, поддерживаемые обоими браузерами. А в нижеприведенном списке представлены свойства, которые являются чрезвычайно «опасными» для использования в таблицах стилей. Если же вы все-таки хотите использовать определенное свойство с соответствующими ему значениями, то это необходимо делать лишь с помощью внутреннего разветвления страницы для соответствующего браузера. Вам также необходимо представлять, что они, в лучшем случае, могут работать с ошибками как в Netscape Navigator, так и в Internet Explorer, поскольку в каждом браузере собственные свойства реализуются различными способами.

#ID (идентификатор Id)
border: <border-width> | <border-style>
border-bottom: <border-bottom-width> | <border-style> | <color>
border-left: <border-left-width> | <border-style> | <color>
border-right: <border-right-width> <border-style> | <color>

```
border-top: <border-top-width> | <border-style> | <color>
display: inline | list-item
float: none | left | right
margin: <length>
margin-left: auto
margin-right: auto
padding: <length> | <percentage>
padding-bottom: <length> <pcrcentage>
padding-left: <length> | <percentage>
padding-right: <length> | <percentage>
padding-top: <length> | <percentage>
text-decoration: overline | underline | blink line-through none
vertical-align: bottom | text-bottom
white-space: nowrap
```

Использование графики

Изображения могут являться лицом web-сайта, а могут его полностью испортить. Хорошие изображения очень важны для зрительного восприятия сайта, но зачастую их большой размер может сделать сайт очень громоздким для посетителей. Для того чтобы размещенные на сайте изображения не вызывали раздражения посетителей, необходимо учесть следующее:

- **Обрежьте изображение.** За счет сокращения размеров вы сократите объем файла, а следовательно, и время его загрузки.
- **Графические элементы должны быть представлены в формате GIF.** GIF-формат позволяет хранить графические элементы в максимально сжатом виде, но с использованием палитры, состоящей из 256 или менее цветов.
- **Фотографии должны быть представлены в формате JPEG.** Этот формат обеспечивает максимальное сжатие с сохранением фотографического качества и палитры из 16 млн цветов.
- **Оптимизируйте палитру.** Сократите число используемых цветов до минимального значения, при котором обеспечивается удовлетворительная четкость и разрешение.
- **Обеспечьте показ изображений с малым разрешением, пока идет загрузка больших изображений.** За счет предварительной загрузки изображений и показа изображений с малым разрешением в ходе разгрузки больших изображений вы сможете удержать клиентов, поскольку ваши страницы будут грузиться быстрее. И в конечном итоге они получат требуемые изображения с высоким разрешением. Хотя общее время загрузки страницы возрастет, посетитель увидит изображение быстрее.
- **Используйте GIF-изображения с чересстрочным форматом.** Версия GIF89a позволяет сохранять файлы с использованием чересстрочной развертки. При использовании этого способа строчки, составляющие изображение, перемежа-

ются. При загрузке такого изображения браузер вначале показывает каждую 8-ю строчку, потом каждую 4-ю, каждую 2-ю и, наконец, представляет полное изображение. При этом посетитель вашей странички сможет понять, что же нарисовано на данной картинке, не дожидаясь ее полной загрузки, что очень удобно. Чересстрочное GIF-изображение содержит приблизительно десять снимков и изображение проявляется частями, а не сверху вниз.

- **Укажите атрибуты height и width элемента <img>.** Это позволит правильно отформатировать загружаемый текст, не ожидая загрузки изображения. Кроме того, это является требованием спецификации HTML 4, хотя невыполнение его все равно приведет к верному представлению документа.
- **Сократите количество анимации.** Максимально сократите использование анимации, чтобы она занимала как можно меньше места и не содержала статичных областей.
- **По возможности используйте пиктограммы изображений.** Использование маленьких, но четких пиктограмм вместо полномасштабных изображений позволит пользователям самим определять необходимость загрузки связанного с ней большого изображения.
- **Разбейте изображения на части.** Там, где возможно, разбейте изображение на несколько частей, которые можно использовать повторно. Повторным использованием изображений вы сможете сократить время, которое потребуется пользователю для загрузки повторяющихся изображений, особенно если в них сделаны лишь незначительные изменения. Только не слишком увлекайтесь этим, будьте благоразумны. Не стоит разбивать на множество частей отдельную черную линию!
- **Обязательно создавайте «альтернативный» текст.** Многие пользователи не желают загружать изображения, а хотят лишь видеть текст документа. С помощью атрибута `alt` элемента IMG вы можете назначить данному изображению текстовое описание. И хотя HTML 4 требует использования этого атрибута, браузеру его существование безразлично. Вы также можете создать исключительно текстовой документ, вообще не используя изображения. Текстовая версия документа должна загружаться с помощью сценария по желанию посетителя.

ПРИМЕЧАНИЕ

При помощи Internet Explorer 5 и Netscape Navigator 6 может воспроизводиться и векторная графика. Эти изображения должны быть представлены в формате масштабируемой векторной графики (Scalable Vector Graphics, SGV), основанной на использовании XML. Такую графику можно создавать с помощью текстовых инструкций. Эти изображения могут составлять значительный объем документа и имеют некоторые сложности в реализации свойства `align` (выравнивание). В настоящее время наилучшим способом создания SVG-изображений является использование графических программ, поддерживающих SVG-формат при прорисовке изображения, и последующее копирование SVG-команд в документ. Поскольку SVG не поддерживается Netscape 4.5 и более старыми версиями, а также Internet Explorer 4 и более старыми версиями, то для вывода SVG или загрузки стандартной графики, обеспечивающей одинаковое представление интерфейса для всех пользователей, необходимо использовать систему проверки используемого браузера. Подробности о SVG-графике можно узнать на сайте W3C по адресу: http://www.w3.org/Graphics/SVG/Overview.html.

Определение слоев

Последним предметом вашего беспокойства при подготовке основы сайта является проблема восприятия и отображения браузерами содержания слоев. Если вы плохо представляете себе структуру слоев, то постарайтесь воспринимать их как стопку стеклянных тарелок с рисунками или текстом на каждой из них. Таким образом, все содержание становится видно одновременно. Размер и содержание каждого отдельного слоя может отличаться. Один может содержать изображение, другой — текст, а третий — и изображение, и текст. До разработки трехмерной модели все web-сайты были «плоскими» и состояли из одного слоя, на котором размещались изображения и текст. Теперь используется «стопка» слоев, каждый из которых может содержать определенную информацию, которая может быть изменена в ходе просмотра. На рис. 6.1 показано, как осуществляется наложение слоев.

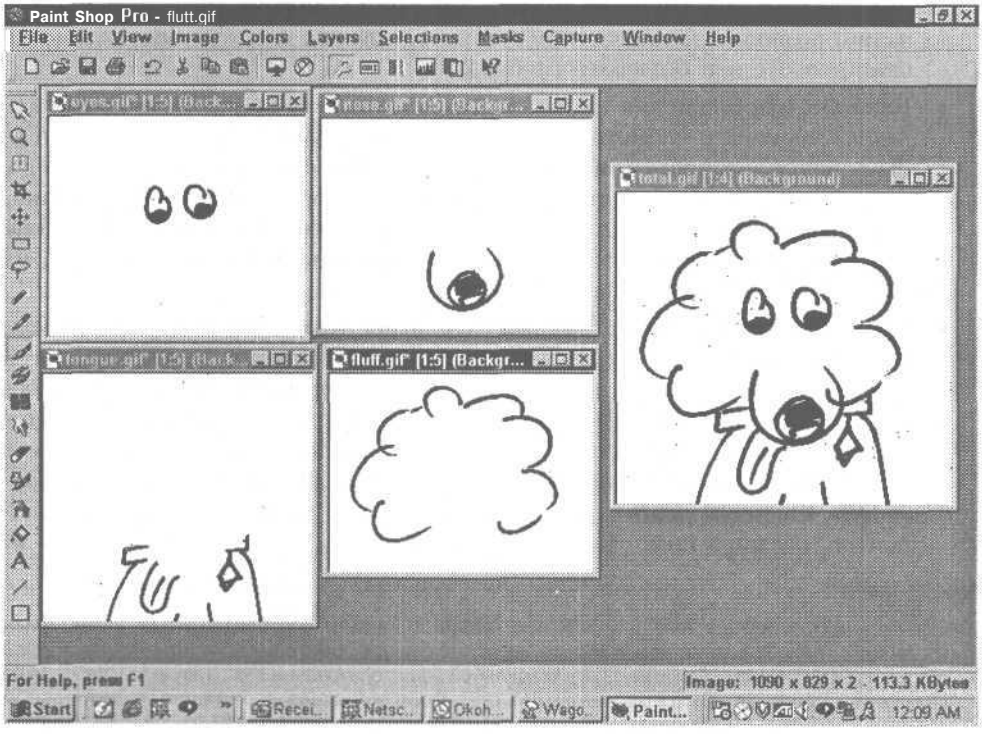


Рис. 6.1. Каждый слой этого изображения содержит часть изображения собаки. Данная технология позволяет вам «собрать собачку» из отдельных частей, помещаемых одна за другой на экране пользователя. Слои также можно использовать в игре, в которой собака будет представлена только в результате решения головоломки

Для создания web-сайта Acropolis потребовалось создать лишь несколько слоев. Давайте рассмотрим их более подробно.

- **Зевс на троне.** Этот слой содержит вращающееся изображение Зевса. Он также перемещается на своем месте.
- **Зевс: контур реплики.** Этот слой будет невидимым большую часть времени. Он просто содержит изображение контура реплики и текст, который изменяется в зависимости от положения указателя мыши.
- **Зевс: зигзаг молнии.** Этот слой проскальзывает по экрану, когда пользователь выберет в меню пункт «получить сведения о других богах или богинях». Этот слой будет невидимым при загрузке документа. Когда этот слой представится на экране, будет осуществлено воспроизведение небольшого аудиофайла, содержащего звук раската грома.
- **Символ Гермеса.** Крылья на его ногах будут двигаться. Остальное изображение останется неподвижным.
- **Гермес: контур реплики.** Когда указатель мыши будет помещен над Гермесом, появится контур реплики, на котором будет представлено сообщение, которое Гермес доставил Зевсу.
- **Меню.** Для корректного размещения и этот слой меню выполнен отдельно.
- **Колоннада.** Этот слой является декорацией к тексту, который будет изменяться в зависимости от выбранного пункта меню.
- **Текст, появляющийся на фоне колоннады.** Этот слой содержит текст, размещаемый между колоннами.
- **Сведения об авторских правах.** Этот текст изменяться не будет, но помещен в отдельный слой, для того чтобы облегчить его позиционирование на экранах с различным размером и разрешениями.

Теперь, когда определены все слои, необходимо разместить их в HTML-документе соответствующим образом, обеспечивая корректное воспроизведение при использовании различных браузеров. Текст документа представлен в листинге 6.2. Как уже упоминалось в главе 1, для идентификации слоев компания Netscape ввела свой элемент — <layer>, но браузеры этой компании могут воспринимать слои, создаваемые элементом . Согласно рекомендациям Интернет-консорциума по CSS-позиционированию корпорация Microsoft при создании слоев поддерживает только элементы и . Поэтому мы всегда будем использовать элемент в качестве контейнера для размещения содержание слоя.

Листинг 6.2. Определение слоев и их содержания

```
<HTML>
<HEAD>
  <SCRIPT  type="text/JavaScript">
If  (NS4 || NS6)  {
   docs =  "document" ;
```

продолжение

Листинг 6.2 *(продолжение)*

```
      styles = " ";
      html = ".document";
      xposition = "e.pageX";
      yposition = "e.pageY";
   } else if (IE4 || IE6) {
      docs =  "document.all";
      styles = ".style";
      html = " ";
      xposition = "event.x";
      yposition = "event.y";
   }
     </SCRIPT>
     <STYLE>
   //Каскадная таблица стиля для сайта Acropolis //
   H1        { color: "red";
               font: "arial italic";
               font-size: "largest";
               text-align: "center";}
   H2        { color: "red";
               font: "italic";
               font-family: "monospace";
               font-size: "largest";
               text-align: "left";
               text-indent: "30px";}
   p:zeustxt { background: "red";
               border: "black";
               color: "black";
               font: "Verdana, normal";
               font-size: "larger";
               text-align: "left";}
   p:hermestxt { background: "yellow";
                 border: "blue";
                 color: "black";
                 font: "Times new Roman, normal";
                 font-size: "smaller";
                 text-align: "left";}
   p:bldgtxt { background: "green";
               border: "blue";
               color: "black";
               font: "Arial, normal";
               font-size: "normal";
               text-align: "left";}
   p:menutxt { background: "black";
               border: "red";
               color: "white";
```

```
            font: "Times new Roman, normal";
            font-size: "larger";
            text-align: "left";}
  </STYLE>
<!-Здесь можно указать ссылки и на другие файлы сценария -->
</HEAD>
<BODY>
   <!-- Zeus in his chair -->
<DIV id="zeuslayer" style = "…">
   <IMG src="images/zeus.gif" alt="Zeus, father of the gods, seated on
   Mt. Olympus" width=100 height=100 lowsrc="zeussm.gif">
</DIV>
   <!-- Контур реплики Зевса -->
<DIV id="zeusbbllayer"
     style = "background: images/zbubble.gif; …"
     align = "center">
      I am Zeus! Be weary of my wrath.
</DIV>
   <!--Zeus's lightning bolt-->
<DIV id="zeusltbltlayer" style = "…">
   <IMG src="images/lightening.gif" alt="Temper! Temper! Temper!"
   width=30 height=100 lowsrc="lightening.gif">
</DIV>
   <!--Символ Гермеса -->
<DIV id="hermeslayer" style = "…">
   <IMG src="images/hermes.gif" alt="Hermes, the Messenger of the Gods."
   width=75 height=75 lowsrc="hermessm.gif">
</DIV>
   <!--Контур реплики Гермеса -->
<DIV id="hermesbbllayer"
     style = "background: images/hbubble.gif; …"
     align = "center">
      Zeus is such a blow hard!
</DIV>
   <!-- Меню -->
<DIV id="menulayer" style = "…">
</DIV>
   <!--The Greek building-->
<DIV id="greeklayer" style = "…">
   <IMG src="images/greek.gif" alt="Imagine a fabulous building sitting
on
   Mt. Olympus" width=400 height=350 lowsrc=greeksm.gif">
</DIV>
   <!-- Текст между колоннами -->
<DIV id="greektext"
     style = "…"
```

продолжение

Листинг 6.2 *(продолжение)*

```
        Here is the holding text for the informational screen
</DIV>
   <!-- Сведения об авторских правах -->
<DIV id="zeus" style = "…">
   <IMG src="images/zeus.gif" alt="Zeus, father of the gods, seated on
   Mt. Olympus" width=100 height=100 lowsrc="zeussm.gif">
</DIV>
</BODY>
</HEAD>
```

Подготовка к реализации взаимодействия

Еще на этапе разработки и планирования сайта закладываются определенные элементы взаимодействия. Иногда некоторые элементы HTML-кода имеют соответствующие отметки или представлены таким образом, что могут выполнять определенные интерактивные задачи. Несмотря на многократные напоминания, разработчики зачастую забывают указывать все глобальные переменные JavaScript, необходимые для функционирования отдельных частей сайта, хотя именно с их помощью обеспечивается взаимодействие или изменение сайта. Если эти переменные не определены, то самая необходимая функция работать не будет. Этап подготовки важен так же, как и собственно этап разработки сценариев. Используемые слои должны быть заранее определены, переменные — настроены, альтернативные документы — оформлены, а изображения — загружены, обеспечивая вывод интерфейсных элементов, представленных изображениями. В данной главе рассматриваются способы настройки HTML-кода, облегчающего организацию взаимодействия, а также определение глобальных переменных, которые понадобятся для запуска уже рассмотренных сценариев.

Установка переменных

В главе 5 рассматривался один из способов идентификации каждого слоя web-сайта для последующего написания сценария. Сейчас мы изучим код, позволяющий идентифицировать все слои с помощью функции `layerSetup`, вызываемой при загрузке документа.

Эта функция запускается событием `onLoad`, относящимся к элементу `<body>` HTML-документа или являющимся первым выражением сценария, позволяющего определить каждый слой при создании их web-браузером.

```
<body onload = "layerSetup(); . . . " . . . >
```

или

```
<script type="text/JavaScript">
   layersetup()
</script>
```

ПРИМЕЧАНИЕ

Если у вас множество сценариев и функций уже связано с событием onLoad, то, вероятно, вместо этого события необходимо использовать элемент <script> (второй вариант).

После вызова этой функций она последовательно определит каждый слой и установит его «вес» на основе предварительно заданных переменных, используемых при создании глобального программного интерфейса, поддерживаемого любым браузером. Каждое из нижеприведенных выражений используется для создания отдельной переменной, к которой могут обращаться все функции, независимо от типа используемого браузера. Это избавляет вас от постоянной необходимости повторно определять слои. Теперь добавить новые или удалить неиспользуемые переменные вы можете в одном месте, и теперь они не будут загромождать HTML-код при дальнейшем развитии вашего сайта. Функция layerSetup будет помещена в файл сценария, который будет доступен любому HTML-документу вашего сайта.

```
function layerSetup () {
   Zeuslyr =  eval(docs + '["zeuslayer"]' + styls);
   ZeusBblLyr =  eval(docs + '["zeusbbllayer"]' + styls);
   ZeusLghtLyr =  eval(docs + '["zeuslightening"]' + styls);
   HermesLyr =  eval(doc + '["hermeslayer"]' + styl);
   HermesBblLyr =  eval(doc + '["hermesbbllayer"]' + styl);
   MenuLyr = eval(doc + '["menulayer"]' + styl);
   Greeklyr = eval(doc + '["greeklayer"]' + styl);
   GreekTxtLyr = eval(doc + '["greektext"]' + styl);
   CopyLyr = eval(doc + '["copyright"]' + styl);
}
```

Доступ к этому файлу может быть осуществлен при помощи команды сценария, помещенной в начале каждого HTML-документа:

```
<script type="text/JavaScript" src="sitescripts.js">
```

Теперь, когда все представленные выше слои определены, имена их переменных и идентификаторы могут использоваться во всех остальных функциях сайта при обращении к любому из этих слоев.

Определение этих переменных может быть осуществлено глобально, как это сделано в предыдущем примере, и за счет использовании сценария в каждом документе для облегчения их чтения остальными функциями. Обратите особое внимание на то, что если вы помещаете идентификаторы слоев в начале вашего документа (см. представленный ниже пример), то их необходимо поместить в начале *каждого* документа вашего сайта. Имена переменных и идентификаторы могут быть размещены в документе с помощью самого первого элемента <script> на каждой странице:

```
<HTML>
<HEAD>
<SCRIPT type="text/JavaScript">
   ZeusLyr =  eval(doc + '["zeuslayer"]' + styls);
   ZeusBblLyr =  eval(doc + '["zeusbbllayer"]' + styls);
   ZeusLghtLyr =  eval(doc + '["zeuslightening"]' + styls);
```

```
    HermesLyr =  eval(doc + '["hermeslayer"]' + styl);
    HermesBblLyr =  eval(doc + '["hermesbbllayer"]' + styl);
    MenuLyr = eval(doc + '["menulayer"]' + styl);
    GreekLyr = eval(doc + '["greeklayer"]' + styl);
    GreekTxtLyr = eval(doc + '["greektext"]' + styl);
    CopyLyr = eval(doc + '["copyright"]' + styl);
</SCRIPT>
</HEAD>
<BODY>
.... Здесь размещается тело документа....
</BODY>
</HTML>
```

ПРИМЕЧАНИЕ

Во избежание повтора информации и, следовательно, уменьшения общего времени загрузки каждой страницы сайта необходимо определить эти переменные как глобальные величины в сценарии, размещенном на основной странице, а не в каждом отдельном HTML-документе. Единственный недостаток использования глобальных значений в сценариях: к ним сложнее обращаться из поддокументов слоя, фрейма или плавающего фрейма.

После того как вы осуществили предварительную загрузку ваших переменных, их смогут использовать все остальные функции. Теперь вам необходимо осуществить предварительную загрузку изображений, которые будут динамически заменяться при перемещении посетителя по сайту.

Предварительная загрузка изображений

Предварительная загрузка изображений требуется по целому ряду причин. Самая основная заключается в том, что присутствующие на сайте изображения должны загружаться быстро, чтобы посетителю не пришлось ожидать их загрузки при перемещении по сайту. Загрузка нескольких изображений на каждой последующей странице оптимизирует общее время загрузки сайта.

Еще одной причиной предварительной загрузки является необходимость динамической замены изображений на одной странице. Например, вы хотите, чтобы при помещении указателя мыши на изображении Зевса он «улыбнулся» или «встал» с трона. Может, вам захочется, чтобы он держал молнию на коленях, или просто изменить фон текста. Все это осуществляется за счет динамической замены изображений. Она может выполняться «гладко», только если изображение уже загружено в кэш-память пользователя. В противном случае замена будет осуществляться скачками и посетитель иногда будет вынужден созерцать пустое пространство.

Вы можете запустить предварительную загрузку изображений различными способами. Первый способ заключается в вызове функции событием onLoad элемента BODY, хотя это и не рекомендуется.

СОВЕТ

Активное использование события onLoad приведет к «запутыванию» функций и может привести к прерыванию выполнения других, более необходимых визуальных эффектов. Если это единственный процесс, выполняемый при загрузке документа, то это — очень хороший способ. В противном случае перед подключением к этому событию дополнительной функции все тщательно взвесьте.

```
<body onload = "preloadImages('images/zeus.gif','images/zbubble1.gif',
'images/hbubble1.gif', 'images/hermesshoe.gif', 'images/greek.gif',
'images/zlightening.gif');" ...>
```

При использовании события onLoad для предварительной загрузки изображений вы можете использовать нижеприведенную функцию, которая может вызываться из любого места документа. Она помогает избежать неразберихи, связанной с активным использованием события onLoad.

```
<script type="text/JavaScript">
preloadImages(''images/zeus.gif')','images/zbubble1.gif', 'images/
hbubble1.gif', 'images/hermesshoe.gif', 'images/greek.gif','images/
zlightening.gif');
</script>
```

Функция, производящая загрузку изображений в массив, очень проста:

```
<script type="text/JavaScript ">
<!--
function preloadImages() { /
  if (document.images) {
    var imgFiles = preloadImages.arguments;
    var preloadArray = new Array();
    for (var i=0; i<imgFiles.length; i++) {
      preloadArray[i] = new Image;
      preloadArray[i].src = imgFiles[i];
    }
  }
}
//-->
</script>
```

ПРИМЕЧАНИЕ

За счет глобального определения переменной preloadArray вы можете непосредственно обращаться к этим изображениям при помощи функции swapImage, рассмотренной в главе 9.

Первое выражение функции — if (document.images) -- проверяет наличие выражений image в документе. Если таковые имеются, то определяются две переменные. Первая — imgFiles — собирает имена, которые переданы функции при ее вызове. Вторая — preloadArray — создает новый массив, содержащий файлы, имена которых были переданы функции preloadImages.

Далее запускается цикл for, который считает все аргументы, переданные функции при ее вызове. Он задает в массиве пространство для размещения изображения и загружает в него указанный файл. Число повторений цикла определяется переменной цикла (i), которая каждый раз после загрузки очередного изображения увеличивается на единицу (i++), пока не достигнет значения числа изображений — imgFiles. При каждом цикле значение индекса массива также увеличивается на единицу. Этот индекс указывает на позицию в массиве, куда должно быть загружено изображение. В ходе этого процесса изображения загружаются в кэш-память компьютера посетителя и индексируются таким образом, что к ним можно обратиться по значению индекса с помощью функции swapImage (эта функция будет подробно рассмотрена в главе 9).

Другим аспектом, о котором вам необходимо позаботиться, являются события, относящиеся к изображениям-картам.

Перехват событий изображений-карт

Изображение-карта (image map) является по существу именно картой, на поверхности которой определены фигуры. Каждая фигура (область, часть изображения) используется в качестве ссылки на другие разделы данного документа или на другие страницы. В большинстве случаев изображение представлено «плоским» файлом формата GIF или JPEG, хотя существует возможность создать изображение-карту с использованием анимированного GIF-изображения. Каждый элемент изображения-карты содержит один или более элементов <area> или <a>, которые определяют геометрические области, являющиеся «активными точками» для обращения к другим документам.

Для обработки событий они должны быть перехвачены. При использовании Netscape Navigator необходимо перехватывать события всего слоя или элемента <a>. Этот браузер не может перехватить событие от элемента или <map>. Internet Explorer позволяет использовать события практически от любого элемента. Вы можете использовать события onMouseOver и onMouseOut элементов , <img> и <a>. Этот список можно продолжить.

Netscape требует, чтобы перехватывались все события объекта. Для выполнения этого требования необходимо написать функции по перехвату всех событий для каждого отдельного объекта. Это означает, что невозможно создать глобальный сценарий, способный перехватывать все события элемента и его содержания. Вам необходимо отдельно перехватывать события каждого объекта, входящего в элемент , и запускать специальную функцию для произошедшего события. Это единственный способ, с помощью которого вы можете запускать определенную функцию при наступлении события изображения, находящегося внутри элемента , и другую — при наступления события элемента <textarea>.

ПРИМЕЧАНИЕ

Если вашим объектом является элемент , определяющий слой, содержащий изображение, то вы будете перехватывать только события, относящиеся к тексту, находящемуся в данном слое.

В приведенном ниже листинге оператор условного перехода (if) сначала проверяет тип используемого браузера. Если используется Navigator, то для идентификации изображения-карты перехватываются события click, mouseover и mouseout. При возникновении этих событий вызывается функция. Например, если выражение Event.CLICK истинно, то будет вызвана функция dragIt. Если произойдет Event.MOUSEOVER, то — swapImage. Приведенный пример обычно помещается с помощью элемента <script> непосредственно перед загрузкой элемента <map>.

```
if (NS4 || NS6) {
  imageMapName.captureEvents(Event.CLICK | Event.MOUSEOVER | Event.MOUSEOUT);
  imageMapName.onClick = dragIt;
  imageMapName.onMouseOver = changeProp('zeusimg','1');
  imageMapName.onMouseOut = changeProp('zeusimg','2');
}
```

Как уже упоминалось, Internet Explorer при создании взаимодействия может использовать события, исходящие непосредственно от объектов <area> и <a> изображения-карты. Основным различием между этими элементами является нали-

чие у элемента <area> атрибута alt, позволяющего задавать альтернативное текстовое сообщение, которое может демонстрироваться вместо изображения. Кроме того, для элемента <area> не обязателен URL target, если использовался атрибут nohref. В составе элемента <a> обязательно должна присутствовать ссылка на URL ресурса.

ПРИМЕЧАНИЕ

Изображение-карта усложняет ориентирование на сайте посетителей, использующих браузер без воспроизведения изображений или речевые синтезаторы. При использовании в документе изображения-карты для облегчения навигации этих пользователей необходимо использовать отдельные ссылки.

При создании изображения-карты два атрибута элемента <img> являются предметом беспокойства, во всяком случае, когда они являются источниками взаимодействия. Это элементы ismap и usemap.

- ismap (ismap): определяет и реализует изображения-карты на стороне сервера.
- usemap (usemap=*uri*): связывает изображение-карту, создаваемую с помощью элемента MAP, и текущее изображение. Значение этого атрибута должно соответствовать значению атрибута name элемента MAP.

Когда используется любой из этих атрибутов изображения, то карта перекрывает исходное изображение. Это позволяет вам обращаться к карте и использовать события, связанные с областью карты, к которой относится данное событие.

```
<img name="imagemapname" src="imagemap.gif" width="800" height="600"
usemap="#interactivemap" alt="Карта изображения навигационной страницы.">
```

Сама по себе карта определяется атрибутом name, поэтому этот атрибут может относиться и к изображению. Можно также использовать атрибуты class и id, что облегчит идентификацию свойств таблицы стилей, относящейся к карге.

- class (class=*«cdata-list»*):назначает элементу имя класса. Браузер пользователя реализует классы в виде групп определенных типов.
- name или id (name=cdata): присваивает элементу <тар> имя, которое будет использоваться как идентификатор при автоматизации обработки событий карты или определения областей карты.

После определения свойств элемента <тар> необходимо обратиться к отдельным элементам <area> и <a>, определяющим координаты областей и их связи. Здесь могут оказаться полезными два атрибута — taborder и accesskey. Поскольку для перемещения между областями карты может использоваться клавиша Tab, необходимо также определить Tab-порядок.

СОВЕТ

Поскольку управлять картой изображения можно с помощью клавиши Tab, необходимо сначала активизировать события и связи.

- accesskey (accesskey=*«character»*): позволяет назначать «горячие» клавиши для обращения к ссылкам элементов <area> и <a>. Эти клавиши позволяют осуществлять быстрый переход непосредственно к нужному элементу.

При использовании элемента form это позволяет пользователю очень быстро вводить информацию, а при использовании элемента link — активизировать требуемую ссылку.

- tabindex (tabindex=*number*): задает позицию данного элемента в общем Tab-порядке документа:

Остальные атрибуты объектов <a> и <area>, на которые вы должны обратить внимание, представлены ниже. Для элемента shape обязательным является атрибут coords. При его отсутствии область не определена.

```
coords ( coords=x1,y1,x2,y2
 /** фигура - прямоугольник, x1y1 = левый верхний угол, x2y2 = нижний
правый угол**/
coords=x1,y1,rad
 /** фигура - окружность, xlyl = центр, rad = радиус **/
coords=x1,y1,x2,y2,... xn уп
 /** фигура - многоугольник , xlyl = первый угол, x2y2 = второй угол,
хпуп= последний угол **/ )
```

Этот пример определяет координаты, позволяющие расположить карту на изображении. Возможно использование любого сочетания фигур и координат. В идеальном случае активные области не должны налагаться друг на друга, хотя это и возможно. При наложении двух активных областей будет использована ссылка той, чьи координаты указаны в списке координат первыми.

После того как вы задали области на изображении-карте, можно приступить к назначению им событий. В листинге 7.1 представлена полная последовательность создания интерактивного изображения-карты. HTML-код использует различные события для вызова функции writeContent с заранее определенным текстом. Это позволяет создать изображение-карту, где текстовое содержание будет автоматически изменяться при перемещении мыши.

Листинг 7.1. Использование карты для «привязки» событий к определенным областям изображения

```
<map name="interactivemap">
  <area shape="poly" coords="531,324,534,328,551,411,526,504,701,502,703,
98,566,98,550,128,692,322,531,325"  href="bad_script.html"  onMouseOver  =
"writeContent("Демонстрация неудачных сценариев")" onMouseOut = "writeContent("")">
  <area shape="poly" coords="297,506,250,406,197,406,196,197,485,199,563,
98,101,98,99,502" href="bad_dom.html" onMouseOver = "writeContent("Демонстрация
неудачной реализации DOM")" onMouseOut = "writeContent("")">
  <area shape="poly" coords="439,260,547,130,688,320,522,322,502,299,483,
283,466,272,439,261"  href="Bad_css.htm"  onMouseOver  =
"writeContent("Демонстрация неудачной реализации CSS")" onMouseOut =
"writeContent("")">
  <area shape="circle" coords="401,400,138" href="Html_tip.htm"
onMouseOver = "writeContent("Демонстрация неудачной реализации подсказок
HTML")" onMouseOut = "writeContent("")">
  <area shape="poly"
coords="199,197,403,197,404,258,354,263,301,294,267,347,254,387,254,404,199,402,199,198
nohref onMouseOver = "writeContent("Демонстрация неудачного html-кода")"
onMouseOut = "writeContent("")">
</map>
```

Этот HTML-код (результат выполнения которого представлен на рис. 7.1) демонстрирует использование нижеприведенных атрибутов.

- href (href="url"): указывает местонахождение документа или источника, к которому относится ссылка.
- name (name=cdata): определяет имя элемента <map>, которое может использоваться как идентификатор при автоматизации обработки событий карты или определения областей карты.
- Nohref: свидетельствует об отсутствии URL, определяющего ссылку. Этот атрибут используется не часто, но его можно использовать для обработки события, которое не требовало бы загрузки нового документа.
- shape (shape= default | circle | rect | poly |): определяет форму области, помещаемой на карте. По умолчанию используется все изображение. Для определения прямоугольника используются лишь две точки (левый верхний и правый нижний углы), для окружности — координаты центра и радиуса, а для многоугольника необходимо задать координаты всех углов.

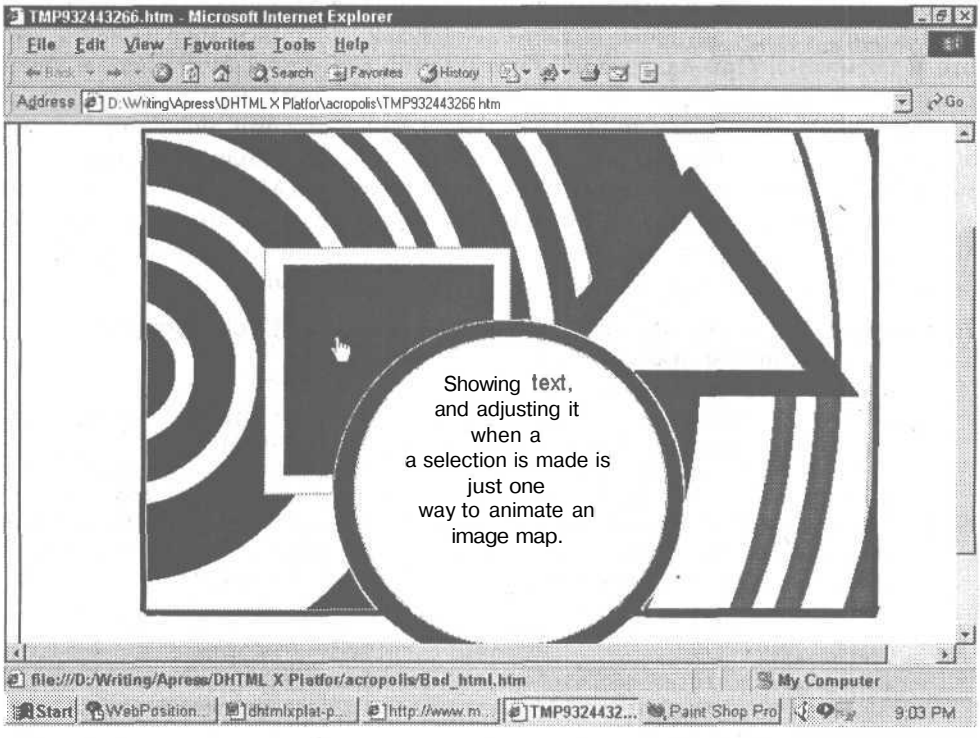


Рис. 7.1. При помещении указателя мыши над различными объектами карты изображения будет происходить изменение текста в окружности. Окружность — это просто пустой прозрачный слой, на который помещается и удаляется текст в зависимости от положения указателя мыши

Использование форм

Формы — это еще одна область наиболее частого использования сценариев. С помощью JavaScript можно изменять содержание открывающихся списков и текстовых полей в зависимости от изменений, выполненных пользователем в других полях формы. Это позволяет создавать динамические формы, отвечающие широкому спектру запросов, не заставляя пользователя отвечать на те вопросы, которые не представляют для него интереса. Сценарии могут использоваться для автоматического выполнения математических действий с содержанием форм или для проверки их содержания на корректность заполнения.

Сценарий, представленный в листинге 7.2, использует язык JavaScript/JScript для изменения содержимого полей в зависимости от выбора, осуществленного в других полях формы. Этот пример позаимствован с web-сайта «JavaScript Cut-N-Paste», обслуживаемого компанией Infohiway. Этот сайт предлагает множество бесплатный сценариев как для личных, так и для коммерческих сайтов. В обмен Infohiway требует отражать в HTML-коде сайта следующую авторскую информацию (при этом она не обязательно должна выводиться на экран):

```
<!-- Cut-N-Paste JavaScript от I5N Toolbox
   Copyright 1998, Infohiway, Inc. Ограниченное использование,
   которое заключается в разрешении использования как на
   коммерческих, так и на личныйх сайтах с условием, что этот код
   не будет являться предметом торговли и внутри HTML-документа
   будет сохранена авторская информация. Мы также будем
   благодарны за размещение ссылки на наш сайт:
   http://www.infohiway.com
   но это не является обязательным условием ;-)    -->

<!-- Если у вас возникли проблемы при модификации данного сценария
   под ваши требования, то высылайте ваши вопросы на форум
   •Self Help Forum Discussion:
   http://www.infohiway.com/javascript/self_help/main.cgi?script=seasons .
-->
```

Сценарий, представленный в листинге 7.2, использует в качестве примера изменения содержания полей формы, осуществляемого с помощью сценария, различные «сферы деятельности» и имена богов для каждого времени года. Этот сценарий легко подстроить под любую конкретную задачу. Он встраивается непосредственно в HTML-код. Форма вызывает сценарий всякий раз при изменении выделения в одном из полей формы. В сценарий включены элементы проверки поддержки сценариев используемым браузером (с оповещением об этом пользователя).

В первых строках сценария производится определение всех объектов и массивов, используемых в сценарии. Условный оператор в начале сценария проверяет версию браузера по поддержке объекта `document.images`. После минимальной проверки версии содержание выражения оператора `if` создает объекты activities («деятельность») и gods (боги). В каждом из этих объектов сценарий создает новые массивы — spring (весна), summer (лето), `fall` (осень) и winter (зима), содержащие списки «ответственности» каждого из перечисленных в соответствую-

щем массиве богов. Содержание этих массивов используется для заполнения открывающихся списков activities и gods в форме, создаваемой HTML-кодом, представленным в листинге 7.3.

Листинг 7.2. Определение массивов, в которых будет помещено содержание для открывающихся списков формы

```
<html>
<head>
   <title>Ваша web-страничка, представляющая жителей Олимпа</title>
</head>
<body bgcolor=white>
<script language="JavaScript">
<!--
if (document.images) {
 activities = new Object0;
 activities.spring = new Array("Love","Lust","Fertility") ;
 activities.summer = new Array("Friendship","War","Harvesting");
 activities.fall = new Array("Hunting","Sailing","Drinking");
 activities.winter = new Array("Death","Starvation","War");
 gods = new Object0;
 gods.spring = new Array("Cupid","Apollo","Aphrodite") ;
 gods.summer = new Array("Hermes","Hera","Mars");
 gods.fall = new Array("Aries","Diana","Baccus");
 gods.winter = new Array("Strife","Hades","Mars");
}
<!-- Сценарий не завершен... пока продолжим чтение
-->
```

В листинге 7.3 видно, что в функцию updateMenus передается лишь одно значение переменной season, которое используется для определения используемого массива для заполнения формы. Если в этой функции выражение document.images окажется истинным, то это будет означать, что мы имеем дело с браузером версии 4.x, и значение индекса выбранного времени года будет назначено переменной sel. Если же выражение document.images окажется ложным, то сценарий выдаст сообщение, информирующее пользователя, что его устаревший браузер не поддерживает функции сценария. В зависимости от того, что будет выбрано – Spring (1), Summer (2), Fall (3) или Winter (4), — будет использовано соответствующее выражение if, и в переменные act и god будут помещены соответствующие «виды деятельности» и боги. Если будут выбраны другие значения, то информация в act и god будет сброшена. После этой последовательности операторов if будет осуществлено измерение длины значений переменных act и god или подсчитано количество элементов каждой переменной. После этого отдельные элементы массивов должны быть определены как отдельный элемент индекса и помещены в открывающиеся списки формы.

Листинг 7.3. Заполнение полей формы

```
function updateMenus(season) {
 if (document.images) {
  sel = season.selectedIndex;
  if (sel == 1) {
   act = activities.spring;
```

```
    god = gods.spring;
   } else if (sel == 2) {
    act = activities.summer;
    god = gods.summer;
   } else if (sel == 3) {
    act = activities.fall;
    god = gods.fall;
   } else if (sel == 4) {
    act = activities.winter;
    god = gods.winter;
   } else {
    act = new Array();
    god = new Array();
   }
   season.form.activities.length = act.length;
   for(i=0;i<act.length;i++)
    season.form.activities.options[i].text = act[i];
   season.form.gods.length = god.length;
   for(i=0;i<god.length;i++)
    season.form.gods.options[i].text = god[i];
  } else {
   alert("Ваш браузер не позволяет изменить списки activities "
   +"и gods. Установите более современный браузер.");
  }
}
// -->
</script>

<!-- Сценарий не завершен... пока продолжим чтение
-->
```

При завершении выполнения сценария в документе будет создана форма. В ней будет три поля со списками. Первое будет называться Время года (season), в нем будет список из пяти пунктов «Выберите время года (0)», «Весна (1)», «Лето (2)», «Осень (3)» и «Зима (4)». Остальные два поля будут пустыми, но при выборе посетителем каких-либо данных произойдет изменение содержания и в этих полях. Изменение значения лишь поля seasons ни к чему не приведет. Его значения выбираются при помощи функции updateMenus всякий раз при изменении содержания выражения season <select> ... <option>, что видно из HTML-кода, создающего эту форму (см, листинг 7.4).

Листинг 7.4. Создание формы на языке HTML

```
<form>
<select name="season" size=1 onChange="updateMenus(this)">
<option>Choose Season
<option>Spring
<option>Summer
<option>Fall
<option>Winter
</select><br><br>

Activities:<br>
<select name="activities" size=1>
```

продолжение

Листинг 7.4 *(продолжение)*

```
<option>                 
</select><br><br>

Gods:<br>
<select name="gods" size=1>
<option>                 
</select><br><br>
</form>
</body>
</html>
<!-- OK... наконец мы достигли конца документа -->
```

Экспериментируя с этим сценарием, можно легко увеличить число массивов или создать многомерные массивы, которые будут обеспечивать более подробные сведения в соответствии с последовательностью выбора других значений. Данная структура может использоваться вами в виде ряда полей, которые могут представлять навигационную систему сайта, или даже в виде теста с рядом многоуровневых ответов. Например, вы можете изменить данный сценарий таким образом, чтобы он выполнял роль меню сайта, позволяя после выбора основной темы поиска «Техническая поддержка» получить список продуктов, для которых можно получить поддержку.

Если же вы хотите сделать этот сценарий более интересным и полезным, то вам необходимо создать простую «витрину» (форму, заполняемую при покупке товара в Интернет-магазине), добавив в форму простейшие функции калькулятора. Нижеприведенный сценарий складывает значения, введенные в первых двух полях, а результат представляет в третьем. Переменной Firstnum присваивается значение текстового поля num1 формы form1. Secondnum — значение текстового поля num2. После чего они складываются и полученная сумма заносится в переменную total, значение которой затем помещается в тестовое поле sum формы form1.

```
<script languge="JavaScript">
  function addValues() {
       firstnum=document.form1.num1.value;
       secondnum=document.form1.num2.value;
       total=firstnum + secondnum;
       document.form1.sum.value=total;
  }
</script>
```

Этот пример сценария может быть вставлен в существующий документ, а также может быть помещен в отдельный файл и вызываться по мере необходимости. HTML-код, создающий форму, используемую в данном примере, приведен ниже. В нем имеется три текстовых ноля, содержащих цифровые значения. При изменении любого из значений производится вызов функции addValues, осуществляющей сложение. Если вы хотите использовать этот сценарий с предыдущим сценарием выбора меню, представленным в листингах 7.2, 7.3 и 7.4, то необходимо будет включить дополнительный сценарий, который будет содержать цифровые значения для каждого бога и его деятельности. Индексные значения этого массива могут быть помещены в дополнительные поля вашего документа.

```
<html>
<head>
<title>Проверка работы сценария в форме</title>
<meta http-equiv="Content-Type" content="text/html; charset=Windows-1251">
</head>

<body bgcolor="#FFFFFF">
<form name="form1" method="post" action="">
  <p>
    <input type="text" name="num1" onChange="addValues()" value="0">
    +
    <input type="text" name="num2" onChange="addValues()" value="0">
    =
    <input type="text" name="sum" value="0">
</form>
</body>
</html>
```

Вы можете использовать представленную последовательность сценариев этой главы для создания собственных «витрин». И хотя сейчас абсолютно бесплатно доступно множество CGI-сценариев, зачастую проще создать собственную небольшую JavaScript-«витрину», которую вы всегда можете изменить на свой вкус.

ПРИМЕЧАНИЕ

Если вы решили осуществлять математические операции в полях формы, то лучше самим вставить все допустимые цифровые значения, чем позволить пользователю вводить их. Дело в том, что сценарий будет работать только с цифрами, а пользователь может по ошибке ввести буквенные значения. Если все-таки необходимо разрешить пользователю вводить данные, то следует проверять, действительно ли введенные значения являются числами. Это можно сделать с помощью функции Isnum, представленной в листинге *7.5*.

Листинг *7.5.* Преобразование символьного представления в числовое

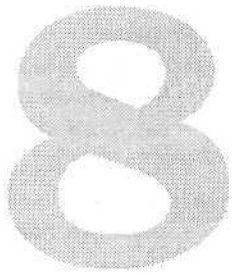
```
function IsNum( numstr ) {
   // Немедленно выйти, если введено неправильное значение
   if (numstr+"" == "undefined" || numstr+"" == "null" || numstr+"" ==
"")
      return false;
   var isValid = true;
   var decCount = 0;      // число десятичных точек в строке

   // преобразовать строку и выполнить сравнение.
   numstr += " ";

   // С помощью цикла организовать проверку каждого символа строки. Если
хоть
   // один из символов не является числом, то вернуть результат false.
   // Допускаются отрицательные числа (первый символ == '-')
   // и десятичная точка (один из символов == '.').
   for (i = 0; i < numstr.length; i++) {
      // отслеживать число десятичных точек в строке
      if (numstr.charAt(i) == ".")
         decCount++;
```

продолжение

Листинг 7.5 *(продолжение)*

```
      if (!((numstr.charAt(i) >= "0") && (numstr.charAt(i) <= "9") ||
              (numstr.charAt(i) == "-") || (numstr.charAt(i) == "."))) {
            isValid = false;
            break;
         } else if ((numstr.charAt(i) == "-" && i != 0) ||
              (numstr.charAt(i) == "." && numstr.length == 1) ||
             (numstr.charAt(i) == "." && decCount > 1)) {
            isValid = false;
            break;
         }
   //if (!((numstr.charAt(i) >= "0") && (numstr.charAt(i) <= "9")) ||
      } // END for
      return isValid;
   }  // end IsNum
```

Реализация взаимодействия



В этой книге уже обращалось ваше внимание на то, что любое взаимодействие страницы с пользователем инициируется с помощью какого-либо катализатора, исходного толчка. В качестве такого катализатора может выступать даже таймер, но в большинстве случаев в качестве инициатора взаимодействия выступает, как правило, сам пользователь. Посетитель может обращаться к сайту посредством нажатия клавиши или перемещения мыши (или другого указывающего устройства). А побуждаться к взаимодействию пользователь может соответствующим расположением изображений, совокупностью поведения элементов сценария, организацией меню и общим видом сайта.

Взаимодействие клиента с сайтом реализуется путем обработки всевозможных событий. Ими могут быть: загрузка и выгрузка объектов или документов, изменение размеров страницы, события мыши и события клавиатуры. Каждое из них может инициировать множество ответных действий различных типов. А можно сделать так, чтобы множество различных событий приводили к выполнению одной функции с одинаковыми или разными параметрами.

Использование событий мыши

Организация взаимодействия с пользователем при помощи событий мыши – это, пожалуй, самый простой из всех способов, которые могут встретиться на сайте. Графическая среда автоматически заставляет большинство пользователей использовать мышь при выполнении какой-либо задачи. Как вы полагаете, сколько «опытных пользователей» при обращении к пунктам меню или кнопке в панели задач воспользуется мышью вместо использования «горячих клавиш»? (Скорее всего, все.)

Перемещение мыши

Для запуска функций сценария, связанных с перемещением мыши, существует три события: `onMouseMove`, `onMouseOut` и `onMouseOver`.

onMouseMove

Основное назначение этого события — отслеживать движение мыши по документу или объекту.

Синтаксис этого события на языке HTML выглядит следующим образом:

```
onMouseMove=script
```

Для непосредственного запуска события из JavaScript, необходимо придерживаться следующего синтаксиса:

```
object.onMouseMove = handler
```

Пример, представленный в листинге 8.1, позволяет использовать событие onMouseMove для показа наименования каждой ячейки или столбца, над которыми перемещается мышь. Результат работы данного примера показан на рис. 8.1. Internet Explorer поддерживает привязку события onMouseMove непосредственно к таблице, в то время как Netscape требует, чтобы был осуществлен перехват всех событий, связанных с элементом, использующим функции сценария. В табл. 8.1 представлены элементы, которые могут перехватывать событие onMouseMove в браузерах компании Netscape. (Эта особенность уже обсуждалась в главе 7.)

Листинг 8.1. Установка событий, перехватывающих перемещение мыши

```
<html>
<TITLE>Пример использования события onMouseMove </TITLE>
<body>
  <table border = "1">
    <tr>
      <td name="r1c1"
          onmousemove="alert(event.srcElement.name)">
         <a href = "#">
<!-- символ # создает пустую, но действительную ссылку.
Она просто осуществит переход к началу web-страницы -->
         <img src="redbutton.gif" name=redbutton>
            Щелкните здесь
         </a>
      </td>
    </tr>
<tr>
      <td name="r2c1"
          onmousemove="alert(event.srcElement.name)">
         <a href = "#">
         <img src="redbutton.gif" name=redbutton>
         </a>
      </td>
    </tr>
<tr>
      <td name="r3c1"
          onmousemove="alert(event.srcElement.name)">
         <a href = "#">
            Щелкните здесь
         </a>
      </td>
    </tr>
  </table>
</body>
</html>
```

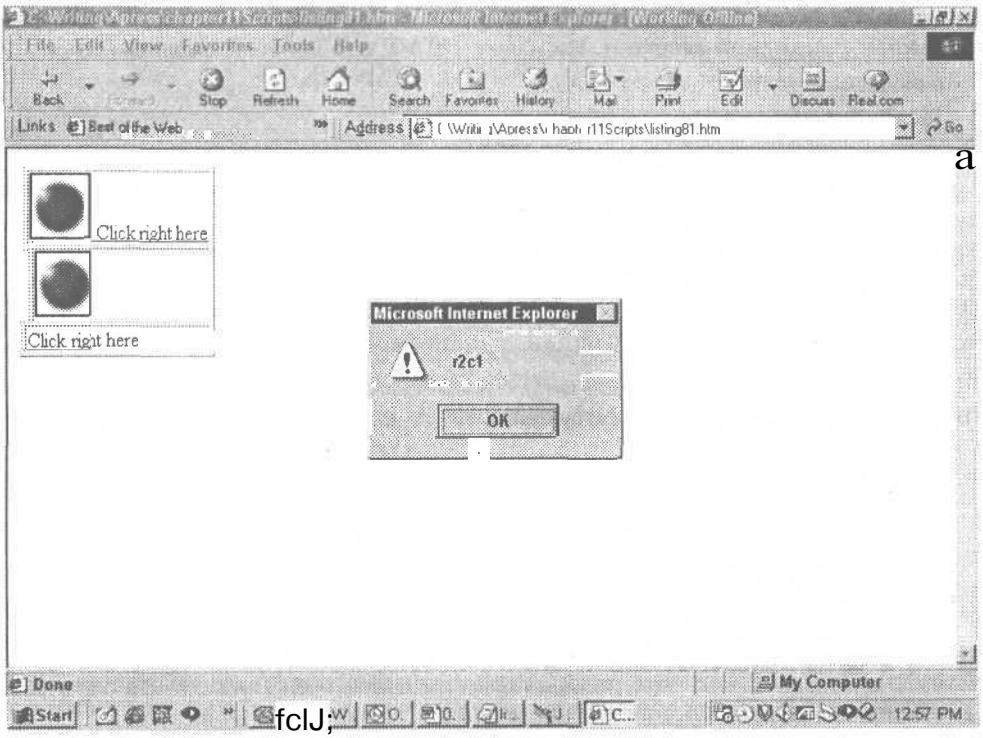


Рис. 8.1. Результат выполнения листинга 8.1 — окно с сообщением, указывающее идентификатор (id) текстового поля, над которым перемещался указатель мыши

Таблица 8.1. Элементы, поддерживающие в Netscape-браузере перехват события onMouseMove

Объект	Navigator 3	Navigator 4	Navigator 6
window	Нет	Да	Да
document	Нет	Да	Да
layer	Нет	Нет	Да, для элемента
link: <a>	Нет	Нет	Да
area	Нет	Нет	Да
image	Нет	Нет	Да
form	Нет	Нет	Да
text, textarea, password	Нет	Нет	Нет
all buttons	Нет	Нет	Нет
select	Нет	Нет	Нет

onMouseOut

Это событие отслеживает выход указывающего устройства за границы текущего элемента. Например, указатель мыши находился над неиспользованной ссыл-

кой, а затем был переведен на другую ссылку. Произойдет событие onMouseOut, связанное с исходной ссылкой. На языке HTML это событие выглядит как:

```
onMouseOut=script
```

JavaScript-синтаксис:

```
object.onMouseOut = handler
```

В листинге 8.2 представлен пример, использующий событие onMouseOut. При перемещении мыши за пределы изображения об этом будет выдано сообщение. Результат представлен на рис. 8.2. Internet Explorer поддерживает привязку события onMouseOut непосредственно к изображению, в то время как Netscape требует, чтобы был осуществлен перехват всех событий, связанных с элементом, использующим функции сценария, или привязка события onMouseOut к области, окружающей изображение.

Листинг 8.2. События мыши могут использоваться для определения имен объектов

```
<HTML>
<HEAD>
<TITLE>Пример использования события onmouseout</TITLE>
</HEAD>
<BODY>
<SCRIPT Language="JavaScript">
  if (document.all) {
     document.write('<DIV name=layer1 style="position:absolute; top:20;
left:20; width:300; height:300" onmouseout="alert(event.srcElement.name)">');
  } else if (document.layers) {
     document.write('<LAYER name=layer1 position=absolute top=20 left=20
width=300 height=300 onMouseOut="alert(this.name)">');
  }
</SCRIPT>

<TABLE name="OurTable"
       border=1>
  <TH name="Table Header">Заголовок таблицы</TH>
  <TR name="Row 1">
    <TD name="R1C1">
      <BUTTON name=redbutton>R1C1</BUTTON>
    </TD>
  </TR>
  <TR name="Row 2">
    <TD name="R2C1">
      <INPUT name=clicktext type=text value="R2C1">
    </TD>
  </TR>
  <TR name="Row 3">
    <TD name="R3C1">
      R3C1
    </TD>
  </TR>
</TABLE>
<SCRIPT Language="JavaScript">
  if (document.all) {
     document.write('</DIV>');
  } else if (document.layers) {
     document.write('</LAYER>');
  }
</SCRIPT>
</BODY>
</HTML>
```

Рис. 8.2. Результат выполнения листинга 8.2 при перемещении мыши в пределах таблицы

В листинге 8.2 также представлен пример использования функции, построенной на событии onMouseOut, позволяющей перехватывать события, связанные со слоями как в браузерах Netscape, так и Microsoft. Эта функция может выдавать сообщения, позволяющие определить пользователю наименование объекта, с которого был убран указатель мыши.

ПРИМЕЧАНИЕ

Этот сценарий Netscape Navigator и Internet Explorer обрабатывают по-разному. Они оба перехватывают событие, но Navigator показывает имя слоя, только когда мышь уходит с него. Explorer показывает имена всех объектов. Так происходит потому, что Navigator отслеживает только события, связанные с объектом «основной слой», и игнорирует события, происходящие с его наследниками. Это делает возможность использования этого типа событий в Netscape очень сомнительной.

В табл. 8.2 представлены элементы, которые могут перехватывать событие onMouseOut в браузере компании Netscape.

Таблица 8.2. Элементы, поддерживающие перехват события onMouseOut в браузерах компании Netscape

Объект	Navigator 3	Navigator 4	Navigator 6
window	Нет	Нет	Нет
layer	Нет	Да, для элемента <layer>	Да, для элемента

продолжение

Таблица 8.2 *(продолжение)*

Объект	Navigator 3	Navigator 4	Navigator 6
ссылка: <a>	Да	Да	Да
area	Да	Да	Да
image	Нет	Нет	Нет
form	Нет	Нет	Да
text, textarea, password	Нет	Нет	Нет
all buttons	Нет	Нет	Нет
select	Нет	Нет	Нет
fileUpload	Нет	Нет	Нет

Из-за различной реализации Netscape Navigator и Internet Explorer я вынуждена прибегнуть к практике, которой обычно стараюсь избегать. Браузер Netscape осуществляет перехват событий только объектов «истинного» слоя, поэтому необходимо написать короткий сценарий, представленный в листинге 8.3, в котором для перехвата событий мыши в разных браузерах используются как элемент <layer>, так и элемент . Подобный вариант написания сценария очень громоздок и сложен для обслуживания, поэтому его следует по возможности избегать.

onMouseOver

Это событие возникает тогда, когда указатель мыши окажется над данным элементом. Например, указатель мыши находился над свободной частью документа, затем был перемещен на изображение или ссылку. При этом инициируется событие_onMouseOver, связанное с данным изображением или ссылкой. Событие наступает до того, как мышь покинет объект. HTML-синтаксис:

```
onMouseOver=script
```

JavaScript-синтаксис:

```
object.onMouseOver = handler
```

Сценарий, представленный в листинге 8.3, использует событие onMouseOver при выводе имени каждой ячейки, над которой будет помещен указатель мыши. Internet Explorer поддерживает привязку события onMouseOver непосредственно к таблице, в то время как Netscape требует, чтобы был осуществлен перехват всех событий, связанных с элементом, использующим функции сценария. В табл. 8.3 представлены объекты, которые могут перехватывать событие onMouseOver в браузере компании Netscape.

Таблица 8.3. Элементы, поддерживающие перехват события onMouseOver в браузерах компании Netscape

Объект	Navigator 3	Navigator 4	Navigator 6
window	Нет	Нет	Нет
layer	Нет	Да — при использовании <layer>	Да — при использовании
link:<a>	Да	Да	Да
area	Да	Да	Да

Объект	Navigator 3	Navigator 4	Navigator 6
image	Нет	Нет	Да
form	Нет	Нет	Да
text, textarea, password	Нет	Нет	Нет
all buttons	Нет	Нет	Нет
select	Нет	Нет	Нет
fileUpload	Нет	Нет	Нет

В программе, представленной в листинге 8.3, каждый раз при перемещении мыши над одной из ссылок ее наименование или текст будут помещаться в окно элемента <textarea>, расположенное ниже. Результат выполнения этого примера представлен на рис. 8.3.

ПРИМЕЧАНИЕ

Internet Explorer поддерживает использование свойства name элемента <a>, а Netscape Navigator — нет. Если для обоих браузеров используется свойство name, то при просмотре примера браузером Netscape Navigator окно элемента <textarea> заполнится последовательностью выражений «undefined» (не определено). Для того чтобы этого не случилось, в сценарий вставлен текст ссылки.

Листинг 8.3. Использование событий для добавления сведений в окно элемента <textarea>

```
<HTML>
<HEAD>
<TITLE>Пример использования события onMouseOver</TITLE>
<script language="JavaScript">
<!--
function showus(obj) {
// Для браузеров IE
  if (document.all) {
    mytext=obj.name;
   //Берет содержание текстовго окна textarea, добавляет в него
   //ссылку и возвращает новый текст
thetextarea.value=thetextarea.value+"\r\n"+mytext;

// Для браузеров Netscape
  } else if (document.layers) {
    mytext=obj;
   // Устанавливает значение области textarea документа
  // и добавляет текст при помощи события MouseOver
document.form1.thetextarea.value=document.form1.thetextarea.value+"\r\n"+mytext;}
}
// -->
</script>
</HEAD>

<BODY>
<!-- Создает ссылку для события MouseOver  -->
<P><A href="#" onmouseover="showus(this)" name="onMouseOver Link">Ссылка
onHouseOver </A>
<!-- Создает ссылку для события  MouseOut -->
<P><A href="#" onmouseover="showus(this)" name="onMouseOut Link">Ссылка
onMouseOut </A>
```

продолжение

Листинг 8.3 *(продолжение)*

```
<!-- Создает ссылку для события mouseDown  -->
<P><A href="#" onmouseover="showus(this)" name="onMouseDown Link">Ссылка
onMouseDown</A>
<!-- Создает ссылку для события  mouseUp  -->
<P><A href="#" onmouseover="showus(this)" name="onMouseUp Link"> Ссылка
OnMouseUp</A>
<!-- Сценарий записывает в окно textarea при использованн IE,
// или в окно textarea, расположенное в форме, при использовании браузера
Netscape. -->
<SCRIPT Language="JavaScript">
  if (document.all) {
     document.write('<P><TEXTAREA name="thetextarea" rows="10"
cols="20">Перемещение было призведено над элементами: </TEXTAREA>');
  } else if (document.layers) {
     document.write('<FORM name="form1"><P><TEXTAREA name="thetextarea"
rows="10" cols="20"> Перемещение было произведено над элементами: </
TEXTAREA></FORM>');
  }
</SCRIPT>
</BODY>
</HTML>
```

ПРИМЕЧАНИЕ

Элемент <form> включает в себя объект <textarea>, поскольку Netscape Navigator не может представлять объекты <input>, <select> и <textarea> вне формы.

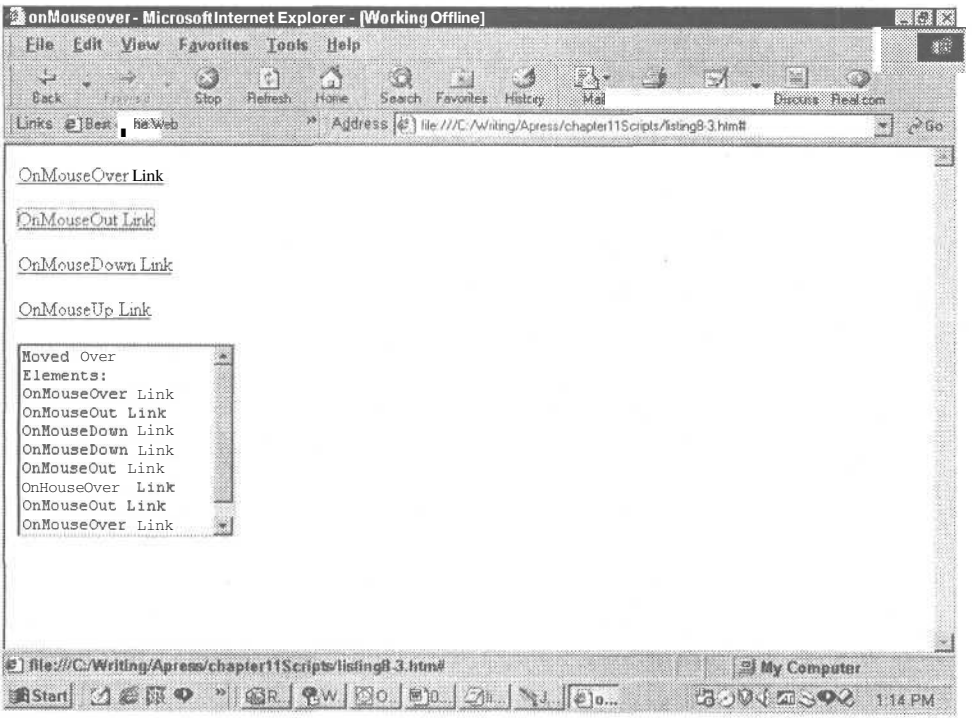


Рис. 8.3. Результат выполнения листинга 8.3: в текстовом окне содержатся имена ссылок, над которыми перемещалась мышь

Нажатие кнопок мыши

Кроме перемещения мыши, событиями также являются нажатие (щелчок) на кнопку мыши и двойной щелчок. В качестве инициаторов событий на DHTML-страницах используются шесть событий: onBlur, onClick, onDblClick, onFocus, onMouseDown и onMouseUp.

onBlur

Этот событие генерируется, когда элемент «выходит из фокуса». Это возможно при переходе в другое окно при помощи мыши или клавиши табуляции, а также при открытии нового окна или приложения. HTML-синтаксис:

```
onBlur=script
```

JavaScript-синтаксис:

```
object.onBlur = handler
```

СОВЕТ

Довольно просто создать бесконечный цикл с использованием события onBlur, связанного с окном, особенно если это *модальное окно* (окно диалога) (которое должно быть закрыто прежде, чем вы сможете возвратиться к основному окну). Обработка диалогового окна будет прервана переключением внимания на основное окно или фрейм, что породит событие onBlur.

В таблице 8.4 представлены объекты, которые могут перехватывать событие onBlur в браузере компании Netscape.

Таблица 8.4. Элементы, поддерживающие перехват события onBlur в браузерах компании Netscape

Объект	Navigator 3	Navigator 4	Navigator 6
window	Да	Да	Да
layer	Нет	Да	Да
link	Нет	Нет	Да
area	Нет	Нет	Да
image	Нет	Нет	Нет
form	Нет	Нет	Нет
text, textarea, password	Да	Да	Да
all buttons	Нет	Нет	Да
select	Да	Да	Да
fileUpload	Да	Да	Да

В примере, представленном в листинге 8.4, событие onBlur используется для изменения цвета фона текущего слоя, когда мышь «переключает внимание». Аналогично работает событие onMouseOut, но оно относится только к использованию мыши и не перехватывает событий клавиатуры. В отличие от него, событие onBlur возникает и при переходе с одного активного элемента на другой клавишей Tab.

Листинг 8.4. Использование события onBlur для изменения цвета фона документа

```
<HTML>
<HEAD>
<TITLE>Пример использования события onBlur</TITLE>
<SCRIPT language="JavaScript">
function changeColor() {
  alert("Вызывает изменение цвета на: " + document.bgColor);
```

продолжение

Листинг 8.4 (*продолжение*)

```
    thiscolor=document.bgColor;
     if (thiscolor=="#ffffff"){
         thiscolor="#ff00ff";
     } else if (thiscolor=="#ff00ff") {
         thiscolor="#ffffff";
     }
     if (document.all) onBlurDoc.bgColor = thiscolor; //IE4
         ocument. layers) document.bgColor = thiscolor; //NN4
}
</SCRIPT>
</HEAD>
<BODY id="onBlurDoc" bgcolor="#ffffff">
Поместите курсор внутрь текстового окна, а затем нажмите клавишу Tab.
<br> Цвет фона окна будет меняться каждый раз при переключении
"внимания".<P>
<SCRIPT Language="JavaScript">
  if (document.all) {
     document.write('<P><TEXTAREA name="thetextarea" rows="10" cols="20"
onblur="changeColor()">Press TAB in Me. </TEXTAREA>');
  } else if (document.layers) {
     document.write('<FORM name="form1"><P><TEXTAREA name="thetextarea"
rows="10" cols="20" onblur="changeColor()"> Нажми на TAB. </TEXTAREA></
FORM>');
  }
</SCRIPT>
</BODY>
</HTML>
```

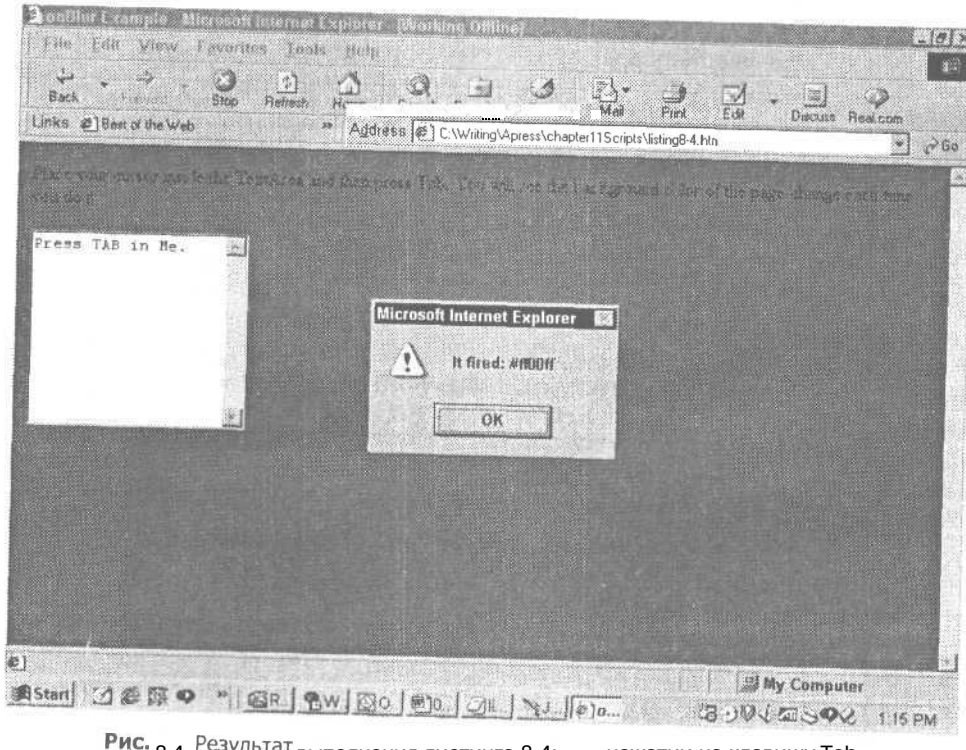


Рис. 8.4 Результат выполнения листинга 8.4: при нажатии на клавишу Tab в текстовом окне цвет фона меняется с белого на серый

onClick

Это событие происходит при щелчке кнопкой мыши на определенном объекте. Оно может быть вызвано нажатием клавиши Enter при заполнении формы или нажатием «горячей клавиши» управления формой, а также выбором пункта из открывающегося списка.

HTML-синтаксис:

```
onClick=script
```

JavaScript-синтаксис:

```
object.onClick = handler
```

В табл. 8.5 представлены объекты, которые могут перехватывать событие onClick в браузере компании Netscape.

Таблица 8.5. Элементы, поддерживающие перехват события onClick в браузерах компании Netscape

Объект	Navigator 3	Navigator 4	Navigator 6
window	Нет	Нет	Да
layer	Нет	Нет	Да
link	Да	Да	Да
area	Нет	Да	Да
image	Нет	Нет	Да
form	Нет	Нет	Нет
text, textarea, password	Нет	Нет	Да
all buttons	Да	Да	Да
select	Нет	Нет	Да
fileupload	Нет	Нет	Да

Пример, приведенный в листинге 8.5, использует последовательность событий onClick для изменения цвета фона в соответствии с цветом фона кнопок. Результат работы примера представлен на рис. 8.5. Поскольку Internet Explorer и Netscape Navigator по разному «понимают» атрибут bgcolor, во избежание ошибок из-за разногласий реализации используется оператор условного перехода. После упрощенной проверки типа браузера реализуются два варианта, «понятные» соответствующему браузеру.

Листинг 8.5. Использование события onClick для изменения цвета фона объекта

```
<html>
<head>
<title>Пример использования события onClick</title>
</head>
<body id="TOC" bgcolor=#6699FF>
<div align="center">
<script>
bg1 = "#6699FF";
bg2 = "#6595AB";
bg3 = "#9797CC";
bg4 = "#FF8080";
bg5 = "#CCFFFF";
bg6 = "#FFFF99";
bg7 = "#C18085";
bg8 = "#76A977";
bg9 = "#83A59A";
function bgChange(bgName) {
```

продолжение

Листинг 8.5 *(продолжение)*

```
      if (document.all) {  // устанавливается цвет фона для  IE4+
        TOC.bgColor = bgName;
      }
      if (document.layers) document.bgColor = bgName;
// устанавливается цвет фона для  NN4
    }
</script>
  <a href="#" onClick="bgChange(bg1)"> <img src="blue.gif"  width="75"
height="42" border="0"></a>
  <a href="#" onclick="bgChange(bg2)"> <img src="steel.gif"  width="80"
height="42" border="0"></a>
  <a href="#" onclick="bgChange(bg3)"> <img src="purple. gif" width="75"
height="42" border="0"></a>
  <a href="#" onclick="bgChange(bg4)"> <img src="rose.gif"  width="75"
height="42" border="0"></a>
  <a href="#" onClick="bgChange(bg5)"> <img src="sky.gif" width="75"
height="42" border="0"></a>
  <a href="#" onClick="bgChange(bg6)"> <img src="yellow.gif" width="75"
height="42" border="0"></a>
  <a href="#" onClick="bgChange(bg7)"> <img src="wine.gif"  width="75"
height="42" border="0"></a>
  <a href="#" onclick="bgChange(bg8)"> <img src="green.gif"  width="75"
height="42" border="0"></a>
  <a href="#" onclick="bgChange(bg9)"> <img src="olive.gif"  width="75"
height="42" border="0"></a>
</div>
</body>
</html>
```

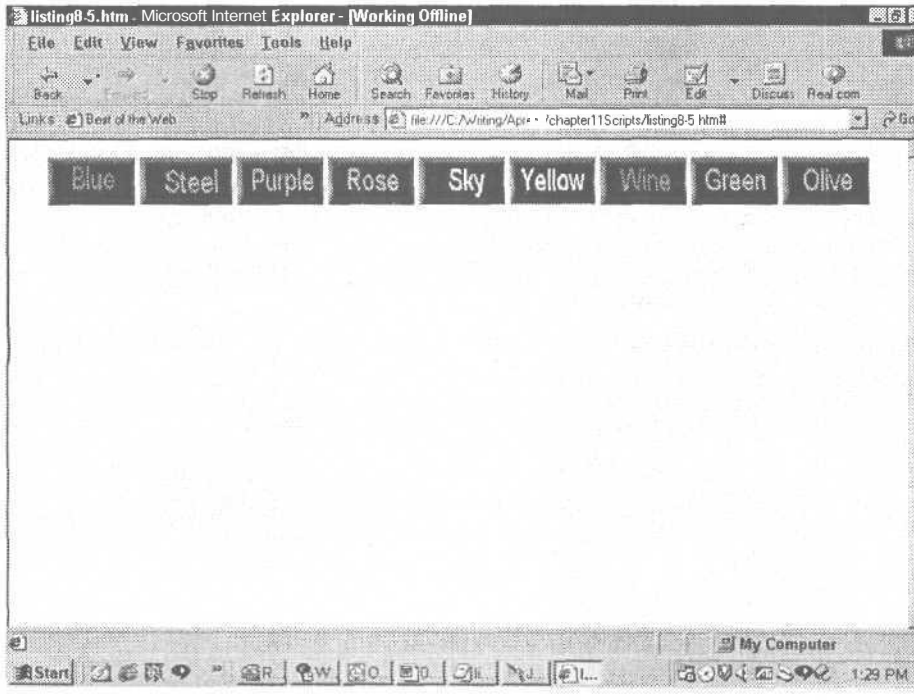


Рис. 8.5. Результат выполнения листинга 8.5: цвет фона изменяется в соответствии с цветом нажатой кнопки

onDblClick

Этот атрибут используется для активизации события в том случае, когда пользователь дважды нажмет левую кнопку мыши. На web-сайтах это свойство используется редко. Ссылки обычно активизируются одним щелчком, так же как и все кнопки и объекты апплетов. HTML-синтаксис:

```
onDblClick=script
```

JavaScript-синтаксис:

```
object.onDblClick = handler
```

В табл. 8.6 представлены объекты, которые могут перехватывать событие `onDblClick` в браузере компании Netscape.

Таблица 8.6. Элементы, поддерживающие перехват события `onDblClick` в браузерах компании Netscape

Объект	Navigator 3	Navigator 4	Navigator 6
window	Нет	Нет	Нет
layer	Нет	Нет	Нет
link	Нет	Да	Да
area	Нет	Нет	Да
image	Нет	Нет	Нет
form	Нет	Нет	Нет
text, textarea, password	Нет	Нет	Нет
all buttons	Нет	Нет	Нет
select	Нет	Нет	Нет
fileupload	Нет	Нет	Нет

В листинге 8.6 представлен пример использования события onDblClick для загрузки нового окна с коротким сообщением, сообщающим пользователю, что нет необходимости активизировать ссылки двойным щелчком. Результат выполнения примера представлен на рис. 8.6.

Листинг 8.6. Создание нового окна при помощи события `onDblClick`

```
<HTML>
<HEAD>
<TITLE>Пример использования события onDblClick</TITLE>
<5CRIPT>
var newWindow
function openNewWindow() {
  alert("Вы выполнили двойной щелчок на ссылке. В этом нет необходимости.
Достаточного одного щелчка");
  window.open("","","height=300, width=300");
}
</SCRIPT>
</HEAD>
<BODY>
<P><A href="#" ondblclick="openNewWindow()">Сведения об обдиночном щелч-
ке</A>
<P><A href="#" ondblclick="openNewWindow()">Сведения о двойном щелчке</A>
<P><A href="#" ondblclick="openNewWindow()">Сведения о перемещении мыши</
A>
<P><A href="#" ondblclick="openNewWindow()">Сведения об использовании
клавиатуры</A>
</BODY>
</HTML>
```

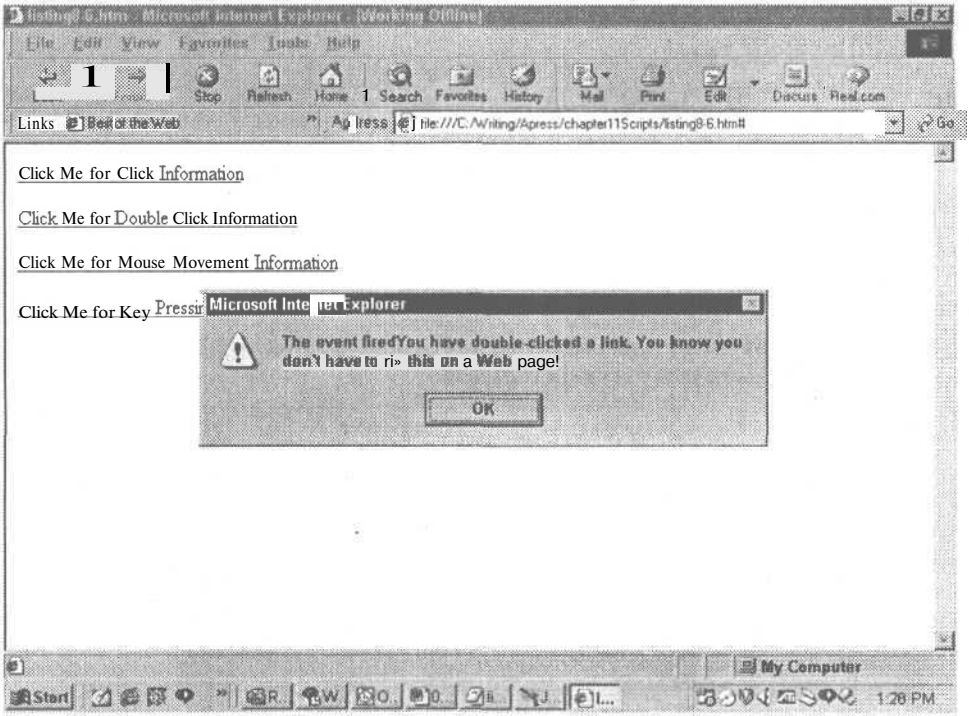


Рис. 8.6. Результат выполнения листинга 8.6: При одиночном щелчке создается новое окно, а при двойном — еще и предупреждение об отсутствии необходимости двойного щелчка

onFocus

Когда внимание пользователя переключается на определенный элемент (щелчок мыши или перемещение с помощью клавиши табуляции), генерируется событие onFocus. Оно также происходит при обращении к элементу с помощью метода focus языка сценария. Это событие обычно используется при работе с формами. При работе с формами по изменению «фокуса» можно изменять содержание подсказки, помогающей корректно заполнить поля формы, а также просто осуществить выделение цветом элемента ввода.

HTML-синтаксис:

```
onFocus=script
```

JavaScript-синтаксис:

```
object.onFocus=handler
```

СОВЕТ

Можно создать бесконечный цикл при использовании события onFocus, связанного с окном, особенно когда используется модальное диалоговое окно. При переключении фокуса с основного окна на обработку запроса диалогового, сразу как только вы вернетесь к использованию основного окна, появится новое диалоговое окно.

В табл. 8.7 представлены объекты, которые могут перехватывать событие onFocus в браузере компании Netscape.

Таблица 8.7. Элементы, поддерживающие перехват события onFocus в браузерах компании Netscape

Объект	Navigator 3	Navigator 4	Navigator 6
window	Да	Да	Да
layer	Нет	Да	Да
link	Нет	Нет	Да
area	Нет	Нет	Да
image	Нет	Нет	Нет
form	Нет	Нет	Нет
text, textarea, password	Нет	Да	Да
all buttons	Нет	Нет	Да
select	Да	Да	Да
fileupload	Да	Да	Да

В листинге 8.7 представлен пример использования события onFocus, которое выводит несколько текстовых полей, обращение к которым приводит к автоматической смене изображения. В данном примере осуществляется предварительная загрузка трех изображений, затем используется простая функция замены изображений в зависимости от выбранного текстового поля. Результат выполнения примера приведен на рис. 8.7.

Листинг 8.7. Замена изображений при обращении к текстовым полям

```
<HTML>
<HEAD>
<TITLE>Пример использования события onFocus</TITLE>
<SCRIPT Language="JavaScript">
TomImg = new Image();          // Предварительная загрузка изображений
TomImg.src = "tomatoe.gif";
PepImg = new Image();
PepImg.src = "pepper.gif";
PeaImg = new Image0 ;
PeaImg.src = "peas.gif";

function showImage(ImageName){
if (document.images) {
    document.veggie.src = eval(ImageName + ".src");
    }
}
</SCRIPT>
</HEAD>
<BODY bgcolor="white">
<IMG name="veggie" src="veggie.gif" alt="Здоровая закуска" width=100
height=100>
<FORM name="FlinstoneForm">
  <INPUT type="Text" value="Помидор" name="tomato"
onfocus="showImage('TomImg')">
  <INPUT type="Text" value="Перец" name="pepper"
onfocus="showImage('PepImg')">
  <INPUT type="Text" value="Горох" name="peas"
onfocus="showImage('PeaImg')">
</FORM>
</BODY>
</HTML>
```

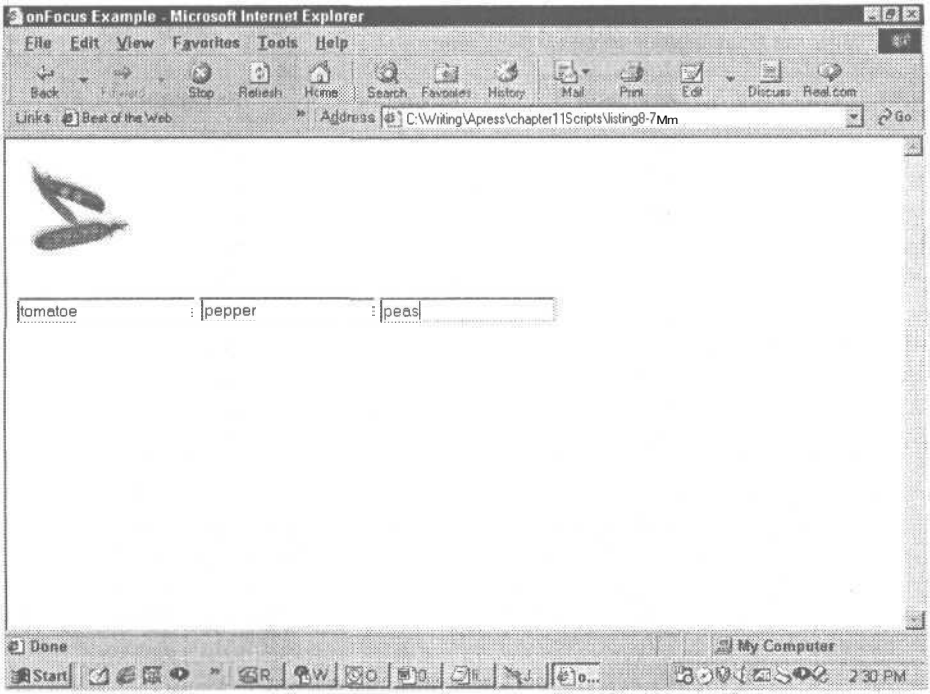


Рис. 8.7. Результат выполнения примера 8.7: переход к определенному текстовому полю вызывает вывод соответствующего изображения

onMouseDown

Это событие возникает при нажатии пользователем кнопки над определенным объектом. Не обязательно выполнять полный щелчок — достаточно просто нажать кнопку. Событие `onClick` происходит при полном щелчке (нажатии и отпускании кнопки), а событие `onMouseDown` наступает уже при нажатии и не дожидается отпускания кнопки. Можно нажать кнопку и, не отпуская ее, перемещаться по объекту, что приведет к запуску как события onMouseDown, так и события onMouseOver.

HTML-синтаксис:

```
onMouseDown=script
```

JavaScript-синтаксис:

```
object.onMouseDown=handler
```

В табл. 8.8 представлены объекты, которые могут перехватывать событие `onMouseDown` в браузере компании Netscape.

Таблица 8.8. Элементы, поддерживающие в Netscape-браузере перехват события onMouseDown

Объект	Navigator 3	Navigator 4	Navigator 6
window	Нет	Нет	Нет
layer	Нет	Нет	Да

Объект	Navigator 3	Navigator 4	Navigator 6
link	Нет	Да	Да
area	Нет	Нет	Да
image	Нет	Нет	Да
form	Нет	Нет	Нет
text, textarea, password	Нет	Нет	Да
all buttons	Нет	Да	Да
select	Нет	Нет	Да
fileUpload	Нет	Нет	Да

В листинге 8.8 приведен пример, использующий событие onMouseDown для запуска «встряски» окна браузера. В примере используется функция moveBy(), осуществляющая перемещение окна относительно его текущего местоположения. Результат выполнения примера представлен на рис. 8.8.

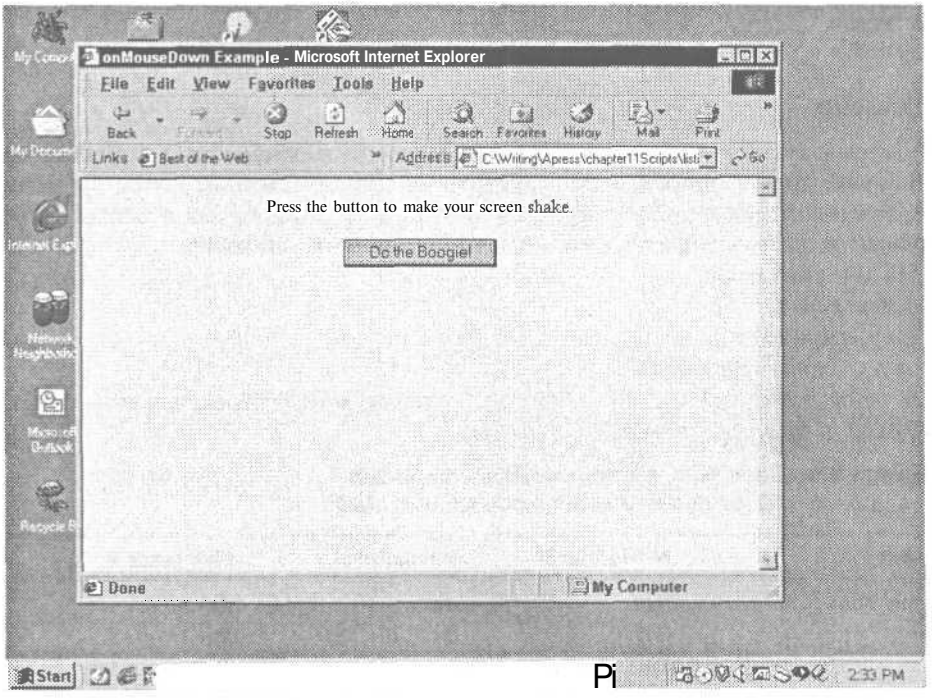


Рис. 8.8. Результат выполнения примера 8.8

Листинг 8.8. Перемещение окна при нажатии на кнопку

```
<HTML>
<HEAD>
<TITLE>Пример использования события onMouseDown</TITLE>
<SCRIPT language="JavaScript">
function getTheShakes(num) {
  if(self.moveBy) {
    for (dist=10; dist>0;dist--) {
```

продолжение

Листинг 8.8 (*продолжение*)

```
        for (i=num; i>0; i--) {
          self.moveBy(0,dist);
          self.moveBy(dist,0);
          self.moveBy(0,-dist);
          self.moveBy(-dist,0);
        }
      }
    }
  }
  </SCRIPT>
  </HEAD>
  <BODY bgcolor="light-brown">
  <DIV align=center>
    <h3>Хочешь быть богом?</h3> <BR>
  Нажми на кнопку, чтобы устроить землетрясение
    <FORM>
      <INPUT type=button onMouseDown="getTheShakes(2) " value="Землетрясение">
    </FORM>
  </DIV>
  </BODY>
  </HTML>
```

onMouseUp

Чтобы вызвать это событие, не обязательно щелкать мышью над определенным элементом, достаточно над ним отпустить кнопку. Другими словами, вы можете нажать кнопку над одной ячейкой таблицы, а отпустить ее над ссылкой. Это не приведет к активизации ссылки, но вызовет событие onMouseUp.

HTML-синтаксис:

```
onMouseUp=script
```

JavaScript-синтаксис:

```
object.onMouseUp=handler
```

В табл. 8.9 представлены объекты, которые могут перехватывать событие onMouseUp в браузере компании Netscape.

Таблица 8.9. Элементы, поддерживающие перехват события onMouseUp в Netscape-браузере

Объект	Navigator 3	Navigator 4	Navigator 6
window	Нет	Нет	Да
layer	Нет	Да	Да
link	Нет	Да	Да
area	Нет	Нет	Да
image	Нет	Нет	Да
form	Нет	Нет	Нет
text, textarea, password	Нет	Нет	Да
all buttons	Нет	Да	Да
select	Нет	Нет	Да
fileUpload	Нет	Нет	Да

В листинге 8.9 приведен пример использования события onMouseUp для запуска воспроизведения MIDI-файла при отправке формы. Вы можете вместо простого

текстового поля создать текстовое поле с открывающимся списком и позволить пользователю выбирать пункт «Продолжить воспроизведение» для прослушивания музыки при отправке формы. Результат выполнения примера представлен на рис. 8.9.

Листинг 8.9. Воспроизведение мелодии при использовании события onMouseUp

```
<HTML>
<HEAD>
  <TITLE>Пример использования события onMouseUp </TITLE>
</HEAD>
<SCRIPT>
<!--
function playSound()  {
    alert("У нас нет этой песни. Попробуйте выбрать другую.");
    document.jukebox.play();
}
-->
</SCRIPT>
<BODY >
  <FORM name="MIDI Form" action="Post">
    Какая у вас любимая песня?    
    <INPUT type="text" value="Mi Vida Loca">
    <INPUT type="button" value="Воспроизведение" onMouseUp="playSound() ">
  </FORM>
  <P>
  <EMBED name="jukebox" src="crow.mid" autostart="false">
</BODY>
</HTML>
```

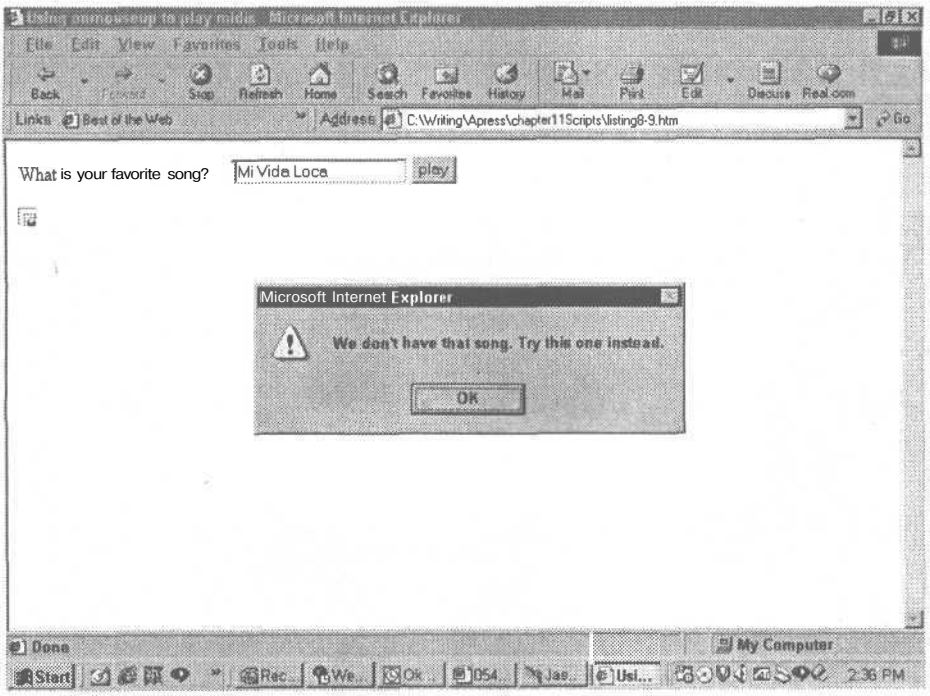


Рис. 8.9. Результат выполнения примера 8.9. При нажатии кнопки Воспроизведение осуществляется воспроизведение мелодии

Изменение размера страницы

При изменении размеров окна или фрейма происходит событие onResize. Его можно использовать для динамического изменения местоположения изображений или слоев на экране (этот вопрос рассматривался в главе 6).

HTML-синтаксис:

```
onresize=script
```

JavaScript-синтаксис:

```
window.onresize=handler
```

ПРИМЕЧАНИЕ

Для того чтобы поддержка этого события осуществлялась как в Netscape Navigator, так и в Internet Explorer, необходимо, чтобы оно было связано только с объектом window.

В листинге 8.10 представлен пример использования события onResize для изменения положения окна. Каждый раз при выборе мышью угла или стороны окна с целью его изменения окно будет перемещаться на новое место. Это сделает изменение окна весьма интересным занятием! Результат выполнения этого примера представлен на рис. 8.10.

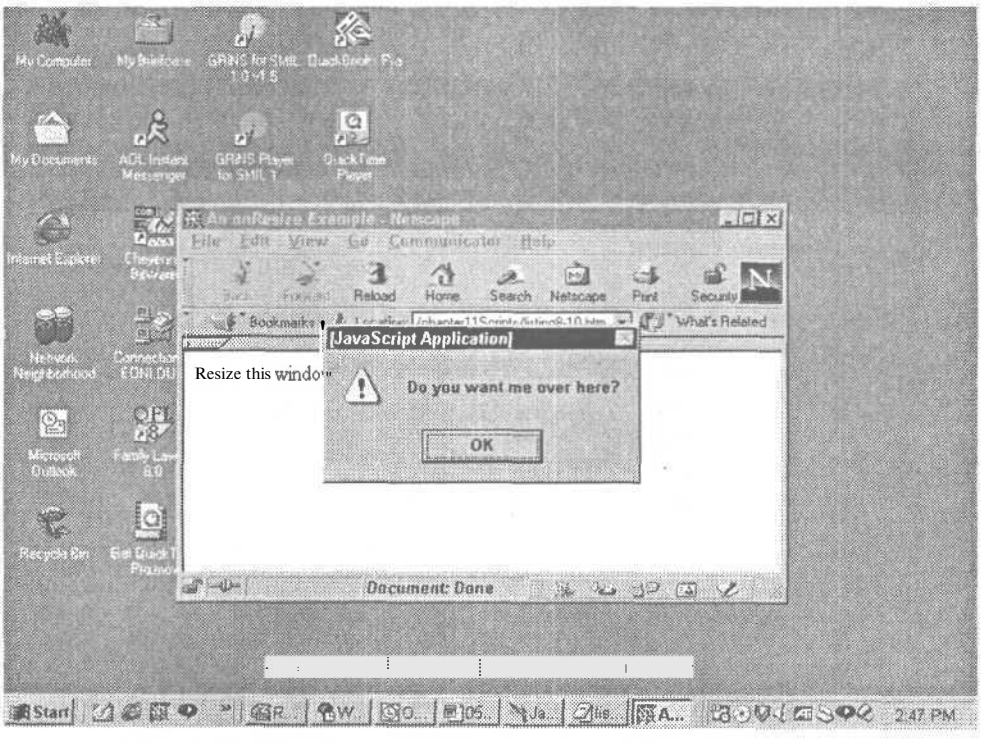


Рис. 8.10. Результат выполнения примера 8.10: при изменении размеров окна оно автоматически перемещается

Листинг 8.10. Использование события onResize

```
<HTML>
<HEAD>
  <TITLE>Пример использования события onResize</TITLE>
<5CRIPT  language="JavaScript">
function amResized(num) {
if(self.moveBy) {
    for (i=2; i>0;i--) {
      for (j=num; j>0;j--) {
        self.moveBy(120,i);
 alert  ("Как я вам здесь?  :-)");
        self.moveBy(i,120);
alert ("А здесь?");
      }
    }
 }
}
</SCRIPT>
</HEAD>
<BODY onResize = "amResized(1)">
Измените размер этого окна и посмотрите, что произойдет!
</BODY>
</HTML>
```

Использование событий клавиатуры

События клавиатуры являются вторым из наиболее часто используемых на web-сайтах типов событий. Для перемещения от ссылки к ссылке или от поля к полю при заполнении формы можно использовать клавишу Tab. Полный ряд событий, связанных с нажатием клавиш клавиатуры, представлен двумя событиями: keyDown и keyUp. Генерация этих событий связана только с клавишей Enter и символьными клавишами стандартной клавиатуры. События функциональных клавиш, клавиш управления курсором и таких клавиш, как Del и Ins, не перехватываются.

ПРИМЕЧАНИЕ

События keyDown и keyUp генерируются до того, как соответствующий символ будет воспроизведен в текстовом поле, поэтому можно осуществить перехват ввода символов с клавиатуры. А это можно использовать для проверки правильности ввода.

Основными событиями клавиатуры, применяемыми для осуществления взаимодействия на web-сайтах, являются onChange, onKeyDown, onKeyPress и onKeyUp. Рассмотрим их подробно.

onChange

Это событие генерируется при изменении содержания текстовых полей text, textarea, password или select. Оно также происходит каждый раз при заполнении пользователем этих типов полей в форме. HTML-синтаксис:

```
onChange=script
```

При использовании JavaScript необходимо указать объект, для которого происходит перехват события:

```
object.onChange=handler
```

Использованная в примере, представленном в листинге 8.11, функция updateLists() использует событие onChange для внесения новой информации в объект SELECT HTML-формы. Функция updateLists()содержится в заголовке документа, а вызывается событием onChange из поля формы. Эта функция осуществляет вывод новой информации в окне выражения SELECT в зависимости от текущего выделения. Результат выполнения примера представлен на рис. 8.11.

Листинг 8.11. Обновление содержания при помощи события onChange

```
<html>
<head>
   <title>Использование события onChange</title>
</head>
<body bgcolor=white>
<SCRIPT language="JavaScript">
<!--
if (document.images) {
 clothes = new Object();
 clothes.football = new Array("Rain jacket", "Paint", "CheeseHat");
 clothes.baseball = new Array("Shorts", "Tank-tops", "Spiked Shoes");
 clothes.hockey = new Array("Skates", "Jeans", "Sweater");
 clothes.basketball = new Array("Jacket", "Tennis Shoes", "Horns");
}
function updateLists(sports) {
 if (document.images) {
  sel = sports.selectedIndex;
  if (sel == 1) {
   cloth = clothes.football;
  } else if (sel == 2) {
   cloth = clothes.baseball;
  } else if (sel == 3) {
   cloth = clothes.hockey;
  } else if (sel == 4) {
   cloth = clothes.basketball;
  } else {
   cloth = new Array();
  }
  sports.form.clothes.length = cloth.length;
  for(i=0;i<cloth.length;i++)
   sports.form.clothes.options[i].text = cloth[i];
 } else {
  alert("Ваш браузер не позволяет выполнить изменение списков");
 }
}
// -->
</script>
<form>
<select name="sports" size=1 onChange="updateLists(this)">
<option>Выберите вид спорта
<option>Футбол
<option>Бейсбол
<option>Хоккей
<option>Баскетбол
</select>
```

```
<p>Clothes:<br>
<select name="clothes" size=1>
<option>                 
</select><br><br>
</form>
</body>
</html>
```

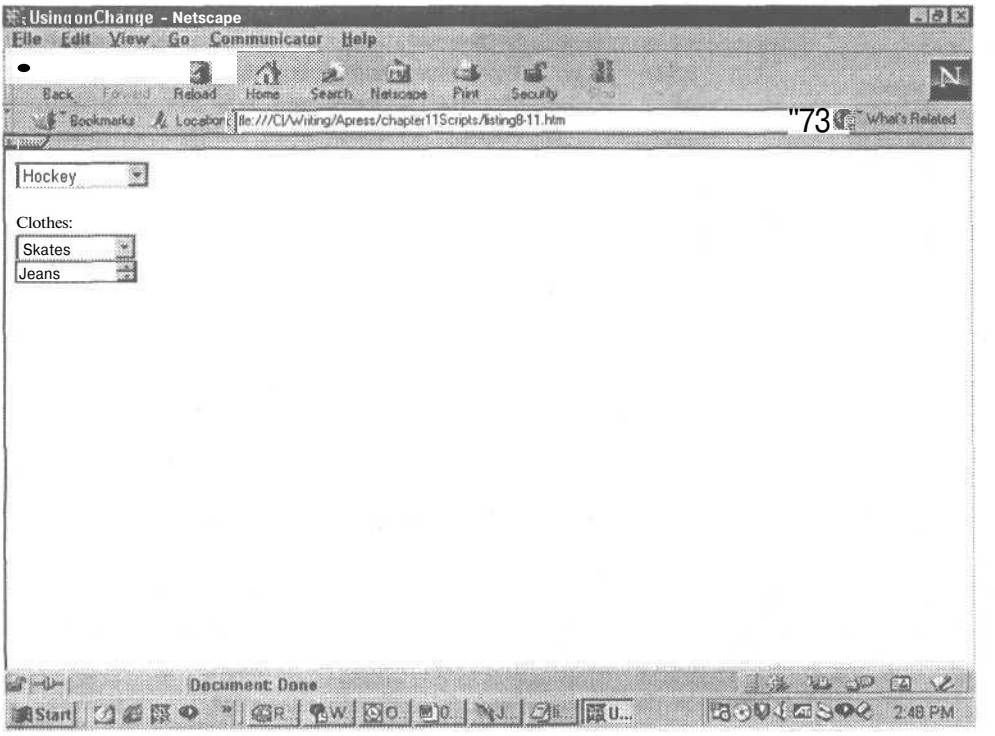


Рис. 8.11. Результат выполнения примера 8.11: содержание списка выбора автоматически изменяется при изменении значения в первом окне выбора

onKeyDown

Генерация события происходит при нажатии клавиши в пределах поля, с которым оно связано, не дожидаясь отпускания клавиши. Передача соответствующего символа в текстовое поле происходит после генерации события, что позволяет произвести проверку значения вводимого символа.

HTML-синтаксис:

```
onKeyDown=script
```

JavaScript-синтаксис:

```
object.onKeyDown=handler
```

В листинге 8.12 приведен модифицированный пример использования события onFocus, в котором обработчик события использует onKeyDown. В этом примере при вводе в текстовом поле происходит вывод соответствующего изображения.

Три изображения предварительно загружаются, а затем при помощи простой функции заменяют друг друга в зависимости от выбора поля ввода. Результат выполнения примера приведен на рис. 8.12.

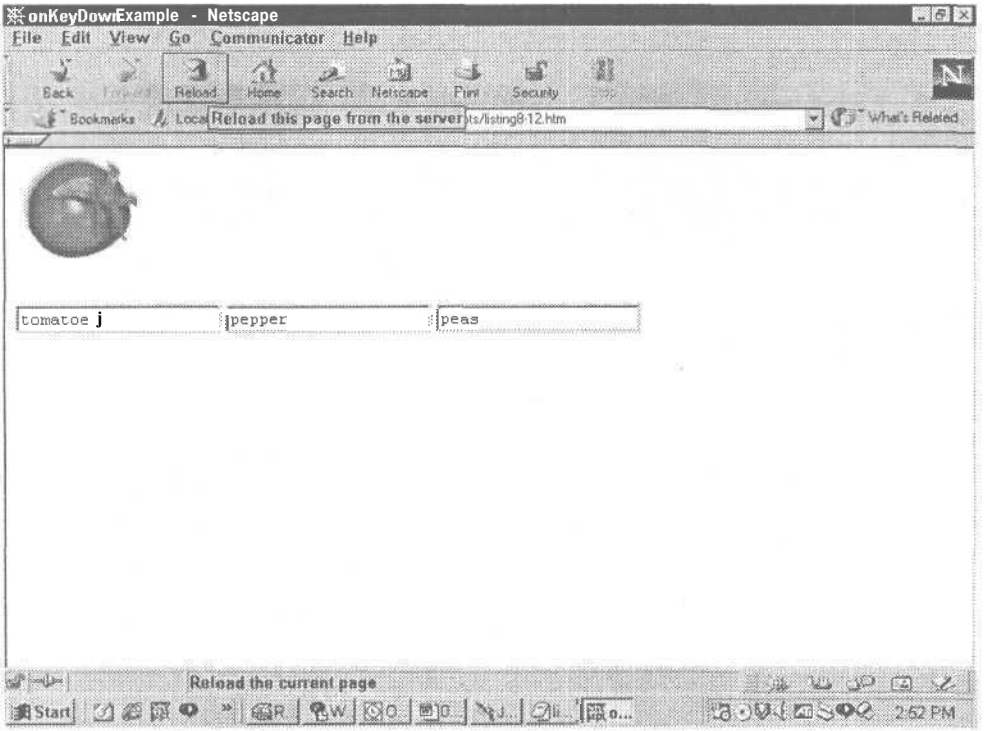


Рис. 8.12. Результат выполнения примера 8.12: изображение овощей изменяется при редактировании соответствующих текстовых полей

Листинг 8.12. Изменение изображений при помощи клавиатуры

```
<HTML>
<HEAD>
<TITLE>Пример использования события onKeyDown</TITLE>
<SCRIPT Language="JavaScript">
TomImg = new Image();  // Предварительная загрузка изображений
TomImg.src = "tomatoe.gif";
PepImg = new Image():
PepImg.src = "pepper.gif";
PeaImg = new Image0;
PeaImg.src = "peas.gif":

function showImage(ImageName){
if (document.images) {
      document.veggies.src = eval(ImageName + ".src");
      }
}
</SCRIPT>
</HEAD>
<BODY bgcolor="white">
```

```
<IMG name="veggies" href="veggie.gif" alt="Здоровая закуска. Измените
название овощей" width=100 height=100>
<FORM name="FlinstoneForm">
  <INPUT type="text" value="Помидор" name="tomatoe" onKeyDown="showImage('TomImg')">
  <INPUT type="text" value="Перец" name="pepper" onKeyDown="showImage('PepImg')">
  <INPUT type="text" value="Горох" name="peas" onKeyDown="showImage('PeaImg')">
</FORM>
</BODY>
</HTML>
```

onKeyPress

Генерация этого события происходит после нажатия и отпускания клавиши. Связанная с ним функция будет запущена только после отпускания клавиши, но до того, как символ будет выведен в текстовом поле. Это событие возвращает значение нажатой клавиши, которое может быть проверено перед выводом.

HTML-синтаксис:

```
onKeyPress=script
```

JavaScript-синтаксис:

```
object.onKeyPress=handler
```

В примере, приведенном в листинге 8.13, требуется нажать клавишу во втором текстовом окне. При этом текст в первом окне будет «перевернут». Переворот осуществляется при помощи метода substring, позволяющего определить длину представленной строки и переставить символы, из которых она состоит, в обратном порядке. Результат выполнения примера приведен на рис. 8.13.

Листинг 8.13. «Переворот» содержания текстового поля при использовании события onKeyPress

```
<HTML>
<HEAD>
<TITLE>Пример использования события OnKeyPress</TITLE>
<SCRIPT language="JavaScript">
function Reversal(form) {
  playtext = "";
  playstring = document.form1.revtext.value;
  for(i=0; i<=playstring.length; i++) {
    playtext=playstring.substring(i, i+1) + playtext;
    document.form1.revtext.value = playtext;
  }
}
</SCRIPT>
</HEAD>
<BODY>
<P>После перехода ко второму текстовому полю нажмите клавишу Enter или
Пробел <br>
для изменения порядка следования символов, составляющих строку в первом
поле
<FORM name="form1">
  <INPUT type="text" value="Пример 8.13" name="revtext">
  <INPUT type="text" value="Перевернуть текст" onKeyPress="Reversal(this.form)">
</FORM>
</BODY>
</HTML>
```

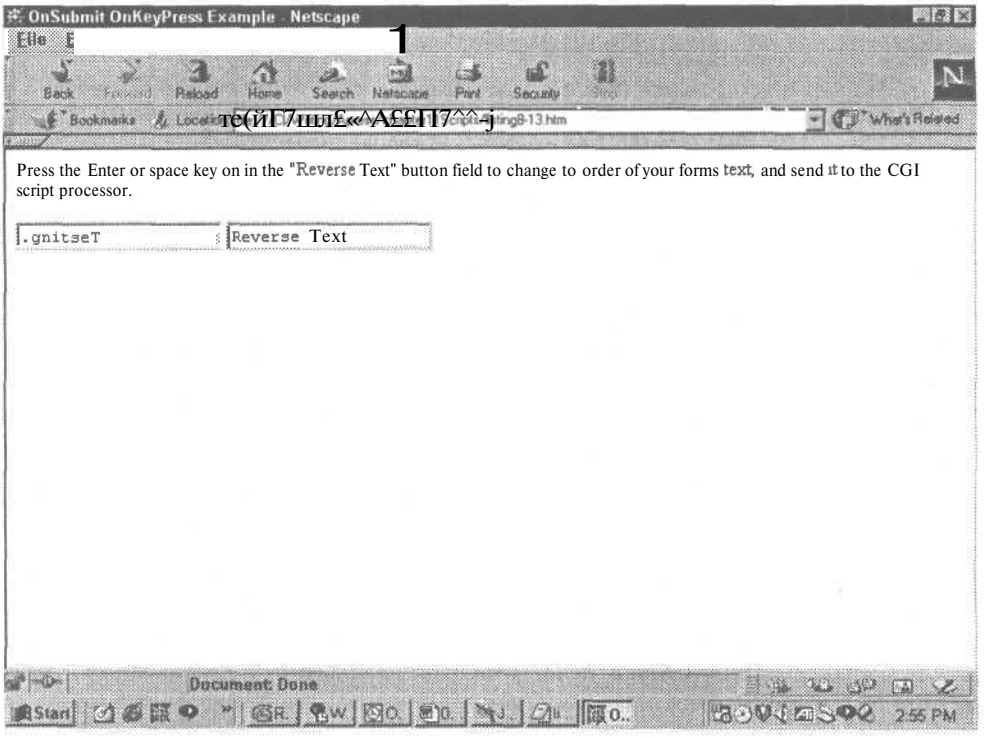


Рис. 8.13. Результат выполнения примера 8.13: строка в первом поле перевернута после нажатия клавиши во втором текстовом поле

onKeyUp

При отпускании клавиши в текстовом поле происходит генерация связанного с ним события onKeyUp. Событие возвращает значение нажатой клавиши, что позволяет выполнить проверку этого значения.

HTML-синтаксис:

```
onKeyUp=script
```

JavaScript-синтаксис:

```
object.onKeyUp=handler
```

Использование событий, связанных с формами

В объекте «форма» для отправки и сброса введенных сведений обычно используются кнопки Submit (Отправить) и Reset (Сброс). Обычно обработку сведений в полях формы осуществляют CGI-сценарии или серверные приложения обработки информации, такие как расширения FrontPage. При помощи JavaScript также можно осуществить захват этих сведений и осуществить их обработку. Для этого необходимо

перехватить события onReset и onSubmit. После перехвата можно запустить функцию, производящую обработку сведений до отправки их CGI-сценарию.

onReset

Генерация этого события происходит при нажатии пользователем на форме кнопки Reset, которая производит очистку формы или восстанавливает исходные значения. При захвате сценарием этого события до отправки на сервер можно заполнить форму определенными значениями, вывести изображение или активизировать звуковое сопровождение, а также просто выполнить любую задачу.

HTML-синтаксис:

```
onReset=script
```

JavaScript-синтаксис:

```
object.onReset=handler
```

Вместо того чтобы при нажатии на форме кнопки Reset заполнить форму исходными значениями, пример, представленный в листинге 8.14, при захвате события onReset осуществляет заполнение формы сведениями, хранимыми в переменных сценария. Результат выполнения примера представлен на рис. 8.14.

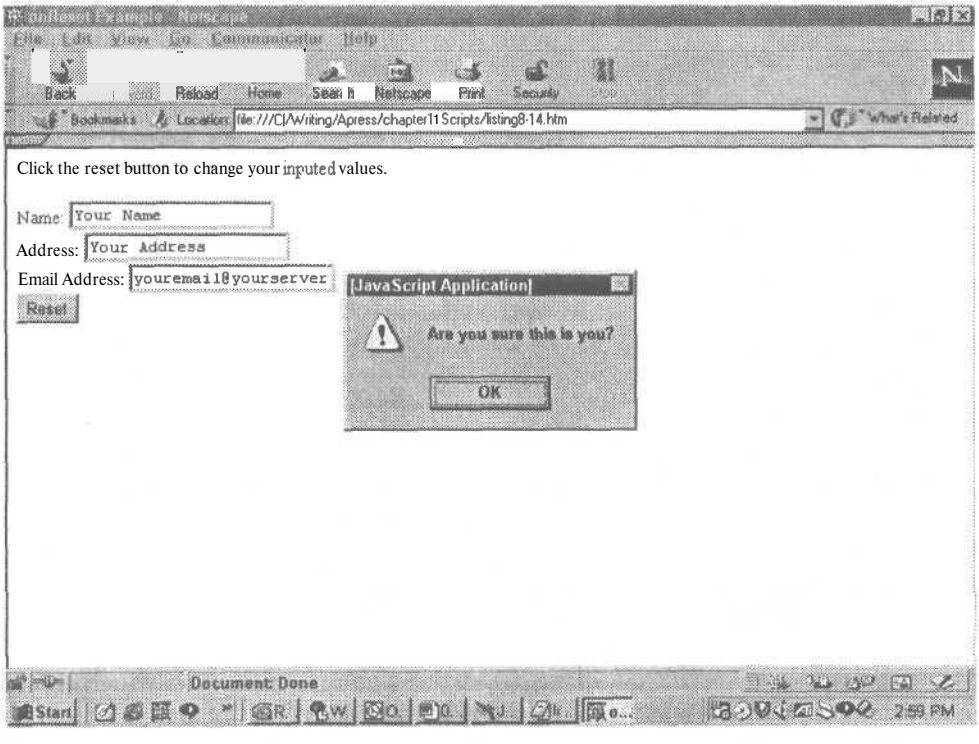


Рис. 8.14. Результат выполнения примера 8.14: при нажатии кнопки Сброс в поля формы вносятся заранее определенные значения

Листинг 8.14. Использование события onReset для загрузки в форму определенных сведений

```
<HTML>
<HEAD>
<TITLE>Пример использования события onReset</TITLE>
</HEAD>
<BODY>
<P>Щелкните на кнопке Сброс для изменения введенных значений</P>
<FORM name="form1" onReset="handleReset(this.form)">
  Имя: <INPUT type="text" value="Иван Петров" name="name"><BR>
  Адрес: <INPUT type="text" value="Санкт-Петербург г/п а/я 777"
name="address"><BR>
  Email-адрес: <INPUT type="text" value="Ivan_Petrov@piter.com" name="email"><BR>
  <INPUT type="Reset" value="Сброс">
</FORM>
<SCRIPT language="JavaScript">
function handleReset(form) {
  document.form1.name.value = "Ваше имя";
  document.form1.address.value = "Ваш адрес";
  document.form1.email.value = "Ваш e-mail";
  alert('Сведения указаны верно?');
}
</SCRIPT>
</BODY>
</HTML>
```

onSubmit

Данное событие активизируется при нажатии пользователем кнопки Submit (Отправить), щелчке на изображении, связанном с кнопкой Submit, или при непосредственном использовании (только в Internet Explorer) элемента <button> или объекта <input type=button>.

HTML-синтаксис:

```
onSubmit=script
```

JavaScript-синтаксис:

```
object.onSubmit=handler
```

Пример, приведенный в листинге 8.15, перехватывая событие onSubmit, выводит простое модальное окно, содержащее сведения из формы. Результат выполнения приведен на рис. 8.15.

Листинг 8.15. Использование события onSubmit позволяет пользователю просмотреть информацию перед отправкой

```
<HTML>
<HEAD>
<TITLE>Пример использования события OnSubmit</TITLE>
</HEAD>
<BODY>
<P>Щелкните на кнопке Отправить.
<FORM name="form1" onSubmit=handleSubmittal(this.form)>
```

```
  Name: <INPUT type="text" value="Иван Петров" name="name"><BR>
  Address: <INPUT type="text" value="Санкт-Петербург г/п а/я 777"
name="address"><BR>
  Email Address: <INPUT type="text" value="Ivan_Petrov@piter.com"
name="email"><BR>
  <INPUT type="submit" value="Отправить">
</FORM>
<SCRIPT language="JavaScript">
function handleSubmittal(form) {
  nametext = document.form1.name.value;
  addytext = document.form1.address.value;
  emailtext = document.form1.email.value;
  alert('Вы собираетесь отправить следующую контактную информацию: ' +
nametext + " адрес " + addytext + " email " + emailtext);
}
</5CRIPT>
</BODY>
</HTML>
```

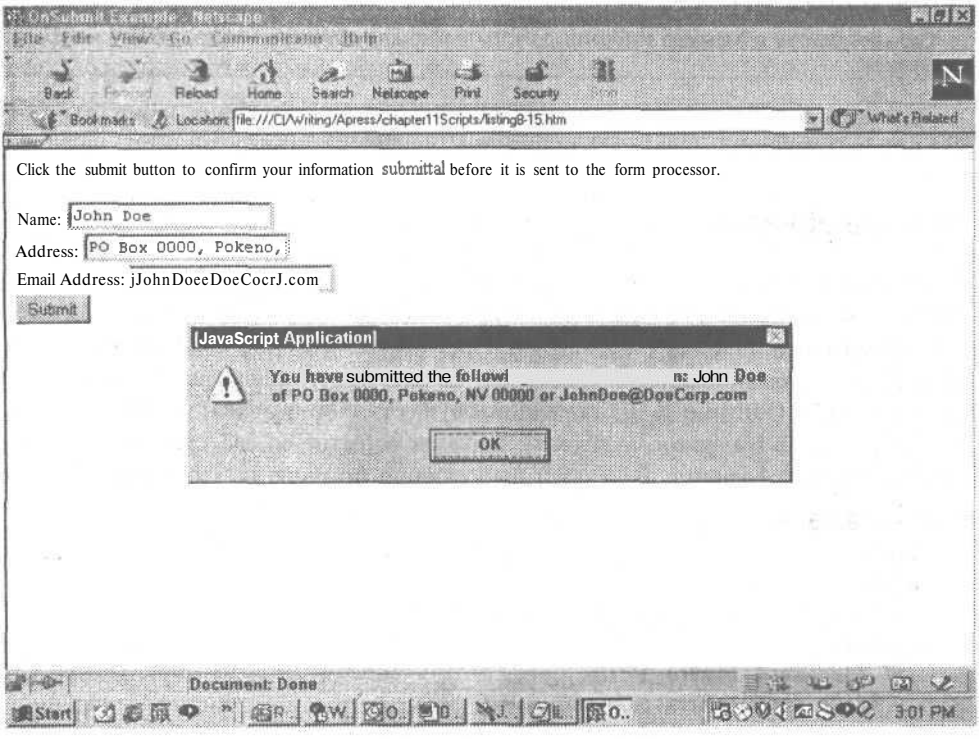


Рис. 8.15. Результат выполнения примера, приведенного в листинге 8.15

Загрузка документа

Для того чтобы пользователь остался доволен первым же посещением вашего сайта, необходимо сразу же предоставить ему интересную информацию. Пользователь не любит ждать, пока загрузится картинка или когда появится текстовая информация. Он хочет сразу знать, что интересного имеется на сайте. За счет применения события onLoad вы можете сразу предоставить пользователю информацию-заставку, с помощью которой он сможет узнать, какую информацию сейчас получит. С помощью этого события вы также можете упорядочить расположение изображений и слоев на экране после того, как они загрузятся.

HTML-синтаксис:

```
onLoad=script
```

При запуске функций, корректирующих ошибки, на языке JavaScript необходимо использовать следующий синтаксис:

```
window.onLoad=script
```

ПРИМЕЧАНИЕ

Событие onLoad в Netscape Navigator действительно для объектов «окно», «слой» и «изображение».

А теперь давайте рассмотрим ряд примеров, использующих перехват события при загрузке пользователем новых документов или объектов в текущее окно.

Загрузка объекта

Загрузка объекта может быть отслежена как в Netscape Navigator, так и в Internet Explorer, но Netscape позволяет выполнять отслеживание загрузки изображения или слоя только начиная с версии 4.x. При помощи Internet Explorer вы можете проверить состояние события onLoad для большинства объектов. В приведенном ниже примере (листинг 8.16) сценарий осуществляет перехват событий загрузки изображений Navigator, а затем использует событие onLoad, связанное с тегом IMG для Internet Explorer. Результат выполнения примера приведен на рис. 8.16.

Листинг 8.16. Загрузка изображений

```
<HTML>
<HEAD>
<TITLE>Пример загрузки изображения</TITLE>
<SCRIPT>
  if (document.layer) {
    image.captureEvents(Events.LOAD);
    image.onLoad=imgLoaded;
  }

function imgLoaded()
{ alert("Изображения загружены");}
</SCRIPT>
```

```
</head>
<BODY>
 Здесь представлено содержание страницы.
<IMG name="veggues" src="veggie.gif" alt="Healthy Snack Time" width=100
height=100 onload="imgLoaded()">
</BODY>
</HTML>
```

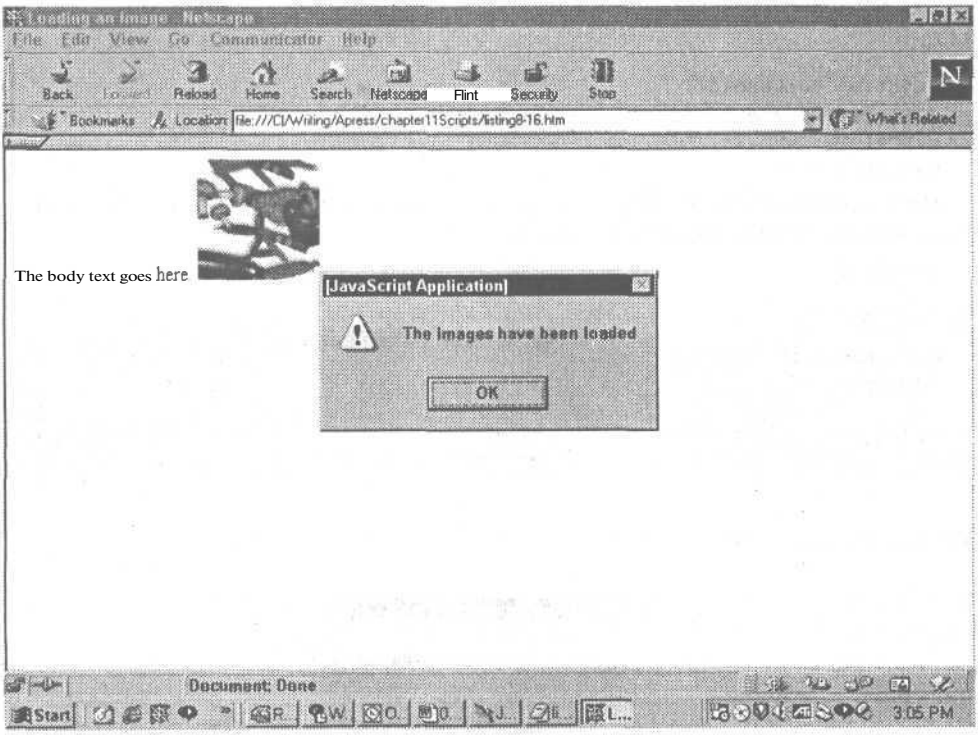


Рис. 8.16. Результат выполнения примера 8.16: после загрузки изображения выдано сообщение

Загрузка страницы

В приведенном ниже листинге 8.17 представлен пример, выводящий окно с сообщением после завершения загрузки страницы. Для продолжения взаимодействия с документом это окно обязательно должно быть закрыто. Результат выполнения примера приведен на рис 8.17. Окно с сообщением не появится до тех пор, пока не будет загружена вся страница и ее объекты. Следовательно, если для полной загрузки вашему документу требуется десять минут (поскольку он содержит базу данных или изображение), то вам придется ожидать появления этого сообщения десять минут. Конечно же, web-страница не должна быть такой громоздкой, если вы не постараетесь разместить на ней все содержание Библиотеки им. Ленина.

Листинг 8.17. Вывод сообщения после полной загрузки документа

```
<HTML>
<HEAD>
<TITLE>Пример использования события OnLoad</TITLE>
<SCRIPT>
  if (document.layer) {
    window.captureEvents(Events.LOAD);
    window.onLoad=bdyLoaded;
  }

function bdyLoaded()
{ alert("Документ полностью загружен");}
</SCRIPT>
</head>
<BODY onload="bdyLoaded()">
  Здесь размещается тело документа.
</BODY>
</HTML>
```

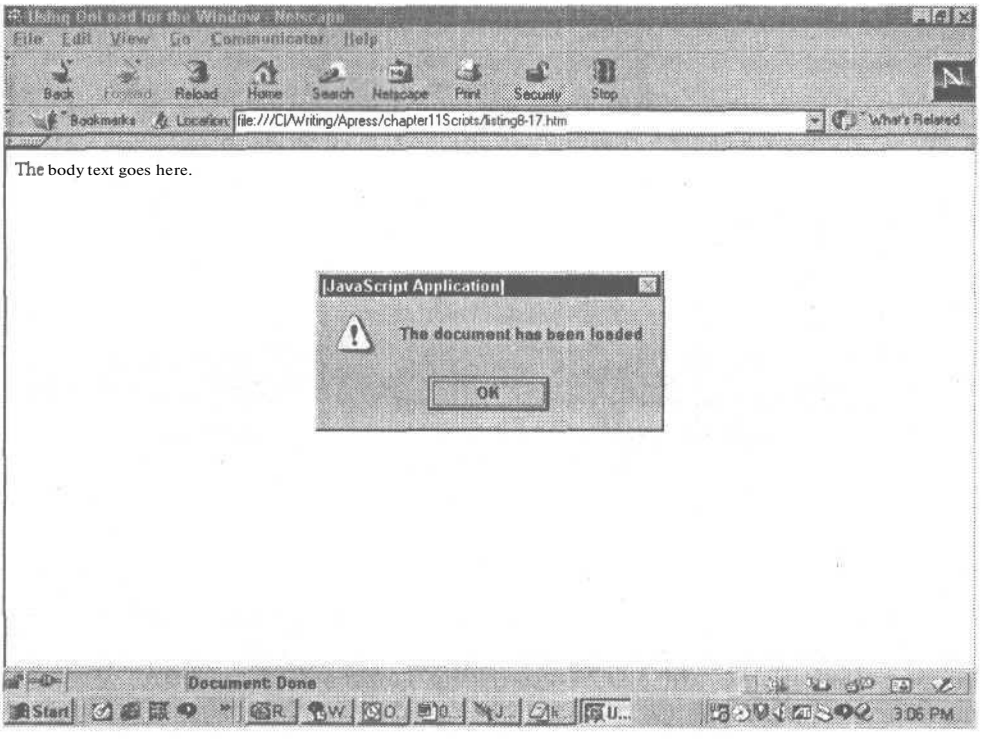


Рис. 8.17. После загрузки всего документа появилось сообщение

Выгрузка (закрытие) документа

Когда пользователь покидает ваш сайт, существует возможность послать ему сообщение с благодарностью за посещение сайта, а также выполнить очистку переменных настройки или записи информации в cookie (историю обращения пользователя к данному серверу). Эти операции можно выполнить именно при выгрузке документа, а не по другим действиям посетителя.

Событие onUnLoad происходит только при закрытии документа. Назначенная функция не будет выполнена, пока пользователь не закроет данное окно.

HTML-синтаксис:

```
onUnload=script
```

JavaScript-синтаксис:

```
window.onUnload=script
```

ПРИМЕЧАНИЕ

В Netscape Navigator событие onUnLoad действительно только в отношении объекта window.

Давайте рассмотрим несколько примеров, которые активизируются при выгрузке (закрытии) всего документа, представленного в окне браузера. Пример из листинга 8.18 является сценарием, который выводит сообщение после закрытия документа. Пока не будет закрыто это окно, загрузка в браузер другой страницы невозможна. Результат выполнения примера представлен на рис. 8.18.

Листинг 8.18. Вывод сообщения при закрытии документа

```
<HTML>
<HEAD>
<TITLE>Пример использования события onUnLoad</TITLE>
<SCRIPT>
  if (document.layer) {
    window.captureEvents(Events.UNLOAD);
    window.onUnLoad=bdyUnLoaded;
  }

function bdyUnLoaded()
{ alert("Документ выгружен");}
</SCRIPT>
</head>
<BODY onunload="bdyUnLoaded()">
  Здесь помещается тело документа.
</BODY>
</HTML>
```

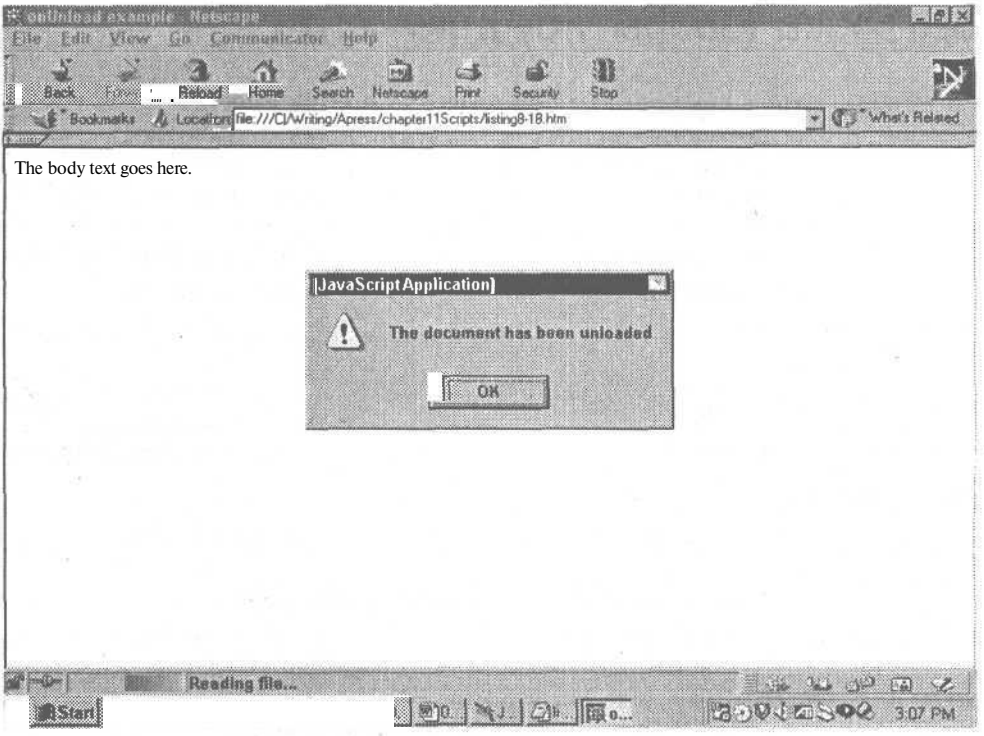


Рис 8.18. После закрытия документа выводится сообщение

Ошибки

При работе с сайтом могут возникать ошибки. Как разработчик динамического web-сайта вы столкнетесь с тем, что, как бы хорошо вы ни разработали код для одного браузера, для безукоризненной работы его в другом браузере придется съесть не один пуд соли. Значительная часть времени уйдет на то, чтобы добиться работоспособности кода сначала на одном браузере. После этого необходимо изменить этот код так, чтобы обойти различия в реализации объектной модели документа двумя основными браузерами. Если вы хотите добиться желаемого эффекта, то можете оставить неизменным код для одного браузера, а затем при помощи события onError обработать ошибки, возникающие при воспроизведении документа другим браузером.

Событие onError происходит при возникновении ошибки при работе с документом. Большинство ошибок происходит в момент выполнения, но необходимо предусмотреть возможность возникновения ошибок еще на этапе загрузки документа или объекта.

HTML-синтаксис:

```
onerror=script
```

JavaScript-синтаксис:

```
window.onerror=script
```

ПРИМЕЧАНИЕ

В Netscape Navigator событие onError действительно только в отношении объекта window.

Пример, представленный в листинге 8.19, «отлавливает» ошибку, специально встроенную в 11-й строке листинга. Происходит вывод окна с сообщением об ошибке, с указанием типа ошибки, строки и URL источника ошибки. Результат выполнения данного примера приведен на рис. 8.19.

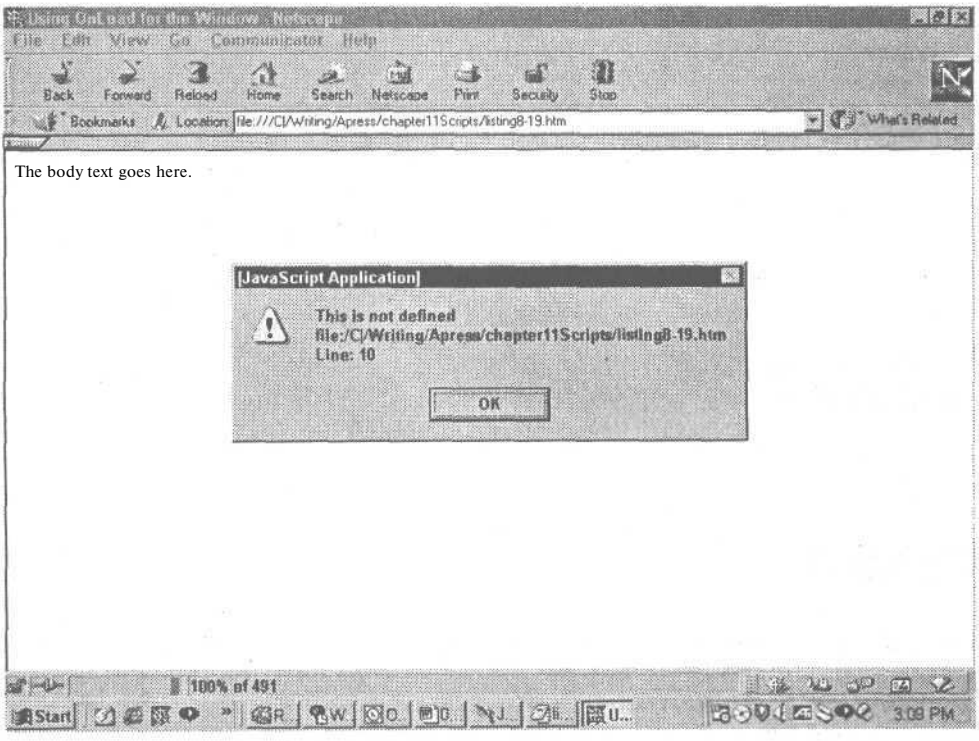


***Рис.* 8.19.** При возникновении ошибки появилось сообщение

Листинг 8.19. Перехват и вывод сообщения об ошибке при использовании события onError

```
<HTML>
<HEAD>
<TITLE>Пример использования события onError</TITLE>
<SCRIPT language="JavaScript">
  function traperror(msg,url,line)
  { alert(msg + "\n" + url + "\nLine: " + line);
  return true;
  // Пресекает индикацию ошибки средствми IE.
  }
```

продолжение ⇩

Листинг 8.19 *(продолжение)*

```
    onError=traperror;
    This.Entry.should_cause_a_scripting_error = 5; //Это ошибка
  </SCRIPT>
  </HEAD>
  <BODY onError="errortrap()">
  <SCRIPT  language="JavaScript">
    if (document.layer) {
      window.captureEvents(Events.ERROR);
      window.onError=errortrap;
    }
  </SCRIPT>
  </BODY>
  </HTML>
```

Работа со слоями

9

В главе 6 проведено определение и описание всех слоев, имеющихся в примере нашего HTML-сайта. Это был только первый шаг к использованию слоев. Нам необходимо правильно разместить их в документе, управлять их видимостью и осуществлять анимационное перемещение. При использовании в документе множества слоев необходимо определить, как они будут взаимодействовать друг с другом и как они предстанут перед пользователем. Давайте рассмотрим, как решаются эти задачи.

Позиционирование и индексирование слоев

Определение позиций слоев является одной из наиболее важных задач в создании «слоеных» HTML-сайтов. Невнимательное упорядочивание слоев сайта может привести к тому, что в одной операционной системе он будет представлен великолепно, а при использовании этого же браузера в другой операционной системе — просто ужасно. Для того чтобы при использовании любого браузера все слои документа были размещены соответствующим образом, необходимо использовать свойства каскадных таблиц стилей, представленные далее в этой главе. Позиционирование элементов может осуществляться относительным, абсолютным и фиксированным позиционированием.

Относительное позиционирование

При первоначальной разбивке документа слои располагаются в соответствии с текущей последовательностью объектов и текста. После установки размера каждого объекта выбирается оптимальное расположение. После завершения этого этапа вы можете сместить объекты с их текущего положения в документе (по-

скольку позиция объекта определяется его размером). Это позиционирование известно как *относительное позиционирование (relative positioning)* (см. рис. 9.1). При использовании относительного позиционирования изменение расположения одного объекта ведет к изменению положения следующих за ним объектов. Это означает, что использование этого вида позиционирования может привести к выходу контейнеров за границы документов или документ может изменить первоначально предполагаемые пропорции. Для того чтобы контейнеры сохранили размеры и расположение, необходимо ввести упорядочивание в соответствии со значениями параметра z-index. (Рассматривалось в главе 6.)

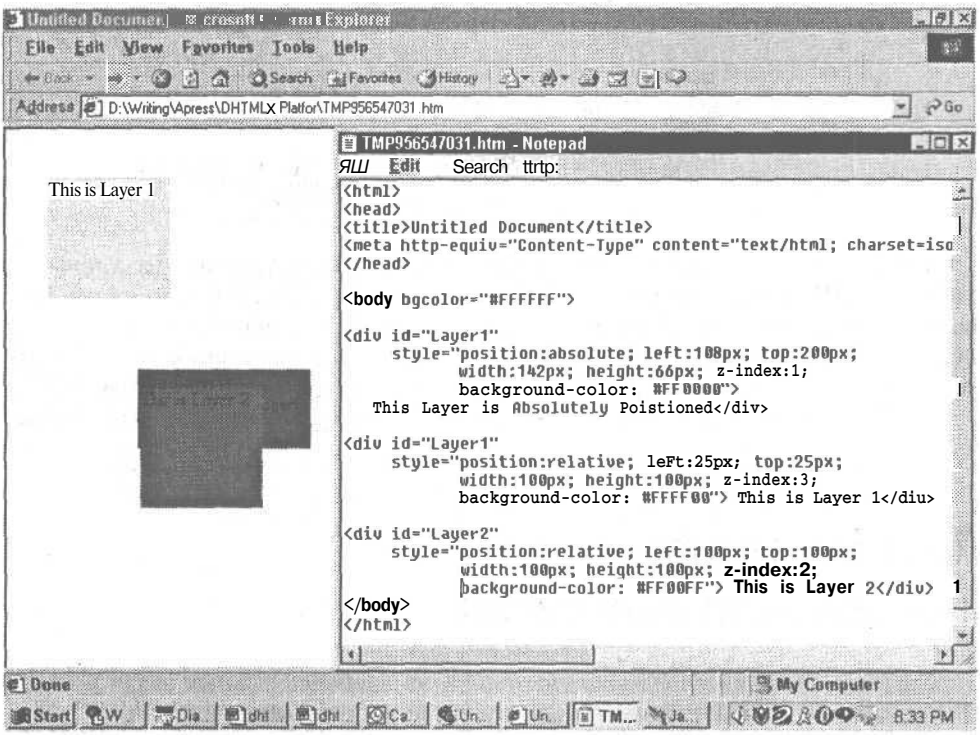


Рис. 9.1. Относительно позиционированные элементы располагаются в документе в соответствии с их реальным положением в документе

Абсолютное позиционирование

Абсолютно позиционированные элементы, представленные на рис. 9.2, помещаются в соответствующий им контейнер. При этом типе позиционирования для последующих элементов будут установлены новые контейнеры. Содержание абсолютно позиционированных элементов не «обтекает» другие контейнеры. Это может привести к перекрытию содержания других контейнеров данного документа. Абсолютно позиционированные элементы не влияют на положение следующих за ними элементов.

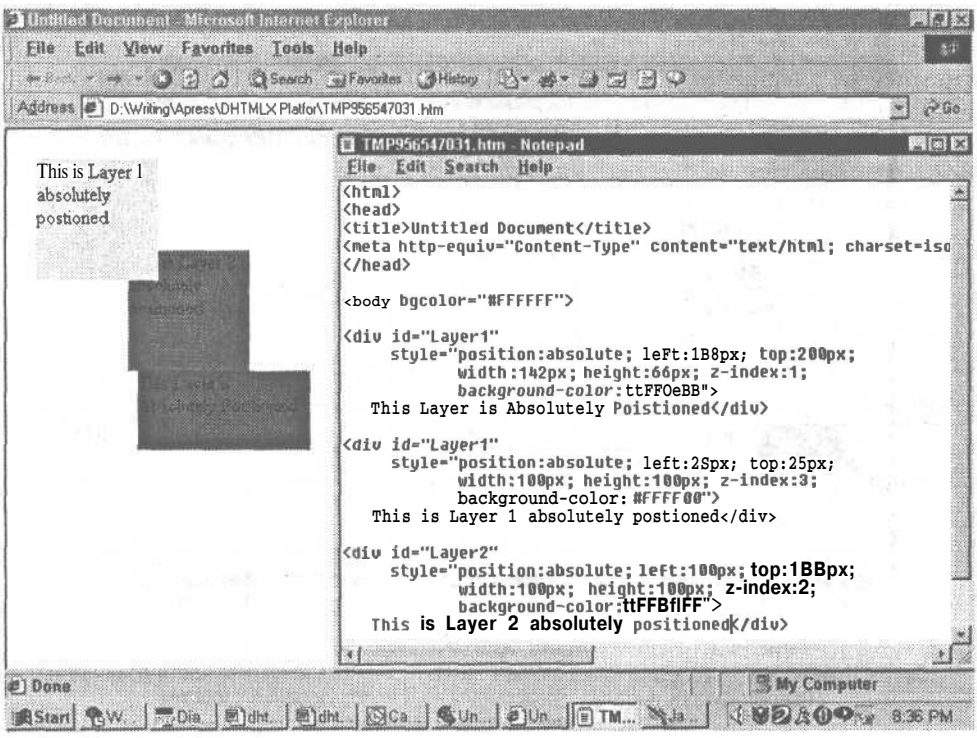


Рис. 9.2. Абсолютно позиционированные элементы появляются на странице в строгом соответствии со значениями параметров `left` и `top` в атрибуте `style`, связанном с элементом, которые создает этот слой

Фиксированное позиционирование

Фиксированное позиционирование — это вариант абсолютного позиционирования, определяющий расположение контейнера относительно окна web-браузера. При просмотре «длинных» документов «зафиксированный» контейнер остается на своем месте даже при прокрутке документа, в то время как абсолютно позиционированный элемент, связанный с началом документа, при прокрутке будет перемещаться. Если расположить фиксированное поле в таком многостраничном документе, как книга, он будет появляться в нижней части каждой страницы. На рис. 9.3 и 9.4 представлен один и тот же документ с прокруткой и без. Данный вид позиционирования позволяет помещать на каждой странице верхний и нижний колонтитул или подпись в конце последовательности одностраничных документов.

ПРИМЕЧАНИЕ

Internet Explorer 4 и 5 не поддерживают фиксированное позиционирование. На настоящий момент этот тип позиционирования полностью поддерживает лишь Netscape Navigator 6.

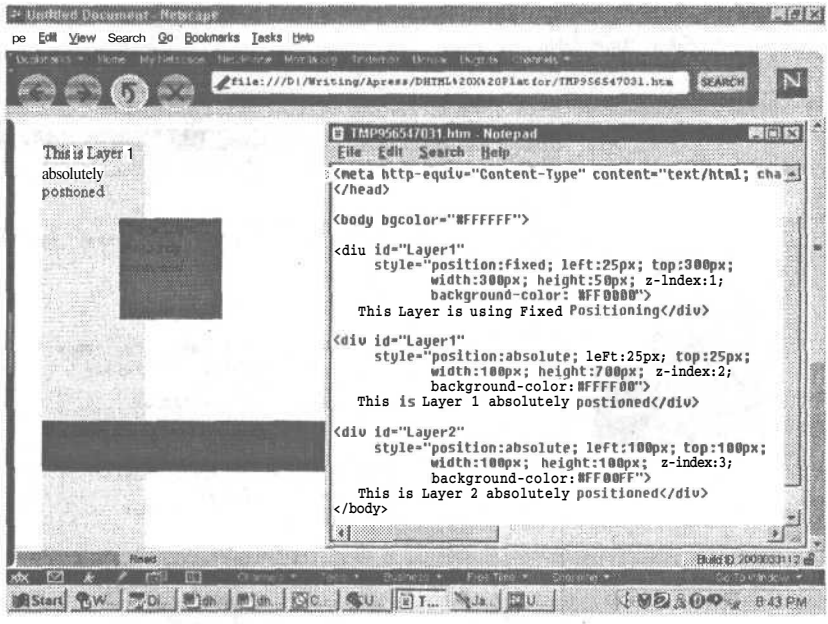


Рис. 9.3. Экран документа, прокрученный к началу страницы. Окно с фиксированным позиционированием находится ближе к нижней части документа. Рядом показан листинг, представляющий использование свойства position: fixed таблицы стилей этот слоя

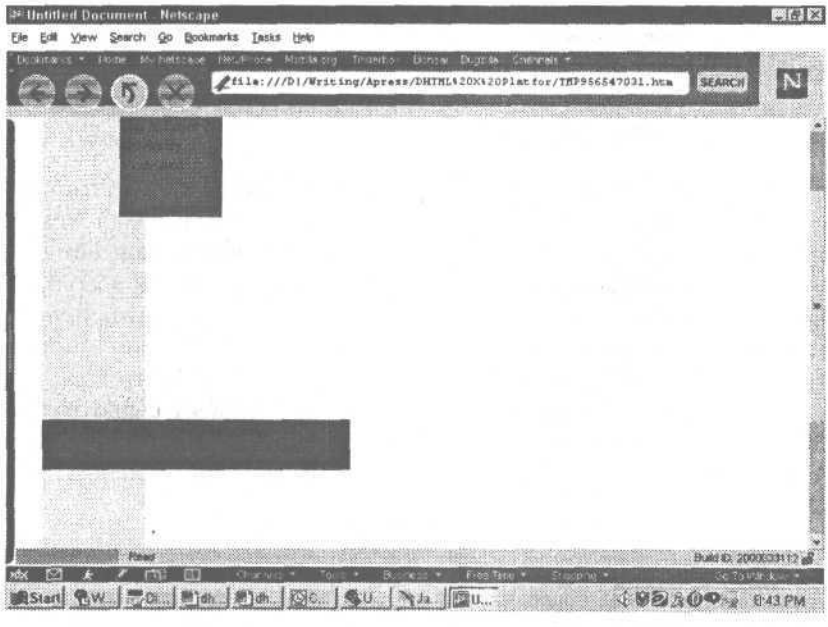


Рис. 9.4. После прокрутки страницы вниз слой Layer 2 поднялся вверх, *а* фиксированный слой остался подвешенным ближе к нижней части страницы

Свойства каскадных таблиц стиля

При размещении слоя необходимо использовать восемь из ста двадцати двух свойств CSS. Первые семь: position, left, top, right, bottom, width и height. Последним и единственным свойством CSS, которое управляет индексированием позиции каждого отдельного слоя, является z-index. Это свойство управляет последовательностью «укладки» слоев, то есть определяет перекрытие содержания других слоев.

СОВЕТ

Вы можете не помнить остальных свойств CSS, но обязательно должны запомнить эти восемь свойств. Они составляют базис DHTML-системы. Наиболее часто в качестве значения этих свойств используются конкретные расстояния — length. Когда вы прочитаете спецификацию различных атрибутов CSS, то заметите, что одно из возможных значений представлено как <length>. Описатель значения <length> относится к различным целым числам со знаком или без, а также к отдельным единицам измерения, в которых может быть представлено это значение. По умолчанию для всех значений единицы измерения расстояний используется положительный знак (+), хотя многие элементы могут иметь отрицательное значение (-). Отрицательные целые числа создают множество проблем с разработкой и форматированием при использовании некоторых браузеров.

Значения расстояния могут быть относительными и абсолютными. При определении позиции объекта в HTML-документе могут применяться *относительные единицы измерения,* которые используют ссылку на другое автоматически масштабируемое свойство (например, высоту символа). Таблицу стиля, использующую относительные единицы, легче преобразовать в различных типах среды. Например, если вы установите все расстояния в пикселах (pixels, px), высотах текстовых элементов (element's text, em) или в высотах буквы x (ex), то независимо от того, в какой среде осуществляется формирование изображения, оно всегда будет представлено верно. Единица em может использоваться для измерения как по вертикали, так и по горизонтали, тогда как при использовании ex-измерения возможно определение только высоты. Обе эти величины оперируют размером шрифта элемента, за исключением того случая, когда они входят в атрибут font-size, где они увеличивают или уменьшают только данный шрифт.

Пикселы — это относительные единицы измерения. Они определяются разрешением устройства просмотра, которым для web-страниц почти всегда является дисплей компьютера. Но иногда для их просмотра страниц может использоваться телевизор с Интернет-подключением (WebTV) или экран портативного устройства. Если плотность пикселов такого устройства будет сильно отличаться от плотности на типовом экране компьютера, то для более точного представления браузер может осуществить перемасштабирование пиксельных значений таблицы стилей.

Абсолютные единицы измерения используют действительную систему мер и представлены в дюймах (inches, in), сантиметрах (centimeters, cm), миллиметрах (millimeters, mm), пунктах (points, pt) и цицеро (picas, pc). При аппроксимации требуемых размеров зачастую браузер пользователя лучше ориентируется в абсолютных значениях.

Свойство position

Свойство position используется для определения способа размещения объекта на странице. Объекты могут быть фиксированными, абсолютными, относительными, статическими и плавающими. *Плавающий объект* — это объект, размещаемый при использовании атрибута float, то есть он интерпретируется иначе, чем объекты, позиционирование которых осуществляется при помощи атрибута position.

Синтаксис этого свойства:

```
position: static | relative | absolute | fixed | inherit
```

Контейнер static располагается в соответствии с обычным расположением документа, то есть контейнер/объект будет перемещаться в зависимости от наличия на экране места, как в обычном HTML-файле. В отношении статически позиционированных объектов не используются атрибуты left и top. Значение relative приводит к размещению объекта в соответствии с обычным расположением документа, но смещает его относительно обычного положения. При использовании значения absolute позиция объекта строго определяется свойствами left, right, bottom и top или лишь свойствами left и top совместно со свойствами width и height. Объекты со значением fixed позиционируются аналогично абсолютно позиционированным объектам, но «привязка» идет к стабильному объекту (тому, который не перемещается). Обычно фиксированный объект устанавливается в неперемещаемом слое в видимой части окна web-браузера. Например, объект, установленный на фиксированную позицию, будет оставаться на своем месте даже при прокрутке документа.

ПРИМЕЧАНИЕ

Объекты со значением inherit будут наследовать значения свойств их «родителей». Например, если родительский элемент использовал абсолютное позиционирование, то данный элемент также будет использовать абсолютное позиционирование.

Свойство left

Свойство left указывает расстояние от левого края объекта до левого края содержащего его контейнера. Как уже говорилось, это может быть внешний край видимой области окна браузера или граница другого элемента, содержащего данный объект. Синтаксис этого свойства:

```
Left: <length> | <percentage> | auto | inherit
```

Когда в качестве значения используется <length>, то указывается конкретное смещение от края контейнера в выбранных единицах измерениях. Это означает, что расстояние от левого края объекта до края содержащего его блока будет составлять x единиц измерения. Значение percentage указывает процентное смещение от соответствующей стороны контейнера. Значение auto является значением по умолчанию. В этом случае производится автоматическое вычисление смещения в зависимости от ширины данного объекта.

ПРИМЕЧАНИЕ

Общей единицей, используемой для позиционирования слоев на web-сайтах, является пиксел. Это, скорее всего, связано с тем, что пиксел является стандартной единицей измерения в компьютерной графике.

Свойство top

Свойство top указывает расстояние от верхнего края контейнера объекта. Этот параметр может использоваться при абсолютном позиционировании, совместно с параметрами left, width и height или left, right и bottom. Для точного позиционирования объекта в браузерах 4-й версии требуется не менее четырех свойств. Синтаксис данного свойства:

top: <length> | <percentage> | auto | inherit

Возможно точное указание расстояния от верхней границы контейнера до данного объекта в выбранных единицах измерения, указание в процентном соотношении, а также автоматическое вычисление смещения на основе высоты текущего объекта.

Свойство right

Свойство right указывает расстояние от правого края объекта до правого края содержащего его блока. Вспомните, что это может быть внешним краем видимой области окна браузера или краем другого элемента, содержащего текущий объект. Синтаксис этого свойства:

right: <length> | <percentage> | auto | inherit

Возможно точное указание расстояния от правой границы контейнера до данного объекта в выбранных единицах измерения, указание в процентном соотношении, а также автоматическое вычисление смещения на основе ширины текущего объекта.

Свойство bottom

Свойство bottom указывает расстояние от нижнего края объекта. Это значение может использоваться при абсолютном позиционировании совместно с параметрами left, right и top. Для точного позиционирования объекта в браузерах 4-й версии требуется не менее четырех свойств. Синтаксис данного свойства:

bottom: <length> | <percentage> | auto | inherit

Значение bottom может быть указано в определенных единицах измерения, что говорит о том, что от нижнего края объекта до нижнего края содержащего его блока будет x единиц. Возможно указание процентного соотношения этого расстояния, а также автоматическое определение смещения в зависимости от высоты объекта.

Свойство width

По умолчанию значение width для каждого элемента контейнера вычисляется автоматически в зависимости от наследуемой ширины элемента. Для «картинки» это значение задано явно, поскольку оно определено художником-дизайнером.

Свойство width предназначено для установки ширины контейнера элемента и, следовательно, к нему можно обращаться с помощью элемента <style> едва ли не каждого элемента. При размещении элемента в слое значение width используется не только для указания размера блока назначения, но также влияет на местоположение остальных относительно или статически позиционируемых объектов. Синтаксис данного свойства:

```
width: <length> | <percentage> | auto | inherit
```

Значение width может быть представлено определенным значением, выраженным в единицах измерения, которые определяют относительное или абсолютное расстояние между элементами. Действительными значениями могут быть, например, 10 px, 1 cm, 2 ex, 10 mm и т. д. При использовании percentage указывается процентное соотношение размера исходного контейнера, в котором вы хотите представить данный объект. При использовании слоя это является процентным соотношением размера родительского контейнера, в качестве которого может выступать как видимая область окна браузера, так и другой слой.

Свойство height

По умолчанию высота каждого контейнера автоматически вычисляется на основе наследуемой высоты содержания элемента. Наследуемая высота изображения — это его физический размер, например 300×600 пикселов. Для текстовых блоков и других объектов, не имеющих заранее определенного размера, наследуемая высота вычисляется на основе значений HTML-свойств min-height, max-height и height. Каждое из этих свойств может перекрываться значением CSS-свойства height.

Свойство height предназначено для перекрытия наследуемой высоты содержания элемента, и, следовательно, к нему можно обращаться с помощью элемента <style> едва ли не каждого элемента. Свойство height определяет высоту контейнера элемента. При использовании слоев свойство height не только задает размер блока, но и влияет на расположение остальных относительно или статически позиционируемых объектов. Его синтаксис:

```
height= <length> | <percentage> | inherit
```

Значение height может быть представлено определенным значением, выраженным в единицах измерения, которые определяют относительное или абсолютное расстояние между элементами. Действительными значения могут быть, например, 3 px, 1 cm, 5 ex и т. д. При использовании percentage указывается процентное соотношение размера исходного контейнера, в котором вы хотите представить данный объект. При использовании слоя это является процентным соотношением размера родительского контейнера, в качестве которого может выступать как видимая область окна браузера, так и другой слой.

Свойство z-index

Это свойство используется только с блоками, имеющими свойства позиционирования. Свойство z-index выполняет две роли. Первая заключается в управле-

нии порядка наложения (stacking order) позиционированных контейнеров. Вторая роль заключается в определении, может ли блок использоваться для определения локального контекста наложения. Синтаксис свойства:

```
z-index = auto | <integer> | inherit
```

Значение этого свойства, представленное целым числом (integer), может иметь как положительные, так и отрицательные значения. Самый нижний слой будет иметь значение индекса 0. Слой с индексом, равным 1, будет располагаться над содержанием слоя с индексом 0. При использовании значений auto или inherit позиция слоя будет соответствовать значению родительского элемента.

Теперь, после того, как вы изучили все свойства позиционирования CSS, необходимо рассмотреть отдельные примеры кода, которые, собственно, и создают в требуемом месте экрана пользователя слои с их содержимым.

Полное определение слоя

Каждый слой документа должен идентифицироваться своим элементом . Этот элемент совместно с HTML-атрибутами id и class обеспечивают общий механизм определения структуры документа. Элемент — это элемент-контейнер, который не отображается на экране. Это делает его идеальным средством для создания слоев, при использовании которого вам не придется беспокоиться об отображаемых границах или о проблемах выравнивания вложенных элементов.

Элемент , согласно требованиям Интернет-консорциума, определен при помощи следующих атрибутов:

- align (align=center | justify | left | right): применяется для управления горизонтальным и вертикальным выравниванием текста и объектов в пределах документа.
- class (class=*"cdata-list"*): предназначен для назначения имени класса данному элементу. Браузер использует эти классы для объединения определенных типов информации для упрощения их последующего использования.
- href (href= *url*): предназначен для идентификации определения Интернет-сайта, связанного с данным документом.
- id (id=*"имя"*): предназначен для назначения элементу имени. Это имя может использоваться для идентификации слоя в сценарии при выполнении с ним различных действий.
- lang (lang=*"код_языка"*): этот атрибут указывает язык, используемый при создании элемента или изменении его значений.
- style (style="*описатели_стиля*"): предназначен для применения определенного стиля к указанному элементу.
- title (title=*текст*): позволяет назначить элементу заголовок.

ПРИМЕЧАНИЕ

Следующие атрибуты действительны в отношении элемента только при использовании IE 4 или 5. Предыдущие атрибуты «работают» с браузерами компаний Netscape и Microsoft начиная с 4-й версии.

- onClick (onClick=*script*): этот атрибут приводит к генерации указанного события при щелчке указывающего устройства на данном элементе.
- onDblClick (onDblClick=*script*): этот атрибут активизируется при двойном щелчке на элементе.
- onKeyDown (onKeyDown=*script*): этот атрибут применяется для генерации события при нажатии клавиши, когда этот элемент находится в фокусе.
- onKeyPress (onKeyPress=*script*): этот атрибут применяется для генерации события при полном нажатии — когда клавиша нажата и отпущена при обращении к данному элементу.
- onKeyUp (onKeyUp=*script*): этот атрибут применяется для генерации события при отпускании клавиши при обращении к данному элементу.
- onMouseDown (onMouseDown=*script*): этот атрибут применяется для генерации события при нажатии клавиши мыши над элементом.
- onmousemove (onmousemove=*script*): этот атрибут применяется для генерации события при перемещении мыши даже не над элементом.
- onMouseOut (onMouseOut=*script*): этот атрибут применяется для генерации события при выходе курсора мыши за пределы элемента.
- onMouseOver (onMouseOver=*script*): этот атрибут применяется для генерации события при перемещении мыши над элементом.
- onMouseUp (onMouseUp=*script*): этот атрибут применяется для генерации события при отпускании клавиши мыши над элементом.

Двумя основными свойствами, позволяющими определить ваши слои, являются id и style. Каждое из этих свойств используется при организации взаимодействия, обеспечивая уникальный идентификатор и метод идентификации настроек различных атрибутов CSS внутри объявления элемента. Свойство id является основным идентификатором объекта. Оно позволяет выбирать и перемещать отдельный слой, не производя никаких изменений с другими объектами. Если свойство id не указано, то вы не сможете изменить содержание слоя и не сможете осуществить его анимацию. Сделать это можно только со слоем, имеющим идентификатор id. В отдельных случаях возможна анимация всех объектов определенного класса (class), но только тогда, когда все объекты должны быть изменены одинаковым образом.

Атрибут style предназначен для обеспечения доступа к свойствам CSS непосредственно при определении элемента. Для этого, конечно, можно использовать атрибут id и отдельную таблицу стиля, но браузеры Netscape иногда игнорируют их подсоединение. В главе 3 мы уже обращали внимание, что основным назначением атрибута style при работе со слоями и другими объектами является определение точной позиции объекта в документе. Но помимо этого данный атрибут может содержать и другие свойства, определяющие внешний вид элемента.

В нашем примере документа определены следующие слои:

Зевс на троне

```
<div id="zeuslayer"
    style="position:absolute;
       left:521px; top:294px;
       width:113px; height:119px;
```

```
        z-index:7">
    <img src="IMAGES/Zeus.gif" width="97" height="129">
  </div>
```

Зевс и контур реплики

```
<div id="zeusbbllayer"
    style="position:absolute;
        left:397px;  top:327px;
        width:150px; height:75px;
        z-index:8;
        background-image:url(IMAGES/zbubble1.gif) ">
    <p align="center"><br>
    This is test<br>
    text.</p>
  </div>
```

Зевс и молния

```
<div id="zeuslightening"
    style="position:absolute;
        left:519px; top:273px;
        width:40px; height:75px;
        z-index:9">
    <img src="IMAGES/zlightening.gif" width="40" height="75">
  </div>
```

Гермес

```
<div id="hermeslayer"
    style="position:absolute;
        left:18px; top:372px;
        width:82px; height:77px;
        z-index:3">
    <img src="hermesshoe.gif" width="90" height="96">
  </div>
```

Гермес и контур реплики

```
<div id="hermesbbllayer"
    style="position:absolute;
        left:47px;  top:312px;
        width:80px; height:100px;
        z-index:6;
        background-image:url(IMAGES/hbubble1.gif) ">
    <p align="center"> This<br>
    is test<br>
    text. </p>
  </div>
```

Меню

```
<div id="menulayer"
    style="position:absolute;
        left:10px;  top:91px;
        width:120px; height:140px;
        z-index:1">.
```

```
    <font color="black" size="+1" style="arial">
    <p><a class=leftMenuMore href="/"
     onMouseOver = "popUp('elMenu1' ,event)"
     onMouseOut = "popDown('elMenu1')">Mythology</a>
    <p><a class=leftMenuMore href="/"
     onMouseOver="popUp('elMenu2',event)"
     onMouseOut="popDown('elMenu2') ">Shopping</a>
    </font>
    <p>
     <font color="black" size="+1" style="arial">
     <a class=leftMenuMore href="/"
       onMouseOver="popUp('elMenu3',event)"
       onMouseOut="popDown('elMenu3') ">Acropolis Staff</a>
     </font></p>
  </div>
```

Колоннада

```
<div id="greeklayer"
    style="position:absolute;
       left:153px; top:13px;
       width:480px; height:400px;
       z-index:2;
       overflow: scroll">
   <img src="Greek.gif">
 </div>
```

Текст между колонн

```
<div id="greektext"
    style="position:absolute;
       left:235px;  top:126px;
       width:318px; height:184px;
       z-index:5">
   This is holding text for placement within the Greek colonnades.
 </div>
```

Авторская информация

```
<div id="copyright"
    style="position:absolute;
       left:156px;  top:431px;
       width:472px; height:38px;
       z-index:4;
       overflow: visible">
   <p align="center">
    <font size="-1">
    Copyright 2001, Heather Williamson. All Rights
    Reserved.<br>
    Distributing any of this information without permission is
    expressly prohibited.</font>
   </p>
 </div>
```

На рис. 9.5 обратите внимание на разграничительные линии, показывающие границы расположения каждого слоя, как их представляет Macromedia Dreamweaver и множество других WYSIWYG HTML-редакторов.

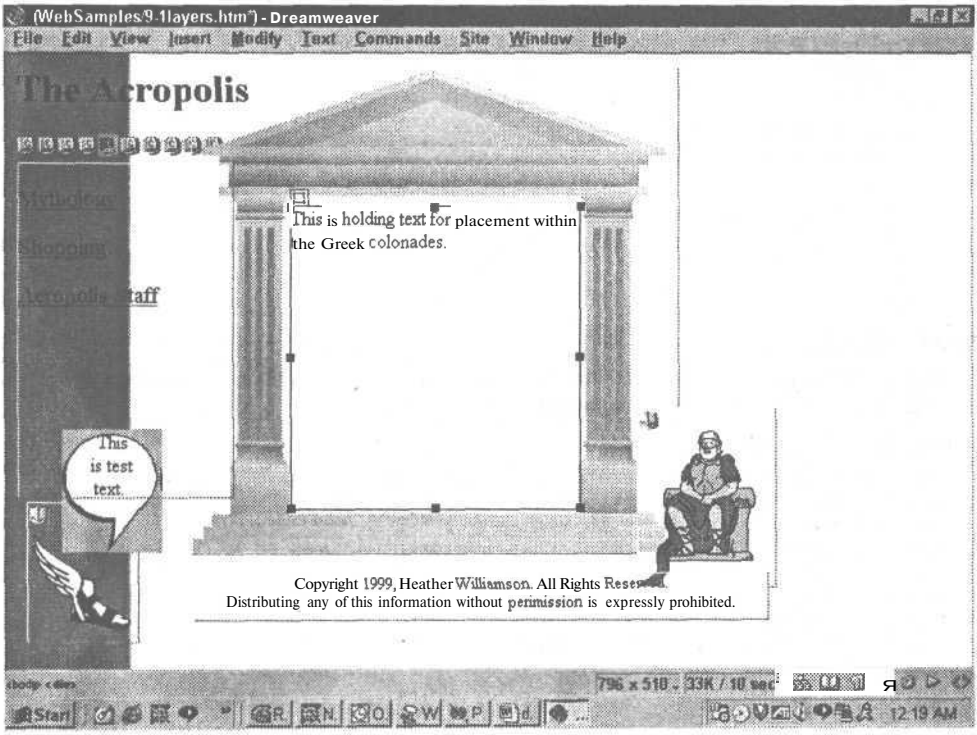


Рис. 9.5. Каждый слой должен быть добавлен в документ в том порядке, в котором вы хотите. Можно удостовериться, что редактор правильно установил значение z-index для каждого слоя

Функция Change Property

Каждое из рассмотренных в данной главе свойств связано с событием, запускающим функцию Change Property (изменить свойство), написанную на языке JavaScript. Эта функция позволяет браузерам вызывать лишь одну функцию и, вместе с этим, с помощью одной функции изменять все доступные свойства. Простейшая функция должна содержать два выражения:

```
function ChangeProp (layerName, theProp, theValue) {
  var usedLyr = eval(doc + '["' + layerName + '"]' + styl);
  eval('usedLyr.' + theProp + '="' + theValue + '"');
}
```

Однако эти выражения не учитывают различия в реализации свойств visible и visibility браузерами Netscape и Microsoft. Поскольку таковое имеется, необходимо предусмотреть дополнительный оператор условного перехода if (для определения, когда используется visible, а когда visibility) и выполнить соответствующие действия для каждого из используемых браузеров.

```
<script language="JavaScript">
<!--
function changeProp(layerName,theProp,theValue) {
```

```
    var usedLyr = eval(doc + '["' + layerName + '"]' + styl);
  if ((theProp == "visibile") || (theProp == "visibility")) {
    if (NS4) {
     theProp = "visible";
    }
    else if (IE4 || IE5 || NS6) {
     theProp = "visibility";
    }
  }
    eval ( 'usedLyr.' + theProp + '="' + theValue + '"');
  }
//-->
</script>
```

ПРИМЕЧАНИЕ

Функция Change Property требует, чтобы браузер был предварительно определен с помощью следующего способа. (Он рассматривался в главе 6.)

```
var bwr = navigator.appName;
var ver = parseInt(navigator.appVersion, 10);
NS4 = ( bwr == "Netscape" && ver == 4 ) ? 1 : 0;
NS6 = ( bwr == "Netscape" && ver == 6 ) ? 1 : 0;
IE4 = ( bwr == "Microsoft Internet Explorer" && ver == 4 ) ? 1 : 0;
IE5 = ( bwr == "Microsoft Internet Explorer" && ver == 5 ) ? 1 : 0;
ver4 = (NS4 || IE4 || NS6 || IE5) ? 1 : 0;
isMac = (navigator.appVersion.indexOf("Mac") != -1) ? 1 : 0;
isDynamic = (NS4 || (IE4 && !isMac) || NS 6 || IE5) ? 1 : 0;

  if (NS4 || NS6) {
   doc = "document";
   styl = "";
  }
  else if (IE4 || IE5) {
   doc = "document.all";
   styl = ".style";
  }
```

После того как вы поместите этот сценарий в подключаемый JavaScript-файл, его можно будет запустить из любого документа сайта. Это позволяет загружать эти функции один раз, а затем, благодаря хранению в кэш-памяти, использовать их непосредственно, а не производить каждый раз их загрузку.

Управление видимостью слоя

В используемом нами в качестве примера сайте имеются девять отдельных слоев, большинство из которых частично перекрываются и перемещаются в ходе выполнения определенных сценариев. Вы должны быть уверены, что при анимации слоев они имеют соответствующую настройку прозрачности. Видимость слоя определяется свойством `visibility`. Оно может использоваться в сценариях, воспроизводимых Netscape Navigator или Internet Explorer, хотя их JavaScript-команды отличаются. Если сделать слой совсем прозрачным, он все равно будет занимать определенное место в вашем документе, поэтому его расположение необходимо учитывать при общей раскладке страницы. Синтаксис свойства `visibility` по умолчанию:

```
visibility: visible | hidden | collapse | inherit
```

Когда это свойство имеет значение visible, содержимое контейнера становится видимым и занимает место в документе. Если используется значение hidden, все содержимое контейнера становится невидимым, но он по-прежнему занимает место в документе и влияет на общую раскладку. Результат использования значения collapse аналогичен результату применения свойства hidden.

ПРИМЕЧАНИЕ

Применение значения collapse к строке или столбцу таблицы приведет к исчезновению строки или столбца, вместо того чтобы сделать их содержание невидимым.

В нашем примере некоторые слои первоначально должны быть невидимы:

- Зевс: контур реплики;
- Зевс: молния;
- Гермес: контур реплики.

СОВЕТ

Когда осуществляется работа с несколькими перекрывающимися слоями, их проще сначала нарисовать на кальке. Тогда легче представить время, когда они должны стать видимыми (или невидимыми), и определить место, в котором они должны появиться.

Иногда эти слои должны быть видимы. Оба контура реплик необходимо показать, когда мышь посетителя окажется над соответствующим символом. Другими словами, когда указатель мыши пользователя окажется над Гермесом, контур реплики должен стать видимым. Аналогично и слой с молнией Зевса должен стать видимым в определенном месте при определенном событии, а затем стать невидимым в другой точке, перед тем как вернуться в исходное положение.

ПРИМЕЧАНИЕ

Каждый слой по умолчанию прозрачный, за исключением области, занимаемой его содержанием. Это значит, что, если вы поместите слой с текстовым содержанием над изображением, текст будет представлен так, как будто он написан на изображении. Если вы решите разместить одно изображение над другим, то часть нижнего слоя будет закрыта верхним слоем. Использование формата GIF с прозрачными фоном — наиболее удачное средство избежать наличия неприятных переходов, образующихся при наложении изображений.

Давайте рассмотрим текст документа, делающий эти слои невидимыми при первоначальной загрузке документа. HTML и CSS могут лишь сообщить браузеру, где необходимо размещать изображения и показывать ли их при первой загрузке. Ни одна из этих технологий не может осуществить перемещение слоя или изменить его видимость в ходе просмотра документа после его загрузки.

Зевс: контур реплики

```
<div id="zeusbbllayer"
    style="position:absolute;
       left:397px; top:327px;
       width:150px; height:75px;
       z-index:8;
       background-image:url(IMAGES/zbubble1.gif);
       visiblity: hidden">
   <p align="center"><br>
   This is test<br>
   text.</p>
</div>
```

Зевс: молния

```
<div id="zeuslightening"
    style="position:absolute;
       left:519px; top:273px;
       width:40px; height:75px;
       z-index:9;
       visiblity: hidden">
   <img src="IMAGES/zlightening.gif" width="40" height="75">
</div>
```

Гермес: контур реплики

```
<div id="hermesbbllayer"
    style="position:absolute;
       left:47px; top:312px;
       width:80px; height:100px;
       z-index:6;
       background-image: url(IMAGES/hbubble1.gif);
       visiblity: hidden">
   <p align="center"> This<br>
   is test<br>
   text. </p>
</div>
```

После того как установлена исходная видимость слоев, можно приступить к составлению сценария, который позволит изменять их видимость в определенное время. Текст сценария будет приведен в следующем разделе.

Обеспечение «динамичности» слоев

После того как определены местоположение и видимость слоев, приступаем к созданию сценария, обеспечивающего интерактивность. Для начала просмотрим список идентификаторов слоев и найдем те, которые имеют динамическое содержание, для того чтобы определить тип требуемого взаимодействия.

Слои без динамического содержания:

- авторская информация;
- Зевс;
- колоннада.

Слои, которые перемещаются:

- Гермес;
- молния Зевса.

Слои, в которых будут изменяться атрибуты:

- Гермес: контур реплики;
- Зевс: контур реплики;
- Зевс: молния.

Слои, в которых будет изменяться текстовое содержание:

- Гермес: контур реплики;
- Зевс: контур реплики.

Просмотрев этот список, можно сделать вывод, что имеется возможность разбить сценарий на группы по реализации различных видов взаимодействия. Можно создать отдельную функцию, которая будет перемещать все слои, функцию, осуществляющую изменение значений отдельных свойств и другие изменения в содержании слоя. Таким образом, требуется создать лишь три основные функции, передавая которым определенные параметры можно осуществлять требуемые изменения в документе.

После определения слоев можно перейти непосредственно к написанию функций, осуществляющих необходимые действия.

Анимация слоев

Когда пользователь представляет себе DHTML-сайт, то зачастую рассчитывает увидеть информацию, перемещающуюся по экрану либо при использовании событий таймера, либо за счет специальных событий. Анимацию можно инициировать с помощью событий загрузки, выгрузки, перемещения мыши, щелчков мышью, клавиатуры или по истечении определенного временного интервала.

Давайте рассмотрим общие принципы перемещения слоя. После того как мы один раз переместим слой, мы легко сможем вызвать аналогичное перемещение, инициируемое новым событием или через определенный интервал времени.

Перемещение

Осуществление перемещения слоя достигается за счет определения левого верхнего угла слоя, а затем установления для этого угла нового местоположения. Этот способ реализуется как при использовании Internet Explorer, так и Netscape Navigator, хотя необходимо внимательно проверить результаты выполнения перемещения каждым браузером.

В начале сценария необходимо определить переменные, с помощью которых будет осуществляться смещение слоя. Смещение — это число пикселов, на которое мы переместим слой с текущей позиции на новую. Смещение будет определяться переменными offsetX и offsetY. Переменная engaged будет использоваться для предотвращения попытки одновременного перемещения данного объекта другими функциями:

```
var engaged = false
var offsetX = 0
var offsetY = 0
```

Основной функцией будет являться функция dragIt. Ее предназначение - установить должное смещение для просмотра документа браузером, а затем создать цикл, при выполнении которого слой будет плавно перемещаться с одной позиции в следующую под углом в 45 градусов. Конечно же, данное движение можно скорректировать, добавив в функцию дополнительные переменные, но мы рассмотрим этот пример в самом простом варианте:

```
function dragIt(e) {
   engaged = true;
   if (IE4 || IE5) { //Fix IEis left and top properties for use as integer
    leftPos = movingLyr.left;
```

```
    topPos = movingLyr.top;
    chkLeftLength = leftPos.length; //Получение значения left
    chkTopLength = leftPos.length; // Получение значения top
    if (chkLeftLength !== -1) { // Установить смещение на 1 для .left
     offsetX = eval(xpos) - leftPos.substring(0,chkLeftLength-2);
    }
    if (chkLeftLength !== -1) { // Установить смещение на 1 для .top
     offsetY = eval(ypos) - topPos.substring(0,chkTopLength-2);
    }
   } else if (NS4 || NS6) { //NS и так использует целые числа.
    offsetX = eval(xpos) - movingLyr.left;
    offsetY = eval(ypos) - movingLyr.top;
   }
   if (engaged) {
    for (var i=0; i<400; i++) {
    moveIt(e);
    }
    offsetx=0;
    offsetY=0;
    engaged = false
   }
}
```

После присвоения переменной engaged значения true, свидетельствующего, что данный слой задействован, необходимо предусмотреть возможность «обхода» разницы определения свойств .left и .top в Netscape Navigator и в Internet Explorer. Браузер Netscape представляет значения обоих свойств в виде целых чисел. А Internet Explorer — в виде строки, содержащей цифровое значение положения слоя, плюс обозначение px. Поскольку браузер Netscape представляет целые числа, то для определения смещения эти результаты можно непосредственно использовать в свойствах layerName.left и layerName.top. В то время как для браузера Microsoft сначала надо с помощью оператора условного перехода удалить единицу измерения и только потом осуществлять математические операции. Данное выражение оператора if осуществляет определение длины и возвращает значение, укороченное на два символа (px), а затем использует полученную переменную в дальнейших выражениях.

После назначения значения смещения необходимо создать цикл или ряд циклов, которые переместят слой к новому месту назначения. Для этого используется выражение for, с переменной цикла i, которая увеличивается на единицу 400 раз. Одновременно она уменьшает значения смещения (размещенные в функции moveIt) на то же значение. Каждый раз при выполнении цикла вызывается функция movelt, которая и осуществляет перемещение и уменьшения значений смещения.

В функции movelt реализована простая пошаговая процедура, которая добавляет сокращенное смещение к текущему положению слоя и осуществляет назначение этих значений свойствам слоя. После чего смещения снова уменьшаются. Заключительное выражение — setInterval — определяет свойство синхронизации для функции movelt таким образом, что вы действительно можете увидеть перемещение слоя по экрану. Если этому параметру присвоить значение fast, то вы не успеете заметить перемещение слоя или просто увидите смазанное пятно, а на некоторых мониторах лишь увидите появление слоя на новом месте. Используемое выражение — t = setInterval ("moveIt()",500) — назначает перемен-

ной t интервал ожидания (полсекунды), через который будет осуществлено повторное выполнение функции moveIt, представленной ниже:

```
function moveIt(e)  {
     movingLyr.top = eval(ypos) + offsetY;
    movingLyr.left = eval(xpos) + offsetX;
    offsetX--;
    offsetY--;
    t = setInterval("moveIt()",500);
 }
```

В предыдущей функции для корректировки отклонений в определении значений .left и .top браузером Internet Explorer пришлось использовать оператор условного перехода. В следующей функции для осуществления перемещения слоя с молнией нам также придется использовать специальное выражение для перехвата событий слоя Zeus (Зевс). Internet Explorer не поддерживает метод captureEvents, поэтому используемое специально для IE событие относится непосредственно к изображению Зевса. Представленный ниже сценарий использует оператор eval для определения перемещаемого слоя и слоя, от которого браузеру Netscape необходимо перехватывать события.

ПРИМЕЧАНИЕ

Перехват событий таким образом позволяет браузеру Netscape Navigator использовать события мыши за пределами обычного ограничения их использования только в ссылках <a>.

```
<SCRIPT LANGUAGE="JavaScript">
 movingLyr = eval(doc + ' ["zeuslightening"]' + styl);
 clickingLyr = eval(doc + ' ["zeuslayer"]' + styl);
 if (NS4 || NS6) {
  clickingLyr.captureEvents(Event.CLICK);
  clickingLyr.onClick = dragIt;
 }
 </SCRIPT>
```

Netscape требует использовать метод captureEvents для перехвата событий, связанных с изображениями или слоями. Internet Explorer может использовать HTML-событие в качестве «атрибута», прикрепленного непосредственно к изображению, но не может использовать события, связанные со слоями. Как видно из нижеприведенной части кода, к элементу <img> для определения способа инициирования перемещения слоя с молнией добавлено выражение onClick="dragIt(event)".

```
<div id="zeuslayer"
    style="position:absolute;
       left:521px; top:294px;
       width:113px; height:119px;
       z-index:5">
   <img src="images/Zeus.gif" width="97" height="129"   onClick="dragIt(event)">
  </div>
```

После того как два данных примера кода будут объединены в одном web-документе, будет создана основа нашего сайта-примера, в котором при каждом щелчке на изображении Зевса по экрану будет перемещаться молния. Еще один из способов изменения внешнего вида сайта заключается в замене одного изображения другим при наступлении определенного события или через определенный временной интервал.

Замена изображений

Одним из распространенных эффектов динамических сайтов является замена изображений (swapping images), вызванная перемещением или щелчком мыши. Реализовать этот эффект довольно просто, но для разных браузеров необходимо воспользоваться различными способами.

Internet Explorer может непосредственно связывать события onClick, onMouseOver и onMouseOut с изображением. Для того чтобы браузеры Netscape могли сделать то же самое, необходимо включить все изображения или области, в которых будет осуществляться замена изображений, в элемент <a>. Для этот надо просто создать пустую ссылку href (href="#"). Если пользователь щелкнет на такой ссылке, то сценарий сможет перехватывать ее, а сама ссылка не вызовет перехода на другую страницу.

Как видно из приведенного ниже листинга, события onMouseOver и onMouseOut связаны с пустой ссылкой, поэтому этот пример будет работать с обоими браузерами. При вызове функции swapImage ей передаются две переменные: name, являющаяся идентификатором изображения, которое должно быть заменено (test), и идентификатор изображения, которым будет произведена замена (0).

```
<a href="#"
    onMouseOver="swapImage('test','0')"
    onMouseOut="swapImage('test','1')">
 <IMG name="test"
    src="helmet1.gif" >
 </a>
```

СОВЕТ

Создавая сайт, одинаково воспроизводимый браузерами как Netscape, так и Microsoft, необходимо помнить, что атрибут id, поддерживаемый Microsoft как часть элемента <img>, не поддерживается браузерами Netscape. Для обеспечения совместимости необходимо использовать атрибут name. Применение обоих атрибутов с точки зрения HTML 4.01 неправильно, но оба браузера нормально обрабатывают такую комбинацию. _

В нижеприведенном примере заменяемые изображения сначала загружаются в массив, а затем при вызове функции обращение к ним осуществляется с помощью второй переменной. Для того чтобы к массиву могли обращаться обе функции (swapImage и preloadImages), необходимо объявить массив глобальной переменной.

```
var preloadArray = new Array()
```

ПРИМЕЧАНИЕ

Существует иной способ предварительной загрузки изображений. В приведенном ниже примере производится отдельная загрузка каждого используемого изображения. Это подходящий способ для сайта, который содержит небольшое количество изображений, но он совершенно недопустим для сайтов с большим числом изображений.

```
img1on = new Image();      // изображения для события onMouseOver
 imglon.src = "image1on.jpg";
 img2on = new Image();
 img2on.src = "image2on.jpg";
 img3on = new ImageO;
 img3on.src = "image3on.jpg";
 img4on = new ImageO;
 img4on.src = "image4on.jpg";
```

```
img5on = new Image();
img5on.src = "image5on.jpg";

img1off = new Image();          // изображения для события onMouseOut
img1off.src = "image1off.jpg";
img2off = new ImageO;
img2off.src = "image2off.jpg";
img3off = new ImageO;
img3off.src = "image3off.jpg";
img4off = new ImageO ;
img4off.src = "image4off.jpg";
img5off = new ImageO;
img5off.src = "image5off.jpg";g
```

При использовании этого метода приведенная далее функция swapImage должна быть изменена соответствующим образом. Для присвоения нового значения атрибуту *src* элемента image в сценарии необходимо использовать следующее выражение: document[imgName].src = eval(imgName +".src"). При данном способе предварительной загрузки изображений при вызове функции необходимо с помощью сценария передавать ей только имя изображения без указания индекса массива: onMouseOver="swapImage(imgName)".

После предварительной загрузки необходимо определиться с заменой картинок «на лету». В представленной функции swaplmage необходимо проверить, содержит ли документ изображения, а затем необходимо определить тип браузера. Если используется IE, то необходимо оценить количество объектов документа и точно определить изображение, которое будет изменяться. После этого необходимо сделать так, чтобы свойство объекта src было равно значению src определенного элемента массива. Netscape Navigator позволяет автоматически одним выражением связывать свойство src элемента массива с объектом.

ПРИМЕЧАНИЕ

Выражение, «понимаемое» Netscape; (document[imgName].src = preloadArray[arrayID].src), будет работать и с IE 5, но ие со всеми версиями IE 4, в которых подразумевается наличие оператора eval в левой части данного выражения.

```
function swapImage(imgName,arrayID) {
  if (document.images) {
   if(IE4 || IE5){
    workingImg = eval(doc + '["' + imgName + '"]');
    workingImg.src = preLoadArray[arrayID].src;
   } else if (NS4 | | NS6) {
    document[imgName].src = preLoadArray[arrayID].src;
    }
  }
 }
```

После загрузки этой функции у вас имеется возможность для замены изображений использовать таймер. Это упростит чередование изображений вашего сайта, а также позволит посетителю увидеть последовательный показ различных изображений. В следующем подразделе использование таймера рассмотрено подробно.

Использование таймера для генерации события

Помимо обычной, основанной на событиях, модели взаимодействия для запуска большинства функций можно использовать таймер. Просмотрите некоторые

функции, которые используют события в качестве отправного толчка, например функцию dragIt (рассмотренную в предыдущем разделе). Большинство из них может запускаться как от таймера, так и от события. Функции, использующие события для обработки информации, могут замедлять работу вашей системы и прерывать вывод оставшейся части документа.

Наиболее простой способ установки временной привязки — идентифицировать массив или последовательность массивов, содержащих сведения, которые необходимо чередовать. Например, это может быть временной интервал, с которым должна «срабатывать» определенная функция, и имя этой функции. Вы можете определить свой массив двумя способами. Первый — объявить два параллельных массива и назначить им соответствующую информацию. Например, в массиве timeArray[0] вы могли бы задать целое значение временного интервала, с которым была бы выполнена первая функция, а в массиве eventArray[0] вы бы поместили вызов функций, которые должны выполняться. В данном случае не составляет особого труда связать два массива, но ввод данных будет осуществляться разработчиком:

```
var timeArray = new Array(5);
var eventArray = new Array(5);
```

Более предпочтительным способом при создании двухмерного массива является написание функции, осуществляющей автоматическое внесение требуемых данных в каждую ячейку массива. Например, нижеприведенная функция берет две переменные и назначает их значения определенным свойствам текущего объекта. Функция вызывается с помощью нескольких выражений, которые сначала помещают информацию в основной массив, в данном примере названный timingArray. Переменная timingArray определена как новый массив из 5 элементов. Число элементов массива для размещения компонента каждого временного события может быть скорректировано.

```
function setTiming(timeFrame, fctnName)
   this.timeFrame = timeFrame;
   this.fctnName = fctnName;
 }
 timingArray = new Array(5)
```

После определения массива в него необходимо внести соответствующую информацию. В двухмерном массиве назначается индекс для основного массива timingArray, а затем осуществляется его заполнение указанием ссылок на свойства, указанные в функции setTiming. Нижеприведенный код осуществляет заполнение массива при помощи функции setTiming.

ПРИМЕЧАНИЕ

В массивах нумерация начинается с 0, поэтому пять элементов массива пронумерованы от 0 до 4.

```
timingArray[0] = new setTiming(i1i, "changeProp('zeuslayer','visible',
'visible')");
 timingArray[1] = new setTiming(i4i, "changeProp('greektexti,'visible',
'visible')");
 timingArray[2] = new setTiming(i10i, "changeProp('hermeslayer',
'visible','visible')");
 timingArray[3] = new setTiming(i15i, "dragIt(event)");
 timingArray[4] = new setTiming(i19i, "swapImage('hermeslayer',i1i)");
```

СОВЕТ

Данный массив построен таким образом, что последовательность размещения элементов в массиве совпадает с временной последовательностью событий. Сценарий построен так, чтобы обращение осуществлялось «строка за строкой» без необходимости просмотра всего массива, иначе может получиться бесконечный цикл.

После того как этот массив заполнен, необходимо обеспечить автоматический вызов функции таймера при помощи события onLoad элемента body. Это позволит запустить таймер сразу после загрузки документа, и вам не придется запускать его отдельно с помощью сценария.

ПРИМЕЧАНИЕ

Использование для инициации событий языка SMIL (*Synchronized Multimedia Integration Language*) — это наиболее надежное и подходящее решение, но использовать его можно только тогда, когда эта технология будет поддерживаться новыми версиями браузеров. SMIL — это основанный на XML язык, позволяющий определять в HTML-документе время появления и воспроизведения изображений, текста, аудио-, видео- и анимационных файлов. Более полную информацию о SMIL можно получить на web-сайте Интернет-консорциума: http://www.w3.org/AudioVideo/. А в данном примере переменные сначала необходимо инициализировать:

```
<BODY onload=startTimer()>
```

Первая из них, timeCount, сбрасывает таймер в 0. Эта переменная отслеживает число прошедших временных интервалов. Переменная i является индексом массива. Уменьшение ее значения производится значительно медленнее, чем значение переменной timeCount, поскольку она получает приращение лишь по заполнению последней. Как видно из нижеприведенной функции, таймер содержится внутри цикла while, который выполняется, пока timerRunning не получит значение false. Переменная timerRunning введена для обеспечения работы цикла. Она устанавливается в false после завершения обработки всех событий. Это предотвращает «зацикливание» сценария, что расходует ресурсы компьютера.

```
timeCount = 0;
 i = 0;
 timerRunning = false;
```

Функция startTimer() используется для запуска цикла, который осуществляет чтение архива до тех пор, пока не будет исчерпана ранее определенная переменная timecount. Необходимо подкорректировать настройки цикла for, чтобы он не закончился до того, как будут запущены все события.

Если событие, находящееся в timingArray.timeframe, будет запущено одновременно с переменной timecount, то функция посчитает, что этот элемент массива уже запущен, и увеличит индекс. Затем, за пределами выражения if, значение таймера будет инкрементировано. Перед увеличением значения таймера будет проверен массив. Если конец массива еще не достигнут, то переменной timecount будет присвоено значение 30 и выполнение цикла for завершится. При существующем значении переменной цикла (100) эта функция будет запускаться каждую десятую долю секунды. Нижеприведенный код управляет временной синхронизацией перемещения элементов 10 раз в секунду.

```
function startTimer() {
  for (timecount=0; timecount<30; timecount++){
   alert('current time: ' + timecount);
   if (timingArray[arraystep].timeFrame == timecount) {
```

```
        eval(timingArray[arraystep].fctnName);
        if (arraystep == (arraylength-1)) {
         timecount=30;
        } else {
         arraystep++;
        }
       }
       t=setInterval("startTimer()",100);
  }
}
```

Обеспечив таким образом интерактивное изменение слоев и изображений, мы только приступили к изучению того, что можно сделать с вашими web-сайтами. В следующем разделе мы рассмотрим, как изменить свойства слоев и их поведение.

Определение «поведения» слоев

Процесс связывания поведения со слоем обычно означает изменение значения свойства или изменение содержимого экрана при возникновении определенного события или наступления события через определенный интервал времени.

Изменение атрибутов

Одно из основных назначений языка JavaScript на динамических сайтах заключается не в назначении атрибутов HTML или CSS, а в их изменении. Изменение атрибутов видимых HTML-объектов или их CSS-свойств позволяет производить изменение внешнего вида сайта в момент выполнения, а следовательно, обеспечивать способ взаимодействия с посетителем. Существует пять областей в CSS и HTML, изменения которых одинаково поддерживаются браузерами Netscape и Microsoft (реализация DOM корпорации Microsoft позволяет изменять и другие HTML-атрибуты и CSS-свойства, но они будут проигнорированы браузерами Netscape (см. приложение 3)):

- видимость;
- размеры;
- значения индексов;
- свойства фона;
- обрезка изображений.

Изменение видимости

Одним из наиболее популярных эффектов динамического сайта является появление и пропадание различных элементов. Этот эффект реализуется за счет изменения свойства `visibility`. Сценарий производит переключение этого свойства в качестве реакции на определенные события.

В Netscape Navigator 4 реализовано свойство `visible`, а в Internet Explorer — visibility. В Netscape Navigator 6 уже тоже используется свойство `visibility`. Следовательно, для каждого браузера необходимо обеспечить свой механизм изменения видимости. Например, добавление к рассмотренной более подробно

в данной главе функции ChangeProperty следующего кода приведет к более точному выполнению:

```
if (NS4) object.visible= visible;
else object.visibility=visible;
```

При использовании исходной функции смены свойств видимость можно изменить следующим выражением:

```
ChangeProp('LayerName','visibility','value');
```

Эта функция может быть вызвана с помощью события, связанного с определенным HTML-элементом. Например, нижеприведенное выражение при помощи ссылки <a> осуществляет вызов функции ChangeProp, когда над ссылкой будет указатель мыши.

```
<a onMouseOver="ChangeProp('hermesbbllayer','visible','visible')"
onMouseOut="ChangeProp('hermesbbllayer','visible','hidden')"
name="bkgbutton"> Add something here</a>
 <a onmouseover="ChangeProp('zeusbbllayer','visibility','visible')"
onMouseOut="ChangeProp('zeusbbllayer','visible','hidden')"
name="bkgbutton"> Add something here</
```

Этот вызов функции сопровождается передачей в нее трех переменных: Layerl — наименование слоя, visibility или visible — название свойства, и hidden — значение, которое должно быть назначено данному свойству. После выполнения функции текущий слой будет «спрятан». Далее с данным слоем можно производить любые изменения, но слой останется невидимым, пока его свойству visibility не будет присвоено значение visible.

Изменение положения

Основными свойствами, определяющими местоположение слоя, являются top и left. Эти свойства определяют, где будет находиться верхний левый угол слоя. С их помощью можно установить исходное местоположение и затем, при необходимости, осуществлять перемещение по экрану.

При помощи рассмотренной ранее функции изменения свойств осуществить перемещение можно выражением:

```
ChangeProp('LayerName','top','value')";
```

Эта функция может быть вызвана событием, связанным с любым HTML-элементом. Например, следующее выражение осуществляет вызов при помощи кнопки <input>:

```
<input type=button onClick="changeProp('ZeusLayer','top','100')"
value="top" name="topbutton">
```

При вызове функции в нее передаются три переменные: ZeusLayer — наименование слоя, top — название свойства, и 100 — новое значение этого свойства. При выполнении этого кода весь слой переместится таким образом, что верхний край слоя будет находиться на расстоянии 100 пикселов от верхней границы документа.

При вызове функции из ссылки будет осуществлено перемещение слоя, в данном случае слоя Гермес, на 100 пикселов вправо от границы видимой области экрана:

```
<a onClick="ChangeProp('HermesLayer','left','available_width - 100')"
name="leftlink"> Move Layer</a>
```

ПРИМЕЧАНИЕ

В данном случае переменная available_width является ранее определенной и ее значение — ширина видимой части окна.

После перемещения слоя, в данном случае после щелчка посетителя на ссылке, слой останется на новом месте до сброса окна или перемещения слоя в другое место.

Использование значения z-index

Если вы хотите поместить в окне несколько слоев и осуществить чередование видимых слоев, то необходимо использовать значение z-index каждого слоя. С его помощью можно определить, какой слой будет находиться «выше» остальных. При частом изменении этого значения можно осуществить «вибрацию» изображений. Также можно осуществить чередование текстового содержания.

Функция ChangeProp может изменять значение свойства z-index при вызове следующим образом:

```
ChangeProp('LayerName', 'zindex', 'value')";
```

В приведенном ниже выражении эта функция помещает слой, содержащий молнию, над слоем, содержащим изображение Зевса. Теперь молния сможет пролететь по экрану, как будто она выпущена Зевсом.

```
<a onClick="ChangeProp('ZeusLightening','zindex','12')"name="zbutton">
</a>
```

После того как этот слон оказался видимым, необходимо использовать соответствующий код для быстрого перемещения его по экрану. А после того, как он достигнет назначения, он должен снова стать невидимым и вернуться в исходное место.

Изменение свойств фона

Зачастую необходимо изменять или устанавливать цвет фона, чтобы сделать прозрачный слой непрозрачным или выделить один слой среди других. Эту процедуру можно использовать для чередования последовательности слоев и изменения цвета фона каждого слоя при указании на части диаграммы. Кроме того, ее также можно использовать для выделения текущего выбора, осуществленного при помощи указателя мыши, и для облегчения восприятия информации. Эти изменения осуществляются с помощью свойства background.

ПРИМЕЧАНИЕ

Netscape поддерживает свойства background и bgcolor, a Internet Explorer — background, backgroundcolor и backgroundimage. Для создания сайта, одинаково воспроизводимого обоими браузерами, необходимо использовать только одно свойство — background или осуществлять изменение названия свойства.

Для смены цвета фона может использоваться следующее выражение:

```
ChangeProp('LayerName', 'background','value')";
```

Для изменения слоя «Гермес: контур реплики» при помещении указателя мыши над изображением Гермеса необходимо использовать следующее выражение:

```
<a onMouseOver="ChangeProp('hermesbbllayer','background','blue')"
name="bkgbutton"> Add something here</a>
```

При использовании этой функции в качестве пункта меню при перемещении мыши над данной ссылкой она изменит цвет контура реплики Гермеса на синий.

Изменение свойства Clipping

Возможность обрезки используется для сокращения размеров исходного изображения. Свойство clip осуществляет корректировку видимой области при просмотре изображения или слоя. Его можно использовать для того, чтобы показать лишь часть изображения. Это удобно при выделении определенной области на схеме или другом изображении для привлечения внимания в ходе рассмотрения, а также для обеспечения анимации изображения. Нижеприведенный вызов функции позволяет вам осуществить обрезку изображения при использовании для инициации этого процесса как события, так и таймера. Для свойства clip предусмотрен только один тип обрезки: rect (top, left, right, bottom).

ПРИМЕЧАНИЕ

Это свойство не работает с Netscape Navigator 4.5 и Internet Explorer 5, но корректно выполняется только Netscape Navigator 6 PreRelease 2.

```
ChangeProp('LayerName','clip','value')";
```

Использование звуков

Звук является наиболее интересным эффектом, который каждым браузером реализуется по-своему. Единственным способом внедрения звука в документ, поддерживаемым обоими браузерами, является использование объекта <embed>. В документ можно встроить сколько угодно звуковых объектов, но каждый из них должен иметь уникальный идентификатор, позволяющий использовать их отдельно. Например, звуки можно назвать в соответствии с исполняемой в них мелодией или в порядке их перечисления на web-сайте:

```
<embed name='siteSound1' src='Titanic.wav'
loop=false autostart=false mastersound hidden=true width=0 height=0>
</embed>
```

Элемент <embed> имеет большой набор атрибутов, представленных ниже с их синтаксисом:

- accesskey: позволяет быстро перейти к требуемому элементу. При использовании формы пользователь сможет быстро ввести необходимые данные. При использовании ссылки — быстро активизировать требуемую ссылку. Для выделения символа, соответствующего «горячей клавише», используется подчеркивание с помощью тега <U></U>.

  ```
  accesskey="символ"
  ```

ПРИМЕЧАНИЕ

Если встраиваемый объект не имеет интерфейса, то данный атрибут функционировать не будет.

- align: определяет позицию объекта относительно границ документа.

  ```
  align= absbottom | absmiddle | baseline | bottom | left | middle |
  right | texttop | top
  ```

- alt: позволяет вставить текст, который будет демонстрироваться при недоступности для пользователя текущего объекта.

  ```
  alt=текст
  ```

- autostart: предназначен для автостарта воспроизведения встроенного объекта, если он поддерживается соответствующим приложением на стороне пользователя.

  autostart= true | false

- bgcolor: управляет цветом фона данного элемента. Можно использовать как название цвета, так и его шестнадцатеричное представление. Bgcolor можно использовать с midi-плейером и другими объектами. Эффект этого атрибута определяется типом объекта.

  bgcolor= colorname | *RGBcolor*

- class: предназначен для назначения элементу имени класса. Браузеры используют классы для объединения определенных типов информации для последующего использования.

  class="*cdata-list*"

- code: указывает имя файла, в котором содержится воспроизводимая информация.

  code= filename

- codebase: определяет основную директорию встроенных объектов, которая используется в качестве основы для указания других адресов.

  codebase= *uri*

- dir: определяет направление текстового потока в документе для того, чтобы web-браузер смог правильно представить текст. Этот атрибут также может использоваться для указания регистра символов.

  dir=LTR | RTL | [CS | CI | CN | CA | CT]

- height: указывает высоту встроенного объекта.

  height= objectHeight

- hspace: определяет ширину боковых интервалов блока, содержащего объект.

  hspace=< *number* >

- id: назначает объекту имя.

  id=" *name* "

- lang: определяет язык, используемый для создания элемента и его значения.

  lang=" language code "

- language: определяет язык, используемый для написания сценария.

  language=javascript | jscript | vbscript | vbs

- loop: определяет возможность повторов выполнения встроенного объекта.

  loop= true | false

ПРИМЕЧАНИЕ

Internet Explorer отказывается от стандартной, общепринятой возможности указания числа повторов, и в нем этот атрибут не работает.

- name: указывает идентифицирующее название приложения или объекта.

  name=" *appname* "

- pluginspage: указывает размещение приложения, которое поддерживает тип вложенного файла. .

  pluginspage=*url*

- src: указывает местонахождение файла, содержащего представляемые данные.

  src=*url*

- style: определяет CSS-свойства для данного элемента.

  style=style descriptors

- tabindex: определяет позицию текущего элемента в общем tab-порядке документа.

  tabindex= *number*

- title: определяет заголовок элемента.

  title= *text*

- units: определяет единицы измерения, используемые при указании значений в атрибутах height и width.

  units= px | em | pt | ex | cm | in | mm | *xx*

- vspace: определяет верхний и нижний пространственные интервалы блока, содержащего объект.

  vspace=< *number* >

- width: определяет ширину блока данного объекта в пикселах и в процентном соотношении к ширине всего окна.

  width=< *number* >

ПРИМЕЧАНИЕ

Не все из перечисленных атрибутов работают в Netscape Communicator

Итак, при помощи <embed>-команды при просмотре документа мы загрузили файл, но воспроизведение указанного звука произойдет только при генерации определенного события. Это можно сделать при помощи вызова такой функции:

```
controlSound('play', 'siteSoundl')
```

Для вызова функции control Sound можно использовать любое событие, поддерживаемое Netscape и Internet Explorer, «привязав» его как к кнопке, так и к ссылке. Полный список «общих» для двух браузеров событий представлен в приложении 3, а здесь приведен список событий, являющихся «наименьшим общим знаменателем», то есть тех событий, которые работают с браузерами начиная с 4-й версии на любой платформе.

onAbort	onError	onLoad	onMouseUp
onBlur	onFocus	onMouseDown	onMove
onClick	onKeyDown	onMouseMove	onResize
onDblClick	onKeyPress	onMouseOut	onUnload
onDragDrop	onKeyUp	onMouseOver	

Функция controlSound вызывается двумя переменными: именем <embed>-элемента, содержащего звук, и командами play или stop. Сценарий сначала проверяет наличие этого объекта, а затем, если посетитель пользуется браузером Netscape, он в соответствии с командой начинает воспроизводить или останавливает воспроизведение указанного звукового объекта. При использовании Internet Explorer звуковой объект «понимает» команды stop или run.

Нижеприведенный сценарий позволяет «обойти» эту разницу:

```
<script language="JavaScript">
<!--
function controlSound(sndAction,sndObj) {
 if (eval(sndObj) != null) {
  if (NS4|| NS6) eval(sndObj+((sndAction=='stop')?'.stop()':'.play(false)'));
  else if (eval(sndObj+".FileName"))
   eval(sndObj+((sndAction=='stop')?'.stop()':'.run()'));
 }
}
//-->
</script>
```

После начала воспроизведения звуковой фрагмент может быть повторен один или множество раз в зависимости от значения атрибута loop во встроенном объекте.

ПРИМЕЧАНИЕ

Многие web-страницы, использующие звуки, выводят небольшую панель управления, представляющую воспроизводящее программное обеспечение, в том месте страницы, где помещен звуковой объект. С помощью этой панели можно остановить воспроизведение звуковых файлов. Если же вы совсем не хотите слышать звуковые объекты, то их необходимо отключить в настройках вашего браузера или выключить колонки.

Использование строки состояния

Текущее состояние может выводиться в строку состояния браузера. Она не всегда замечается пользователем, поэтому туда не стоит помещать информацию, которую посетитель *обязательно должен* увидеть. По умолчанию строка состояния используется для вывода адреса, с которым в настоящее время осуществляется связь. Зачастую строка состояния используется для вывода подсказок к пунктам меню или значкам, которые должны использоваться для перемещения по сайту. Она также используется для предоставления дополнительной информации о происходящих на сайте событиях и авторской информации.

Нижеприведенная функция может быть связана с любым событием или временным интервалом и позволит избежать многократного повторения выражения window.status в вашем документе.

```
showStatusMessage ("Check out the new information in the Charts Area.");
```

Функция showStatusMessage использует одну переменную, с помощью которой в функцию передается текст сообщения, который необходимо поместить в строке состояния окна браузера. Теоретически нельзя использовать кавычки и другие специальные символы.

В нижеприведенной таблице показано, как можно вывести в строку состояния специальные, «запрещенные» символы.

Символ	Результат
\"	Двойная кавычка
V	Одиночная кавычка
\\	Обратная косая черта
\b	Забой
\t	Табуляция
\п	Переход на новую строку
\r	Возврат каретки (Enter)
\f	Переход на новую страницу

В приведенном ниже примере сценария используются отдельные команды для помещения сообщений в строку состояния окна. За счет помещения этих выражений в отдельную функцию их можно вызывать из любого документа без необходимости предварительной загрузки информации. За счет такой возможности вы сократите время на обслуживание сайта при необходимости изменить сообщения или при модификации сценария.

```
<script language="JavaScript">
<!--
function showStatusMessage(messageStr) {
 window.status=messageStr;
 }
//-->
</script>
```

Использование предупреждений

Иногда информация, которую вы хотите сообщить пользователю, может быть им пропущена. Наилучшим способом избежать этого является использование окон сообщений, «выскакивающих» окон. Неправильное использование окон сообщений может вызвать раздражение посетителей, особенно, когда их содержание носит насмешливый характер. Однако эти окна очень полезны, когда вам необходимо сообщить пользователю важную информацию. Например, известить пользователя, что сайт рассчитан на работу с браузерами версии 4.0 и выше и что сейчас ему будет представлен упрощенный вариант страницы.

Представленный ниже вызов функции включает одну переменную, передаваемую в эту функцию. Эта переменная содержит текст, который будет представлен в окне сообщения:

```
PopupMsg ("You do not have a 4.0 browser, so will be taken to an
alternate site.");
```

Как вы уже видели, функция `alert(theMessage)` состоит из одного выражения. После создания функции, содержащей это выражение, вы можете вызывать ее с помощью любого события или через определенный временной интервал. Это облегчит обслуживание всего сайта и добавление новых сообщений.

```
<script language="JavaScript">
<!--
```

```
function popupMsg(theMessage) {
 alert(theMessage);
}
//-->
</script>
```

ПРИМЕЧАНИЕ

«Выскакивающие» сообщения могут явиться большой проблемой для пользователей некоторых браузеров, которые могут блокировать их обработку. А если пользователь использует только текстовый режим, то может оказаться невозможным удалить сообщение с экрана обычным образом, то есть щелчком на кнопке OK.

Изменение содержания слоя

Использование DHTML позволяет изменить содержание отдельных слоев. Вы можете изменить адрес изображения или загрузить на его место новое изображение при щелчке на кнопке или в соответствии с расписанием. На сайте Acropolis мы столкнемся только с изменениями, основанными на событиях. Вы можете использовать вызов тех же функций для изменения содержания при помощи таймера по определенному расписанию.

Существует несколько способов изменения содержания слоя. В Internet Explorer для этого могут использоваться команды innerText, innerHTML, outerText и outerHTML. В браузерах Netscape изменять текстовое содержание слоя также возможно различными способами. Изменять существующее содержание можно с помощью метода layerobject.write. Более совершенными способами являются использование src-свойства слоя и использование функции layer.load().

Давайте рассмотрим два различных способа загрузки содержания в слой. Первый — проще. Он предназначен для записи строк текста в слой. Он не может вставлять содержание из Интернета, только из данной строки. В приведенном ниже примере объект contentBubble определен «на лету» как плавающий слой для контуров реплик Зевса и Гермеса, в которых будет содержаться передаваемый текст.

```
function loadBblContent(newText, layerName) {
  contentBubble = eval (doc + '["' + layerName + '"]' +html);
  if(NS4 | |NS6) {
   contentBubble.write(newText);
   contentBubble.close();
  else {
   contentBubble.innerHTML = newText;
  }
}
```

Как видно, для выполнения соответствующего кода необходимо использовать оператор условного перехода. Браузеры Netscape для размещения информации на слое используют метод .write, а браузеры Microsoft — используют innerHTML. Оба этих способа позволяют помещать только небольшие объемы информации и не позволяют работать с большими объемами (размером с документ). При необходимости замены большого количества информации следует

обратить внимание на более совершенные возможности Internet Explorer и Netscape Navigator.

Для замены больших текстовых фрагментов необходимо использовать src-свойство слоев. Как уже упоминалось, Netscape позволяет загрузить информацию из другого документа в элемент или <layer>. Internet Explorer пока не поддерживает такой возможности. Для использования содержания из другого файла для размещения в слое необходимо добавить новый элемент , содержащий элемент <iframe>.

ПРИМЕЧАНИЕ

Netscape Navigator не поддерживает элемент <iframe>.

Элемент <iframe>, используемый для идентификации внутреннего фрейма (плавающего фрейма), имеет те же атрибуты, что и элемент , но еще и атрибут src, позволяющий загрузить в документ новое содержание. Внутренний фрейм размещается в блоке объекта или в блоке текста, а его содержимое может быть выровнено по окружающему тексту. Браузеры, которые не поддерживают фреймы или имеют настройки, запрещающие их вывод, *не смогут* показать элементы <iframe>.

Приведенный ниже HTML-код создает слой с помощью элемента , идентифицирует его как ieContent и позиционирует его так же, как слой greektext. Новый слой будет использоваться только Internet Explorer. Сначала он загружается как невидимый текст и становится видимым при необходимости загрузки нового содержания.

```
<div id="ieContent"
   style="position:absolute;
      left:235px; top:126px;
      width:238px; height:251px;
      z-index:2"
      visibility="hidden">
 <iframe id="ieframe" frameborder=1 width=1 height=l scrolling=no src="">
</iframe>
```

После создания «заготовки» для браузеров Netscape и Microsoft можно приступить к созданию функции, которая загружает в них содержание. Эта функция — loadGrkContent — позволяет загрузить в слой текстовое содержание из любого документа, указав его URL. В этом случае требуемый URL должен быть передан в документ при помощи пункта меню.

Вначале эта функция определяет используемый браузер. Если им является Internet Explorer, то для загрузки используется элемент <i frame>, но сначала определяется его ширина на экране. Если используется браузер Netscape, то для определения содержания и его размера используется метод layer.load.

```
function loadGrkContent(doc_url) {
  if (IE4 || IE5) {
contentGT.width= 238;
document.all.ieContent.document.frames["ieframe'].document.location.href
= doc_url;
  } else if(NS4 || NS6) {
```

```
    contentGT.load(doc_url, 238);
  }
}
```

Предыдущая функция позволяет разделить содержание вашего сайта на несколько документов, в каждом из которых содержится часть информации, составляющей общее содержание сайта. Нижеприведенный сценарий запускается как часть сайта, поэтому использует ранее определенные слои и переменные. Если эти переменные не будут определены, сценарий работать не будет!

ПРИМЕЧАНИЕ

При помещении этого кода в документ необходимо предварительное определение используемого браузера. Работа функции ChangeProperty основывается на использовании переменных, определенных в данном фрагменте кода!

```
var bwr = navigator.appName;
   var ver = parseInt(navigator.appVersion, 10);
NS4 = ( bwr == "Netscape" && ver == 4 ) ? 1 : 0;
NS6 = ( bwr == "Netscape" && ver == 6 ) ? 1 : 0;
IE4 = ( bwr == "Microsoft Internet Explorer" && ver == 4 ) ? 1 : 0;
IE5 = ( bwr == "Microsoft Internet Explorer" && ver == 5 ) ? 1 : 0;
ver4 = (NS4 || IE4 || NS6 || IE5) ? 1 : 0;
isMac = (navigator.appVersion.indexOf("Mac") .!= -1) ? 1 : 0;
isDynamic = (NS4 || (IE4 && !isMac) || NS 6 || IE5) ? 1 : 0;

  if (NS4 | | NS6) {
   doc = "document";
   styl = "";
   html = " .document";
   xpos = "e.pageX";
   ypos = "e.pageY";
  }
  else if (IE4 | | IE5) {
   doc = "document.all";
   styl = ".style";
   html = "";
   xpos = "event.x";
   ypos = "event.y";
  }
  contentGT = eval(doc + ' ["greektext"]' +html);
```

Общий вид сайта-примера

После того как мы рассмотрели все фрагменты кода, осуществляющие перемещение слоев, анимацию и изменение содержания, настало время взглянуть на весь сайт целиком. Внешний вид сайта представлен на рис. 9.6.

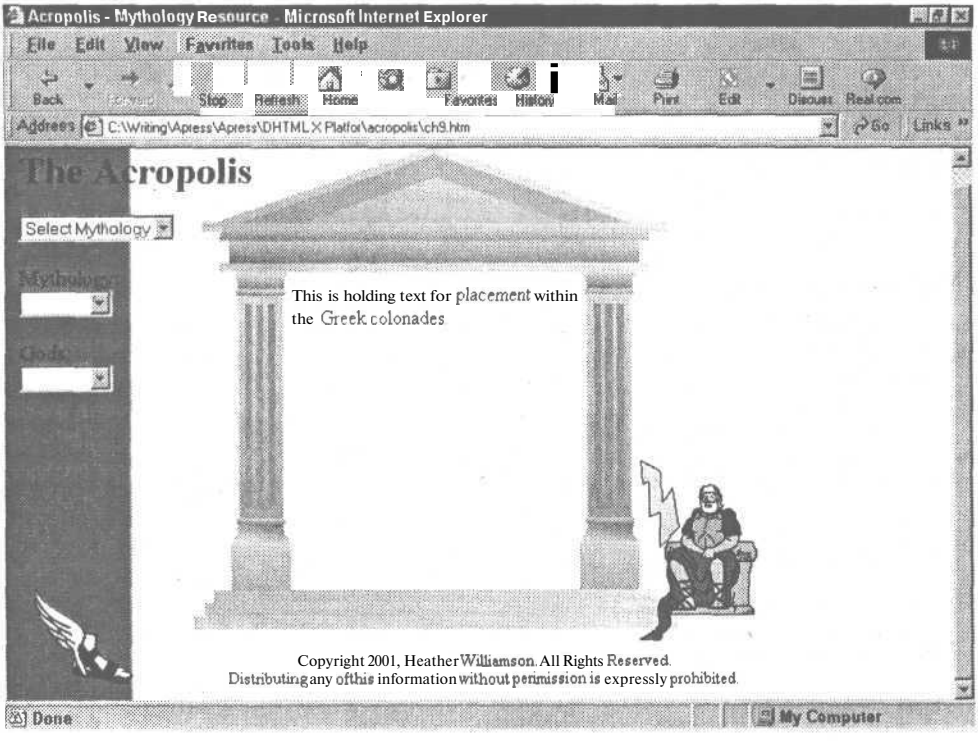


Рис. 9.6. Внешний вид сайта Acropolis

С добавлениями, произведенными в данной главе, а также с рассмотренными в предыдущих, код нашего сайта-примера выглядит следующим образом:

```
<HTML>
<HEAD>
<TITLE> Acropolis - Mythology Resource</TITLE>
<SCRIPT LANGUAGE="JavaScript">
   var bwr = navigator.appName;
   var ver = parseInt(navigator.appVersion, 10);
   alert (bwr + ver);
N54 = ( bwr == "Netscape" && ver == 4 ) ? 1 : 0;
NS6 = ( bwr == "Netscape" && ver == 6 ) ? 1 : 0;
IE4 = ( bwr == "Microsoft Internet Explorer" && ver == 4 ) ? 1 : 0;
IE5 = ( bwr == "Microsoft Internet Explorer" && ver == 5 ) ? 1 : 0;
ver4 = (NS4 || IE4 || NS6 || IE5) ? 1 : 0;
isMac = (navigator.appVersion.indexOf("Mac") != -1) ? 1 : 0;
isDynamic = (NS4 || (IE4 && !isMac) || NS6 || IE5) ? 1 : 0;

 if (NS4 ||N56) {
   doc = "document";
   styl = "";
   html = ".document";
   xpos = "e.pageX";
   ypos = "e.pageY";
  }
  else if (IE4 || IE5) {
```

```
    doc = "document.all";
    styl = ".style";
    html = "";
    xpos = "event.x";
    ypos = "event.y";
   }
contentGT = eval(doc + ' ["greektext"] ' + html);
var engaged = false
var offsetX = 0
var offsetY = 0

function dragIt(e) {
  engaged = true;
  if (IE4 || IE5) {
   leftPos = movingLyr.left;
   topPos = movingLyr.top;
   chkLeftLength = leftPos.length;
   chkTopLength = leftPos.length;
   if (chkLeftLength !== -1) {
    offsetX = eval(xpos) - leftPos.substring(0,chkLeftLength-2);
   }
   if (chkLeftLength !== -1) {
    offsetY = eval(ypos) - topPos.substring(0,chkTopLength-2);
   }
  } else if (NS4 || NS6) {
     offsetX = eval(xpos) - movingLyr.left;
     offsetY = eval(ypos) - movingLyr.top;
  }
  if (engaged) {
   for (var i=0; i<400; i++) {
   moveIt(e);
   }
   offsetX=0;
   offsetY=0;
   }
}

function moveIt(e) {
        movingLyr.top = eval(ypos) + offsetY;
   movingLyr.left = eval(xpos) + offsetX;
   offsetX--;
   offsetY--;
   t = setInterval("moveIt()",500);
}

function changeProp(layerName,theProp,theValue) {
  var usedLyr = eval(doc + '["' +layerName + '"]'+ styl);
if ((theProp == "visibile") || (theProp == "visibility")) {
  if (NS4) {
   theProp = "visible";
  }
  else if (IE4 || IE5 || NS6) {
   theProp = "visibility";
  }
}
  eval('usedLyr.' + theProp + '="' + theValue + '"');
 }

function loadBblContent(newText,layerName) {
```

```
  contentBubble = eval (doc + '["' + layerName + '"]' +html);
  if (NS4 || NS6) {
   contentBubble.write(newText);
   contentBubble.close();
  else {
   contentBubble.innerHTML = newText;
  }
}

function loadGrkContent(doc_url) {
  if (IE4 || IE5) {
    document.all.ieContent.document.frames["ieframe'].document.location.href=
doc_url;
  } else if(NS4 | | NS6) {
   contentGT.load(doc_url, gtext_width);
  }
}

function controlSound(sndAction,sndObj) {
 if (eval(sndObj) != null) {
  if (NS4|| NS6) eval(sndObj+((sndAction=='stop')?'.stop()' : ' .play(false)'));
  else if (eval(sndObj+".FileName"))
   eval(sndObj+((sndAction=='stop')?'.stop()':'.run()'));
 }
}

</SCRIPT>
</HEAD>
<BODY background="images/acbkgrnd.gif"
   text="#000000" link="#660066"
   vlink="#FF0000" alink="#FFFF00">
<script language="Javascript">
 {
mythology = new Object();
mythology.Roman= new Array("Land", "Sea", "Death", "Love", "War");
mythology.Greek= new Array("Land", "Sea", "Death", "Love", "War");
mythology.Hindu= new Array("Land", "Sea", "Death", "Love", "War");
gods = new ObjectO;
gods.Roman = new Array("Cupid", "Apollo", "Aphrodite");
gods.Greek = new Array("Hermes", "Hera", "Mars");
gods.Hindu = new Array("Aries", "Diana", "Bacchus");

function updateMenus(myths) {
 if (document.images) {
 sel = myths.selectedIndex;
 if (sel == 1) {
  myth = mythology.Roman;
  god = gods.Roman;
 } else if (sel == 2) {
  myth = mythology.Greek;
  god = gods.Greek;
 } else if (sel == 3) {
  myth = mythology.Hindu;
  god = gods.Hindu;
 } else {
  myth = new Array();
  god = new Array();
 }
 myths.form.mythology.length = myth.length;
```

```
 for(i=0;i<myth.length;i++) {
  myths.form.mythology.options[i].text = myth[i];}
 myths.form.gods.length = god.length;
 for(i=0;i<god.length;i++) {
  myths.form.gods.options[i].text = god[i];}
 }
 }
}
</script>
<H1> The Acropolis</H1>
<P>
 <font color="black" size="+1" style="arial">
<form>
<select name="myths" size=1 onChange="updateMenus(this)">
<option>Select Mythology
<option>Roman
<option>Greek
<option>Hindu
</select><br><br>

Mythology:<br>
<select name="mythology" size=1>
<option>                   
</select><br><br>

Gods:<br>
<select name="gods" size=1>
<option>                   
</select><br><br>
</form>
</font>

 <div id="greeklayer"
   style="position:absolute;
      left:153px; top:13px;
      width:401px; height:401px;
      z-index:1; overflow: hidden"> <img src="images/Greek.gif" width="400"
height="400">
</div>
<div id="hermeslayer"
   style="position:absolute;
      left:18px; top:372px;
      width:82px; height:77px;
      z-index:4">
 <a onmouseover="changeProp('hermesbbllayer','visibility','visible')"
   onmouseout="changeProp('hermesbbllayer','visibility','hidden')"
   name="bkgbutton">
  <img src="images/hermesshoe.gif" width="90" height="96">
 </a>
</div>
<div id="ieContent"
   style="position:absolute;
      left:235px; top:126px;
      width:238px; height:251px;
      z-index:2";
      visibility="hidden">
 <iframe id="ieframe" frameborder=1 width=1 height=1 scrolling=no src="">
</iframe>
</div>
```

```
<div id="greektext"
   style="position:absolute;
      left:235px;  top:126px;
      width:238px; height:251px;
      z-index:3">
 This is holding text for placement within the Greek colonades.
</div>
<div id="hermesbbllayer"
   style="position:absolute;
      left:47px;  top:312px;
      width:80px; height:100px;
      z-index:6;
      background-image: url(images/hbubble1.gif);
      visibility: hidden">
 <p align="center"> This<br>
  is test<br>
  text. </p>
</div>
<div id="zeuslayer"
   style="position:absolute;
      left:521px;  top:294px;
      width:113px; height:119px;
      z-index:5">
  <a onmouseover="changeProp('zeusbbllayer','visibility','visible')"
   onmouseout="changeProp('zeusbbllayer','visibility','hidden')"
   name="bkgbutton">
  <img src="images/Zeus.gif" width="97" height="129"  onClick="dragIt(event)">
  </a>
</div>
<div id="zeusbbllayer"
   style="position:absolute;
      left:397px;  top:327px;
      width:150px; height:75px;
      z-index:7;
      background-image: url(images/zbubble1.gif);
      visibility: hidden">
 <p align="center"><br>
  This is test<br>
  text.</p>
</div>
<div id="copyright"
   style="position:absolute;
      left:156px;  top:431px;
      width:472px; height:38px;
      z-index:8;
      overflow: visible">
 <p align="center">
  <font size="-1">
   Copyright 2001, Heather Williamson. All Rights Reserved.<br>
   Distributing any of this information without perimission is expressly
prohibited.
  </font></p>
</div>
<div id="zeuslightening"
   style="position:absolute;
      left:519; top:273;
      width:60; height:100;
      z-index:9;
      visiblity: visible">
```

```
  <img src="IMAGES/zlightening.gif" width="40" height="75">
</DIV>
<SCRIPT LANGUAGE="JavaScript">
movingLyr = eval(doc + ' ["zeuslightening"]' + styl);
clickingLyr = eval(doc + ' ["zeuslayer"]' + styl);
if(NS4| |N56) {
 clickingLyr.captureEvents(Event.CLICK);
 clickingLyr.onClick = dragIt;
}
//-->
</5CRIPT>

</BODY>
</HTML>
```

Теперь, когда у нас есть функционирующий сайт, эти сценарии-примеры можно использовать и в других документах. В главе 10 мы рассмотрим множество сценариев данного сайта и попробуем использовать их на других сайтах.

Еще один вариант использования имеющихся возможностей

10

В данной главе рассматриваются сценарии, которые использовались при создании web-сайта Webravin.com. Эти сценарии представлены в следующих файлах:

- index.htm
- Webravinmain.htm
- webravin.js
- wrscrolling.js
- refs.htm

Для получения более полного представления о DHTML в данной главе мы подробно рассмотрим исходный текст этих фалов. Некоторые специфичные функции в данной книге мы рассматривать не будем. Вы узнаете еще один способ перемещения объектов и изменения текстового содержания. Внимательное изучение представленных сценариев поможет вам в создании собственных динамических сайтов. Каждая из функций представлена в наиболее простом и компактном виде, что облегчает ее понимание и обновление, которое может потребоваться, если разработчики браузеров придумают что-то новое.

Титульная страница: index.htm

Титульная страница сайта WebRavin представлена на рис. 10.1. Она позволяет вам перейти к остальной части сайта, но можно модифицировать страницу таким образом, что остальная часть сайта будет открываться автоматически. Для разметки этого документа, так же как и для других страниц сайта, использованы таблица и одинаковое для всех страниц фоновое изображение, что приводит к ускорению загрузки фона и создает некоторую общность всех страниц сайта. Текст кода этой страницы представлен ниже.

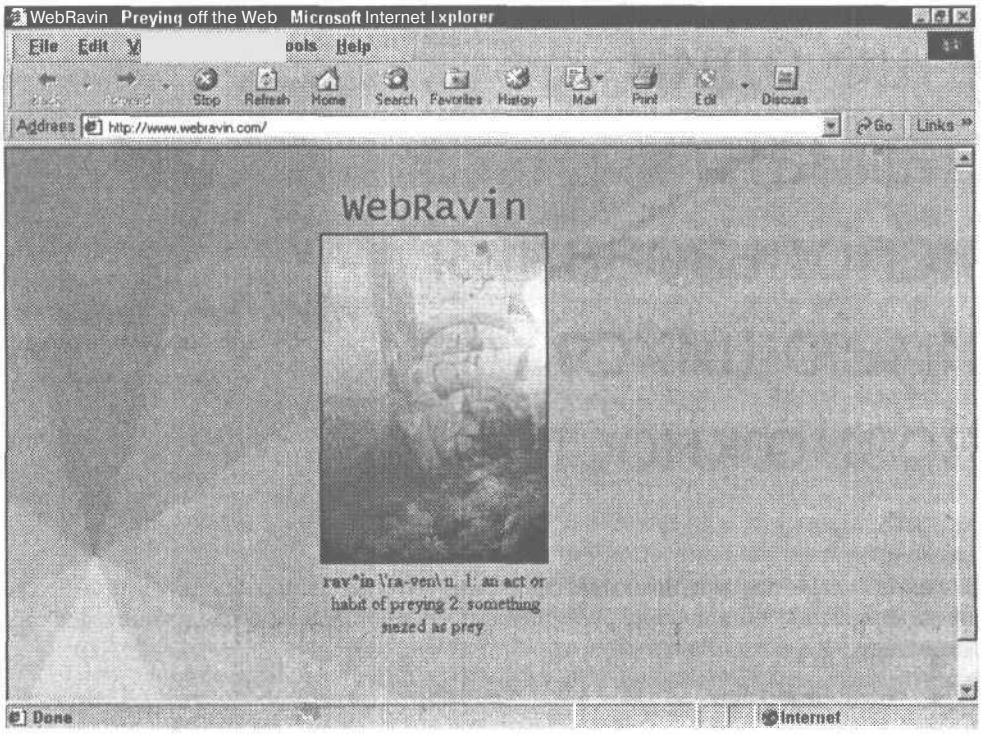


Рис. 10.1. Первая страница web-сайта WebRavin: таблица и два изображения. Чем проще, тем лучше!

```
<html>
<head>
<title>WebRavin - Preying off the Web</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#FFFFFF" background="images/wbrbkgnd.jpg" text="#000000"
link="#006699" vlink="#0033CC" alink="#3399FF">
<script language="JavaScript">
function openSesame() {
  WebRavinWindow = window.open("webravinmain.htm","WebRavin",
"scrollbars=no, width=800, height=600, toolbar=no, location=no,
status=yes, resizable=no");
}
</script>
<table border="0" width="75%" align="center" vspace="10" hspace="10">
  <tr valign="bottom">
    <td height="50" width="100"> </td>
    <td height="50" width="200" align="center">
      <div align="center"><font size="+3" face="Lucida Console">WebRavin</
font></div>
    </td>
    <td height="50" width="100"> </td>
  </tr>
  <tr valign="top">
    <td height="200"> </td>
```

```
    <td height="200" width="200" align="center">
      <div align="center">
        <p><a href="javascript:openSesame()"><img src="images/
logoimg.jpg" width="185" height="270" border="2"></a></p>
      </div>
    </td>
    <td height="200"> </td>
  </tr>
  <tr align="center" valign="top">
    <td height="100" width="100"> </td>
    <td height="100" width="200" align="center"><b>rav*in</b> \'ra-ven\
n. 1:
      an act or habit of preying 2: something seized as prey.</td>
    <td height="100" width="100"> </td>
  </tr>
</table>
<p align="center"><font size="-2" color="#0099CC">Copyright 1999, <a
href="mailto:webmaster@catsback.com">Heather
  Williamson</a>. All Rights Reserved. Any use of graphics, or text from
this
  site is expressley prohibited.</font></p>
</body>
</html>
```

Основной документ: webravinmain.htm

Это основной HTML-файл сайта. Его изображение представлено на рис. 10.2. Он состоит из множества слоев, каждый из которых некоторым образом «оживлен». Основной слой — `contentLayer` — содержит информацию и ссылки. Остальные слои содержат текст, а также графику, которая может перемещаться при активизации мышью или событиями таймера. Каждый из слоев создается с помощью элемента и «привязан» с помощью абсолютного позиционирования. За счет этого как Netscape Navigator, так и Internet Explorer могут одинаковым образом обращаться ко всем слоям и осуществлять взаимодействие. Ниже представлен полный текст кода этой страницы.

```
<HTML>
<HEAD>
<TITLE>WebRavin - General DHTML Topics</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <!- Загрузка сценариев управления свойствами объектов и временной
синхронизации-->
<script language="JavaScript">
  NS4 = (document.layers) ? 1 : 0;
  IE4 = (document.all) ? 1 : 0;
  ver4 = (NS4 || IE4) ? 1 : 0;
  isMac = (navigator.appVersion.indexOf("Mac") != -1) ? 1 : 0;
  isDynamic = (NS4 || (IE4 && !isMac)) ? 1 : 0;

  if (NS4) {
      doc = "document";
      styl = "";
      html = ".document";
      xpos =  "e.pageX";
      ypos =  "e.pageY";
   }
```

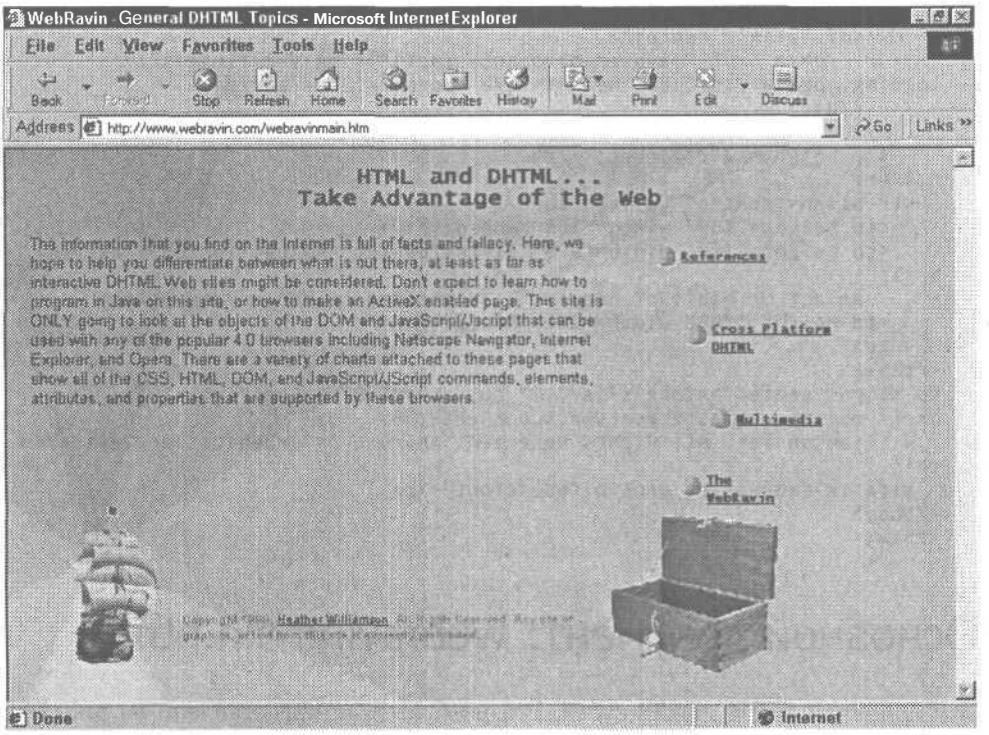


Рис. 10.2. Основная страница сайта WebRavin ограничена размером экрана 800x600 для того, чтобы посетители, которые имеют большее разрешение экрана, не раскрывали окно web-браузера на полный экран

```
    else if (IE4) {
        doc = "document.all";
        styl = ".style";
        html = "";
        xpos =  "event.x";
        ypos =  "event.y";
    }
</script>
<script language="JavaScript" src="webravin.js"></script>
<style type="text/css">
<!--
body {  font-family: Ari'al, Helvetica, sans-serif;
        font-size:  10pt;
        background-attachment: fixed;
        background-color:  #CCFFFF;
        background-image: url(images/wbrbkgnd.jpg);
        background-repeat: no-repeat;
        background-position: left top}
-->
</style>
</HEAD>

<BODY BGCOLOR="#FFFFFF" background="images/wbrbkgnd.jpg" onload="startTimer()">
  <script language="JavaScript">
```

```
   preloadImages('images/roseship.gif','images/roseshipr.gif','images/
buttonbase1.gif', 'images/buttonbase2.gif', 'images/chest2.gif', 'images/
chest3.gif')
  </script>

  <div name="contentLayer" id="contentLayer"
       style="position:absolute;
       left:20px;   top:70px;
       width:475px; height:300px;
       z-index:2;   visibility:hidden;">
  <h3 align="center">
    <font face="Lucida Console">
    HTML and DHTML...<br> Take Advantage of the Web
    </font>
  </h3>
    The information that you find on the Internet is full of facts and
    fallacies. At this site, we hope to help you differentiate between
what is true and what is not out there,
    at least for interactive DHTML Web sites. Don't expect to learn how
to program in Java on this
    site or how to make an ActiveX enabled page. On this site we'll only
discuss the objects of the DOM and JavaScript/Jscript
    that can be used with any of the popular 4.0 browsers, including
    Netscape Navigator, Internet Explorer, and Opera. A
    variety of charts attached to these pages show the
    CSS, HTML, DOM, and JavaScript/JScript commands, elements,
    attributes, and properties that are supported by these browsers.
   </div>
    <iframe id="bufferFrame" style="display:none">
    </iframe>

<div name="shipLayer" id="shipLayer"
     style="position:absolute;
            left:10px;  top:12px;
            width:164px; height:183px; z-index:0">
  <img src="images/roseshipr.gif" width="151"
       height="175" name="roseship">
</div>

 <div name="chestlayer" id="chestlayer"
       style="position:absolute;
       left:459px; top:296px;
       width:191px; height:147px;
       z-index:4">
  <img src="images/chest1.gif" width="162"
       height="145" name="chest" align="right">
  </div>

   <div name="slideLayer" id="slideLayer"
       style="position:absolute;
         left:700px;   top:10px;
         width:40px; height:55px;
         z-index:4; visibility=visible">
  <img src="images/mapdn.gif" width="40"
       height="55" name="scroll" align="right"
      alt="Drag to scroll">
  </div>
   <div name="scrlEvntLayer" id="scrlEvntLayer"
       style="position:absolute;
```

```
          left:700px;  top:0px;
          width:60px; height:1026px;
          z-index:10; visibility:visible;"
           onMouseDown = "engageScroll()"
           onMouseMove = "layerScroll()"
           onMouseUp = "endScroll() ">
  </div>
  <div name="Xplat" id="Xplat"
     style="position:absolute;
            left:511px; top:350px;
            width:132px; height:30px;
            z-index:1; visibility: hidden">
   <ahref="javascript:loadContent('xplat.htm')"
         onMouseOver="swapImage('Xplat','crosspic','3')"
         onMouseOut="swapImage('Xplat','crosspic','2')">
  <img src="images/buttonbase1.gif"
       width="20" height="20"
       name="crosspic" align="left" border="0">
   <font face="Lucida Console, Courier" size="-2">
    <B>Cross Platform<br>DHTML</B></font> </a>
  </div>

  <div name="Refs" id="Refs"
       style="position:absolute;
              left:507px; top:357px;
              width:121px; height:30px;
              z-index:2; visibility: hidden">
   <a href="javascript:loadContent('refs.htm')"
         onMouseOver="swapImage('Refs','refspic','3')"
         onMouseOut="swapImage('Refs','refspic','2')">
   <img src="images/buttonbase1.gif"
         width="20" height="20" name="refspic"
         align="left" border="0">
   <font face="Lucida Console, Courier" size="-2"><B>References</B></
font></a>
 </div>

 <div name="Multimedia" id="Multimedia"
      style="position:absolute;
             left:514px; top:363px;
             width:121px; height:31px;
             z-index:2; visibility: inherit">
   <a href="javascript:loadContent('multi.htm')"
         onMouseOver="swapImage('Multimedia','multipic','3')"
         onMouseOut="swapImage('Multimedia','multipic','2')">
  <img src="images/buttonbase1.gif"
        width="20" height="20" name="multipic"
        align="left" border="0">
  <font face="Lucida Console, Courier" size="-2"><B>Multimedia</B></
font></a>
</div>
<div name="Info" id="Info"
     style="position:absolute;
            left:509px;  top:358px;
            width:70px; height:30px;
            z-index:2; visibility: visible">
   <a href="javascript:loadContent('info.htm')"
         onMouseOver="swapImage('Info','infopic','3')"
         onMouseOut="swapImage('Info','infopic','2')">
```

```
   <img src="images/buttonbase1.gif"
       width="20" height="20"
       name="infopic" align="left" border="0">
  <font face="Lucida Console, Courier" size="-2"> <B>The WebRavin</B></
font></a>
</div>
<p> </p>
 <script language="JavaScript" src="wrscrolling.js"></script>
 <script language="JavaScript">
  if (NS4) {
  document.scrlEvntLayer.captureEvents(Event.MOUSEDOWN | Event.MOUSEUP |
Event.MOUSEMOVE);
     document.scrlEvntLayer.onMouseDown = engageScroll();
     document.scrlEvntLayer.onMouseMove = layerScroll();
     document.scrlEvntLayer.onMouseUp = endScroll();
  }
</script>
</BODY>
</HTML>
```

Основной файл сценариев: webravin.js

Это основной файл со сценариями сайта WebRavin. В нем находятся большинство функций, используемых сайтом, и при необходимости к нему можно обратиться с любой страницы сайта. Поскольку на стороне клиента осуществляется кэширование, не имеет значения, что на каждой странице вы будете использовать не все функции. Обращаться можно только к тем, которые нужны.

```
/*==========================================================================

ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ

НАЗНАЧЕНИЕ: определить все переменные, которые могут использоваться
различными функциями

========================================================================== */
//Переменные массива изображений
var preloadArray = new Array();
```

Первая функция в этом файле — `preloadImages()`. Она производит загрузку изображений, используемых в документе. Все изображения, участвующие в интерактивной замене изображений, должны быть загружены до их использования, чтобы замена происходила «гладко».

```
/* ==========================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ: preloadImages()

НАЗНАЧЕНИЕ: предварительная загрузка всех изображений, используемых при
организации интерактивного интерфейса сайта. Изображения хранятся
в массиве, а обращение к ним производится по индексу массива с помощью
функции swapImage().

========================================================================== */
function preloadImages() {
  if (document.images) {
    var imgFiles = preloadImages.arguments;
```

```
    for (var i=0; i<imgFiles.length; i++) {
      preloadArray[i] = new Image;
      preloadArray[i].src = imgFiles[i];
    }
  }
}
```

Функция swapImages() предназначена для осуществления замены предварительно загруженных изображений HTML-документа. При вызове этой функции в качестве параметров в нее передаются две переменные. Первая — имя IMG-элемента, которое будет заменено, второе — индекс, под которым в массиве хранится предварительно загруженное изображение, которое будет использоваться при замене.

```
/* ==================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ:  swapImages()

НАЗНАЧЕНИЕ: замена изображений, которые были предварительно загружены
с помощью функции preloadImages().

==================================================================*/
function swapImage(layerName,imgName,arrayID) {
  if (document.images) {
   if(IE4){
     workingImg = eval(doc + '["' + imgName + '"]');
     workingImg.src = preloadArray[arrayID].src;
   } else if (NS4) {
     document[layerName].document[imgName].src =
preloadArray[arrayID].src;
     }
 }
}
```

Следующая функция, setTiming(), предназначена для заполнения массива, используемого функцией startTiming. Созданный массив использует один индекс для пары элементов. Первый из них — timingArray[i].timeFrame — определяет временной интервал, через который должен быть выполнена функция, на которую ссылается второй элемент — timingArray[i].fctnName.

```
/* ==================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ: setTiming()

НАЗНАЧЕНИЕ: предназначена для заполнения массива timingArray. Каждое
событие, введенное в массив, должно соответствовать временной последовательности. Нарушение этого порядка приведет к зацикливанию функции.

================================================================== */
function setTiming(timeFrame, fctnName) {
   this.timeFrame = timeFrame;
   this.fctnName = fctnName;
}
var timingArray = new Array()
timingArray[0] = new setTiming("1",
"swapImage('shipLayer','roseship','0')");
timingArray[1] = new setTiming("4",
"moveLayer('shipLayer','10,61,112,163,214,265,314,360,404,445','12,19,25,32,38,45,52,60,69,83')");
timingArray[2] = new setTiming("6",
"swapImage('shipLayer','roseship','1')");
```

```
timingArray[3] = new setTiming("8",
"moveLayer('shipLayer','424,394,365,338,312,286,259,233,206,184','
102,111,118,124,130,136,142,149,157,172')");
timingArray[4] = new setTiming("10", "swapImage('shipLayer','roseship','0')");
timingArray[5] = new setTiming("14",
"moveLayer('shipLayer','210,240,270,300,331,362,395,430,466,497',
'184,191,196,200,205,209,213,218,224,237')");
timingArray[6] = new setTiming("16", "swapImage('shipLayer','roseship','1')");
timingArray[7] = new setTiming("20",
"moveLayer('shipLayer','455,410,364,315,266,215,164,113,62,11','245,
249,253,256,258,261,263,266,268,271')");
timingArray[8] = new setTiming("22", "swapImage('shipLayer','roseship','0')");
timingArray[9] = new setTiming("26", "changeProp('contentLayer',
'visibility', 'visible')");
timingArray[10] = new setTiming("35", "swapImage('chestlayer','chest','4')");
timingArray[11] = new setTiming("40", "swapImage('chestlayer','chest','5')");
timingArray[12] = new setTiming("50", "changeProp('Xplat', 'visibility',
'visible')");
timingArray[13] = new setTiming("55",
"moveLayer('Xplat','511,514,517,520,524,527,530,533,536,539,542,546,549,
552,555','350,338,326,315,303,291,279,267,256,233,210,190,175,160,145')");
timingArray[14] = new setTiming("60", "changeProp('Refs', 'visibility',
'visible')");
timingArray[15] = new setTiming("65",
"moveLayer('Refs','507,508,509,510,512,513,514,516,518,520,522,524,526,
528,530','409,389,370,350,330,310,291,271,251,232,200,170,140,110,80')");
timingArray[16] = new setTiming("70", "changeProp('Multimedia',
'visibility', 'visible')");
timingArray[17] = new setTiming("75",
"moveLayer('Multimedia','514,518,523,527,531,536,540,545,549,553,558,562,
566,571,575','363,354,344,335,326,316,307,295,280,270,255,245,235,225,215')");
timingArray[18] = new setTiming("80", "changeProp('Info', 'visibility',
'visible')");
timingArray[19] = new setTiming("85",
"moveLayer('Info','509,512,515,518,521,524,527,530,532,535,538,541,544,
547,550','358,353,348,342,337,332,327,322,316,308,300,292,285,277,270')");
```

Функция startTimer() использует функцию timingArray для запуска каждого индекса массива и выполнения определенной функции в установленное время. С помощью различных переменных можно установить таймер и отслеживать индекс функции и точное время, когда она должна быть выполнена. Метод setTimeout используется для управления каждым временным интервалом, что приводит к последовательному представлению сайта, а не просто к его «появлению».

```
/* ======================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ:  StartTimer0

НАЗНАЧЕНИЕ: предназначена для запуска таймера, требуемого для запуска
списка событий, происходящих на данной странице. Эта функция сравнивает
текущие значения таймера и временных значений в массиве. При их совпаде-
нии сценарий вызывает обработку этого события. Если совпадения нет, то
сценарий продолжает ожидать совпадений. Эта функция использует оператор
условного перехода для выхода из цикла setTimeout, чтобы не произошло
зацикливания сценария.

====================================================================== */
```

```
//Переменные функций таймера
arraystep = 0;
arraylength = timingArray.length;
timecount = 0;

function startTimer() {
 if (timecount>=100) {
    if (timingArray[arraystep].timeFrame == timecount) {
      eval(timingArray[arraystep].fctnName);
      if (arraystep == (arraylength-1)) {
        timecount=85;
      } else {
        arraystep++;
      }
   }
   timecount++;
   setTimeout("startTimer()",100);
  }
  }
```

Функция changeProp позволяет при помощи имени слоя и имени свойства изменять значение этого свойства. Если имя свойства указано верно, то выражение eval изменит определенное свойство определенного слоя определенным значением.

```
/*  ==================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ: changeProp()

НАЗНАЧЕНИЕ: эта функция предназначена для изменения основных свойств
слоев и изображений в HTML-документе.

================================================================== */

function changeProp(layerName,theProp,theValue) {
   var usedLyr = eval(doc + '["' + layerName + '"]' + styl);
   eval('usedLyr.' + theProp + '="' + theValue + '"');
}}
```

Функция moveLayer() отличается от функции перемещения слоя, рассмотренной ранее в данной книге. Вместо использования математических вычислений при перемещении слоя в определенное место данная функция позволяет задать последовательность точек слоя (координат верхнего левого угла), в которые данный слой будет последовательно помещен. Ускорить или замедлить перемещение можно при совместном использовании с этой функцией метода setTimeOut.

```
/* ==================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ: moveLayer()

НАЗНАЧЕНИЕ: Эта функция может использоваться для перемещения определенного
слоя «из точки А в точку В» при помощи пар координат верхнего левого
угла слоя.

================================================================== */

function moveLayer(layerName, leftList, topList) {
  movingLyr = eval(doc + '["' + layerName + '"]' + styl);

  var topArray = topList.split(",");
```

```
    var leftArray = leftList.split(",");
    var i=0;
    for (var i=0;i<topArray.length;i++) {
        movingLyr.top = topArray[i];
        movingLyr.left = leftArray[i] ;
    }
}
```

Функция loadContent() осуществляет загрузку информации двумя различными способами для Netscape и Internet Explorer. При использовании Netscape Navigator информация загружается с указанного URL непосредственно в элемент с помощью метода load(). Поскольку Internet Explorer не поддерживает этот метод для элемента , то для загрузки информации приходится использовать элемент <iframe>, а затем, для помещения содержания <iframe> в , — функцию finishIEload().

```
/* ================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ: loadContent()

НАЗНАЧЕНИЕ: предназначена для загрузки информации в слой, размещенный
в середине экрана. Поскольку IE не позволяет обращаться к содержанию
элемента <div> через свойство src, для размещения содержания с помощью
атрибута src необходимо использовать отдельный элемент <iframe>. Netscape
позволяет обращаться к содержимому слоя непосредственно через атрибут
src.

================================================================ */
function loadContent(docurl) {
  if (IE4) {
    parent.bufferFrame.location = docurl;
  } else if(NS4) {
    document.contentLayer.load(docurl, 400);
  }
}
```

Функция finishIEload() требуется для помещения информации из объекта <iframe> в объект . Для определения использования браузера Internet Explorer используется оператор условного перехода. Для обращения к этой функции она должна быть включена в начало каждого документа, загружаемого в слой. При обращении к этой функции из дочернего документа ссылка должна иметь следующий вид: parent.finishIEload(). Работу этой функции можно наблюдать в документе refs.htm, рассмотренном далее в этой главе.

```
/* ================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ: finishIEload()

НАЗНАЧЕНИЕ: предназначена для загрузки информации в объект <div>, разме-
щенный в середине страницы. Поскольку IE не позволяет обращаться непо-
средственно к содержанию элемента <div> через атрибут src, приходится
вначале загружать новый документ в скрытый элемент <iframe>. А затем
в начале каждого загружаемого документа помещать этот сценарий для завер-
шения процедуры загрузки и копирования текста в элемент <div> при помощи
команды innerHTML, поддерживаемой браузером Internet Explorer.

================================================================ */
```

```
function finishIEload() {
  if (IE4) {
   contentLayer.innerHTML = parent.bufferFrame.document.body.innerHTML
  }
}
```

Содержание прокручиваемого слоя: wrscrolling.js

Содержащиеся в этом файле функции определяют возможность прокрутки информации, загруженной в слой contentLayer. При отсутствии этих функций любая информация, которая не помещается на экране, не сможет быть показана с помощью Netscape Navigator. Всего в файле хранится три функции. Собственно перемещение слоя осуществляется функцией layerScroll(). Для прокрутки информации в ней используется значение ожидаемой длины страницы, определяемой объемом информации, содержащейся в документе.

```
/* ==================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ: layerScroll()

НАЗНАЧЕНИЕ: предназначена для создания «линейки прокрутки» для Internet
Explorer и Netscape Navigator вместо обычной линейки IE. Netscape не
поддерживает прокрутку элемента <div>, поэтому данная функция обеспечивает совместимость.

================================================================== */

//Переменные функций прокрутки
var engaged = false;
var mouseY = 0;
pageheight=null;
compLyr=eval(doc+ ' ["slideLayer"]' + styl);
contentLyr=eval(doc + '["contentLayer"]' +styl);

function layerScroll(e) {
  mouseY = eval(ypos);
  if (engaged) {
    if (mouseY>5 && mouseY<590) {
      contentLyr.top=pageheight* -(mouseY/600)+20;
      compLyr.top=mouseY - 10;

    }
  }
}
```

Основное назначение второй функции — engageScroll() — убедиться, что кнопка мыши нажата до того, как она помещена над областью прокрутки. Кроме того, она вычисляет высоту слоя, для того чтобы вы могли знать шаг прокрутки.

```
/* ==========================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ: engageScroll()

НАЗНАЧЕНИЕ: предназначена для установки переменных и запуска прокрутки
содержания слоя.

========================================================================== */
function engageScroll(evt) {
  engaged=true;
  pageheight=(NS4)? document.contentLayer.document.height:
document.all.contentLayer.clientHeight;

}
```

Третья функция файла — wrscrolling.js — назначает переменной engaged значение false для того, чтобы любое перемещение мыши над значком прокрутки не вызывало прокрутки страницы.

```
/* ==========================================================================

НАИМЕНОВАНИЕ ФУНКЦИИ: endScroll()

НАЗНАЧЕНИЕ: предназначена для завершения процесса прокрутки.

========================================================================== */
function endScroll(evt) {
  engaged=false;
}
```

Работа с остальными документами: refs.htm

Файл refs.htm — это небольшой документ, который будет представлен далее. Используемые в нем ссылки подобны тем, которые используются в основном документе сайта. Но если обратить внимание на эти ссылки, то можно отметить, что они открывают связанные с ними документы в новом окне, поскольку их содержание значительно больше места, выделенного для представления в основном документе. Элемент <body> содержит ссылку onload для запуска функции finishlEloado. Ее наличие требуется в начале каждого загружаемого в слой документа. Без нее невозможно перемещение информации из элемента <iframe> в элемент в браузерах Microsoft.

```
<html>
<head>
  <tit\e> Reference Source List </title>
</head>
<body onload="parent.finishIEload()">
<img src='images/buttonbase1.gif' width='48' height='47' align='left'>
<A HREF="javascript:openWindow('HTMLchart.htm') ">HTML Element Support</
A><br>
<img src='images/buttonbase1.gif' width='48' height='47' align='left'>
<A HREF="javascript:openWindow('CSSchart.htm')">CSS Property Support</
A><br>
```

```
      src='images/buttonbase1.gif' width='48' height='47' align='left'
<imgHREF="javascript:openWindow('DOMchart.htm')">DOM Property Support</
A><br>
<img src='images/buttonbase1.gif' width='48' height='47' align='left'>
   HREF="javascript:openWindow('scriptchart.htm')">JScript/JavaScript
Support</A><br>
<script language="JavaScript">
function openWindow(destFile) {
  newWindow = window.open("",null,"height=200,width=400,status=yes,
toolbar=no,menubar=no,location=no");
}
</script>
<P> </p><P> </p>
<P> </p><P> </p>
<P> </p><P> </p>
<P> </p><P> </p>
<P> </p><P> </p>
</body>
</html>
```

Создание web-сайта

11

Самым главным этапом разработки web-сайта является этап планирования. Например, плотник не может приступить к строительству дома, пока не увидит чертежа, на котором показано, как соединяются различные части дома, чтобы они смогли выдержать самую сильную бурю. Если архитектор еще не спроектировал дом, то плотник не сможет его построить. Работа web-дизайнера подобна работе архитектора. Его задача — организовать работу таким образом, чтобы разработчики смогли «сложить» сайт, который бы безукоризненно работал с любым браузером. Дизайнер и разработчик должны тесно сотрудничать, для того чтобы клиент и посетители клиента были довольны результатами.

Web-сайт, так же как и дом, является для его владельца вложением капитала. В бизнесе web-сайт — это место, где каждый потраченный доллар должен использоваться на 100 %. Сайт даже может являться единственным источником дохода компании. Сколько бы денег не ушло на разработку и каков был бы доход от эксплуатации сайта (а некоторые сайты могут быть бесценны), web-сайты разрабатываются для получения выгоды, независимо от того, приносят ли они реальные деньги или просто являются элементом рекламной кампании. Если сайт вызывает негативную реакцию покупателей из-за плохой организации или непрезентабельного внешнего вида, то ваши доходы снизятся.

В любом бизнесе отсутствие прибыли недопустимо. Персональная ли это страничка или коммерческий сайт, вы не скажете, что вас не интересует, будет ли ваша страница вызывать внимание аудитории или нет. Тем более если сайт носит коммерческий характер, вы должны знать, для кого вы реализуете свой материал, кто его смотрит и как гарантировать, что ваши клиенты будут возвращаться на эту страничку.

Команда по созданию сайта

Для разработчика сайта, а особенно в команде разработчиков для охвата всех вопросов необходимо распределить обязанности и определить, кто будет руко-

водить определенным этапом разработки. В крупных фирмах, занимающихся web-дизайном, в команде можно выделить следующие роли:

- продюсер сайта;
- разработчики сайта;
- художники;
- администраторы сервера;
- тестеры.

ПРИМЕЧАНИЕ

Эти должности могут иметь и другие названия. В каждой компании названия должностей могут быть разными.

Каждая должность требует определенных знаний и навыков. Они могут быть несколько шире рамок названия должности, но ответственность за сайт ложится на всю команду. В маленьких компаниях можно ограничиться двумя сотрудниками. Один из них одновременно будет выполнять роль продюсера, разработчика и web-администратора, а еще один — осуществлять тестирование и выполнять художественное оформление. Просто помните, что работу можно разделить на множество людей, как это происходит в крупных компаниях.

Продюсер сайта

Продюсер сайта осуществляет общий надзор за разработкой сайта. Эту роль не обязательно должен выполнять лучший «сценарист» или лучший художник. Однако он должен четко представлять все вопросы разработки сайта, чтобы на соответствующем уровне общаться со всеми заинтересованными сторонами, организовать взаимодействие и обеспечить конечный результат, удовлетворяющий заказчика. Независимо от размера вашей компании без продюсера вам не обойтись.

При поступлении заказа на новый продукт продюсер должен учитывать сотни и даже тысячи различных вопросов. Наиболее опытный продюсер заранее конспективно записывает все идеи и вопросы — одни используют для этого офис и пишут на стенках кабинета, другие держат все на диске, третьи используют скоросшиватель. Я пользуюсь всеми способами.

Продюсер составляет бюджет и график выполнения работ. Он общается с клиентом и с командой, для того чтобы гарантировать точное выполнение требований клиента в определенные сроки и за определенную стоимость, устраивающую разработчиков. Он должен действовать расторопно, иначе разработчики почувствуют, что их запросы не учитываются, а это ухудшит и замедлит их работу. Этот человек постоянно держит руку на пульсе и отвечает за то, чтобы каждый из участвующих в проекте соответствовал самым современным требованиям. Продюсер обеспечивает доведение до всех членов команды любых изменений в графике проекта или запросов клиента, а также осуществляет стимулирование как из средств клиента, так и за счет внутренних резервов.

Можно сказать, что продюсер — это посредник, занимающийся всеми сторонами проекта. Он способствует взаимодействию разработчиков с заказчиком, и только он один знает, о чем думают другие. Продюсер также является един-

ственным человеком, который занимается решением всех неувязок. Например, команда считает, что заказчик, который хочет, чтобы его web-сайт был выполнен в фиолетово-оранжевых цветах, — идиот. Продюсер вежливо преподносит клиенту обоснованные беспокойства группы. Он должен освободиться от эмоциональной стороны вопроса. Аналогично, если заказчику не нравится работа отдельных элементов сайта, включенных в проект разработчиками, то продюсер имеет дело с клиентом, изолируя разработчиков от вмешательства в работу, кроме тех случаев, когда этот действительно необходимо.

Разработчик сайта

Разработчик сайта (site developer) создает HTML-код и сценарии, обеспечивающие работу сайта в соответствии с заявленными требованиями. Совместно с продюсером и художником он реализует интерфейс сайта, а также систему перемещения по сайту. Он осуществляет размещение сайта на web-сервере и, совместно с администратором сервера, проверяет программную и аппаратную совместимость своего детища.

Художник

Художник (graphic artist) совместно с продюсером и разработчиком обеспечивает соответствие графики запросам и предпочтениям клиента (например, конкретную цветовую схему). Он также должен принимать во внимание ограничения размеров и скорости, определяемой системой доставки. Художнику иногда приходится работать непосредственно с заказчиком, чтобы тот мог представить себе, что же он получит, а также для того, чтобы готовая графика устраивала web-разработчиков.

Администратор web-сервера

Администратор web-сервера (web server administrator) обеспечивает физическое размещение и круглосуточную безошибочную программную поддержку HTML-документов, баз данных, электронного магазина и «картинок». Продукт общего труда можно посмотреть и попробовать лишь благодаря усилиям web-администратора. Если в работе web-администратора будут допущены серьезные ошибки, то заказчик будет явно разочарован невозможностью воспроизведения графики или неправильной работой системы навигации.

Тестер

Тестер (quality assurance (QA) tester) участвует в работе лишь на заключительном этапе, хотя время проверки сайта может занимать столько же или даже больше времени, чем разработчик затратил на его создание. Тестеру необходимо убедиться, что сайт работает при просмотре любыми браузерами под любыми операционными системами. Он смотрит, чтобы любая база данных, используе-

мая сайтом, корректно предоставляла требуемые данные, независимо от того, как составлен запрос.

Тестер проверяет возможность нахождения на сайте информации пользователем любой квалификации. Например, если сайт специализируется на продаже подков, тестер должен убедиться, что на сайте без труда можно найти любой требуемый вид подков. Для того чтобы создать действительно хороший web-сайт, требуется профессиональный подход со знанием специфики, независимо от того, является ли сайт интерактивным или чисто информационным. Не имеет значения, есть ли у вас дополнительные обязанности, или вы один совмещаете несколько должностей, или вы один обслуживаете магазин, — для того чтобы ваш web-сайт привлекал посетителей и заставлял их потом неоднократно возвращаться, необходимо выполнить, и выполнить качественно, все перечисленные мероприятия. Только тогда ваш web-сайт можно будет назвать удачным.

Заповеди успеха продюсера

Приведем список правил и советов, которые продюсер должен держать в голове. Я их называю «Мои заповеди по успешному выполнению работы»:

- ни один вопрос не является бессмысленным;
- будьте учителем, проповедником и пацифистом;
- надо все записывать;
- организация — это не шутка;
- будьте последовательны;
- у вас есть рот — пользуйтесь им;
- подгоняйте, льстите и находите причины, но ни за кого не делайте его работу;
- телепатия — не мистика;
- принесите пиво и останьтесь до конца вечеринки;
- это ваша лодка, а вы — ее капитан;
- с прокаженными не общаются;
- подпишите;
- улыбнитесь;
- утро вечера мудренее;
- отдыхать тоже иногда надо;
- регулярно проводите резервное копирование своих данных.

Ни один вопрос не является бессмысленным

Задавайте как можно больше вопросов. Записывайте вопросы, которые пришлось задать одному клиенту, чтобы затем сразу задать их следующему. Не все вопро-

сы подходят различным клиентам, но вы не получите нужные вам ответы, не задавая их. Полученные ответы помогут сформировать границы проекта, порядок выполнения, а также сроки и стоимость его реализации. Для хранения вопросов-ответов выделите отдельный раздел в вашем блокноте (или подкаталог в каталоге ПРИМЕЧАНИЯ). Эта практика поможет вам лучше обслуживать заказчика, поскольку у вас всегда под рукой будет источник важной информации о его web-сайте.

Будьте учителем, проповедником и пацифистом

Необходимо воспринимать как аксиому, что большинство клиентов склонны погубить ваш спланированный и вполне готовый проект. Чтобы не дать клиенту помешать работе, вам необходимо обучить каждого клиента, что ему можно делать, а что нельзя. Технология «рытья земли носом» в сети пока не используется, поэтому не позволяйте заказчику прибегать к ней.

Читайте заказчикам проповеди о том, как важна их роль в создании web-сайта. Если они своевременно не могут предоставить вам необходимую информацию, вы выбьетесь из графика. Вы можете помочь им точно определиться, что должен содержать их web-сайт, но вы не должны брать содержание с потолка. Заказчик должен полностью обеспечивать вас информацией. Убедитесь, что заказчик знает роль каждого члена группы разработки, особенно вашу. Точно определитесь, где и когда будет осуществляться взаимодействие. Вы же не хотите, чтобы ваш заказчик представил вас своему начальнику с неприглядной стороны.

Сделать так, чтобы все остались довольны, — в этом и заключается ваша работа. Команда разработчиков должна быть благодарна заказчику, а тот, в свою очередь, признателен им за конечный результат. А вы — тот счастливчик, которому удалось благополучно свести цифры и факты, и поэтому вы благодарны обеим сторонам.

Надо все записывать

Вы когда-нибудь теряли номер мобильного телефона, когда вам необходимо срочно связаться с заказчиком? А вы когда-либо забывали наименование графического файла и приходилось воссоздавать его лишь из-за того, что вы не смогли найти в сети, а потом вы обнаруживали его неделю или две спустя? Если такое случалось, то я предлагаю все изменить. Ну а если с вами такое еще не случалось, то я хочу предупредить, что подобные промахи свидетельствуют о непрофессионализме и могут сильно навредить вам.

Вы должны просто пристраститься к ведению записей. Составьте список всех членов команды. Контактную информацию по каждому клиенту держите в одном месте. Записывайте все важные сведения. Ведите список всего «имущества» — записывайте каждый файл, базу данных, информацию для базы данных, каталоги и графику, которые требуются для создания web-сайта.

Технические требования, блок-схемы, инструкции, карты страниц и структура навигации должны быть проработаны в мельчайших подробностях. Они должны

явиться основой всей работы, первым шагом при создании интерактивного или статического web-сайта.

Организация — это не шутка

Для того чтобы все шло замечательно, необходимо немного понервничать. При отслеживании каждой мелочи web-проекта не помешает быть навязчиво-принудительным. Невозможно создать хороший web-сайт, не учитывая все нюансы. Прямо сейчас соберите все документы и файлы, которые вы создали в ходе планирования, и запомните, куда вы их положили.

Будьте последовательны!

При образовании наименований всех проектов, и не только данного, будьте последовательны. Если вы хотите, чтобы файлы на вашем диске были упорядочены по наименованию копании заказчика, соблюдайте это правило для каждого проекта. Если вы называете изображение кнопки возврата на домашнюю страницу сайта компании XYZ `btn-home-xyz.gif`, то такое же изображение для компании ABC необходимо называть `btn-home-abc.gif`.

Сохраняйте копии документов на каждой стадии разработки. Если после первой версии вы вынуждены скорректировать документ, оставьте оригинал и просто укажите версию в новом имени файла. Никогда не заменяйте данные их новыми копиями, пока на 100 % не будете уверены, что они вам больше не понадобятся. В большинстве случаев эта информация может быть вам не нужна в течение нескольких месяцев, а потом внезапно понадобится.

СОВЕТ

Вы никогда не должны соглашаться на выполнение проекта или составление бюджета, пока у вас не будет точно просчитанного плана. Наиболее дорогостоящая часть любого проекта — устранение ошибок, а непродуманное планирование может привести к их значительному увеличению, что отрицательно скажется на вашем бюджете и доходе. Если ваша команда работает одновременно над восьмью проектами, не стоит информировать об этом заказчика, сославшись, что у группы достаточно времени и для девятого. Просмотрите календарь. Проверьте график отпусков. Исходите из того, что любой человек может заболеть, ему может понадобиться внезапно решить свои проблемы или у него просто спад активности. Допустите возможность неожиданного возникновения проблем, и вам понадобится вдвое больше времени для завершения проекта, чем вы изначально рассчитывали. Вы можете использовать специальное программное обеспечение, которое поможет вам рассчитать среднюю длительность выполнения любой задачи. Подстройте его под задачи своей группы и применяйте в повседневной деятельности.

У вас есть рот — пользуйтесь им

Разговаривать с людьми никогда не было грехом. Не упускайте возможности даже в самые напряженные периоды работы пообщаться с заказчиком, командой и менеджерами. Говорите обо всем. Удачи и неудачи должны быть обсуждены с кем-либо из участвующих в проекте. Не принимайте решения, которое может

сказаться на заказчике, не посоветовавшись с ним. Если решение может сказаться на интересах вашей команды, никогда не осуществляйте его, не обговорив это с членами команды. Не надо использовать их советы, но согласование позволяет избежать ошибок и поможет вам остаться необходимым как клиенту, так и заказчику.

Подгоняйте, льстите и находите причины, но ни за кого не делайте его работу

Если вы занимаетесь продюсированием сайта, то вы должны «дирижировать» процессом. Никогда не занимайтесь составлением кода, разработкой дизайна, оформлением графики или оформлением авторских прав. Составлением программ занимаются разработчики. Графикой занимаются художники. При необходимости вы можете выполнять эти роли, но не вместо кого-то. Вы должны заниматься своей работой. Если вы выполняете все функции в одиночку, то сложнее отделить продюсерские функции от остальных, но чем лучше вы сможете разделить свои обязанности, тем более эффективной будет ваша работа.

Если вы истинный продюсер, работающий с командой разработчиков, то позвольте им выполнять их работу и не вмешивайтесь в нее. Они знают свое дело лучше вас. Если художник и заказчик пришли к единому мнению, не стоит навязывать свое. Если у вас спрашивают ваше мнение, будьте искренны, но не воздвигайте стену между вами и дизайнерами, залезая туда, куда не просят. Работа продюсера заключается в организации, поддержании порядка, распределении приоритетов и посредничестве между заказчиком и командой. Ответственности хватает на каждого.

Телепатия — не мистика

Если вы считаете, что телепатии не существует, то вы не сможете долго заниматься разработкой web-сайтов. Вы должны знать, что происходит на каждом этапе. Хорошо, если вы знаете, где найти ответы на возникающие вопросы, но вы также должны предвидеть, какие вопросы может задать заказчик, а еще лучше — быть готовым ответить на них. Если вы не знаете ответа, то будьте чистосердечны и расскажите клиенту все, что считаете нужным. А затем убедитесь, что вы оправдали его ожидания.

СОВЕТ

Один из способов удержаться во главе своей группы и достойно предстать перед заказчиком — постоянно быть в курсе новых технологий и всех новейших программных решений, совершенствуя свой уровень в период наименьшей нагрузки. Используйте все новшества технологий в работе, это может облегчить работу всей команде. Вы не должны стремиться изменить используемые инструментальные средства, не посоветовавшись с членами команды, тем не менее необходимо непрерывно обеспечивать конструктивное решение проблем, с которыми они могут столкнуться.

Принесите пиво и останьтесь до конца вечеринки

Другими словами, не занимайтесь сверхурочной работой сами и не заставляйте заниматься ею своих подопечных. Если же люди вынуждены работать помимо основного рабочего времени для того, чтобы уложиться в график, то это именно вы испортили все дело. Не бросайте людей одних. Принесите что-нибудь выпить, чипсы и останьтесь с ними до завершения работы над проектом.

СОВЕТ

Прежде чем вы сможете требовать с других, вы должны сами создать свой web-сайт от начала до конца. Перед тем как заниматься планированием, вы должны самостоятельно выполнить каждый этап, чтобы иметь представление о его реальной продолжительности. Если у вас нет идеи, чем сейчас должны заниматься члены вашей группы, то вы не можете быть движущей силой, лидером или игроком команды!

Это ваша лодка, а вы — ее капитан

Капитан лайнера отвечает за жизнь не только пассажиров, но и своего экипажа. Если вам не нравятся пассажиры, то экипаж не должен знать об этом. Если вам не нравится член экипажа, об этом не должны догадываться пассажиры. Не подвергайте проект опасности, нагнетая напряженность. Вы, наоборот, должны ее снять. Подливая масла в огонь, вы только усложните и увеличите свою работу.

С прокаженными не общаются

Догадайтесь, почему? Вы должны выступать в роли политика. При продюсировании web-сайтов в ходе общения как с заказчиком, так и с командой необходимо проявить искусство дипломата. Накануне четырехдневных выходных нельзя заявлять сотрудникам, что *они отработают* 180 часов на компанию XYZ Corp. к следующей среде. Если от вас отворачиваются заказчики или члены вашей команды, то, вероятно, проблема кроется в отсутствии нормальных взаимоотношений.

Подпишите

Если вы когда-либо случайно перепутали два файла, то вы представляете насколько сложно вернуть все на свои места. Чтобы избежать случайностей, подписывайте название компании или условное наименование работы на каждом документе данного проекта или добавляйте условное наименование проекта к имени каждого файла.

Улыбнитесь!

Чем лучше ваше отношение к работе, тем лучше она будет получаться. Чем лучше вы работаете, тем более счастлив будет заказчик.

Утро вечера мудренее

Если вы не уверены в результате проекта или не знаете, как лучше преподнести его заказчику, то отложите его, чтобы вернуться к этому позже. У вас может появиться запал проработать всю ночь, но лучше оставить все до утра. На свежую голову лучше думается и, соответственно, принимаются более правильные решения.

Отдыхать тоже иногда надо

Спланируйте свой отпуск, желательно между проектами. При необходимости его можно запланировать и в ходе работы над очередным проектом. Напомню еще раз, что свежая голова работает лучше.

Регулярно проводите резервное копирование своих данных

Не допускайте, чтобы аппаратные сбои сводили на нет часы, дни или недели работы. Если вы «ведете» очень важный проект, проводите резервное копирование как можно чаще. Я использую Zip-драйв и провожу резервное копирование каждый час.

Составление плана работы

Любой сайт начинается с предложений клиенту. При разработке плана web-сайта убедитесь, что он отражает следующие моменты:

- общее описание предназначения сайта,
- обсуждение маркетингового плана,
- конкуренция и чем ваше предложение лучше,
- потенциальные посетители — для кого предназначен сайт,
- блок-схема вашего сайта и система навигации,
- полная схема содержания сайта,
- план дальнейшего развития сайта,
- график разработки,
- общий расчет стоимости.

После подготовки плана вы и ваш заказчик должны подписать договор, включающий общее предложение и расчетную стоимость. Некоторые web-дизайнеры ограничиваются лишь устной договоренностью. Этим ребятам не хватает деловой хватки. В настоящее время устная договоренность не приветствуется.

В договоре необходимо отразить хотя бы основные моменты:

- определение сторон,
- условия оплаты (сроки платежей),

- условия окончания действия контракта,
- содержание работ.

Даже простой контракт поможет вам получить заработанные деньги через суд. Конечно же, наличие контракта еще не является гарантией победы в юридической борьбе, но, по крайней мере, вы будете в лучшей форме.

СОВЕТ

Я советую получить с заказчика аванс в размере хотя бы 30 % от расчетной суммы. Обычно клиент, который сразу пожелает внести значительную часть суммы, так же пунктуально оплачивает остальную часть гонорара по мере выполнения проекта. Внесите в контракт график выплаты оставшейся части гонорара.

Вопросы, на которые необходимо ответить перед созданием схемы

Когда потенциальный заказчик переходит к обсуждению схемы web-сайта, я предлагаю ему ответить на представленные ниже вопросы. Получив ответы на все, я буду уверена, что ничто не будет упущено при разработке плана для данного заказчика.

1. Какова цель данного web-сайта?
2. На кого рассчитан этот сайт?
3. Как вы планируете продвигать этот сайт?
4. Какие типы взаимодействия с клиентами предполагаются?
5. Сколько предполагается выделить времени на тестирование сайта?
6. Как предполагается «заманивать» клиентов на сайт?
7. Какова вероятность, что посетители сайта используют браузеры версий 4.x? (Заказчик должен представлять техническую оснащенность своих клиентов.)
8. Сколько графики должно быть на сайте?
9. Как часто будет меняться содержание сайта?
10. Предполагаемая степень роста сайта.
11. Как вы предполагаете, за сколько этапов может быть завершена работа над сайтом? Большинство клиентов желает, чтобы сначала был готов информационный сайт, а затем он развивался бы до сайта, осуществляющего электронные сделки. Это изменение может основываться на популярности сайта и наличии постоянной клиентуры, а также способности привлечь новых посетителей.
12. Должны ли на сайте использоваться видеоролики и аудиофайлы?
13. Какой вид хранения информационного содержания сайта будет использоваться?

14. Какой ожидается трафик?
15. Какой художественный стиль должен использоваться на сайте?
16. Как должна будет осуществляться доставка информации? Active Server Pages? Cold Fusion? Стандартный HTML?
17. Какая цветовая схема должна использоваться?
18. Сайт будет использоваться для продаж товаров или оказания платных услуг?
19. Как заказчик желает осуществлять переписку с посетителями?
20. Какой вид информации будет распределяться и как?
21. Каким образом необходимо организовать перемещение посетителей по сайту?
22. Должен ли обрабатываться счет-фактура?
23. Какие типы оплаты принимаются?
24. Как будет осуществляться доставка товаров и по какому графику?
25. Какой тип коммерческого счета будет использован?
26. Как часто будет производиться смена ассортимента товара?
27. Кто будет вести учет товаров?
28. Какой, по-вашему, потребуется вид обучения для обслуживающего персонала?
29. Сколько информации о компании должно быть представлено на сайте?
30. Где и как будут размещены пресс-релизы?
31. Сколько информации о владельцах компании или менеджерах будет доступно на сайте?
32. Требуется ли разместить на сайте чат? Кто в нем сможет принимать участие? Кто им будет управлять и кто будет за ним наблюдать?

Я надеюсь, что вышеперечисленный список вопросов явится хорошим примером того, какие моменты должны быть учтены, прежде чем вы приступите к разработке полного схематического и функционального плана динамического сайта. Это, конечно, не исчерпывающий список, но довольно полный, и вам лишь необходимо включить в него более конкретные вопросы.

Основы дизайна: разработка схемы

При разработке web-сайта необходимо сначала все тщательно продумать. Вы можете посвятить планированию web-сайта столько же времени, сколько и собственно реализации. А после того, как будет принято решение разрабатывать сайт, ориентируясь на использование браузеров версии 4.x, вы сможете увеличить время разработки сайта и его сценариев на сотни часов по сравнению с временем разработки сайта, ориентированного только на использование последнего поколения браузеров.

После завершения составления плана и подписания договора можно приступать к составлению схемы сайта. На этом этапе возьмите общую схему, блок-схему и навигационную схему, которые вы ранее разработали с заказчиком, и уточните все детали. Необходимо сделать так, чтобы ваша схема была прототипом окончательного варианта сайта. Это последний шаг перед реальным написанием кода страниц сайта.

Я думаю, это надо немножко разъяснить. Та схема, о которой я говорю, — это не блок-схема, не график и тем более не схема соединений или принципиальная электрическая схема. *Схема web-сайта* включает всё от графика до бюджета, от блок-схемы до устного описания дизайна сайта, от контуров до временных вариантов графики.

Давайте рассмотрим схему, созданную при разработке сайта, который мы рассмотрим далее.

План-схема сайта Acropolis (пример)

Разработан: 1 сентября
Автор: Heather Williamson

Описание

Целью данного сайта является распространение всех типов информации и «пустячков», относящихся к греческим и римским богам. Сайт позволит посетителям приобрести статую, почитать мифы, оригиналы и копии документов, найденных при раскопках храмов в Греции и Италии и т. д.

Предназначение сайта — двойное: он предназначен для информирования посетителей, а также для осуществления продаж. Для достижения этих целей нам нужны полные копии мифов и легенд, относящихся к данным богам. Для более полного создания образа потребуются изображения богов в различных позах, включая те, которые являются классическими (например, Зевс, мечущий молнии).

Доступные адреса

mythsnmythology.com
zeusorjupiter.com
acropolisgods.com

Конкуренция

Хотя существует множество сайтов, посвященных римским и греческим богам, ни на одном из них нет более полного описания богов и полубогов обеих религий. Кроме того, ни на одном из сайтов нет «витрины» (электронного магазина). Этот факт делает предполагаемый сайт уникальным как с образовательной, так и с коммерческой точки зрения.

СОВЕТ

Можно вставить список сайтов с адресами и кратким описанием, позволяющим клиенту проверить наличие конкуренции и по возможности представить новые идеи на основе того, что удачно реализовано на этих сайтах.

Маркетинговые задачи

- Необходимо находиться в верхних строчках всех основных поисковых систем, таких как AltaVista, HotBot, Excite, Yahoo, GoTo, WebCrawler, AOL/NetFind, LinkMonster, LinkStar, Excite, Lycos, Infoseek, Lexicon, StartingPoint, E!Net Tradewave Galaxy и LookSmart.
- Разместить рекламные баннеры на других сайтах схожей тематики, таких как PinkMonkey.com, cliffs.com, historybookclub.com и т. д. Подобные сайты привлекают людей, интересующихся историческими материалами и являющихся потенциальными покупателями соответствующих «товаров».
- Разместить рекламу в таких конференциях, как alt.mythology.
- Разместить web-адрес на всех исходящих документах и материалах, таких как бланки писем, конверты, и в почтовых рассылках.

Аудитория

Этот сайт будет привлекать людей, интересующихся греко-римскими богами. Они будут обращаться сюда за научной информацией или просто для того, чтобы сделать подарок любимому профессору истории или близкому другу.

Рост

Структура сайта позволяет легко расширить его содержание за счет других древних и современных религий. Сотни богов были и продолжают оставаться предметом поклонения различных культур на Земле, так что это предприятие может в будущем достичь гигантского успеха. Программное обеспечение «shopping-cart» (корзинка, куда складываются покупки) должно поддерживать бесчисленное множество различных товаров, ограниченное лишь аппаратными средствами сервера. Следовательно, должна существовать возможность обновления сервера в зависимости от потребностей сайта.

Эскиз сайта

ПРИМЕЧАНИЕ

Большинство эскизов сайта более подробны, чем представленный здесь. В эскиз сайта должны включаться наброски всех основных информационных разделов каждой страницы web-сайта.

1. Заглавная страница:
   - ссылки на другие страницы с помощью меню,
   - обсуждение основных групп продуктов,
   - приглашение на сайт и в компанию.
2. Сведения о компании:
   - пресс-релизы,
   - цели компании,
   - история компании,
   - контактная информация (включая карты),
   - возможности трудоустройства.

3. Продаваемые товары:
   1) греческая символика:
   - серебряные скульптуры,
   - оловянные скульптуры,
   - книги,
   - одежда:
     - современный стиль,
     - исторический стиль,

   2) римская символика:
   - серебряные скульптуры,
   - оловянные скульптуры,
   - книги,
   - одежда:
     - современный стиль,
     - исторический стиль.

4. История и мифы:
   - основные боги:
     - Зевс (Юпитер),
     - Посейдон (Нептун),
     - Гадес (Плутон),
     - Арес (Марс),
     - Аполлон (Феб - бог солнца),
     - Гермес (Меркурий).
   - основные богини:
     - Гера (Юнона),
     - Геста (Веста),
     - Афина (Минерва),
     - Артемида (Диана),
     - Афродита (Венера),
   - младшие боги,
   - младшие богини.

ПРИМЕЧАНИЕ

Поскольку всего в греко-римской мифологии насчитывается более пятидесяти младших богов и богинь, я не стала перечислять их всех, но здесь надо было поместить всю информацию целиком.

График

Начало реализации проекта: 1 июля.
Срок завершения: 1 сентября.
Группа разработки: три человека.
Графика: два человека.
Тестирование: два человека.

Начало работы с архивными документами: 1 июля.
Тестирование прототипа сайта: 10 июля.
Представление прототипа заказчику: 15 июля.
Реализация изменений прототипа: 20 июля.
Начало разработки: 20 июля.
Проверка: 25 июля.
Завершение создания графики: 1 августа.
Представление «черновика» заказчику: 15 августа.
Оформление уточнений по графике: 18 августа.
Завершение окончательного тестирования: 25 августа.
Маркетинг: 25 августа.
Учет изменений по первым откликам посетителей: к 1 сентября.

Предполагаемый бюджет

ПРИМЕЧАНИЕ

Различные компании представляют клиентам бюджеты различным образом. Одни просто разбивают проект на этапы с описанием того, что будет выполняться в течение каждого, а затем представляют общую стоимость. Другие сразу делят бюджет на части, позволяя клиенту оценить почасовую оплату и расчет часов на выполнение каждого этапа проекта. Я представила здесь относительно подробный бюджет, хотя такая подробность может и не потребоваться, а может, напротив, для скрупулезного клиента потребуется еще более подробная раскладка.

Основные затраты

Элемент	Стоимость работ (трудоемкость)	Длительность (в часах)	Расчетная стоимость
Вводная и остальные страницы, ссылки на другие сайты	$75 час	40	$3000
Графика	$100 час	40	$4000
Разработка информационного бюллетеня	$75 час	12	$900
Система продаж — система безопасности при работе с кредитными карточками	$75 час	40	$4000
Программное обеспечение продаж и SSL-сертификации	$725	-	$725
Тестирование	$75 час	80	$8000
Итого		132	$20 625

Дополнительные затраты

Размещение сайта на сервере			$600 год
Настройка DNS и основного сервера	-	-	$50

DNS-регистрация			$70
Обслуживание и поддержка	$75 час	10 часов в месяц	$9000

В итоге общая стоимость разработки данного сайта лежит в пределах $20625, а дополнительное обслуживание в течение года составит примерно $9600. Кроме того, через два года придется платить ежегодно $35 компании Network Solutions за обслуживание регистрации имени домена, выбранного заказчиком.

ПРИМЕЧАНИЕ

Цифровые значения, использованные в данном примере, позаимствованы у некоторых мелких компаний-разработчиков. Реальная стоимость создания web-сайта может сильно отличаться в зависимости от компании-разработчика, вашего местоположения и требований к сайту.

Планирование сайта разработчиком

Планирование web-сайта является довольно длительным процессом, состоящим из множества этапов. Этот процесс включает следующие компоненты:

- разработка основной страницы;
- определение способа перемещения по сайту;
- создание графики;
- определение слоев;
- планирование интерактивности;
- планирование тестирования.

Только после того, как будут выполнены эти этапы, можно приступать к написанию собственно кода страниц web-сайта.

Разработка основной страницы: предварительная подготовка

Вы можете посчитать, что если вы представили web-сайт в своей голове, то сможете описать его заказчику, боссу, жене или специалисту отдела реализации, а они смогут его представить себе в том же виде. Такое может случиться только во сне, поэтому рассчитывать на это не приходится. Человек или группа людей, которым должен быть представлен сайт, должны увидеть его. Хотя бы в набросках. Им надо показать блок-схему и четкое описание. Они хотят увидеть, как в конечном итоге будет выглядеть проект. Только после этого они могут с уверенностью подписать договор, дающий вам свободу творчества.

Большинство заказчиков сами не знают, чего они хотят. А я для них должна создать чудесный сайт. Если вы устно опишете, как будет выглядеть сайт, и не представите даже эскиза основных графических объектов и готовой цветовой схемы, то заказчик никогда не будет доволен ни графикой, ни цветовой схемой, ни способом перемещения посетителей по сайту. Он неизбежно вернется и скажет:

«Я думаю, что было бы лучше добавить вот этого цвета», или «Почему я должен щелкнуть здесь, чтобы перейти туда?», или «Я передумал. Давайте все переделаем». Когда подобные вопросы возникают после окончания согласования или уже в ходе разработки, то это приводит к значительному увеличению сроков выполнения работы. Такие заказчики постоянно будут «стоять над душой», чтобы быть уверенными, что все делается так, как им хочется. Они могут не знать, чего они хотят, но они не дадут вам спокойно заняться вашим делом. Это клиенты, которые хотят, чтобы вы рвали на себе волосы. Вот именно из-за таких клиентов и необходимо создать предварительную блок-схему и согласовать все детали.

Я уже говорила, что создание основной страницы — это искусство, которого избегает молодежь. Хотя все очень просто.

Сядьте с будущим владельцем web-сайта и запишите список цветов, которые нравятся вам и дают хозяину сайта почувствовать, чего он хочет от сайта. Мы знаем, что цвета могут вызывать различные чувства: как тепла и холода, так и веселья и печали. Поэтому не стоит об этом забывать при выборе цветов, отражающих чувства, которые, по мнению заказчика, должен вызывать сайт.

После того как выбраны цвета, возьмите в руки бумагу, карандаш или ручку и примерно изобразите, как, по-вашему, должна выглядеть основная страница сайта. Это будет первая страница, которую увидят посетители, поэтому она должна оставлять очень приятное впечатление, иначе посетители просто покинут ее. Взгляните, например, на сайт http://www.hunting-fishing-vacations.com, представленный на рис. 12.1. Этот сайт посвящен охоте, седланию лошадей, снаряжению для рыбной ловли и путешествиям по труднопроходимым местам Орегона.

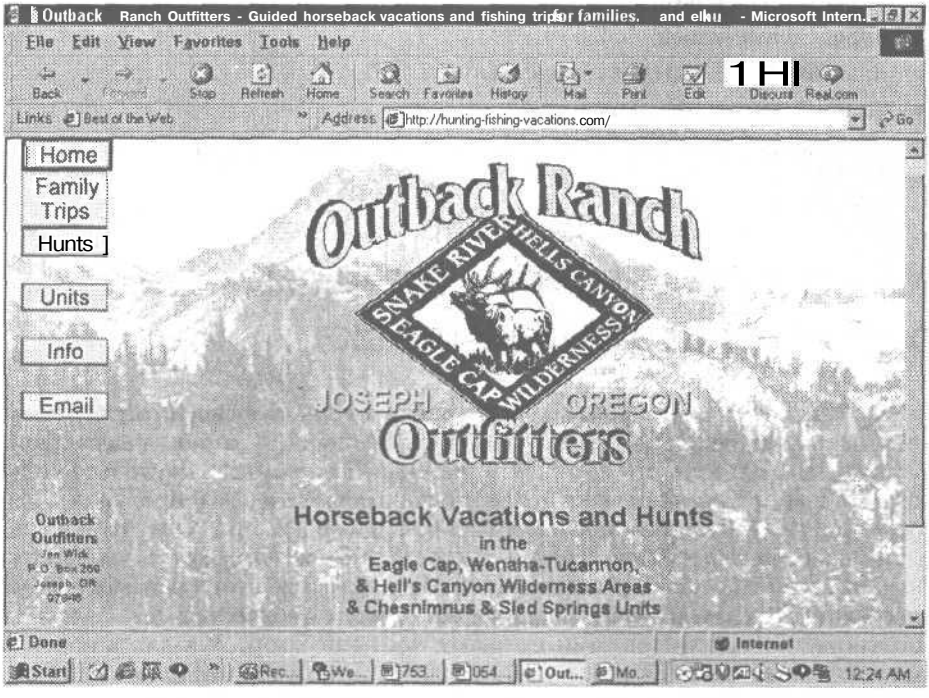


Рис. 12.1. Эта главная страница позволяет убедиться, что вы точно знаете, для чего вам нужен рюкзак

С другой стороны, такой сайт, как http://www.MoonlightChronicles.com, представленный на рис. 12.2, является причудливым, приятным и веселым. Вы сразу понимаете, что этот сайт выглядит как серия журналов, написанных художником. И если графика в верхнем левом углу не вызывает у вас этого впечатления, то подпись точно все объясняет.

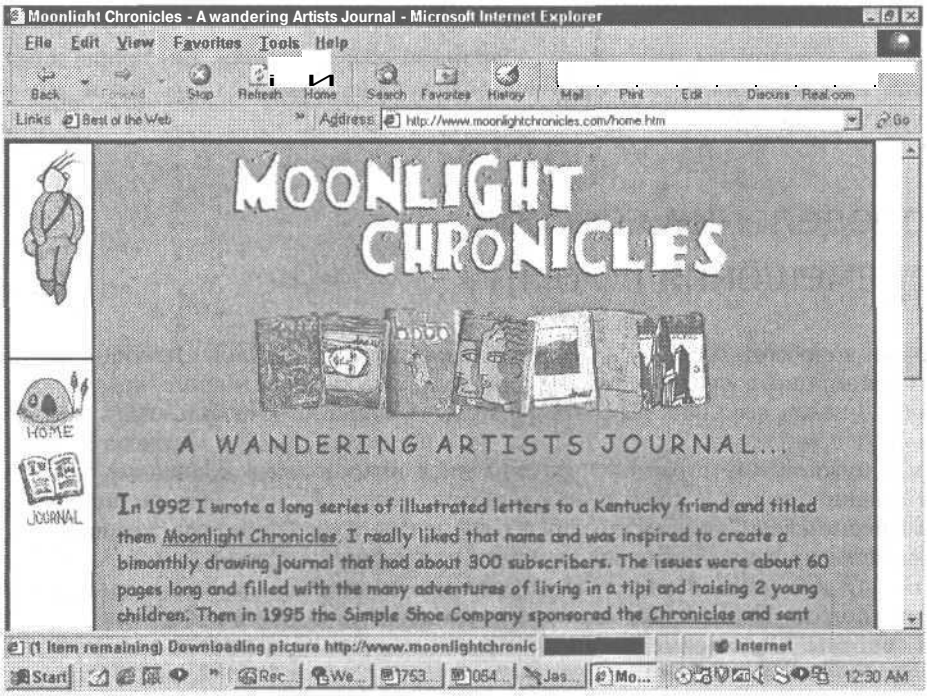


Рис. 12.2. Более причудливый сайт, вызывающий улыбку при взгляде на буквы, украшающие экран. И вместе с тем все серьезно

Каждый из этих сайтов имеет свою направленность: первый должен вызвать в нас жажду путешествий, а второй — заставить улыбнуться и приготовиться к развлечениям. Сначала эти сайты были нарисованы на бумаге, эскиз за эскизом, постепенно приближаясь к требуемому впечатлению, которое должен вызывать сайт. При создании сайта необходимо просмотреть сотни картинок и графических фрагментов, для того чтобы найти то, что вам точно подходит, даже если найденные изображения и не появятся в окончательном варианте сайта.

Еще одним шагом является определение основных частей сайта. Вы не должны разбираться во *всей* информации, помещаемой на сайте, но вам необходимо знать, какие разделы и основные категории будут использоваться. Например, на сайте Moonlight Chronicles существуют три простые категории: e-Journal (электронный журнал), Books (книги) и Links (ссылки). На главной странице представлена информация о владельце сайта, а также о том, как пользоваться журналом, поэтому нет необходимости добавлять категорию About. Перечислив основные типы информации, которые вы хотите представить на сайте, вы можете распределить ее по категориям, а затем найти наилучший способ ее размещения на сайте.

СОВЕТ

Если у вас имеется стол или свободное пространство на полу, то вырежьте копии каждого изображения и заголовков статей, которые вы собираетесь поместить на сайте, и потрите их о войлок, чтобы они наэлектризовались. Теперь вы легко можете перемещать их, не занимаясь этим на компьютере. После того как вы с заказчиком найдете удачное сочетание, вы можете составить эскиз на бумаге. Иногда заказчику легче представить себе сайт подобным образом, чем на листе писчей бумаги.

Таким образом вы избавитесь от лишнего разочарования: так проще компоновать изображения, художественное оформление и блок-схемы. Вы и ваш заказчик в конце концов останетесь довольны.

Определение способа перемещения по сайту

Первым вопросом, который я решаю после создания основной страницы, является навигационная система. Должна ли я сделать интерактивно открывающиеся меню? Должна ли я использовать фреймы? Должна ли я использовать заменяющие друг друга изображения? Должна ли я растянуть процесс перехода пользователя к нужной ему странице? Должна ли я использовать поисковые машины? Вот именно здесь я и должна определиться, поскольку это решение повлияет на размещение слоев, количество других элементов взаимодействия, которые будут размещены на данной странице, а также на графические элементы.

Выбор желаемой системы из огромного множества навигационных систем очень сложен. Необходимо одновременно держать в уме все особенности каждой системы. Основными навигационными системами являются: раскрывающиеся меню (Pull-down Menus), непосредственные ссылки (Direct Links) и средства поиска (Search Interface).

Раскрывающиеся меню

- Удобны для сайтов с большим числом подразделов. Например, на сайте, посвященном уходу за кожей, в категории «Лосьон» возможно наличие подкатегорий «для рук», «для тела» и «общий».
- Занимают мало места, даже если содержат очень много информации. Раскрывающаяся часть не влияет на раскладку страницы.
- Просты при создании форм, поддерживаемых большинством браузеров.
- Реализуются за счет довольно короткого сценария.

Непосредственные ссылки

- Подходят для сайтов с небольшим числом ссылок.
- Легко интегрируются с внешним видом сайта без использования полей форм и других излишеств.
- Ссылки могут быть текстовыми и графическими.
- Реализуются стандартными средствами HTML без применения сценариев.
- Можно «оживить» с помощью сценариев на JavaScript с помощью кода «Image-rollover», доступного в большинстве современных HTML-редакторов.

Средства поиска

- Поиск позволяет найти любую информацию на сайте, но посетитель должен точно знать, что он ищет.
- Поиск легче реализовать с помощью множества существующих CGI-сценариев или при использовании различных ресурсов web-сервера, таких как ColdFusion и ASP.
- Поиск позволяет посетителю найти именно ту информацию, которая ему нужна, не «перерывая» весь сайт.

Не важно, какую систему навигации вы выберете в качестве основной, добавление возможности поиска всегда украсит ваш сайт. Большим недостатком поисковых систем, используемых в качестве основной навигационной системы, является то, что они не позволяют посетителю представить структуру сайта. Наилучший вариант — предоставить возможность использования как утилиты поиска, так и основной системы навигации, базирующейся на меню или ссылках, показывающих положение пользователя в структуре сайта. Это облегчит ориентирование пользователя и избавит его от необходимости возвращаться на главную страницу.

На рис. 12.3 представлена система меню, в которой не отображается ваше текущее положение на сайте. При использовании посетителем средств поиска для нахождения требуемой страницы в данном примере он не сможет определить, в каком месте сайта оказался. Это вынуждает возвращаться на главную страницу и осуществлять дальнейший поиск.

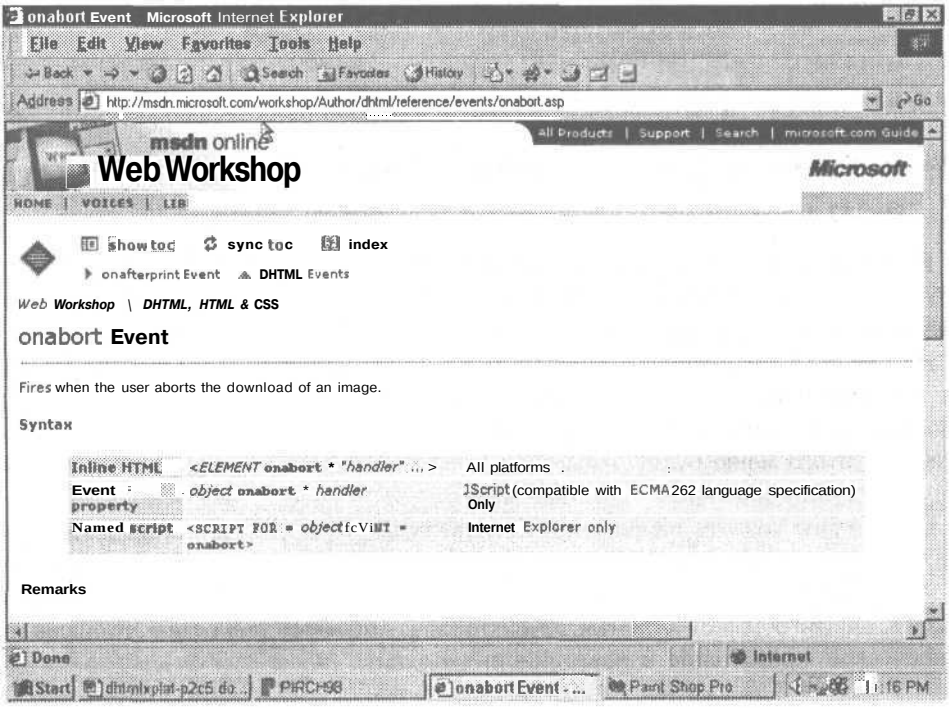


Рис. 12.3. В этом случае посетитель не представляет своего местонахождения на карте сайта

На рис. 12.4 представлена страничка сайта McAfee. В данном случае меню открыто в соответствующем месте карты сайта и в нем выделен пункт Support. Сайты, подобные этому, позволяют с помощью бокового меню определить свое местонахождение, независимо от того, как вы попали на данную страницу.

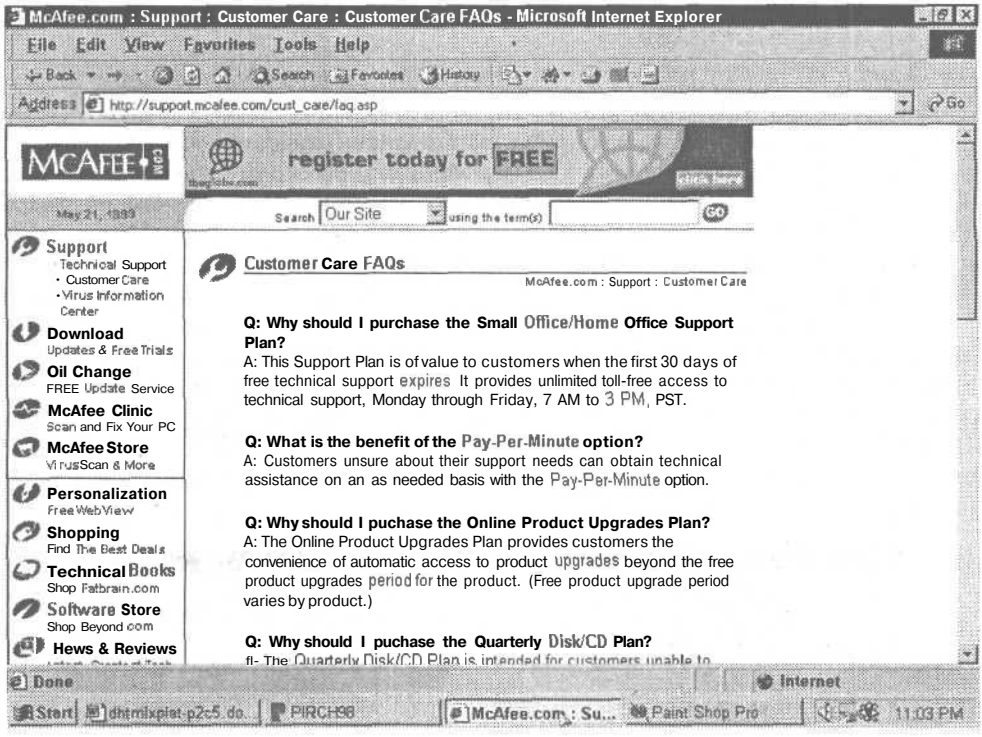


Рис. 12.4. В данном случае боковое меню подсказывает вам ваше местонахождение

В хорошо организованной системе навигации посетитель сразу видит, где он находится. При сложности получения этой информации посетители могут погрязнуть в массе данных и графики, не имеющей никакого порядка.

На рис. 12.5 представлена система меню сайта Acropolis, рассмотренного ранее в данной книге (это просто набросок, а не функционирующая система меню). Программисты могут использовать такие заготовки для определения того, как с помощью этого меню будет осуществляться взаимодействие и какие параметры будут доступны в каждом заголовке меню.

Вы должны создать наброски всей структуры меню прежде, чем приступите к программированию. Это позволяет сделать разработку сайта менее трудоемкой.

При помощи этих подсказок и изображений разработчик может заранее предсказать, каким образом клиенты, художники и продюсер пожелают реализовать графическое оформление и навигационную схему.

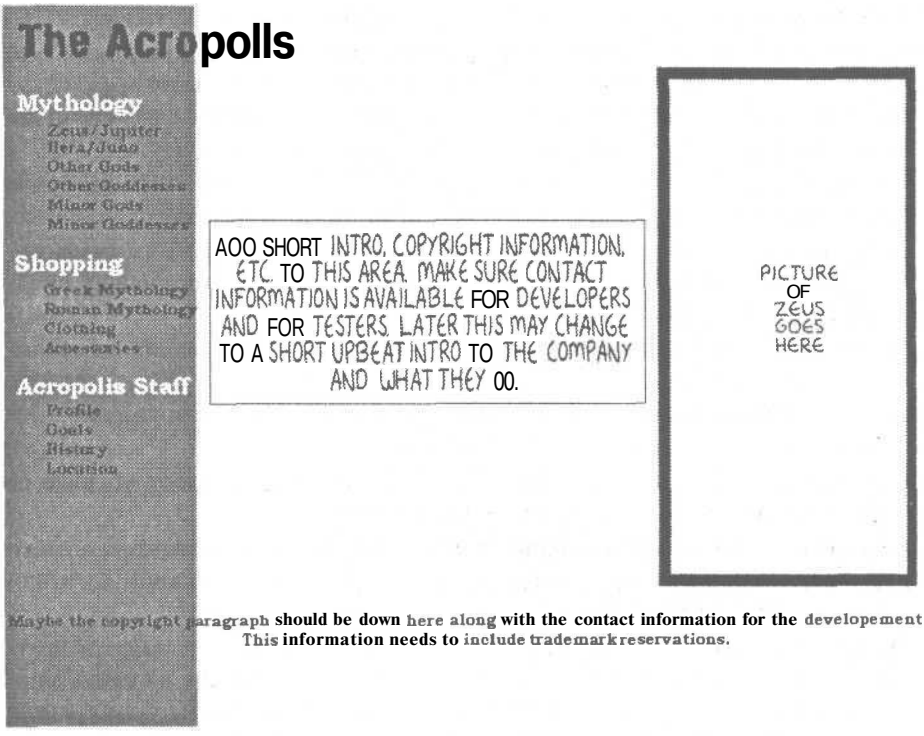


Рис. 12.5. Меню на этих набросках может использоваться программистами в качестве подсказки

Создание графики

Готовое решение по графике может быть первым законченным элементом, который увидит заказчик. Хотя беседа с продюсером должна изменить мнение клиента, большая часть исходной идеи должна оставаться целой. В противном случае окончательный результат может не понравиться клиенту.

В начале процесса создания web-сайта необходимо очень тщательно продумать художественное оформление. Графика должна вызывать интерес к представляемой информации. Посмотрите на рис. 12.1. Оформление этого сайта вызывает желание оказаться на открытом воздухе. А графика сайта, представленного на рис. 12.2, настраивает на что-то забавное.

Сайт Moonlight Chronicles имеет чувство света и притягивает читателей. За счет представления необычной информации и причудливого характера хозяина сайта (который тоже участвовал в художественном оформлении) простая, карикатурная графика обеспечивает основу всего сайта. Эти карикатуры сочетаются с фотографиями книг и журналов, продаваемых компанией Moonlight Chronicles. Эта комбинация создает резкий контраст, который заставляет посетителей снова и снова возвращаться на сайт.

Приведем несколько правил использования графики на вашем сайте:

- **Графика должна соответствовать содержанию сайта и дополнять его.** Людям никогда не понравится сайт, на котором расположено множество графики, которая требует загрузки в течение длительного времени и не приносит никакой практической пользы. Помните следующие советы:
  - для разделения больших потоков текста используется горизонтальная линия,
  - кнопки позволяют пользователю перемещаться с одного места на другое,
  - графики и диаграммы применяются для более интенсивного осмысления пользователем обсуждаемых вопросов или для уточнения представляемой информации,
  - изображения могут использоваться для обращения внимания на предомет обсуждения,
  - художественные элементы интерфейса обычно служат для указания посетителю на определенную информацию.

  Если нет основательных причин размещать на сайте графику, уберите ее и посмотрите, как без нее будет выглядеть сайт.
- **Старайтесь использовать графические элементы как можно меньшего размера.** Большие графические элементы требуют длительной загрузки и раздражают посетителей.
- **Не используйте слишком много цветов.** Вы когда-нибудь видели пеструю рубашку, на которую больно смотреть? Использование на сайте множества цветов имеет такой же эффект. Если вы хотите, чтобы у посетителя снова возникло желание посетить ваш сайт, не шокируйте его ошеломляющими цветами.
- **Проверьте ширину изображений и размеры шрифтов при любом разрешении.** Компьютер с разрешением 640x480 будет иметь значительно меньше пространства для размещения изображений, чем при разрешении 1024x768. Не рассчитывайте на использование посетителем разрешения 1024x768, пока вы не предусмотрите возможность замены изображений их уменьшенными копиями.

Планирование взаимодействия

Поскольку здесь мы ведем разговор о динамических сайтах, то вопрос взаимодействия с посетителем необходимо продумать более подробно, чем при создании простого статического web-сайта. В рассмотренном в данной книге примере существует несколько информационных слоев с анимацией, соответствующей тематике. Молнии Зевса летают по экрану, а Гермес переплывает по экрану для доставки свежего сообщения.

Обычно приходится довольно часто общаться с хозяином сайта, чтобы решить, что необходимо «оживить» и как осуществить реакцию сайта на действия посетителя, особенно сколько перемещений необходимо воспроизвести. Интерактивность вашего сайта может быть реализована следующими элементами:

- раскрывающиеся меню,
- перемещающиеся графические элементы,
- GIF-анимация,
- текущее время и дата, автоматически выводимые на экран,
- игры, которые позволят вам создать веселые лица и фигурки.

Модель изображения, представленная на рис. 12.6, показывает, как осуществляется планирование внешнего вида сайта Acropolis.

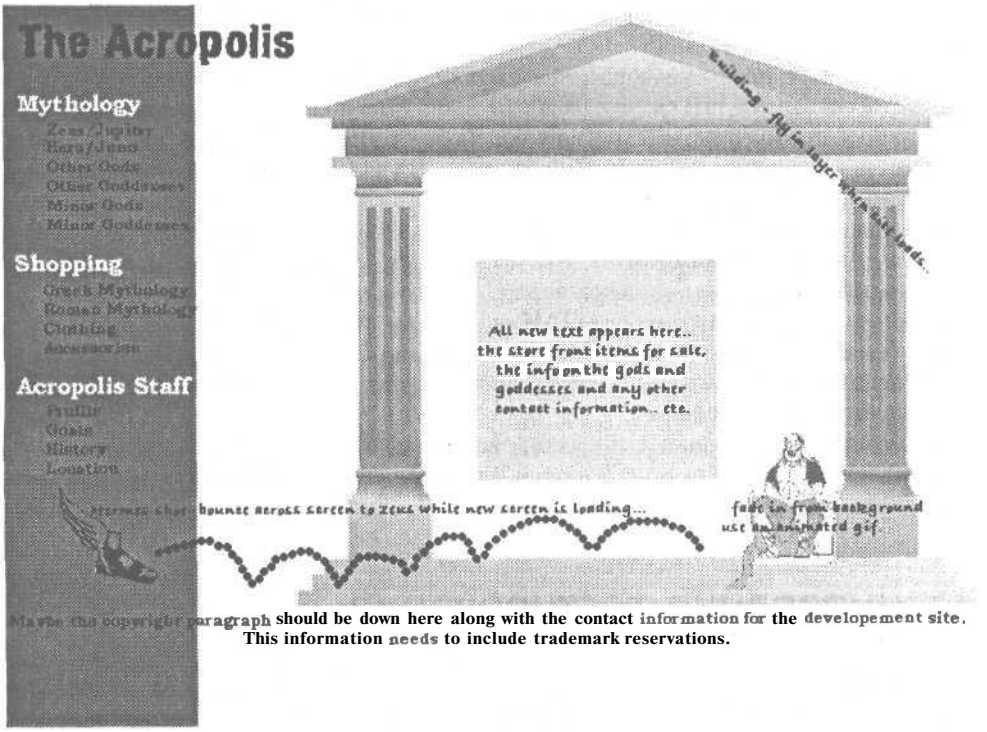


Рис. 12.6. Примечания на данном экране позволяют разработчику выбрать тип взаимодействия, связанный с данными изображениями

После того как наметились возможные варианты осуществления взаимодействия, вы переходите к определению отдельных слоев, позволяющих упорядочить изображения.

Определение слоев

Когда разработчики собираются создавать интерактивный сайт, им приходится манипулировать с целым рядом слоев, для того чтобы разместить все объекты в желаемом порядке. Для того чтобы изображения появлялись или текст формировался «на лету», необходимо использовать слои с изменяемыми характерис-

тиками. Существует множество технологий, например изменение цвета или src-атрибуты изображения, при использовании которых не требуется обязательное размещение объекта на отдельном слое.

Определение и организация требуемых слоев являются главным шагом. Принимая во внимание, что у вас есть свой логотип компании, перелетающий в документе с левой стороны вашего экрана в правый угол, вы можете сделать так, чтобы наименование компании тоже перемещалось от правого края к левому. Оба этих объекта, логотип и название компании, потребуют использования различных слоев с управлением сценарием. При одновременном перемещении эти слои будут перекрывать друг друга. Иногда перекрытие слоев специально требуется.

При работе с перекрывающимися слоями, имеющими определенное положение, вы должны убедиться, что каждый слой имеет соответствующий z-индекс и настройку прозрачности (см. главу 6). Любое изображение слоя должно иметь требуемую прозрачность краев. В противном случае изображения могут закрыть части слоев.

Вы можете представить слои в виде кусков прозрачного пластика, сложенных один над другим. На каждом из них имеется изображение. Если вы не укажете компьютеру с помощью z-индекса порядок, в котором они должны появляться, то вы должны расположить их в HTML-коде в соответствующем порядке. Видимость слоев определяется порядком их следования, который может неправильно воспроизводиться браузерами более старых версий, поэтому использование z-индекса является обязательным.

Как только вы завершите проектирование расположения слоев схемы и каждого объекта, вы можете приступать к планированию необходимого объема тестирования сайта.

Планирование тестирования

Тестирование — это один из этапов создания web-сайта, которым зачастую пренебрегают. Разработчики программного обеспечения знают, что перед тем, как завернуть программу в красивую обертку и положить на прилавок, ее необходимо тщательно протестировать. Почему же web-разработчики забывают об этом важном этапе? Если вы считаете, что это должен делать заказчик, то вы ужасно ошибаетесь. Web-сайты должны тестироваться человеком, не входящим в компанию, но не заказчиком, который и так потратил свои деньги на создание сайта.

Наибольшее внимание при тестировании необходимо обратить на удобство и простоту использования. Обязательно должна быть проверена вся система навигации. Необходимо убедиться, что она функционирует на каждой странице сайта и у посетителя всегда имеется возможность вернуться в исходную точку без активизации кнопки Back, применяемой для просмотра браузера. При использовании единой системы навигации в пределах всего сайта необходимо проверить ее удобство.

При подготовке к тестированию убедитесь, что в вашем распоряжении имеются все необходимые программные и аппаратные ресурсы или внешние источники, которые могут помочь. Тестирование можно считать завершенным, если

сайт проверен различными браузерами, установленными на различных операционных системах:

- Macintosh с Netscape Navigator 3
- Macintosh с Netscape Navigator 4
- Macintosh с Netscape Navigator 4.7x
- Macintosh с Netscape Navigator 6
- PC с Netscape Navigator 3
- PC с Netscape Navigator 4
- PC с Netscape Navigator 4.7x
- PC с Netscape Navigator 6
- UNIX или Linux с Netscape Navigator 3
- UNIX или Linux с Netscape Navigator 4
- UNIX или Linux с Netscape Navigator 4.7x
- UNIX или Linux с Netscape Navigator 6
- Macintosh с Internet Explorer 3
- Macintosh с Internet Explorer 4
- Macintosh с Internet Explorer 5
- PC с Internet Explorer 3
- PC с Internet Explorer 4
- PC с Internet Explorer 5
- UNIX или Linux с Internet Explorer 3
- UNIX или Linux с Internet Explorer 4
- UNIX или Linux с Internet Explorer 5

Как простой человек, вы не можете иметь доступ ко всем этим аппаратным средствам, но можете выполнить необходимое тестирование с помощью друзей через сеть или обратившись в специальную компанию, специализирующуюся на тестировании. Кроме того, при разработке динамических сайтов вы в первую очередь ориентируетесь на использование браузеров 4-й и более свежих версий, а также на корректное взаимодействие с браузерами 3-й версии. Для того чтобы ваши сайты не вызывали сильного расстройства у пользователей браузеров третьей версии, убедитесь, что сайт способен посылать им сообщение о том, как им следует просматривать данный сайт, и о том, что пора обновить программное обеспечение. В противном случае сайт должен высылать им упрощенный вариант страниц, который они смогут воспроизвести.

При тестировании убедитесь, что вы проверили *всё*. Не ограничивайтесь проверкой только основного меню, предполагая, что и остальные ссылки работают. Не рассчитывайте, что, поскольку в последний раз при просмотре страницы все работало нормально, теперь все так и осталось. Разработчики имеют обыкновение путать X и Y. При проверке страниц задайте себе следующие вопросы:

1. Я проверил все ссылки на каждом браузере?
2. Я смог вернуться на главную страницу с любой страницы?

3. Я выполнил все сценарии на каждой странице в каждом браузере?
4. Могу ли я приобрести любой из рекламируемых товаров и выбрать все его параметры с помощью каждого браузера?
5. Не происходили ли ошибки в выполнении сценариев, если щелкнуть на чем-нибудь при их выполнении?
6. Все ли мои сценарии правильно определяют тип браузера и подстраиваются под него?
7. Работают ли сертификаты безопасности в каждом браузере?
8. Всегда ли можно определить, в каком месте сайта я нахожусь?
9. Я записал все отмеченные ошибки?
10. Я перепроверил все после исправления ошибок?

СОВЕТ

Не забудьте включить время проверки в ваш график работы и в ваш бюджет. Если вы забудете включить этот пункт в бюджет, вы можете заметить, как ваш доход вылетит в трубу.

Тестирование должно начаться с момента начала разработки сайта. После того как вы нарисовали блок-схему, изучите ее внимательно и убедитесь, что вы не забыли ни одного пункта меню и ни одной ссылки. После этапа тестирования блок-схемы необходимо тестирование на этапе написания HTML-кода и окончательной реализации.

Попросите других людей, работающих с вами, просмотреть блок-схему и эскиз. Свежий взгляд поможет обнаружить незаметные ошибки. Не стоит одному проводить тестирование и исправление всех ошибок. Ваше чувство «собственника» по отношению к сайту может привести к пропуску досадных ошибок интерфейса или неуклюжести навигационных схем, поскольку вы привыкли выполнять определенные действия характерным для вас образом.

И наконец, необходимо допустить, что web-сайт неустойчив, до тех пор, пока не будет доказана его стабильность. После того как вы это осознаете, вы сможете взглянуть на сайт другими глазами. В действительности всякий, кто зайдет на сайт, осуществляет его проверку. Заказчик рассматривает его для того, чтобы убедиться, что сайт удовлетворяет всем выдвинутым требованиям. Разработчик просматривает его на предмет функционирования и правильности выполнения определенных операций. Но реально продюсер или заказчик не в состоянии проверить сайт. Он не может быть проверен в одиночку ни разработчиками, ни покупателем. Если все не поработают над улучшением сайта, вы можете потерять множество клиентов из-за неудовлетворенности посетителя.

13 Общедоступность

Многие люди используют Интернет в качестве неисчерпаемого источника информации. Это наилучшее средство отыскания тысяч различных фактов и мнений, связанных с использованием новейших технологий. Существующие средства могут обеспечить наивысшую общедоступность, позволяющую использовать Интернет не только слепым, но и тем, кто не может перемещать мышь или другое координатное устройство. Эти средства могут помочь любому приобщиться к миру глобальной сети. Вы когда-либо представляли себе, насколько трудно найти информацию в сети, если английский язык не является для вас родным? Или если вы можете войти в Интернет только с помощью устаревшего браузера Linux?

Сколько раз вам приходилось отключать «картинку», загружающуюся в браузер, чтобы можно было посмотреть на документ, автор которого не позаботился об определении атрибутов height и width тега <img>? Как узнать, какое изображение необходимо загрузить, а какое можно полностью игнорировать, если не указан атрибут alt? В этом атрибуте должно быть представлено описание, позволяющее представить, что собой представляет это изображение или для чего оно требуется. Например, вы можете указать в нем «переход на главную страницу», и, независимо от видимости изображения, вы сможете узнать, куда можно переместиться, щелкнув на этом месте. Я не единственная, кто хотел бы иметь возможность просматривать страницу без «картинок».

Web-мастера могут поместить на созданные ими страницы значок доступа к сети национального центра CPB/WGBH National Center for Accessible Media's (NCAM). Наличие на сайте этого значка свидетельствует о том, что дизайн сайта позволяет обращаться к нему пользователям с ограниченными функциональными возможностями. Значок сопровождается следующим описанием: «A globe, marked with a grid, tilts at an angle. A keyhole is cut into its surface». A alt-текст значка сообщает: «Web Access Symbol (for people with disabilities)». Значок можно найти на страничке http://www.wgbh.org/wgbh/pages/ncam.

Руководящие принципы при создании общедоступных сайтов

При создании сайтов, доступных для людей с ограниченными возможностями, необходимо руководствоваться следующими принципами:

1. Не используйте таблицы стилей.
   - Убедитесь, что ваши страницы будут верно воспроизводиться даже без поддержки браузером таблиц стиля, или создайте для него альтернативные страницы.
   - Убедитесь, что после отключения таблиц стилей информация на страницах не повторяется.
2. Обязательно используйте альтернативный (сопроводительный) текст для изображений.
   - Используйте alt-атрибут для каждой «картинки». Это является требованием спецификации HTML *4.x*.
   - Используйте изображения только для представления важной информации, а для сложных изображений с помощью атрибута `longdesc` установите ссылку на отдельный документ, содержащий более подробное описание.
   - Упрощайте изображения. В качестве маркеров списка используйте звездочку (*), вместо изображений цифр используйте сами цифры, а вместо невидимых изображений, используемых для отступа, вставляйте символ пробела.
3. Корректно применяйте изображение-карту.
   - Помимо карты, обязательно разместите на странице текстовые ссылки (как встроенные, так и внизу страницы).
   - Воспользуйтесь кэшированием изображений-карт на стороне клиента, поскольку его сейчас поддерживает большинство клиентов.
   - Для присвоения наименования объектам <area> изображения-карты используйте атрибут `title`.
4. В тексте ссылки должна содержаться полезная информация.
   - Текст должен быть лаконичным.
   - Не создавайте односложные ссылки. Не стоит указывать «Щелкните здесь» или «Приготовься», когда остальной сайт посвящен обсуждению вкуса мороженного. Необходимо использовать короткие фразы, например «Список видов мороженого», или «Рецепты мороженого», или даже «Изготовление мороженого».
5. Обеспечьте возможность перемещения по сайту с использованием клавиатуры.
   - Используйте для перехода от ссылки к ссылке или от области к области изображения-карты клавишу Tab, определив с помощью HTML Tab-

порядок. Как правило, пользователь перемещает курсор слева направо и сверху вниз. При необходимости определите свойство `tabindex` для определения порядка перемещения курсора.

- При помощи атрибута accesskey определите «горячие клавиши» для обращения ко всем элементам управления и к ссылкам. «Горячие клавиши» должны быть подчеркнуты в подписях элементов управления. Для браузеров, которые не делают это автоматически, воспользуйтесь тегом `<u>`.

6. Обеспечьте альтернативные средства для всех элементов управления и апплетов.
   - Создайте альтернативные страницы с использованием стандартного HTML.
   - Для присвоения имени каждому элементу управления используйте атрибуты `label` и/или title. Хорошо бы также использовать атрибут id для идентификации каждого объекта на странице.
7. Альтернативные страницы не должны использовать фреймов.
   - Используйте элемент `<noframes>`, чтобы показать страницу без использования фреймов, или обеспечьте ссылку на страницу без фреймов.
   - Задайте название каждого фрейма атрибутом `title`.
8. Альтернативные страницы не должны содержать сложных таблиц.
   - Альтернативные страницы вообще должны быть без таблиц, поскольку в случае игнорирования табличных тегов прочесть представленную информацию будет просто невозможно.
   - Убедитесь, что таблицы представлены корректно. Слева направо и сверху вниз.
   - Для назначения описания строкам и столбцам таблицы используйте атрибут `title`. Например, «Продажи третьего квартала» или «Стоимость грузовиков Tonka».
9. Поддерживайте параметры форматирования пользователя.
   - Не используйте форматирование с помощью специальных шрифтов, цветов или размеров, которые воспроизводятся *только на вашем компьютере*.
   - Используйте объекты HTML по их прямому назначению. Например, <hl> — для оформления заголовков, а <p> — для выделения абзацев.
   - Всегда исходите из того, что весь внешний вид может измениться при изменении размеров окна.
10. Не используйте «бегущую строку».
    - Не стоит применять «бегущую строку» (`<marquee>`), поскольку воспроизвести ее может только Internet Explorer.
    - Если же вы все-таки хотите использовать «бегущую строку», то создайте и альтернативную страницу со статическим текстом для посетителей, у которых отключена функция воспроизведения анимации.
    - Никогда не используйте «бегущую строку» в качестве ссылки.

При использовании этих принципов независимо от настроек браузера или ограниченных возможностей пользователя ваш сайт станет доступным для большинства посетителей. Конечно же, все необходимо многократно протестировать, чтобы быть уверенным в окончательном результате.

Использование атрибутов alt, title и longdesc

В предыдущем списке я упоминала эти атрибуты. Они предназначены для пре доставления посетителю дополнительной информации. В одних случаях эта ин формация будет воспроизводиться вместо изображения или апплета. В други случаях — только как текст подсказки. А иногда — для определения ссылки дл перехода посетителя к другому документу для более подробного изучения мат риала.

alt

Первым и самым известным является атрибут `alt`. Этот атрибут обеспечив; сопроводительный текст для таких элементов, как `<applet>`, <area>, `<img>` и `<inpu` которые могут не всегда воспроизводиться браузером пользователя. При нев можности представления одного из этих объектов на его месте будет предст лен текст, содержащийся в атрибуте `alt`. Он позволяет пользователю получ сведения об общем содержании документа, а также не пропустить важную формацию, относящуюся к данному объекту. В случае использования объек `<img>` или <area> текст представит ссылку на другой документ. С помощью ; тернативного текста можно представить полные сведения о содержании до мента для пользователей, использующих текстовые терминалы, для тех, браузеры не поддерживают формы, для посетителей, использующих рече синтезатор для воспроизведения содержания страницы, для людей с ослаб ным зрением или для посетителей, у которых отключена функция воспрои дения графики.

Текст, помещаемый в alt-атрибут, должен быть «полезным», а не раздра пользователя банальными фразами «Щелкните здесь», которые только доб ют путаницу. Примером грамотного сопроводительного текста может явл такой: «Переход па сайт WallowaValley.com». Формат атрибута `alt` очень пр

```
alt="строка_сопроводительного_текста"
```

ПРИМЕЧАНИЕ

В HTML 4.x этот атрибут является обязательным для тегов `<img>` и `<area>`. А для эле `<applet>` и `<input>` — дополнительным.

title

Атрибут `title` отличается от работы атрибута `alt` (который автоматичес производит сопроводительный текст, если нет возможности показать эл

Содержание атрибута title не выводится на место, занимаемое элементом, но обеспечивает представление его названия, которое воспроизводится рядом браузеров. Наличие этого атрибута очень полезно для тех, кто использует браузер для поиска информации. Синтаксис этого атрибута:

```
title= "текст заголовка"
```

longdesc

Атрибут longdesc подобен атрибуту alt, поскольку его основная задача также заключается в том, чтобы представить описание объекта, но только он представляет непосредственный адрес (URL) этого описания, а не само описание. Этот атрибут действителен лишь для трех элементов HTML: <frame>, <iframe> и <img>. Причем элементы <frame> и <iframe> воспринимают этот атрибут несколько иначе, чем элемент <img>.

При использовании в элементах <frame> и <iframe> атрибута longdesc он обеспечивает ссылку на файл с дополнительным описанием содержания фрейма. Это описание дополняет содержание атрибута title и особенно полезно при использовании браузеров, не воспроизводящих графические изображения, поскольку позволяет пользователю понять структуру документа.

При использовании данного атрибута в элементе <img> он также определяет ссылку на дополнительное описание изображения, но это описание является дополнением к alt-атрибуту изображения. При использовании этого атрибута с изображением-картой он должен содержать описание всей карты, а не отдельной области.

Для всех элементов, использующих атрибут longdesc, синтаксис одинаков:

```
longdesc = "url"
```

При разработке сайта, с которым смогут работать люди с ограниченными возможностями, необходимо внедрить в документы специальные акустические таблицы стилей, позволяющие управлять работой синтезаторов речи.

Акустические таблицы стилей и синтезаторы речи

Вы бы хотели услышать чтение произведений Шекспира с английским акцентом? А как насчет рекламы техасского соуса для жаркого с чисто западным произношением? Вы когда-нибудь пытались слушать докладчика с ровным монотонным голосом? Смог бы этот звук завлечь вас в роскошный ресторан или отправиться прокатиться верхом в диких прериях? Меня бы нет! Столкнувшись с этим, мы не можем больше забыть о звуковом оформлении web-страниц.

Я не говорю о дополнительном хриплом фоновом звуке или воспроизведении неприятных звуков, сопровождающих щелчок на кнопке. Я говорю об акустических таблицах стилей, которые при правильном сочетании с речевым синтезатором позволяют сделать Интернет «живым».

При постоянном росте числа социальных групп, пользующихся сетью, Интернет-консорциум принял решение о необходимости внедрения в таблицу стилей

элементов, обеспечивающих необходимую поддержку для людей с ограниченным зрением или незрячих. При попытке акустического воспроизведения текстового документа необходимо сочетать технологию синтеза речи и различных аудиосимволов. Зачастую для акустического синтеза текста вы должны преобразовать документ в текстовой файл и затем воспроизвести его с помощью встроенного «диктора». Такой вариант является менее информативным, поскольку визуальная структура документа не сохраняется. На это и был сделай основной упор в работе Интернет-консорциума, которая увенчалась разработкой приведенных ниже акустических свойств таблиц стиля, обеспечивающих как визуальное, так и акустическое предоставление документа, что позволяет полностью осознать содержание документа. Каждое описание сопровождается представлением синтаксиса атрибута.

ПРИМЕЧАНИЕ

Акустические (звуковые) таблицы стилей могут применяться не только для людей с ограниченным зрением. Они могут использоваться для сообщения информации пользователям, управляющим машиной, пользователям, не умеющим читать, при создании промышленной или корпоративной системы документации, а также для различных развлечений.

- Azimuth: определяет угол, с которого должен исходить звук синтезатора речи. Этот атрибут может использоваться совместно с атрибутом elevation, указывая точное пространственное расположение голоса для документов, состоящих из множества частей.

  azimuth=<angle> | [[left-side | far-left | left | center-left | center | center-right | right | far-right j right-side] || behind] | leftwards | rightwards | inherit

- Elevation: определяет высоту (угол) места, откуда должен исходить звук.

  elevation=<angle> | below | level | above | higher | lower | inherit

- Cue-before: определяет звук, который будет предшествовать воспроизведению содержания элемента.

  cue-before'=<*uri*> \ none | inherit

- Cue-after: определяет звук, который будет воспроизведен после «озвучивания» содержания элемента.

  cue-after'=<*uri*> \ none | inherit

- Cue: предназначен для более быстрого установления значений атрибутов cue-before и cue-after.

  cue=[<'cue-before'> || <'cue-after'>] | inherit

- Pause-before: определяет длительность паузы, которая должна быть выдержана перед воспроизведением содержания элемента.

  pause-before=<time> | <percentage> | inherit

- Pause-after: определяет длительность паузы, которая должна быть выдержана после воспроизведения содержания элемента.

  pause-after=<time> | <percentage> | inherit

- Pause: предназначен для более быстрого установления значений атрибутов pause-before и pause-after.

  ```
  pause=[ [<time> | <percentage>]{1,2} ] | inherit
  ```

- Pitch: определяет высоту тона «голоса» синтезатора.

  ```
  pitch=<frequency> | x-low | low | medium | high | x-high | inherit
  ```

- Pitch-range: определяет допустимый диапазон «голосов» синтезатора.

  ```
  pitch-range=<number> | inherit
  ```

- Play-during: определяет фоновый звук, исполняемый при воспроизведении содержания элемента.

  ```
  play-during=<url> | mix? repeat? | auto | none | inherit
  ```

- Richness: определяет качество воспроизводимого голоса.

  ```
  richness=<number> | inherit
  ```

- Speak: определяет, должен ли текст воспроизводится устно, и если да, то каким образом.

  ```
  speak= normal | none | spell-out | inherit
  ```

- Speak-numeral: определяет способ воспроизведения чисел.

  ```
  speak-numeral=digits | continuous | none | inherit
  ```

- Speak-punctuation: определяет способ воспроизведения знаков пунктуации.

  ```
  speak-punctuation-code | none | inherit
  ```

- Speech-rate: определяет скорость, с которой синтезатор будет произносить текст.

  ```
  speech-rate=<number> | x-slow | slow | medium | fast | x-fast | faster
  | slower | inherit
  ```

- Stress: определяет силу ударения. При использовании атрибута Stress в английском языке, в котором ударение используется для выделения главных слов предложения, вы можете выделить первостепенное, второстепенное и третичное ударения.

  ```
  stress=<number> | inherit
  ```

- Voice-family: представляет собой разделенный запятыми, упорядоченный по приоритетам список семейств голосов, определяющих голоса, используемые при воспроизведении текста. Примером семейства голосов может быть «male», «female» и даже «grandmother» и «grandfather».

  ```
  voice-family=[[<specific-voice> | <generic-voice> ],]* [<specific-
  voice> | <generic-voice> ] | inherit
  ```

- Volume: определяет среднюю громкость. Другими словами, корректирует динамический диапазон, которого придерживается синтезатор.

  ```
  volume=<number> | <percentage> | silent | x-soft | soft | medium |
  loud | x-loud | inherit
  ```

Использование этих свойств в акустической таблице стилей позволяет управлять устным воспроизведением документов. Так же как и CSS, акустические таблицы стилей можно подключать к множеству документов. А устное воспроизведение web-сайта можно осуществить последовательно с его визуальным появлением.

UNICODE и использование шрифтов

Еще одним способом, используемым при создании web-сайтов с последовательным появлением, является язык UNICODE. UNICODE — это система кодирования символов, подобная ISO-10646, содержащая буквы большинства языков мира. Поскольку каждая буква закодирована отдельным кодом, два компьютера, использующие UNICODE, могут осуществлять обмен информации символами, не допускающими двоякого представления. UNICODE — это не разновидность наборов символов, он содержит точное описание изображения каждой буквы или языковых определений. Это 16-битная универсальная система кодирования символов.

За счет использования специальных языков в HTML- и CSS-документах, использующих символы в UNICODE-кодировке, возможно точное воспроизведение изображений букв любого языка, а не «мусора». При определении языка браузер пользователя может получить утилиты, необходимые для корректного воспроизведения документа, или представить пользователю сообщение об ошибке воспроизведения документа с точным указанием ее причины. Это позволяет рассчитывать на возможность интернационализации разработки web-проектов. Поддержка глобализации в сети является элементом зарождения международного сообщества. UNICODE является средством представления в сети информации на различных языках, а не только на английском, являющимся в настоящее время доминирующим.

Язык HTML может быть интернациональным, поскольку он за счет явного определения языка текста позволяет представлять слова, из которых состоит документ, различными наборами символов. Базовым набором символов, изначально определенным для HTML-документов, был ISO Latin 1. Теперь используется ISO 10646 или UNICODE, который содержит примерно 34 тысячи символов, используемых во всех языках мира (более 50 000 символов). Для представления текстовых элементов с помощью символов указанного языка в HTML используются атрибуты `lang`, `dir` и `align`. Они могут быть применены ко всему документу.

Можно создать один документ на различных языках. Различные версии документов доступны и могут автоматически загружаться браузером при помощи элемента `<link>`. Поскольку некоторые языки могут использовать двунаправленный текст (помимо слеванаправо еще и справаналево) или текст, располагаемый только справа налево, то для данного вида языков помимо атрибута dir применяются специальные параметры <bdo> и <pre>.

ПРИМЕЧАНИЕ

Если необходимо расположить текст сверху вниз, как во множестве азиатских языков, то придется создать XML-документ и форматировать его с помощью языка XSL.

За счет определения кодировки символов перед представлением документа HTTP-серверы также могут помочь браузеру посетителя в «интернационализации». Протокол HTTP может осуществлять распределение схем кодировки символов, воспринимаемых браузером клиента. При наличии множества версий

документа на различных языках он может сообщить серверу, какая версия документа требуется.

ПРИМЕЧАНИЕ

В настоящее время Интернет-консорциум занимается разработкой международного формата документов. Более подробные сведения по этому вопросу можно получить по адресу: http://www.w3.org/International/Activity.html.

Основная цель применения UNICODE — позволить web-браузерам представлять документ символами именно того языка, на котором он написан. Если вы укажете язык и направление, в котором текст должен быть представлен символами UNICODE, то это в значительной степени снизит вероятность представления пользователю содержания вашего документа в виде «мусора».

При помощи HTML-атрибута `lang` можно указать код множества языков. Некоторые из этих кодов не соответствуют принятому двухбуквенному коду, представленному в табл. 13.1 Часть из них, представленные ниже, используют несколько иную систему обозначения представленного языка:

`en-UK`: английский (Великобритания);

`en-cockney`: английский, версия кокни;

`i-navajo`: язык навахо, используемый американскими туземцами;

`x-klingon`: основной элемент x указывает на экспериментальный язык. В данном случае — язык клингон, придуманный в фильме «Звездный путь».

ПРИМЕЧАНИЕ

Все языки программирования, такие как Си, Java, JavaScript, Basic и Pascal, называемые few, явно ограждаются от использования программного кода.

Таблица 13.1. Установленные коды языков

Язык	Код языка	Языковая группа/подгруппа/семья
Абхазский	ab	Иберийско-кавказские
Азербайджанский	az	Тюркские
Аймара	ay	Индейские
Албанский	sq	Индоевропейские
Амхарский	am	Эфиосемитские
Английский	en	Германские
Арабский	ar	Семитские
Армянский	hy	Индоевропейские
Ассамский	as	Индийские
Афар	aa	Кушитские
Африкаанс	af	Германские
Баскский	eu	Баскский
Башкирский	ba	Тюркские
Белорусский	be	Восточнославянские
Бенгальский	bn	Индийские
Бирманский	my	Тибето-бирманские
Бирхор	bh	Аустроазиатские
Бислама	bi	Пиджин (англоязычный)
Болгарский	bg	Южнославянские

продолжение

Таблица 13.1 *(продолжение)*

Язык	Код языка	Языковая группа/подгруппа/семья
Бретонский	br	Кельтские
Бхотия	dz	Китайско-тибетские
Валлийский	cy	Кельтские
Венгерский	hu	Финно-угорские
Волапюк	vo	Международные искусственные
Волоф	wo	Западноатлантические
Вьетнамский	vi	Вьетмыонгские
Галисийский	gl	Романские
Гренландский	kl	Эскимосско-алеутские
Греческий	el	Греческие
Грузинский	ka	Иберийско-кавказские
Гуарани	gn	Тупи-гуарани
Гуджарати	gu	Индийские
Гэльский (шотландский)	gd	Кельтские
Датский	da	Скандинавские
Зулу	zu	Банту
Иврит	he	Семитские
Идиш	yi	Германские
Индонезийский	id	Австронезийские
Интерланг	ie	Международные искусственные
Интерлингва	ia	Международные искусственные
Инуктитут	iu	Эскимосско-алеутские
Инупик	ik	Эскимосско-алеутские
Ирландский	ga	Кельтские
Исландский	is	Скандинавские
Испанский	es	Романские
Итальянский	it	Романские
Йоруба	yo	Ква
Казахский	kk	Тюркские
Каннада	kn	Дравидийские
Каталанский	ca	Романские
Кашмири	ks	Дардские
Кечуа	qu	Индейские
Киньяруанда	rw	Банту
Киргизский	ky	Тюркские
Кирунди	rn	Банту
Китайский	zh	Китайско-тибетские
Корейский	ko	Корейские
Корсиканский	co	Романские
Коса	xh	Банту
Курдский	ku	Иранские
Кхмерский	km	Монкхмерские
Лаосский	lo	Тайские
Латинский	la	Италийские
Латышский	lv	Балтийские
Лингала	ln	Банту

Язык	Код языка	Языковая группа/подгруппа/семья
Литовский	lt	Балтийские
Македонский	mk	Южнославянские
Малагасийский	mg	Австронезийские
Малайский	ms	Австронезийские
Малаялам	ml	Дравидийские
Мальтийский	mt	Семитские
Маори	mi	Полинезийские
Маратхи	mr	Индийские
Молдавский	mo	Романские
Монгольский	mn	Монгольские
Науру	na	Микронезийские
Немецкий	nl	Германские
Немецкий	de	Германские
Непальский	ne	Индийские
Норвежский	no	Скандинавские
Окситанский	oc	Романские
Орано (галла)	om	Кушитские
Ория	or	Индийские
Пенджаби	pa	Индийские
Персидский (Фарси)	fa	Иранские
Польский	pl	Западнославянские
Португальский	Pt	Романские
Пушту	PS	Иранские
Ретороманский	rm	Романские
Румынский	ro	Романские
Русский	ru	Восточнославянские
Самоа	sm	Полинезийские
Санго	sg	Адамауа-восточные
Санскрит	sa	Индийские
Сербохорватский (хорватосербский)	sh	Южнославянские
Сербский	sr	Южнославянские
Сесото	st	Банту
Сингальский	si	Индийские
Синдхи	sd	Индийские
Сисвати	ss	Банту
Словацкий	sk	Западнославянские
Словенский	sl	Южнославянские
Сомали	so	Кушитские
Суахили	sw	Банту
Сунданский	su	Австронезийские
Тагальский	tl	Филиппинские
Таджикский	tg	Иранские
Тайский	th	Тайские
Тамильский	ta	Дравидийские
Татарский	tt	Тюркские
Телугу	te	Дравидийские
Тибетский	bo	Китайско-тибетские
Тив	tw	Бантоидные

продолжение

Таблица 13.1 *(продолжение)*

Язык	Код языка	Языковая группа/подгруппа/семья
Тиграй	ti	Эфиосемитские
Тонга	to	Полинезийские
Тсвана	tn	Банту
Тсонгский	ts	Банту
Турецкий	tr	Тюркские
Туркменский	tk	Тюркские
Узбекский	uz	Тюркские
Уйгурский	ug	Тюркские
Украинский	uk	Восточнославянские
УРДУ	ur	Индийские
Фарерский	fo	Скандинавские
Фиджи	fj	Океанийские
Финский	fi	Прибалтийско-финские
Французский	fr	Романские
Фризский	fy	Германские
Хауса	ha	Чадские
Хинди	hi	Индийские
Хорватский	hr	Южнославянские
Чешский	cs	Западнославянские
Чжуанский	za	Тайские
Шведский	sv	Скандинавские
Шона	sn	Банту
Эсперанто	eo	Международные искусственные
Эстонский	et	Прибалтийско-финские
Яванский	Jv	Австронезийские
Японский	ja	Японский

Эти коды, как и множество других стандартных данных, определены Международной организацией по стандартам (International Standards Organization, ISO). ISO объединяет усилия национальных организаций стандартизации более сотни стран и до сих пор не имеет отношения ни к одному правительственному органу. Ее единственная задача — обеспечить распространение стандартов по всему миру, способствуя международному обмену товаров и услуг. Интернет становится одним из основных средств предоставления услуг и информации в международном деловом сообществе. ISO также помогает развить кооперацию в интеллектуальной, научной, технологической и экономической сферах общения. Результат работы этой организации представляется в виде международных соглашений, публикуемых в качестве международных стандартов. Именно этими стандартами определяются стандарт ISO 9000, с помощью которого осуществляется перемещение документов, и UNICODE-стандарт, действительно способствующий глобализации Интернета.

Последние подсказки, уловки и хитрости

14

Существует множество вещей, которые позволяют разработчику сделать web-сайт более привлекательным для посетителей. Можно реализовать все: от подвижных изображений до отслеживания предпочтений пользователя и персонализации страниц. При этом необходимо осознавать, что некоторым посетителям это может нравиться, а других — просто раздражать. Помните, что представленные здесь уловки и хитрости не могут постоянно применяться на каждой созданной вами странице, по они иногда могут облегчить их создание и сделать их более веселыми.

ПРИМЕЧАНИЕ

Некоторые из приведенных здесь уловок и хитростей могут вызвать проблемы у посетителей, использующих синтезатор речи и прочие устройства для людей с ограниченными функциональными возможностями. Перед добавлением объемных сценариев это необходимо обязательно учитывать.

Советы по завоеванию популярности у посетителей

Один из основных принципов Интернета — для того чтобы посетитель снова и снова возвращался на ваш сайт, необходимо произвести положительное впечатление и доставить ему удовольствие от общения с сайтом. Статистические данные свидетельствуют, что в среднем пользователь, до того как найдет то, что ему нужно, посещает семь сайтов. Как его можно заставить вернуться именно к вам?

Это вопрос со множеством ответов, ни один из которых не дает стопроцентной гарантии. Если у вас имеется настоящий магазин, то большинство из нижеприведенных принципов, используемых в обычном магазине также, необходимо реализовать и в электронном магазине web-сайта:

- Общая обходительность и вежливость. Эта позиция — необходимое условие, которое ведет к тому, что посетитель захочет сюда вернуться.

- Распродажи и скидки могут привлечь клиентов. Можно определить скидки на второй товар при приобретении клиентом первого. Организуйте БЕСПЛАТНУЮ доставку.
- Справочный отдел позволит ответить на все вопросы покупателя, которому не придется куда-либо звонить или обращаться по электронной почте.
- Очень важна персонализация, осуществляющая идентификацию покупателя, уточняющая, как он сюда попал, и отслеживающая, каким товарам он отдает предпочтение.

А вот еще несколько моментов, над которыми необходимо задуматься, чтобы добиться соответствия перечисленным выше советам.

Подумайте, пожалуйста, об этикете

Необходимо, чтобы вы работали в интересах своих пользователей, а не боролись с ними. При реализации DHTML на своих страницах не забывайте следовать простым правилам этикета:

1. **Не жертвуйте производительностью компьютера ради всяких излишеств.** Мы живем в век удовольствий, и Интернет только способствует этому. Учтите это при разработке web-сайтов. Конечно же, иногда сайты разрабатываются именно для того, чтобы разместить различные украшения, такие как фильмы, музыку и интерактивные игры, но они включаются туда по просьбе клиента. Хорошо! Но только убедитесь, что задержки выполнения являются допустимым, продуманным и спланированным результатом, а не являются указанием на необходимость увеличения мощности компьютера.
2. **Сделайте так, чтобы динамическое содержание несло смысловую нагрузку.** Вы когда-нибудь заходили на очень яркий и «шумный» сайт, на котором все эффекты являются лишь ненужным украшением, не имеющим никакого отношения к содержанию web-сайта? Такой сайт можно сравнить с пением «Боже, царя храни» при игре в бейсбол. Наилучшим способом реализации художественных возможностей DHTML является использование их «со смыслом». Например, при реализации системы навигации в качестве средства перемещения между сценами или для привлечения внимания к определенным областям страницы. Конечно, летающий дракон выглядит действительно впечатляюще, но все-таки избегайте использования подобного рода эффектов. Оставьте этот эффект для другой странички, к которой он подходит!
3. **Постоянно повторяющееся движение раздражает.** Вы, конечно же, представляете, насколько неприятно звучание некоторых wav или midi-файлов, когда они включены в режиме непрерывного воспроизведения. Почему же мы считаем, что объекты, выполняющие повторяющиеся движения, не вызывают раздражения? А теперь об элементе <blink>. При чтении абзаца или заголовка мигающий текст более неприятен, чем грязь, попавшая из лужи в ваши туфли!
4. **Будьте учтивы.** Возможно, ваша мама говорила вам: «Mind your P's and Q's». Вам необходимо вспомнить эти слова при размещении информации, которая станет доступной всему миру. В некоторых случаях из-за оскор-

бительных высказываний вы можете лишиться работы или даже будете привлечены к уголовной ответственности. Подумайте о тех, кто увидит ваши страницы. Будут ли они доступны лишь пользователям корпоративной сети или станут доступны из Интернета? В первом случае вы четко представляете круг «зрителей», но даже в этом случае вы не знаете пола, национальности и религиозных убеждений каждого посетителя. Политическая корректность как обязательное требование должна присутствовать во всем, и в конечном итоге она «сыграет на вас».

5. **Проверять! Проверять! Проверять!** Это правило очевидно, но иногда оно игнорируется. Не важно, какие эффекты вы используете, не важно, как они реализуются. Главное, что ваши страницы должны безукоризненно воспроизводиться на любой системе, на которой они будут просматриваться. В отношении Интернета имеется в виду, что DHTML-документы с использованием CGI или JavaScript/JScript-сценариев должны корректно воспроизводиться всеми основными браузерами, начиная с версии 4.0, включая даже те, которые еще и не разработаны. В корпоративной сети достаточно лишь использовать команды одного языка (JScript или JavaScript), поскольку вы знаете, что в компании используются только браузеры Microsoft или Netscape.

Поскольку я только что сказала «всеми основными браузерами, начиная с версии 4.0», то вы, вероятно, удивитесь: а что случится с теми, кто до сих пор использует браузеры версий от 1.0 до 3.x? Сейчас объясню. Если вы занимаетесь программированием на DHTML, то вы рассчитываете только на людей, использующих современное программное обеспечение, позволяющее воспроизвести результат вашего труда, но ваши сценарии должны быть «вежливы» в отношении и тех браузеров, которые вас не «понимают». Если вы хотите угодить старым браузерам, то создайте отдельный набор документов, в которых используется лишь стандартный HTML.

Этикет написания сценариев также важен. При разработке сценариев имеется необходимость сотрудничества с другими разработчиками, которые, возможно, будут использовать ваш сценарий в будущем. Всякий раз, создавая web-сайт, оставьте сценарий доступным и пользователю, что позволит ему увеличить свой опыт.

При реализации сценариев не забывайте о следующих правилах:

1. **Сначала надо подумать.** Добавление интерактивных сценариев увеличивает время загрузки страниц. При использовании сценариев убедитесь, что задержки выполнения соответствуют расчетным и не требуют увеличения мощности компьютера.
2. **Осуществляйте поддержку браузеров старшего поколения.** Нет ничего хуже, чем разочарование ваших посетителей от неработающего сценария. Ваш сценарий прекрасно работает только на браузерах 4.x, а как быть с их предшественниками? Сценарии, использующие новые технологии, могут не работать на браузерах старшего поколения. Проверьте работу сайта на всех браузерах. Если они у вас не установлены, обратитесь к друзьям или родственникам, которые еще пользуются старым программным обеспечением. Ваш сценарий не обязательно должен работать на старых браузерах, но он должен ими корректно игнорироваться! Не позволяйте вашему коду приводить к зависанию компьютера, если вы не предусмотрели представление старым браузерам альтернативных страниц.

3. **Вставляйте комментарии.** Если вы никогда не пытались разобраться в части кода после того, как кто-нибудь «покопался» в нем, то вы не поймете важности наличия комментариев. Если же вы сталкивались с кодом без комментариев, то данная рекомендация не требует разъяснения. Комментарии не должны быть объемными, достаточно просто включать в строку небольшую фразу: «Эта функция форматирует ваш жесткий диск».
4. **Придерживайтесь стандартов.** При создании HTML-кода не отклоняйтесь от стандартов. Всякий раз, когда вы делаете то, что работает только на одном браузере или поддерживается только одной версией языка сценария, вы ограничиваете универсальность вашего сайта или приложения. Правильность вашего HTML-кода можно проверить на сайте Интернет-консорциума.
5. **Проверяйте работу сайта.** Очевидное и часто игнорируемое правило определяет разницу между сайтом, который разочаровывает и отталкивает клиентов, и сайтом, на который приятно вернуться. CGI и JavaScript/JScript-сценарии должны корректно выполняться всеми основными браузерами, начиная с версии 4.x.

Добавление функций электронного магазина

Электронная коммерция является перспективным направлением использования Интернета. Электронные магазины в Интернете создаются быстрее, чем в реальном мире. На самом деле большинство новых «домашних» предприятий представлены только в Интернете, поскольку это самый эффективный и требующий наименьших затрат способ реализации товаров, ориентируемых на огромное число потенциальных покупателей. Существует огромное количество программного обеспечения, которое позволяет оформлять «витрину» магазина в сети. Некоторые из них позволяют создавать «торговые ряды», а другие — только отдельные магазины. Одни позволят оперативно оформить покупку (online), а для других требуются оформление заказа в автономном релсиме и отправка его на сайт магазина. Ассортимент некоторых электронных магазинов ограничен сотнями товаров, тогда как другие представляют широчайший выбор товара, исчисляемый несколькими тысячами наименований. Вы можете найти бесплатные версии электронных магазинов, а можете затратить на их создание тысячи долларов. Вам просто необходимо найти то, что вас устроит.

В табл. 14.1 представлено краткое сравнение двух типов электронных магазинов.

Таблица 14.1. Сравнение программного обеспечения электронных магазинов

	Extropia's Web Store	**Miva Merchant**
Интерфейс администратора	Отчет по электронной почте или самостоятельное чтение CGI и HTML-файлов	Web-интерфейс
Число продуктов	До 500 (число наименований товаров ограничено базой данных, представленной в виде текстового файла. Приемлемая скорость поиска в ней определяется мощностью сервера)	Не ограничено

	Extropia's Web Store	Miva Merchant
Стоимость лицензии	Бесплатно	$500 за «торговый ряд» из одного магазина и по $100 за каждый дополнительный магазин
Возможность настройки	При помощи HTML и CGI. Но требуются специальные знания	С помощью HTML-форм с помощью сценариев. Требуется знание HTML в пределах команд форматирования страниц
Система безопасности	Поддержка SSL	Поддержка SSL
web-адрес	http://www.extropia.com	http://www.miva.com

Эти примеры позволяют вам заглянуть в мир электронных магазинов. Работают они практически одинаково. Вы можете импортировать базу данных, созданную в Extropia Web Store, в Miva Merchant. Поскольку Extropia Web Store базируется на открытых CGI-сценариях, написанных на языке Perl, вы можете переделать этот магазин по своему усмотрению. Можно добавлять и удалять наименования товаров без всякого обучения и не зная языка. Miva Merchant также можно модифицировать, но для этого необходимо изучить язык HTMLScript компании Miva. Магазин также можно изменить, добавив новые функции и программные модули. Новые расширения Miva Merchant можно приобрести на их web-сайте (http://miva.com) или сайте их адептов — компании StarBase 21 (http://www.starbase21.com).

Запись и чтение cookies

Cookies — это один из способов проследить, кто посещает ваш web-сайт. При посещении сайта производится запоминание всех личных настроек, произведенных на сайте каждым посетителем. Например, если вы когда-нибудь посещали страницы Excite (http://www.excite.com) и производили настройки своей учетной записи, то вы могли увидеть этот процесс в действии.

Самый простой способ настроить cookie при помощи JavaScript и JScript заключается в использовании следующей команды:

```
document.cookie = "cookieName=cookieValue";
```

Свойства cookie-документа выражены строчными значениями, содержащими наименование и значение cookie. Например, содержание свойства document.cookie будет:

```
CookieName= George; expires= May 5 2005
```

После того как вы установили это значение cookie, вам необходимо найти возможность прочитать его при следующем визите посетителя. Проведение анализа строки с целью нахождения в ней точного наименования cookie, хранимого в переменной name, представлено в примере:

```
Var name = "cookieName"
  var prefix = name + "=";
  var begin = document.cookie.indexOf("; " + prefix);
  if (begin == -1) {
    begin = document.cookie.indexOf(prefix);
    if (begin != 0) return null;
```

```
} else
  begin += 2;
var end = document.cookie.indexOf(";", begin);
if (end == -1)
  end = document.cookie.length;
return unescape(document.cookie.substring(begin + prefix.length, end));
```

Печать

Одно из самых неприятных впечатлений посетителя может быть связано с обрывом половины изображения или даже части строк текста при печати страниц на некоторых принтерах. Устранить эту проблему можно добавлением в HTML-документы специальных команд печати.

Сначала необходимо использовать атрибут MEDIA="print" в объекте <style> HTML-документа для назначения типа среды, к которой будут относиться новые CSS-атрибуты. Затем при помощи описателя класса настроить свойства page-break-before или page-break-after для всех элементов данного класса. Пример применения данного способа представлен ниже:

```
<HTML>
<HEAD>
<STYLE MEDIA="print">
<!--
page { page-break-before: always }
//-->
</STYLE>
</HEAD>
<BODY>
<H1>Это первая страница</H1>
<P CLASS = "page">Это текст на первой странице</P>
<H1>Это начало второй страницы</H2>
<H2 CLASS = "page">А теперь мы на третьей странице!</H2>
</BODY>
</HTML>
```

Эти команды относятся к рекомендациям CSS 2 и будут недоступны при использовании браузеров версии более старой, чем 4.0. Описатель MEDIA="print" объекта <style> и назначение class="page" используются только при отправке документа на принтер.

Уловки, позволяющие привлечь внимание посетителей

Постоянство посетителей относительно. TANSTAAFL (There ain't no such thing as a free lunch!) — такого выражения нет в словаре посетителя. Они — свободные птицы! Они все хотят получить бесплатную доставку. Они стремятся снизить цену. Но как владелец сайта вы должны объяснить им, что они могут получить бесплатно, а что нет.

Посетители также хотят узнать, кто является производителем, как ваш товар или услуга облегчат их жизнь, помогут сберечь время или деньги, на которые потом можно отправиться в двухнедельный отпуск в Европу.

Оформление текста

Даже если вы не изучали шесть (и более!) лет стили написания, способы воздействия, технологии, тонкости и психологию маркетинга, как этому учат в университетах, — вы, скорее всего, сами придете к нужному решению.

Вот несколько правил, которые помогут вам завладеть покупательской аудиторией лишь при помощи текстового содержания сайта:

- Помните о том, что вам необходимо продать. Это можно сделать, лишь убедившись, что конкретный человек, желательно вице-президент по продажам, занимается содержанием сайта.
- Используйте знания маркетологов при модификации содержания для того, чтобы перехватить потенциальных покупателей и удержать их на сайте.
- Участвуйте в составлении всех документов и переписке.
- Сообщите посетителю, чем ваш товар или услуга могут быть полезны именно для *него, а* не просто для чего он нужен.
- Найдите предлог, чтобы посетитель оставил вам свое имя и электронный адрес. Например, можно предложить ему ежемесячно или раз в квартал получать сведения о новинках вашего магазина или важные новости, касающиеся вашего предприятия. Если кто-нибудь подпишется на эту услугу, то они будут получать информацию, которая будет напоминать им о вас.

Вот еще несколько источников, в которых можно почерпнуть более подробную информацию о технологиях web-маркетинга, а также различные советы, подсказки и хитрости:

- **Wilson Internet Services** (http://www.wilsonweb.com). Этот сайт содержит множество оригинальных идей по web-рекламе, маркетингу и управлению трафиком web-сайта.
- **Sell It! Your Web Commerce Resource** (http://www.sellitontheweb.com). Этот сайт представляет продуманную систему рассылки новостей, которая позволит вам изучить множество неафишируемых подробностей новых маркетинговых подходов, которые уже работают или еще не реализованы на web-сайтах.
- **Click-Z** (http://www.clickz.com). Этот сайт ежедневно посылает большое информационное письмо с советами.
- **I-Sales Digest** (http://www.audettemedia.com/i-sales/). Это список подписчиков форума по передовым технологиям маркетинга в сети.
- **Search Engine Watch** (http://www.searchenginewatch.com). Этот сайт содержит множество самой свежей информации по способам сохранения высокого положения и поиску аналогичных средств.
- **Internet World** (http://www.iw.com/digest.html). Значительная часть сайта посвящена разработке приложений электронной коммерции и их оптимальному использованию.
- **C-Net** (http://www.cnet.com). Данный сайт содержит последние новости по всем технологиям, а также информацию по их применению в конкретных условиях.
- **eRetail.net** (http://www.eRetail.net). Содержит информацию об электронной коммерции в Интернете.

- **Jacob** Neilson's Alert Box (http://www.useit.com/alertbox). Профессионал по практическому применению обеспечивает рассылку новостей, в которых представлены «все углы и закоулки» создания и использования привлекательных web-сайтов.

Не связывайтесь с фреймами

Наилучший способ рассердить посетителя — позволить ему щелкнуть на ссылке и открыть эту ссылку внутри одного из фреймов вашего сайта. Вы не хотите, чтобы посетитель покинул вас, и открываете ссылку «на вашем сайте». Этого можно избежать, лишь добавив на каждой странице небольшой сценарий на JavaScript, представленный ниже:

```
if (self != top)
top.location = self.location
```

Этот сценарий следит за тем, чтобы текущая страница вашего сайта загружалась в оригинальное окно (top), а не во фрейм. Этот сценарий необходимо привязать к событию onload тела документа или с помощью элемента <script> поместить его в заголовок документа <head>.

Старайтесь не использовать кнопку Submit

Вам приятно искать и нажимать мышью кнопку Submit после заполнения полей формы? Данное неудобство можно обойти за счет использования самого факта выбора значений для запуска сценария.

В приведенном ниже HTML-документе представлена форма с выражением <select> <option>, предлагающая загрузить различные web-сайты. Сценарий использует выбранное в нем значение в функции loadDocument для определения ответвления, подставляемого в выражение case. Свойство selectedIndex используется для определения значения индекса (0, 1, 2...) каждого выделения и затем сравнивает его с идентификаторами выражения case. В этом выражении для автоматической загрузки указанной страницы используется простая команда: window.location.href.

```
<html>
<head>
<script language="JavaScript" TYPE="text/javascript">
<!-- функция loadDocument
function loadDocument(site) {
  switch (site) {
    case 0: return
    break;
    case 1: window.location.href =
         "http://www.webravin.com.com"
    break
    case 2: window.location.href = "http://www.catsback.com"
    break
  }
}
// -->
</script>
```

```
</head>
<body>
  <form name = "selector">
    <select name="titles" size="1" onChange =
          "loadDocument(this.selectedIndex)">
      <option>Load Site...</option>
      <option>WebRavin</option>
      <option>Cats Back</option>
    </select>
  </form>
</body>
</html>
```

Размещайте текст в нескольких колонках

Можно выиграть немного свободного пространства для размещения текста и изображений, сосредоточив важную информацию перед посетителем. За счет применения колонок ее можно упорядочить и представить в более компактном виде. Это осуществляется при помощи табличного форматирования групп колонок (элементы <colgroup> и <col>) без создания строк:

```
<table cols = 3 cellspacing = 2>
  <colgroup span = 3>
    <col width = "30%">
    <col width = "30%">
    <col width = "30%">
  </colgroup>
  <td>Column 1 text...</td>
  <td>Column 2 text...</td>
  <td>Column 3 text...</td>
</table>
```

ПРИМЕЧАНИЕ

Большое количество колонок усложнит восприятие вашей страницы. Если текст целиком помещается на экране, лучше разместить его в двух колонках, а в третьей колонке поместить заголовки и изображения.

Изменить число колонок, границы, наполнения, интервалы и выравнивания можно с помощью тех же свойств и атрибутов, которые использовались во всех версиях ваших документов.

Что ждет нас в будущем

В будущем открывается много потенциальных возможностей. Существует большое разнообразие характеристик для разработки web-страниц и представления информации. К ним относятся новые разработки HTML, каскадных таблиц стилей, объектной модели документа и расширяемого языка разметки (Extensible Markup Language, XML).

Помните, что, если вы ведете разработки с учетом будущих или даже сегодняшних стандартов, вы не сможете реализовать все потенциальные возможности, поскольку программное обеспечение просто не способно угнаться за стандартами.

Новшества в HTML

Интернет-консорциум (http://www.w3c.org) непрерывно работает над развитием и увеличением возможностей HTML. Эти улучшения касаются открытости и модульности HTML, которые за счет конкретных соглашений и стандартов XML позволяют использовать его в различных типах документов. XHTML - это просто очередная попытка объединить модульность и возможность расширения XML- и HTML-конструкциями, с которыми вы уже знакомы.

Каскадные таблицы стилей

Каскадные таблицы стилей уже претерпели две ревизии, и к выходу в свет этой книги готовится третья. При изменениях в HTML и в его дочерних языках XML развитие CSS и других языков таблиц стилей позволит быстро создавать и изменять документы. Новые разработки CSS позволят осуществлять интерактивное форматирование элементов управления, изменять цветовые схемы, пространства имен, размещение информации в нескольких колонках. Кроме того, они позволят осуществить дополнительную поддержку среды и управление печатью, а также улучшение системы поддержки национальных шрифтов и их расположения. Расширение CSS уже практически готово, но полностью поддерживающее его программное обеспечение появится не ранее чем через год после опубликования рекомендаций. Точно так же, как современные пятые и шестые версии браузеров обеспечили поддержку CSS 2 лишь через год после их опубликования.

Объектная модель документа

После реализации модели DOM Level 1 предстоящий выпуск DOM Level 2 позволит охватить не только HTML и XML. Модель DOM Level 2 включает управление CSS и вообще всеми таблицами стиля. Она облегчит включение событий документа в HTML- и XML-документы. Но необходимо учитывать, что программная поддержка появится также не ранее чем через год после официального опубликования.

Разработка расширяемого языка разметки

XML является одной из мощных и быстроразвивающихся спецификаций. Помимо исходной спецификации XML существуют новые спецификации и прорабатываемые черновики: XPointer, XLink, XML Schemas, XML Inclusions, XML Base и XML Signatures. Текущая разработка языка XML Query обеспечит использование интерактивных запросов в XML-документах. При добавлении встроенного языка запросов вы можете с помощью XML-документов показывать соответствующую запросу информацию в желаемом виде.

Будущее языка XML может стать будущим Интернет-систем обмена документами.

Приложение 1. Элементы HTML, поддерживаемые новыми браузерами

Профессиональный вид web-страниц можно обеспечить с помощью множества HTML-элементов и атрибутов, которые позволяют осуществлять форматирование документа. Представленная ниже таблица позволяет разобраться, какие элементы HTML поддерживаются каждым браузером.

ПРИМЕЧАНИЕ

Часть элементов не поддерживаются некоторыми браузерами ввиду того, что в них осуществлена поддержка собственных аналогичных элементов.

Таблица П1.1. Элементы и атрибуты

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
A	accesskey, class, coords, dir, href, hreflang, id, lang, name, onblur, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, rel, rev, shape, style, tabindex, target, title, type	Да	Да	Да	Да	Да	Да	Да	Да
ABBR	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
ACRONYM	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	*И/в*	Да	Да	Да	Н/в	Да	Да

продолжение

Таблица П1.1 *(продолжение)*

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
ADDRESS	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
APPLET	align, alt, archive, class, code, codebase, height, hspace, id, name, object, style, title, vspace, width	Да	Да	Да	Да	Да	Да	Да	Да
AREA	accesskey, alt, class, coords, dir, href, id, lang, nohref, onblur, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, shape, style, tabindex, target, title	Да	Да	Да	Да	Да	Да	Да	Да
Ош.	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
BASE	href, target	Да	Да	Да	Да	Да	Да	Да	Да
BASEFONT	color, face, id, size	Нет	Да	Да	Да	Да	Да	Да	Да
BDO	class, dir, id, lang, style, title	Да	Да	Да	Да	Да	Да	Да	Да
BIG	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
BLINK	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Да	Нет	Нет	Да	Да	Нет	Нет
BLOCKQUOTE	cite, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
BODY	align, background, bgcolor, class, dir, id, lang, link, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onload, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onunload, sytle, text, title, vlink	Да	Да	Да	Да	Да	Да	Да	Да
BR	class, clear, id, style, title	Да	Да	Да	Да	Да	Да	Да	Да
BUTTON	accesskey, class, dir, disabled, id, lang, name, onblur, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, tabindex, title, type, value	Нет	Да	Да	Да	Нет	Да	Да	Да
CAPTION	align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
CENTER	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
CITE	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
CODE	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
COL	align, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, span, style, title, valign width	Да	Н/в	Да	Да	Да	Н/в	Да	Да

продолжение

Таблица П1.1 *(продолжение)*

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
COLGROUP	align, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, span, style, title, valign, width	Да	Н/В	Да	Да	Да	Н/в	Да	Да
DD	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
DEL	cite, class, datetime, dir, id, lang,onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Нет	Н/в	Да	Да	Нет	Н/в	Да	Да
DFN	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
DIR	class, compact, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
DIV	align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Да	Да	Да	Да	Да	Да	Да
DL	class, compact, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
DT	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
EM	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/В	Да	Да	Да	Н/в	Да	Да
FIELDSET	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Нет	Н/в	Да	Да	Нет	Н/в	Да	Да
FONT (сокращенно)	class, color, dir, face, id, lang, size, style, title	Да	Да	Да	Да	Да	Да	Да	Да
FORM	accept-charset, action, class, dir, enctype, id, lang, method, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onreset, onsubmit, style, target, title	Да	Да	Да	Да	Да	Да	Да	Да
FRAME	class, frameborder, id, longdesc, marginheight, marginwidth, name, noresize, scrolling, src, style, title	Да	Да	Да	Да	Да	Да	Да	Да
FRAMESET	class, cols, id, onload, onunload, rows, style, title	Да	Да	Да	Да	Да	Да	Да	Да
H1 – H6	align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
HEAD	dir, lang, profile	Да	Да	Да	Да	Да	Да	Да	Да
HR	align, class, id, noshade, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, size, style, title, width	Да	Н/в	Да	Да	Да	Н/в	Да	Да
HTML	dir, lang, version	Да	Да	Да	Да	Да	Да	Да	Да
I	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да

продолжение

Таблица П1.1 *(продолжение)*

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IES
IFRAME	align, dass, frameborder, height, id, longdesc, marginheight, marginwidth, name, scrolling, src, style, title, width	Нет	Ош.	Да	Ош.	Нет	Ош.	Да	Да
IMG	align, alt, border, class, dir, height, hspace, id, ismap, lang, longdesc, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, src, style, title, usemap, vspace, width	Да	Н/в	Да	Да	Да	Н/в	Да	Да
INPUT	accept, accesskey, align, alt, checked, class, dir, disabled, id, lang, maxlength, name, onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onselect, prompt, size, src, style, tabindex, title, type usemap, value	Да	Да	Да	Да	Да	Да	Да	Да
INS	cite, class, datetime, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Нет	Н/в	Да	Да	Нет	Н/в	Да	Да
ISINDEX	class, dir, id, lang, readonly, style, title	Да	Да	Да	Да	Да	Да	Да	Да
KBD	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
LABEL	accesskey, class, dir, for, id, lang, onblur, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Да	Да	Да	Да	Да	Да	Да
LAYER	above, background, below, bgcolor, clip, height, left, name, src, top, visibility, width, z-index	Да	Да	Нет	Нет	Да	Да	Нет	Нет

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
LEGEND	accesskey, align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Нет	Ош.	Нет	Да	Нет	Ош.	Нет	Да
LI	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, type, value	Да	Н/в	Да	Да	Да	Н/в	Да	Да
LINK	class, dir, area, hreflang, id, lang, media, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, rev, rel, style, tqarget, title, type	Да	Да	Да	Да	Да	Да	Да	Да
MAP	class, dir, id, lang, name, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Да	Да	Да	Да	Да	Да	Да
MARQUEE	behavior, bgcolor, class, datafld, dataformatas, datasrc, direction, height, hspace, id, lang, language, loop, onafterupdate, onblur, onbounce, onclick, ondblclick, ondragstart, onfinish, onfocus, onhelp, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onresize, onrowenter, onrowexit, onselectstart, onstart, scrollamount, scrolldelay, style, title, truespeed, vspace, width	Нет	Ош.	Да	Да	Нет	Ош.	Да	Ош.
MENU	class, compact, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
META	content, dir, http-equiv, lang, name, scheme	Да	Да	Да	Да	Да	Да	Да	Да

продолжение

Таблица П1.1 *(продолжение)*

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
NOFRAMES	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
NOSCRIPT	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Да	Да	Да	Да	Да	Да	Да
OBJECT	align, archive, border, class, classid, codebase, codetYe, data, declare, dir, height, hspace, id, lang, name, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, standby, style, tabindex, title, type, usemap, vspace, width	Да	Да	Да	Да	Да	Да	Да	Да
OL	class, compact, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, start, style, title, type	Да	Н/в	Да	Да	Да	Н/в	Да	Да
OPTGROUP	class, dir, disabled, id, label, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Да	Ош.	Да	Да	Да	Да	Да
OPTION	class, dir, disabled, id, label, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, selected, style, title, value	Да	Да	Да	Да	Да	Да	Да	Да
P	align, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
PARAM	id, name, type, value, valuetype,	Да	Да	Да	Да	Да	Да	Да	Да
PRE	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, width	Да	Н/в	Да	Да	Да	Н/в	Да	Да
Q	cite, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
S	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
SAMP	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
SCRIPT	defer, language, src, type	Да	Да	Да	Да	Да	Да	Да	Да
SELECT	class, dir, disabled, id, lang, multiple, name, onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, size, style, tabindex, title	Да	Да	Да	Да	Да	Да	Да	Да
SMALL	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
SPAN	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
STRIKE	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да

продолжение

Таблица П1.1 (*продолжение*)

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
STRONG	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
STYLE	dir, lang, media, title, type	Да	Да	Да	Да	Да	Да	Да	Да
SUB	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
SUP	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
TABLE	align, bgcolor, border, cellpadding, cellspacing, class, dir, frame, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, rules, style, summary, title, width	Да	Н/в	Да	Да	Да	Н/в	Да	Да
TBODY	align, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, valign	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
TD	abbr, align, axis, bgcolor, char, charoff, class, colspan, dir, headers, height, id, lang, nowrap, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, scope, style, title, valign, width	Да	Да	Да	Да	Да	Да	Да	Да
TEXTAREA	acesskey, class, cols, dir, disabled, id, lang, name, onblur, onchange, onclick, ondblclick, onfocus, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onselect, readonly, rows, style, tabindex, title	Да	Да	Да	Да	Да	Да	Да	Да

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
TFOOT	align, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, valign	Нет	*И/в*	Да	Да	Нет	Н/в	Да	Да
TH	abbr, align, axis, bgcolor, char, charoff, class, colspan, dir, headers, height, id, lang, nowrap, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, scope, style, title, valign, width	Да	Н/в	Да	Да	Да	Н/в	Да	Да
THEAD	align, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, valign	Нет	Н/в	Да	Да	Нет	Н/в	Да	Да
TITLE	dir, lang	Да	Да	Да	Да	Да	Да	Да	Да
TR	align, bgcolor, char, charoff, class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, valign	Да	Н/в	Да	Да	Да	Н/в	Да	Да
TT	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
U	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да
UL	class, compact, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title, type	Да	Н/в	Да	Да	Да	Н/в	Да	Да

продолжение

Таблица П1.1 (*продолжение*)

Элемент	Атрибуты	Win 95/98/NT				Macintosh			
		Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
VAR	class, dir, id, lang, onclick, ondblclick, onkeydown, onkeypress, onkeyup, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, style, title	Да	Н/в	Да	Да	Да	Н/в	Да	Да

Сокращения в таблице:

Ош. — имеются ошибки в реализации, приводящие в отдельных случаях к непредвиденным результатам.

Н/в — поддерживаются не все атрибуты и события.

Приложение 2. Совместимость каскадных таблиц стилей

Каскадные таблицы стилей (Cascading Style Sheets, CSS) позволяют осуществлять форматирование содержания web-страниц таким образом, что оно будут выглядеть одинаково или почти одинаково при их просмотре большинством web-браузеров. Приведенные в данном приложении таблицы содержат сведения о поддержке различными браузерами свойств каскадных таблиц стиля, регламентированных спецификацией CSS 2 (Cascading Style Sheets Level 2). Таблицы упорядочены по CSS-категориям, что позволит вам быстрее находить требуемые сведения.

Применяемые сокращения:

Ош. — существуют некоторые проблемы в реализации, и поэтому они не всегда работают при определенных условиях.

П — планируется реализовать.

Операционная система	Применяемое в ОС сокращение	Полное наименование
Windows 95, Windows 98, Windows NT	Nav4	Netscape Navigator 4.x
	Nav6	Netscape Navigator 6
	IE4	Internet Explorer 4
	IE5	Internet Explorer 5.x
Macintosh	Nav4	Netscape Navigator 4.x
	Nav6	Netscape Navigator 6
	IE4	Internet Explorer 4
	IE5	Internet Explorer 5

CSS позволяет внедрять стили в web-страницы различными способами. В табл. П2.1 представлены различные способы привязки стилей к HTML-элементам.

Таблица П2.1. Основные концепции и @-правила

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Комментарии: /*комментарий */	Да	Да	Да	Да	Да	Да	Да	Да
Возможность включения в HTML-документы								
LINK	Да	Да	Да	Да	Да	Да	Да	Да
<STYLE...> </STYLE>	Да	Да	Да	Да	Да	Да	Да	Да
@import	Нет	Да	Ош.	Да	Нет	Да	Да	Да
<xElement style= "declaration;">	Да	Да	Да	Да	Да	Да	Да	Да
Селектор группирования: a, b, c {объявление;}	Да	Да	Да	Да	Да	Да	Да	Да
Контекстуальный селектор: a b c { объявление; }	Да	Да	Да	Да	Ош.	Да	Да	Да
Наследование	Ош.	Да	Да	Да	Ош.	Да	Да	Да
@-правила:								
@charset	Нет	П	Нет	Нет	Нет	П	Нет	Да
@fontface	Да	Да	Ош.	Да	Да	Да	Ош.	Да
@import	Да	Да	Да	Да	Да	Да	Да	Да
@media	Нет	Да	Да	Да	Нет	Да	Да	Да
@page	Нет	П	Нет	Да	Нет	П	Нет	Да
селектор .class	Да	Да	Ош.	Да	Да	Да	Да	Да
селектор #id	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
!important	Нет	Да	Да	Да	Нет	Да	Нет	Да
Порядок каскадного включения								
Сортировка «по весу» (Weight sorting)	Ош.	Да	Да	Да	Ош.	Да	Да	Да
Сортировка по происхождению (Origin sorting)	Ош.	Да	Да	Да	Ош.	Да	Да	Да
Сортировка по специфичности (Specificity sorting)	Ош.	Да	Да	Да	Ош.	Да	Да	Да
Сортировка по порядку (Order sorting)	Ош.	Да	Да	Да	Ош.	Да	Да	Да

В спецификации CSS 2 существуют различные псевдоэлементы и классы, которые функционируют как обычные элементы и классы, но.не отражены в DOM. Они представлены в табл. П2.2.

Таблица П2.2. Псевдоэлементы и псевдоклассы

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
:active	Нет	Да	Да	Да	Нет	Да	Да	Да
:after	Нет	Да	Нет	Нет	Нет	Да	Нет	Да
:before	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
:first	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
:first-child	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
:first-line	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
:first-letter	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
:focus	Нет	Да	Нет	Нет	Нет	Да	Нет	
:hover	Нет	Да	Да	Да	Нет	Да	Да	Да
:lang	Нет	Да	Нет	Да	Нет	Да	Нет	
:left	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
:link	Да	Да	Да	Да	Да	Да	Да	Да
:right	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
:visited	Нет	Да	Да	Да	Нет	Да	Да	Да

К основным CSS-свойствам относятся свойства, управляющие внешним видом текста на страницах. В табл. П2.3 представлен полный список CSS-свойств шрифтов и текста.

Таблица П2.3. Свойства шрифтов и текста

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Шрифт (font)								
<font-family>	Ош.	Да	Да	Да	Да	Да	Да	Да
<line-height>	Ош.	Да	Да	Да	Ош.	Да	Да	Да
<font-size>	Да	Да	Да	Да	Да	Да	Да	Да
<font-style>	Ош.	Да	Да	Да	Да	Да	Да	Да
<font-variant>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
<font-weight>	Ош.	Да	Да	Да	Да	Да	Да	Да
caption	Нет	П	Да	Да	Нет	П	Да	Да
icon	Нет	П	Да	Да	Нет	П	Да	Да
menu	Нет	П	Да	Да	Нет	П	Да	Да
message-box	Нет	П	Да	Да	Нет	П	Да	Да
small-caption	Нет	П	Да	Да	Нет	П	Да	Да
status-bar	Нет	П	Да	Да	Нет	П	Да	Да
Семейство шрифтов (font-family)								
<family-name>	Да	Да	Да	Да	Да	Да	Да	Да
<generic-family>	Ош.	Да	Да	Да	Да	Да	Да	Да
... serif	Да	Да	Да	Да	Да	Да	Да	Да

продолжение

Таблица П2.3 *(продолжение)*

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
... sans-serif	Да	Да	Да	Да	Да	Да	Да	Да
... cursive	Нет	Да	Да	Да	Да	Да	Да	Да
... fantasy	Нет	Да	Да	Да	Да	Да	Да	Да
... monospace	Да	Да	Да	Да	Да	Да	Да	Да
Размер шрифта (font-size)								
Абсолютный размер (<absolute-size>)								
минимальное значение-максимальное значение	Да	Да	Ош.	Да	Да	Да	Да	Да
Относительный размер								
больше и меньше	Да	Да	Да	Да	Да	Да	Да	Да
< length >	Да	Да	Да	Да	Да	Да	Да	Да
< percentage >	Да	Да	Да	Да	Да	Да	Да	Да
Корректировка размера шрифта (<font-size-adjust>)								
<number>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
none	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
«Растягивание» шрифта (font-stretch)								
normal	Да	Да	Да	Да	Да	Да	Да	Да
wider	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
narrower	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
ultra-condensed	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
extra-condensed	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
condensed	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
semi-condensed	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
semi-expanded	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
expanded	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
extra-expanded	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
ultra-expanded	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
Стиль шрифта (font-style)								
normal	Да	Да	Да	Да	Да	Да	Да	Да
italic	Да	Да	Да	Да	Да	Да	Да	Да
oblique	Нет	Да	Да	Да	Нет	Да	Да	Да
Разновидности шрифтов (font-variant)								
normal	Нет	Да	Да	Да	Нет	Да	Да	Да
small-caps	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
«Полнота» шрифта (font-weight)								
normal	Да	Да	Да	Да	Да	Да	Да	Да
bold	Да	Да	Да	Да	Да	Да	Да	Да

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
bolder	Да	Да	Да	Да	Нет	Да	Да	Да
lighter	Нет	Да	Да	Да	Нет	Да	Да	Да
Интервал между буквами (letter-spacing)								
100-900	Да	Да	Да	Да	Да	Да	Да	Да
normal	Нет	Да	Нет	Да	Нет	Да	Да	Да
<length>	Нет	Да	Нет	Да	Нет	Да	Да	Да
Высота строки (line-height)								
normal	Да	Да	Да	Да	Да	Да	Да	Да
<number>	Ош.	Да	Да	Да	Ош.	Да	Да	Да
<length>	Ош.	Да	Да	Да	Ош.	Да	Да	Да
<percentage>	Ош.	Да	Да	Да	Ош.	Да	Да	Да
Выравнивание текста (text-align)								
left	Да	Да	Да	Да	Да	Да	Да	Да
right	Да	Да	Да	Да	Да	Да	Да	Да
center	Да	Да	Да	Да	Да	Да	Да	Да
jsutify	Ош.	Да	Да	Да	Ош.	Да	Нет	Ош.
<строка>	Ош.	Да	Нет	Ош.	Ош.	Да	Нет	Ош.
Дополнительные элементы оформления (text-decoration)								
none	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
underline	Ош.	Да	Ош.	Ош.	Ош.	Да	Ош.	Ош.
overline	Нет	Да	Да	Да	Нет	Да	Да	Да
line-through	Да	Да	Да	Да	Да	Да	Да	Да
blink	Да	Да	Нет	Ош.	Да	Да	Нет	Ош.
«Отрывная» часть текста (text-indent)								
<length>	Да	Да	Да	Да	Да	Да	Да	Да
<percentage>	Да	Да	Да	Да	Да	Да	Да	Да
Тень (text-shadow)								
none	Да	Да	Да	Да	Да	Да	Да	Да
<length>	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
<color>	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
Преобразование текста (text-transform)								
capitalize	Да	Да	Да	Да	Да	Да	Да	Да
uppercase	Да	Да	Да	Да	Да	Да	Да	Да
lowercase	Да	Да	Да	Да	Да	Да	Да	Да
none	Да	Да	Да	Да	Да	Да	Да	Да
Вертикальное выравнивание (vertical-align)								
baseline	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
sub	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да

продолжение

Таблица П2.3*(продолжение)*

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
super	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
top	Нет	Ош.	Нет	Да	Нет	Ош.	Да	Да
text-top	Нет	Ош.	Нет	Да	Нет	Ош.	Да	Да
middle	Нет	Ош.	Ош.	Да	Нет	Ош.	Да	Да
bottom	Нет	Ош.	Нет	Да	Нет	Ош.	Ош.	Да
text-bottom	Нет	Ош.	Нет	Да	Нет	Ош.	Ош.	Да
< length >	Нет	Да	Нет	Да	Нет	Да	Ош.	Да
<percentage>	Нет	Ош.	Нет	Да	Нет	Ош.	Ош.	Да
Интервал между словами (word-spacing)								
<length>	Нет	Да	Нет	Да	Нет	Да	Да	Да
normal	Нет	Да	Нет	Да	Нет	Да	Да	Да

Наиболее часто используемыми также являются свойства управления цветом фона. В табл. П2.4 представлены все свойства, которые позволяют управлять фоном документов.

Таблица П2.4. Свойства фона и цвета

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Фон (background)								
< background-attachment>	Нет	Да	Да	Да	Нет	Да	Да	Да
<background-color>	Ош.	Да	Да	Да	Ош.	Да	Да	Да
<background-image>	Ош.	Да	Да	Да	Ош.	Да	Да	Да
< background-position >	Нет	Да	Да	Да	Нет	Да	Да	Да
<background-repeat>	Ош.	Да	Ош.	Да	Ош.	Да	Да	Да
Положение фона (background-attachment)								
fixed	Нет	Да	Да	Да	Нет	Да	Да	Да
scroll	Нет	Да	Да	Да	Нет	Да	Да	Да
Цвет фона (background-color)								
<colorname>	Да	Да	Да	Да	Да	Да	Да	Да
<RGB Value>	Да	Да	Да	Да	Да	Да	Да	Да
transparent	Ош.	Да	Да	Да	Ош.	Да	Да	Да
Фоновое изображение (background-image)								
<URL>	Да	Да	Да	Да	Да	Да	Да	Да
none	Да	Да	Да	Да	Да	Да	Да	Да
Размещение фона (background-position)								
<percentage>	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
< length >	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
bottom	Нет	Да	Да	Да	Нет	Да	Да	Да
center	Нет	Да	Да	Да	Нет	Да	Да	Да
left	Нет	Да	Да	Да	Нет	Да	Да	Да
right	Нет	Да	Да	Да	Нет	Да	Да	Да
top	Нет	Да	Да	Да	Нет	Да	Да	Да
Повторное использование изображения при заполнении фона (background-repeat)								
no-repeat	Да	Да	Да	Да	Да	Да	Да	Да
repeat	Да	Да	Да	Да	Да	Да	Да	Да
repeat-x	Ош.	Да	Да	Да	Ош.	Да	Да	Да
repeat-y	Ош.	Да	Да	Да	Ош.	Да	Да	Да
Цвет (color)								
<colorname>	Да	Да	Да	Да	Да	Да	Да	Да
<RGB-Value>	Да	Да	Да	Да	Да	Да	Да	Да
transparent	Ош.	Да	Да	Да	Ош.	Да	Да	Да

При использовании CSS 2 возможно создание содержания «на лету». В табл. П2.5 представлен список всех свойств, поддерживающих генерацию содержания и манипуляции с курсором.

Таблица П2.5. Генерация содержания и манипуляции курсором

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Содержание (content)								
<string>	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
<URL>	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
<counter>	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
attr(x)	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
close-quote	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
open-quote	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
no-close-quote	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
no-open-quote	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
Увеличение значений счетчика (counter-increment)								
< identifier>	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
<integer>	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
none	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
Сброс счетчика (counter-reset)								
<identifier>	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
<integer>	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
none	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет

продолжение

Таблица П2.5 *(продолжение)*

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Курсор (cursor)								
<URI>	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
auto	Да	Да	Да	Да	Да	Да	Да	Да
crosshair	Нет	Да	Нет	Да	Нет	Да	Нет	Да
default	Нет	Да	Нет	Да	Нет	Да	Нет	Да
e-resize	Нет	Да	Нет	Да	Нет	Да	Нет	Да
help	Нет	Да	Нет	Да	Нет	Да	Нет	Да
n-resize	Нет	Да	Нет	Да	Нет	Да	Нет	Да
ne-resize	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
nw-resize	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
pointer	Нет	Да	Нет	Да	Нет	Да	Нет	Да
s-resize	Нет	Да	Нет	Да	Нет	Да	Нет	Да
se-resize	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
sw-resize	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
text	Нет	Да	Нет	Да	Нет	Да	Нет	Да
wait	Нет	Да	Нет	Да	Нет	Да	Нет	Да

Множество объектов, используемых для создания HTML-документов, представлены блоками (контейнерами). Зачастую возникает необходимость повлиять на свойства этих блоков. Все CSS-свойства, влияющие на внешний вид контейнеров, представлены в табл. П2.6.

Таблица П2.6. Свойства контейнеров

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Рамка (border)								
<border-color>	Ош.	Да	Да	Да	Ош.	Да	Да	Да
<border-style>	Ош.	Да	Да	Да	Ош.	Да	Да	Да
< border-width >	Да	Да	Да	Да	Да	Да	Да	Да
Разрушение рамки (border-collapse)								
collapse	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
separate	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
Цвет рамки (border-color)								
<colorname>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
<RGBvalue>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
border-spacing: <length>	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
Стиль рамки (border-style)								
dashed	Нет	Да	Нет	Да	Нет	Да	Да	Да
dotted	Нет	Да	Нет	Да	Нет	Да	Да	Да

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
double	Да	Да	Да	Да	Да	Да	Да	Да
groove	Да	Да	Да	Да	Да	Да	Да	Да
inset	Да	Да	Да	Да	Да	Да	Да	Да
none	Да	Да	Да	Да	Да	Да	Да	Да
outset	Да	Да	Да	Да	Да	Да	Да	Да
ridge	Да	Да	Да	Да	Да	Да	Да	Да
solid	Да	Да	Да	Да	Да	Да	Да	Да
Верхняя граница рамки (border-top)								
<border-top-color>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
<border-top-style>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
<border-top-width>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
Правая граница рамки (border-right)								
<border-right-color>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
<border-right-style>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
<border-right-width>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
Левая граница рамки (border-left)								
<border-left-color>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
<border-left-style>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
<border-left-width>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
Нижняя граница рамки (border-bottom)								
<border-bottom-color>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да.
<border-bottom-style>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
<border-bottom-width>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
Цвет верхней границы рамки (border-top-color)								
<colorname>	Да	Да	Да	Да	Да	Да	Да	Да
<RGBvalue>	Да	Да	Да	Да	Да	Да	Да	Да
transparent	Да	Да	Да	Да	Да	Да	Да	Да
Цвет правой границы рамки (border-right-color)								
<colorname>	Да	Да	Да	Да	Да	Да	Да	Да
<RGBvalue>	Да	Да	Да	Да	Да	Да	Да	Да
transparent	Да	Да	Да	Да	Да	Да	Да	Да
Цвет левой границы рамки (border-left-color)								
<colorname>	Да	Да	Да	Да	Да	Да	Да	Да
<RGBvalue>	Да	Да	Да	Да	Да	Да	Да	Да
transparent	Да	Да	Да	Да	Да	Да	Да	Да
Цвет нижней границы рамки (border-bottom-color)								
<colorname>	Да	Да	Да	Да	Да	Да	Да	Да
<RGBvalue>	Да	Да	Да	Да	Да	Да	Да	Да

продолжение

Таблица П2.6 (*продолжение*)

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
transparent	Да	Да	Да	Да	Да	Да	Да	Да
border-top-style: <border-style>	Ош.	Да	Ош.	Ош.	Ош.	Да	Ош.	Ош.
border-right-style: < border-style>	Ош.	Да	Ош.	Ош.	Ош.	Да	Ош.	Ош.
border-left-style: <border-style>	Ош.	Да	Ош.	Ош.	Ош.	Да	Ош.	Ош.
border-bottom-style: <border-style>	Ош.	Да	Ош.	Ош.	Ош.	Да	Ош.	Ош.
Толщина верхней границы рамки (border-top-width)								
medium	Да	Да	Ош.	Да	Да	Да	Ош.	Да
thick	Да	Да	Ош.	Да	Да	Да	Ош.	Да
thin	Да	Да	Ош.	Да	Да	Да	Ош.	Да
<length>	Да	Да	Ош.	Да	Да	Да	Ош.	Да
Толщина правой границы рамки (border-right-width)								
medium	Да	Да	Ош.	Да	Да	Да	Ош.	Да
thick	Да	Да	Ош.	Да	Да	Да	Ош.	Да
thin	Да	Да	Ош.	Да	Да	Да	Ош.	Да
<length>	Да	Да	Ош.	Да	Да	Да	Ош.	Да
Толщина левой границы рамки (border-left-width)								
medium	Да	Да	Ош.	Да	Да	Да	Ош.	Да
thick	Да	Да	Ош.	Да	Да	Да	Ош.	Да
thin	Да	Да	Ош.	Да	Да	Да	Ош.	Да
<length>	Да	Да	Ош.	Да	Да	Да	Ош.	Да
Толщина нижней границы рамки (border-bottom-width)								
medium	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
thick	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
thin	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
<length>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
Толщина всех границ рамки (border-width)								
medium	Да	Да	Ош.	Да	Да	Да	Ош.	Да
thick	Да	Да	Ош.	Да	Да	Да	Ош.	Да
thin	Да	Да	Ош.	Да	Да	Да	Ош.	Да
<length>	Да	Да	Ош.	Да	Да	Да	Ош.	Да
Расстояние до нижнего края рамки (bottom)								
<length>	Нет	Да	Да	Да	Нет	Нет	Да	Да
<percentage>	Нет	Да	Нет	Ош.	Нет	Нет	Нет	Ош.
auto	Да	Да	Да	Да	Да	Да	Да	Да

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Очистка (clear)								
both	Да	Да	Да	Да	Да	Да	Да	Да
left	Ош.	Да	Ош.	Да	Ош.	Да	Да	
none	Да	Да	Да	Да	Да	Да	Да	Да
right	Ош.	Да	Ош.	Да	Ош.	Да	Да	Да
Вырезание (clip)								
<shape>	Ош.	Да	Ош.	Ош.	Ош.	Да	Ош.	Ош.
auto	Да	Да	Да	Да	Да	Да	Да	Да
Плавающая область (float)								
left	Ош.	Да	Ош.	Да	Ош.	Да	Да	Да
none	Да	Да	Да	Да	Да	Да	Да	Да
right	Ош.	Да	Ош.	Да	Ош.	Да	Да	Да
Высота (height)								
auto	Нет	Да	Да	Да	Нет	Да	Да	Да
<length>	Нет	Да	Да	Да	Нет	Да	Да	Да
<percentage>	Нет	Да	Да	Да	Нет	Да	Да	Да
Расстояние до левого края рамки (left)								
<length>	Да	Да	Да	Да	Да	Да	Да	Да
<percentage>	Да	Да	Да	Да	Да	Да	Да	Да
auto	Да	Да	Да	Да	Да	Да	Да	Да
Отступы: (margin-width)								
<length>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
<percentage>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
auto	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
Нижний отступ (margin-bottom)								
<length>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
<percentage>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
auto	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
Левый отступ (margin-left)								
<length>	Ош.	Да	Ош.	Да	Да	Да	Ош.	Да
<percentage>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
auto	Нет	Да	Нет	Да	Ош.	Да	Ош.	Да
Правый отступ (margin-right)								
<length>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
<percentage>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
auto	Нет	Да	Нет	Да	Нет	Да	Ош.	Да
Верхний отступ (margin-top)								
<length>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
<percentage>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
auto	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да

продолжение

Таблица П2.6 *(продолжение)*

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Отметки (marks)								
crop	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
cross	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
none	Да	Да	Да	Да	Да	Да	Да	Да
Максимальная высота (max-height)								
<length>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
<percentage>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
none								
Максимальная ширина (max-width)								
<length>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
<percentage>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
none	Нет	Да	Нет	Ош.	Нет	Да	Нет	Да
Минимальная высота (min-height)								
<length>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
<percentage>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
Минимальная ширина (min-height)								
<length>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
<percentage>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
orphans: <integer>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
Контур (outline)								
<outline-color>	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
<outline-style>	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
<outline-width>	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
Цвет контура (outline-color)								
<colorname>	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
<RGBvalue>	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
transparent	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
Стиль контура (outline-style)								
dashed	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
dotted	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
double	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
groove	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
inset	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
none	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
outset	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
ridge	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
solid	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Ширина контура (outline-width)								
medium	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
thick	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
thin	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
<length>	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
Переполнение (overflow)								
auto	Да	Да	Да	Да	Да	Да	Да	Да
hidden	Да	Да	Да	Да	Да	Да	Да	Да
scroll	Да	Да	Да	Да	Да	Да	Да	Да
visible	Да	Да	Да	Да	Да	Да	Да	Да
Заполнение: (margin-width)								
<length>	Ош.	Да	Ош.	Да	Ош.	Да	Ош,	Да
<percentage>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
auto	Да	Да	Да	Да	Да	Да	Да	Да
Заполнение внизу (padding-bottom)								
<length>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
<percentage>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
auto	Да	Да	Да	Да	Да	Да	Да	Да
Заполнение слева (padding-left)								
<length>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
<percentage>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
auto	Да	Да	Да	Да	Да	Да	Да	Да
Заполнение справа (padding-right)								
<length>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
<percentage>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
auto	Да	Да	Да	Да	Да	Да	Да	Да
Заполнение сверху (padding-top)								
<length>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
<percentage>	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
auto	Да	Да	Да	Да	Да	Да	Да	Да
Страница (page)								
<identifier>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
auto	Нет	Да	Нет	Да	Нет	Да	Нет	Да
Верхний колонтитул (page-break-after)								
auto	Нет	Да	Нет	Да	Нет	Да	Нет	Да
always	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
avoid	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
left	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
right	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.

продолжение

Таблица П2.6 *(продолжение)*

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Нижний колонтитул (page-break-before)								
auto	Нет	Да	Нет	Да	Нет	Да	Нет	Да
always	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
avoid	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
left	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
right	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
Расположение (position)								
static	Да	Да	Да	Да	Да	Да	Да	Да
relative	Да	Да	Да	Да	Да	Да	Да	Да
absolute	Да	Да	Да	Да	Да	Да	Да	Да
fixed	Да	Да	Да	Да	Да	Да	Да	Да
quotes: <string>	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
Расстояние от правого края (right)								
<length>	Нет	Да	Да	Да	Нет	Да	Да	Да
<percentage>	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
auto	Да	Да	Да	Да	Да	Да	Да	Да
Размер (size)								
<length>	Нет	Да	Да	Да	Нет	Да	Да	Да
auto	Да	Да	Да	Да	Да	Да	Да	Да
landscape	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
portrait	Нет	Ош.	Нет	Ош.	Нет	Ош.	Нет	Ош.
Расстояние от верхнего края (top)								
<length>	Да	Да	Да	Да	Да	Да	Да	Да
<percentage>	Да	Да	Да	Да	Да	Да	Да	Да
auto	Да	Да	Да	Да	Да	Да	Да	Да
Видимость (visibility)								
visible	Да	Да	Да	Да	Да	Да	Да	Да
hidden	Да	Да	Да	Да	Да	Да	Да	Да
collaps	Да	Да	Да	Да	Да	Да	Да	Да
widows: <integer>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
Ширина (width)								
<length>	Ош.	Да	Ош.	Да	Ош.	Да	Да	Да
<percentage>	Ош.	Да	Ош.	Да	Ош.	Да	Да	Да
auto	Ош.	Да	Ош.	Да	Ош.	Да	Да	Да
Z-индекс (z-index)								
<length>	Да	Да	Да	Да	Да	Да	Да	Да
none	Да	Да	Да	Да	Да	Да	Да	Да

При работе с HTML-документами возникают моменты, когда объект должен использовать дополнительные характеристики. Такое происходит при использовании свойств систематизации, представленных в табл. П2.7.

Таблица П2.7. Свойства систематизации

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Направление (direction)								
ltr	Да	Да	Да	Да	Да	Да	Да	Да
rtl	Да	Да	Да	Да	Да	Да	Да	Да
Показать (display)								
block	Ош.	Да	Да	Да	Ош.	Да	Ош.	Да
compact	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
inline	Нет	Да	Да	Да	Нет	Да	Да	Да
inline-table	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
list-item	Ош.	Ош.	Ош.	Да	Ош.	Ош.	Ош.	Да
marker	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
none	Да	Да	Да	Да	Да	Да	Да	Да
run-in	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
table	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
table-caption	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
table-cell	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
table-column	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
table-column-group	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
table-footer-group	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
table-header-group	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
table-row	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
table-row-group	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
Список стилей (list-style)								
<list-style-image>	Нет	Да	Да	Да	Нет	Да	Да	Да
<list-style-position>	Нет	Да	Ош.	Да	Нет	Да	Ош.	Да
<list-style-type>	Да	Да	Да	Да	Ош.	Да	Да	Да
list-style-image								
<URI>	Нет	Да	Да	Да	Нет	Да	Да	Да
none	Нет	Да	Да	Да	Нет	Да	Да	Да
list-style-position								
inside	Нет	Да	Да	Да	Нет	Да	Ош.	Да
outside	Нет	Да	Да	Да	Нет	Да	Да	Да
list-style-type								
disc	Да	Да	Да	Да	Да	Да	Да	Да
circle	Да	Да	Да	Да	Да	Да	Да	Да
square	Да	Да	Да	Да	Да	Да	Да	Да
decimal	Да	Да	Да	Да	Да	Да	Да	Да
decimal-leading-zero	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
lower-roman	Да	Да	Да	Да	Да	Да	Да	Да

продолжение

Таблица **П2.7** *(продолжение)*

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
upper-roman	Да	Да	Да	Да	Да	Да	Да	Да
lower-greek	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
lower-alpha	Да	Да	Да	Да	Да	Да	Да	Да
upper-alpha	Да	Да	Да	Да	Да	Да	Да	Да
lower-latin	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
upper-latin	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
hebrew	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
aremenian	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
georgian	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
cjk-ideographic	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
hiragana	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
katakana	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
hiragana-iroha	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет
none	Да	Да	Да	Да	Ош.	Да	Да	Да
Двунаправленный текст (Unicode-bidi)								
normal	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
embed	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
bidi-override	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
Разделитель (white-space)								
normal	Да	Да	Нет	Ош.	Да	Да	Нет	Ош.
pre	Да	Да	Нет	Ош.	Да	Да	Нет	Ош.
no-wrap	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.

В качестве основной структуры форматирования HTML-страниц используются таблицы. Их CSS-свойства представлены в табл. П2.8.

Таблица П2.8. Свойства таблиц

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Размещение заголовка (caption-side)								
bottom	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
left	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
right	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
top	Нет	Ош.	Нет	Нет	Нет	Ош.	Нет	Нет
Пустые ячейки (empty-cells)								
show	Нет	Да	Нет	Ош.	Нет	Да	Нет	Ош.
hide	Да	Да	Нет	Ош.	Да	Да	Нет	Ош.

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Расположение таблицы (table-layout)								
auto	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
fixed	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет

При использовании CSS вы можете обратить внимание на то, что каждое свойство имеет свой список значений или единиц измерения, применимых к нему. В таблице П2.9 приведен список всех единиц измерений, поддерживаемых как Netscape Navigator, так и Internet Explorer.

Таблица П2.9. Единицы измерения

Элемент	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
Расстояния (length)								
em (буква m - как единица измерения)	Да	Да	Да	Да	Да	Да	Да	Да
ex	Ош.	Да	Ош.	Да	Ош.	Да	Ош.	Да
Пикселы (px)	Да	Да	Да	Да	Да	Да	Да	Да
Дюймы (in)	Да	Да	Да	Да	Да	Да	Да	Да
Сантиметры (cm)	Да	Да	Да	Да	Да	Да	Да	Да
Миллиметры (mm)	Да	Да	Да	Да	Да	Да	Да	Да
Точки (pt)	Да	Да	Да	Да	Да	Да	Да	Да
pc	Да	Да	Да	Да	Да	Да	Да	Да
Процентное отношение <percentage>	Да	Да	Да	Да	Да	Да	Да	Да
Название цвета (ColorName)								
Список цветов HTML 4.0: Aqua, Black, Blue, Fuchsia, Gray, Green, Lime, Maroon, Navy, Olive, Purple, Red, Silver, Teal, White, Yellow	Ош.	Да	Да	Да	Ош.	Да	Да	Да
Расширение списка цветов, используемое Microsoft*	Нет	Да	Да	Да	Нет	Да	Да	Да
RGB-значение (RGB-Value)								
(RRR,GGG,BBB)	Да	Да	Да	Да	Да	Да	Да	Да
(R%, G%, Ош.%)	Да	Да	Да	Да	Да	Да	Да	Да
#xxx	Да	Да	Да	Да	Да	Да	Да	Да
#xxxxxx	Да	Да	Да	Да	Да	Да	Да	Да
URI's — <URI>	Ош.	Да	Да	Да	Ош.	Да	Да	Да

Примечание к таблице: Расширение списка цветов, используемое Microsoft

AliceBlue	Azure	BlanceDalmond
AntiqueWhite	Beige	BlueViolet
Aquamarine	Bisque	Brown

Burlywood
CadetBlue
Chartreuse
Chocolate
Coral
Cornflower
Cornsilk
Crimson
Cyan
DarkBlue
DarkCyan
DarkGoldenrod
Darkgray
DarkGreen
DarkKhaki
DarkMagenta
DarkOliveGreen
DarkOrange
DarkOrchid
DarkRed
DarkSalmon
DarkSeaGreen
DarkSlateBlue
DarkSlateGray
DarkTurquoise
DarkViolet
DeepPink
DeepSkyBlue
DimGray
DodgerBlue
FireBrick
FloratWhite
ForestGreen
Gainsboro
GhoseWhite
Gold
GoldenRod
GreenYellow
HoneyDew
HotPink
IndianRed
Indigo
Ivory
Khaki
Lavender
LavenderBlush
LawnGreen
LemonChiffon
LightBlue
LightCoral
LightCyan
LightGoldenrodYellow
LightGreen
LightSkyBlue
LightSlateGray
LightSteelBlue
LightYellow
LimeGreen
Linen
Magenta
MediumAquamarine
MediumBlue
MediumOrchid
MediumPurple
MediumSeaGreen
MediumSlateBlue
MediumSpringGreen
MediumTurquoise
MediumVioletRed
MidnightBlue
MintCream
MistyRose
Moccasin
NavajoWhite
Navy
OldLace
Olive
OliveDrab
Orange
OrangeRed
Orchid
PaleGoldenRod
PaleGreen
PaleTurquoise
PaleVioletRed
PapayaWhip
PeachPuff
Peru
Pink
Plum
PowderBlue
Purple
Red
RosyBrown
RoyalBlue
SaddleBrown
Salmon
Sandy Brown
SeaGreen
Seashell
Sienna
Silver
SkyBlue
SlateBlue
SlateGray
Snow
SpringGreen
SteelBlue
Tan
Teal
Thistle
Tomato
Turquoise
Violet
Wheat
White
WhiteSmoke
Yellow
YellowGreen

Приложение 3. Совместимость объектных моделей документа

Объектная модель документа (Document Object Model, DOM) является средством доступа к большинству элементов, содержащихся в HTML-документах. Сведения, относящиеся к реализации DOM, представлены в таблицах по категориям: объекты, принадлежность, события и методы. В каждой таблице имеется информация о поддержке каждого элемента браузерами Internet Explorer и Netscape Navigator.

Таблицы представлены в соответствии с частотой использования элементов DOM. Это облегчит вам поиск требуемых сведений.

Применяемые сокращения:

Ош. — существуют некоторые проблемы в реализации, и поэтому элементы не всегда работают при определенных условиях.

Операционная система	**Применяемое** в ОС сокращение	Полное наименование
Windows 95,	Nav4	Netscape Navigator 4.x
Windows 98,	Nav6	Netscape Navigator 6
Windows NT	IE4	Internet Explorer 4
	IE5	Internet Explorer 5.x
Macintosh	Nav4	Netscape Navigator 4.x
	Nav6	Netscape Navigator 6
	IE4	Internet Explorer 4
	IE5	Internet Explorer 5

Основная работа DOM заключается в обеспечении связи с объектами HTML-страницы. В табл. П3.1 представлен список всех объектов DOM, поддерживаемых Internet Explorer и Netscape Navigator.

Таблица П3.1 Объекты

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
A	Да	Да	Да	Да	Да	Да	Да	Да
ACRONYM	Да	Да	Да	Да	Да	Да	Да	Да
ADDRESS	Да	Да	Да	Да	Да	Да	Да	Да
all	Нет	Нет	Да	Да	Нет	Нет	Да	Да
APPLET	Да	Да	Да	Да	Да	Да	Да	Да
applets	Да	Да	Да	Да	Да	Да	Да	Да
anchors	Да	Да	Да	Да	Да	Да	Да	Да
AREA	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
areas	Да	Да	Да	Да	Да	Да	Да	Да
Ош.	Да	Да	Да	Да	Да	Да	Да	Да
BASE	Да	Да	Да	Да	Да	Да	Да	Да
BASEFONT	Да	Да	Да	Да	Да	Да	Да	Да
BGSOUND	Нет	Нет	Да	Да	Нет	Нет	Да	Да
BIG	Да	Да	Да	Да	Да	Да	Да	Да
BLOCKQUOTE	Да	Да	Да	Да	Да	Да	Да	Да
BODY	Да	Да	Да	Да	Да	Да	Да	Да
BR	Да	Да	Да	Да	Да	Да	Да	Да
BUTTON	Ош.	Ош.	Да	Да	Ош.	Ош.	Да	Да
CAPTION	Да	Да	Да	Да	Да	Да	Да	Да
cells	Нет	Нет	Да	Да	Нет	Нет	Да	Да
CENTER	Да	Да	Да	Да	Да	Да	Да	Да
children	Нет	Нет	Да	Да	Нет	Нет	Да	Да
CITE	Да	Да	Да	Да	Да	Да	Да	Да
CODE	Да	Да	Да	Да	Да	Да	Да	Да
COL	Да	Да	Да	Да	Да	Да	Да	Да
COLGROUP	Да	Да	Да	Да	Да	Да	Да	Да
<!-- COMMENT -->	Нет	Нет	Да	Да	Нет	Нет	Да	Да
DD	Да	Да	Да	Да	Да	Да	Да	Да
DEL	Да	Да	Да	Да	Да	Да	Да	Да
DFN	Да	Да	Да	Да	Да	Да	Да	Да
DIR	Да	Да	Да	Да	Да	Да	Да	Да
DIV	Да	Да	Да	Да	Да	Да	Да	Да
DL	Да	Да	Да	Да	Да	Да	Да	Да
document	Да	Да	Да	Да	Да	Да	Да	Да
DT	Да	Да	Да	Да	Да	Да	Да	Да
elements	Нет	Да	Да	Да	Нет	Да	Да	Да
EM	Да	Да	Да	Да	Да	Да	Да	Да
EMBED	Да	Да	Да	Да	Да	Да	Да	Да
embeds	Да	Да	Да	Да	Да	Да	Да	Да
event	Да	Да	Да	Да	Да	Да	Да	Да

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
external	Нет	Нет	Да	Да	Нет	Нет	Да	Да
FIELDSET	Да	Да	Да	Да	Да	Да	Да	Да
filters	Нет	Нет	Да	Да	Нет	Нет	Да	Да
FONT	Да	Да	Да	Да	Да	Да	Да	Да
FORM	Да	Да	Да	Да	Да	Да	Да	Да
forms	Да	Да	Да	Да	Да	Да	Да	Да
FRAME	Да	Да	Да	Да	Да	Да	Да	Да
frames	Да	Да	Да	Да	Да	Да	Да	Да
FRAMESET	Да	Да	Да	Да	Да	Да	Да	Да
H1	Да	Да	Да	Да	Да	Да	Да	Да
H2	Да	Да	Да	Да	Да	Да	Да	Да
H3	Да	Да	Да	Да	Да	Да	Да	Да
H4	Да	Да	Да	Да	Да	Да	Да	Да
H5	Да	Да	Да	Да	Да	Да	Да	Да
H6	Да	Да	Да	Да	Да	Да	Да	Да
HEAD	Да	Да	Да	Да	Да	Да	Да	Да
history	Да	Да	Да	Да	Да	Да	Да	Да
HR	Да	Да	Да	Да	Да	Да	Да	Да
HTML	Да	Да	Да	Да	Да	Да	Да	Да
I	Да	Да	Да	Да	Да	Да	Да	Да
IFRAME	Да	Да	Да	Да	Да	Да	Да	Да
images	Да	Да	Да	Да	Да	Да	Да	Да
IMG	Да	Да	Да	Да	Да	Да	Да	Да
INPUT	Да	Да	Да	Да	Да	Да	Да	Да
Input Types: button checkbox file hidden image password radio reset submit text	Да	Да	Да	Да	Да	Да	Да	Да
INS	Да	Да	Да	Да	Да	Да	Да	Да
KBD	Да	Да	Да	Да	Да	Да	Да	Да
LABEL	Да	Да	Да	Да	Да	Да	Да	Да
LAYER	Да	Да	Нет	Нет	Да	Да	Нет	Нет
LEGEND	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
LI	Да	Да	Да	Да	Да	Да	Да	Да
LINK	Да	Да	Да	Да	Да	Да	Да	Да
links	Да	Да	Да	Да	Да	Да	Да	Да
LISTING	Нет	Нет	Да	Да	Нет	Нет	Да	Да
location	Да	Да	Да	Да	Да	Да	Да	Да
MAP	Да	Да	Да	Да	Да	Да	Да	Да
MARQUEE	Нет	Нет	Да	Да	Нет	Нет	Да	Да
MENU	Да	Да	Да	Да	Да	Да	Да	Да
META	Да	Да	Да	Да	Да	Да	Да	Да

продолжение

Таблица П3.1 *(продолжение)*

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
navigator	Да	Да	Да	Да	Да	Да	Да	Да
NEXTID	Да	Да	Да	Да	Да	Да	Да	Да
OBJECT	Да	Да	Да	Да	Да	Да	Да	Да
OL	Да	Да	Да	Да	Да	Да	Да	Да
OPTION	Да	Да	Да	Да	Да	Да	Да	Да
options	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
P	Да	Да	Да	Да	Да	Да	Да	Да
PLAINTEXT	Да	Да	Да	Да	Да	Да	Да	Да
plugins	Да	Да	Да	Да	Да	Да	Да	Да
PRE	Да	Да	Да	Да	Да	Да	Да	Да
Q	Да	Да	Да	Да	Да	Да	Да	Да
rows	Нет	Да	Да	Да	Нет	Да	Да	Да
rules	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
S	Да	Да	Да	Да	Да	Да	Да	Да
SAMP	Да	Да	Да	Да	Да	Да	Да	Да
screen	Да	Да	Да	Да	Да	Да	Да	Да
SCRIPT	Да	Да	Да	Да	Да	Да	Да	Да
scripts	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
SELECT	Да	Да	Да	Да	Да	Да	Да	Да
selection	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
SMALL	Да	Да	Да	Да	Да	Да	Да	Да
SPAN	Да	Да	Да	Да	Да	Да	Да	Да
STRIKE	Да	Да	Да	Да	Да	Да	Да	Да
STRONG	Да	Да	Да	Да	Да	Да	Да	Да
STYLE	Да	Да	Да	Да	Да	Да	Да	Да
style	Да	Да	Да	Да	Да	Да	Да	Да
stylesheet	Нет	Да	Да	Да	Нет	Да	Да	Да
stylesheets	Нет	Да	Да	Да	Нет	Да	Да	Да
SUB	Да	Да	Да	Да	Да	Да	Да	Да
SUP	Да	Да	Да	Да	Да	Да	Да	Да
TABLE	Да	Да	Да	Да	Да	Да	Да	Да
TBODY	Да	Да	Да	Да	Да	Да	Да	Да
TD	Да	Да	Да	Да	Да	Да	Да	Да
TEXTAREA	Да	Да	Да	Да	Да	Да	Да	Да
TextRange	Нет	Нет	Да	Да	Нет	Нет	Да	Да
TFOOT	Да	Да	Да	Да	Да	Да	Да	Да
TH	Да	Да	Да	Да	Да	Да	Да	Да
THEAD	Да	Да	Да	Да	Да	Да	Да	Да
TITLE	Да	Да	Да	Да	Да	Да	Да	Да
TR	Да	Да	Да	Да	Да	Да	Да	Да

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
TT	Да	Да	Да	Да	Да	Да	Да	Да
U	Да	Да	Да	Да	Да	Да	Да	Да
UL	Да	Да	Да	Да	Да	Да	Да	Да
userProfile	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
VAR	Да	Да	Да	Да	Да	Да	Да	Да
window	Да	Да	Да	Да	Да	Да	Да	Да
XMP	Да	Да	Да	Да	Да	Да	Да	Да

Как Internet Explorer, так и Netscape Navigator используют DOM для поддержки коллекций, связанных со всеми объектами определенного типа. В табл. П3.2 представлен список всех коллекций объектов, поддерживаемых Internet Explorer и Netscape Navigator.

Таблица П3.2. Коллекции

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
all	Нет	Нет	Да	Да	Нет	Нет	Да	Да
anchors	Да	Да	Да	Да	Да	Да	Да	Да
applets	Да	Да	Да	Да	Да	Да	Да	Да
areas	Да	Да	Да	Да	Да	Да	Да	Да
cells	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
children	Нет	Нет	Да	Да	Нет	Нет	Да	Да
classes	Да	Да	Нет	Нет	Да	Да	Нет	Нет
elements	Нет	Да	Да	Да	Нет	Да	Да	Да
embeds	Да	Да	Да	Да	Да	Да	Да	Да
filters	Нет	Нет	Да	Да	Нет	Нет	Да	Да
forms	Да	Да	Да	Да	Да	Да	Да	Да
frames	Да	Да	Да	Да	Да	Да	Да	Да
id	Да	Да	Да	Да	Да	Да	Да	Да
images	Да	Да	Да	Да	Да	Да	Да	Да
imports	Нет	Нет	Да	Да	Нет	Нет	Да	Да
links	Нет	Ош.	Да	Да	"Нет	Ош.	Да	Да
options	Да	Да	Да	Да	Нет	Да	Да	Да
plugins	Да	Да	Да	Да	Да	Да	Да	Да
rows	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
rules	Нет	Нет	Да	Да	Нет	Нет	Да	Да
scripts	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
stylesheets	Нет	Нет	Да	Да	Нет	Нет	Да	Да
tags	Да	Да	Нет	Нет	Да	Да	Нет	Нет
tbodies	Нет	Нет	Да	Да	Нет	Нет	Да	Да

Объектная модель документа обеспечивает множество событий, поддерживаемых Internet Explorer и Netscape Navigator, позволяющих обеспечить взаимодействие, определяемое поведением посетителя страницы. В табл. П3.3 представлен список событий, распознаваемых DOM, а также реализация их поддержки браузерами Internet Explorer и Netscape Navigator.

Таблица П3.3. События

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
onabort	Да	Да	Да	Да	Да	Да	Да	Да
onafterupdate	Нет	Нет	Да	Да	Нет	Нет	Да	Да
onbeforeunload	Нет	Нет	Да	Да	Нет	Нет	Да	Да
onbeforeupdate	Нет	Нет	Да	Да	Нет	Нет	Да	Да
onblur	Да	Да	Да	Да	Нет	Да	Да	Да
onbounce	Нет	Нет	Да	Да	Нет	Нет	Да	Да
onchange	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
onclick	Да	Да	Да	Да	Да	Да	Да	Да
ondataavailable	Нет	Нет	Да	Да	Нет	Нет	Да	Да
ondatasetchanged	Нет	Нет	Да	Да	Нет	Нет	Да	Да
ondatasetcomplete	Нет	Нет	Да	Да	Нет	Нет	Да	Да
ondblclick	Да	Да	Да	Да	Да	Да	Да	Да
ondragstart	Нет	Нет	Да	Да	Нет	Нет	Да	Да
onDragDrop	Да	Да	Нет	Да	Ош.	Да	Нет	Ош.
onerror	Да	Да	Да	Да	Да	Да	Да	Да
onerrorupdate	Нет	Нет	Да	Да	Нет	Нет	Да	Да
onfilterchange	Нет	Нет	Да	Да	Нет	Нет	Да	Да
onfinish	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
onfocus	Да	Да	Да	Да	Да	Да	Да	Да
onhelp	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
onkeydown	Да	Да	Да	Да	Да	Да	Да	Да
onkeypress	Да	Да	Да	Да	Да	Да	Да	Да
onkeyup	Да	Да	Да	Да	Да	Да	Да	Да
onload	Да	Да	Да	Да	Да	Да	Да	Да
onmousedown	Да	Да	Да	Да	Да	Да	Да	Да
onmousemove	Да	Да	Да	Да	Да	Да	Да	Да
onmouseout	Да	Да	Да	Да	Да	Да	Да	Да
onmouseover	Да	Да	Да	Да	Да	Да	Да	Да
onmouseup	Да	Да	Да	Да	Да	Да	Да	Да
onmove	Да	Да	Нет	Ош.	Да	Да	Нет	Ош.
onreadystatechange	Нет	Нет	Да	Да	Нет	Нет	Да	Да
onreset	Нет	Да	Да	Да	Нет	Да	Да	Да
onresize	Да	Да	Да	Да	Да	Да	Да	Да
onrowenter	Нет	Нет	Да	Да	Нет	Нет	Да	Да
onrowexit	Нет	Нет	Да	Да	Нет	Нет	Да	Да

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
onscroll	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
onselect	Нет	Да	Да	Да	Нет	Да	Да	Да
onselectstart	Нет	Нет	Да	Да	Нет	Нет	Да	Да
onstart	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
onsubmit	Нет	Да	Да	Да	Нет	Да	Да	Да
onunload	Да	Да	Да	Да	Да	Да	Да	Да

Все объекты DOM имеют описывающие их свойства. В табл. П3.4 представлены свойства, связанные с объектами, поддерживаемые DOM, Internet Explorer и Netscape Navigator.

Таблица П3.4. Свойства

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
above	Да	Да	Нет	Нет	Да	Да	Нет	Нет
accesskey	Нет	Да	Да	Да	Нет	Да	Да	Да
action	Да	Да	Да	Да	Да	Да	Да	Да
activeElement	Нет	Нет	Да	Да	Нет	Нет	Да	Да
align	Да	Да	Да	Да	Да	Да	Да	Да
aLink	Нет	Да	Да	Да	Нет	Да	Да	Да
alinkColor	Да	Да	Да	Да	Да	Да	Да	Да
alt	Нет	Да	Да	Да	Нет	Да	Да	Да
altHTML	Нет	Нет	Да	Да	Нет	Нет	Да	Да
altKey	Нет	Нет	Да	Да	Нет	Нет	Да	Да
anchor	Да	Да	Нет	Нет	Да	Да	Нет	Нет
anchors	Да	Да	Нет	Нет	Да	Да	Нет	Нет
appCodeName	Да	Да	Да	Да	Да	Да	Да	Да
applet	Да	Да	Нет	Нет	Да	Да	Нет	Нет
applets	Да	Да	Да	Да	Да	Да	Да	Да
appMinorVersion	Ош.	Да	Да	Да	Ош.	Да	Да	Да
appName	Да	Да	Да	Да	Да	Да	Да	Да
appVersion	Да	Да	Да	Да	Да	Да	Да	Да
area	Да	Да	Нет	Нет	Да	Да	Нет	Нет
availHeight	Да	Да	Да	Да	Да	Да	Да	Да
availWidth	Да	Да	Да	Да	Да	Да	Да	Да
background	Нет	Да	Да	Да	Нет	Да	Да	Да
backgroundAttachment	Нет	Нет	Да	Да	Нет	Нет	Да	Да
backgroundColor	Да	Да	Да	Да	Да	Да	Да	Да
backgroundImage	Да	Да	Да	Да	Да	Да	Да	Да
backgroundPosition	Нет	Нет	Да	Да	Нет	Нет	Да	Да
backgroundPositionX	Нет	Нет	Да	Да	Нет	Нет	Да	Да

продолжение

Таблица П3.4 (*продолжение*)

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
backgroundPositionY	Нет	Нет	Да	Да	Нет	Нет	Да	Да
backgroundRepeat	Нет	Нет	Да	Да	Нет	Нет	Да	Да
balance	Нет	Нет	Да	Да	Нет	Нет	Да	Да
behaviour	Нет	Нет	Да	Да	Нет	Нет	Да	Да
below	Да	Да	Нет	Нет	Да	Да	Нет	Нет
bgColor	Да	Да	Да	Да	Да	Да	Да	Да
bgProperties	Нет	Нет	Да	Да	Нет	Нет	Да	Да
border	Да	Да	Да	Да	Да	Да	Да	Да
borderBottom	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderBottomColor	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderBottomStyle	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderBottomWidth	Да	Да	Да	Да	Да	Да	Да	Да
borderColor	Да	Да	Да	Да	Да	Да	Да	Да
borderColorDark	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderColorLight	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderLeft	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderLeftColor	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderLeftStyle	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderLeftWidth	Да	Да	Да	Да	Да	Да	Да	Да
borderRight	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderRightColor	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderRightStyle	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderRightWidth	Да	Да	Да	Да	Да	Да	Да	Да
borderStyle	Да	Да	Да	Да	Да	Да	Да	Да
borderTop	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderTopColor	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderTopStyle	Нет	Нет	Да	Да	Нет	Нет	Да	Да
borderTopWidth	Да	Да	Да	Да	Да	Да	Да	Да
borderWidth	Да	Да	Да	Да	Да	Да	Да	Да
bottom	Да	Да	Нет	Нет	Да	Да	Нет	Нет
bottomMargin	Нет	Нет	Да	Да	Нет	Нет	Да	Да
browserLanguage	Нет	Нет	Да	Да	Нет	Нет	Да	Да
bufferDepth	Нет	Нет	Да	Да	Нет	Нет	Да	да
button	Да	Да	Да	Да	Да	Да	Да	Да
cancelBubble	Нет	Нет	Да	Да	Нет	Нет	Да	Да
caption	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
cellIndex	Нет	Да	Да	Да	Нет	Да	Да	Да
cellPadding	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
cellspacing	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
charset	Да	Да	Да	Да	Да	Да	Да	Да
checkbox	Да	Да	Да	Да	Да	Да	Да	Да

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
checked	Да	Да	Да	Да	Да	Да	Да	Да
class	Нет	Да	Да	Да	Да	Ош.	Да	Да
className	Нет	Да	Да	Да	Нет	Да	Да	Да
clear	Да	Да	Да	Да	Да	Да	Да	Да
clientHeight	Нет	Нет	Да	Да	Нет	Нет	Да	Да
clientInformation	Нет	Нет	Да	Да	Нет	Нет	Да	Да
clientLeft	Нет	Нет	Да	Да	Нет	Нет	Да	Да
clientTop	Нет	Нет	Да	Да	Нет	Нет	Да	Да
clientWidth	Нет	Нет	Да	Да	Нет	Нет	Да	Да
clientX	Нет	Да	Да	Да	Нет	Да	Да	Да
clientY	Нет	Да	Да	Да	Нет	Да	Да	Да
clip	Да	Да	Да	Да	Да	Да	Да	Да
closed	Да	Да	Да	Да	Да	Да	Да	Да
code	Нет	Нет	Да	Да	Нет	Нет	Да	Да
CodeBase	Нет	Нет	Да	Да	Нет	Нет	Да	Да
codeType	Нет	Нет	Да	Да	Нет	Нет	Да	Да
color	Да	Да	Да	Да	Да	Да	Да	Да
colorDepth	Да	Да	Да	Да	Да	Да	Да	Да
cols	Нет	Да	Да	Да	Нет	Да	Да	Да
colSpan	Нет	Да	Да	Да	Нет	Да	Да	Да
compact	Нет	Да	Да	Да	Нет	Да	Да	Да
complete	Да	Да	Да	Да	Да	Да	Да	Да
connectionSpeed	Нет	Нет	Да	Да	Нет	Нет	Да	Да
content	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
cookie	Да	Да	Да	Да	Да	Да	Да	Да
cookieEnabled	Нет	Нет	Да	Да	Нет	Нет	Да	Да
coords	Нет	Да	Да	Да	Нет	Да	Да	Да
cpuClass	Нет	Нет	Да	Да	Нет	Нет	Да	Да
cssText	Нет	Нет	Да	Да	Нет	Нет	Да	Да
ctrlKey	Нет	Нет	Да	Да	Нет	Нет	Да	Да
current	Да	Да	Нет	Нет	Да	Да	Нет	Нет
cursor	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
data	Нет	Нет	Да	Да	Нет	Нет	Да	Да
dataFld	Нет	Нет	Да	Да	Нет	Нет	Да	Да
dataFormatAs	Нет	Нет	Да	Да	Нет	Нет	Да	Да
dataPageSize	Нет	Нет	Да	Да	Нет	Нет	Да	Да
dataSrc	Нет	Нет	Да	Да	Нет	Нет	Да	Да
defaultCharset	Нет	Нет	Да	Да	Нет	Нет	Да	Да
defaultChecked	Да	Да	Да	Да	Да	Да	Да	Да
defaultSelected	Да	Да	Да	Да	Да	Да	Да	Да
defaultStatus	Да	Да	Да	Да	Да	Да	Да	Да
defaultValue	Да	Да	Да	Да	Да	Да	Да	Да

продолжение

Таблица П3.4 *(продолжение)*

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
defer	Нет	Да	Да	Да	Нет	Да	Да	Да
dialogArgumens	Нет	Нет	Да	Да	Нет	Нет	Да	Да
dialogHeight	Нет	Нет	Да	Да	Нет	Нет	Да	Да
dialogLeft	Нет	Нет	Да	Да	Нет	Нет	Да	Да
dialogTop	Нет	Нет	Да	Да	Нет	Нет	Да	Да
dialogWidth	Нет	Нет	Да	Да	Нет	Нет	Да	Да
Direction/dir	Нет	Да	Да	Да	Нет	Да	Да	Да
disabled	Нет	Да	Да	Да	Нет	Да	Да	Да
display	Да	Да	Да	Да	Да	Да	Да	Да
domain	Да	Да	Да	Да	Да	Да	Да	Да
dynsrc	Нет	Нет	Да	Да	Нет	Нет	Да	Да
elements	Да	Да	Нет	Нет	Да	Да	Нет	Нет
encoding	Да	Да	Да	Да	Да	Да	Да	Да
event	Нет	Да	Да	Да	Нет	Да	Да	Да
expando	Нет	Нет	Да	Да	Нет	Нет	Да	Да
face	Нет	Да	Да	Да	Нет	Да	Да	Да
fqColor	Да	Да	Да	Да	Да	Да	Да	Да
fileCreatedDate	Нет	Нет	Да	Да	Нет	Нет	Да	Да
fileModifiedDate	Нет	Нет	Да	Да	Нет	Нет	Да	Да
fileSize	Нет	Нет	Да	Да	Нет	Нет	Да	Да
fileUpload	Да	Да	Нет	Нет	Да	Да	Нет	Нет
fileUpdatedDate	Нет	Нет	Да	Да	Нет	Нет	Да	Да
filter	Нет	Нет	Да	Да	Нет	Нет	Да	Да
font	Да	Да	Да	Да	Да	Да	Да	Да
fontFamily	Да	Да	Да	Да	Да	Да	Да	Да
fontSize	Да	Да	Да	Да	Да	Да	Да	Да
fontSmoothingEnabled	Нет	Нет	Да	Да	Нет	Нет	Да	Да
fontStyle	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
fontVariant	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
fontWeight	Да	Да	Да	Да	Да	Да	Да	Да
form	Да	Да	Да	Да	Да	Да	Да	Да
frame	Нет	Да	Да	Да	Нет	Да	Да	Да
frameBorder	Нет	Да	Да	Да	Нет	Да	Да	Да
frameSpacing	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
fromElement	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
hash	Да	Да	Да	Да	Да	Да	Да	Да
height	Да	Да	Да	Да	Да	Да	Да	Да
hidden	Да	Да	Да	Да	Да	Да	Да	Да
host	Да	Да	Да	Да	Да	Да	Да	Да
hostname	Да	Да	Да	Да	Да	Да	Да	Да
href	Да	Да	Да	Да	Да	Да	Да	Да

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
hspace	Да	Да	Да	Да	Да	Да	Да	Да
htmlFor	Нет	Да	Да	Да	Нет	Да	Да	Да
htmlText	Нет	Нет	Да	Да	Нет	Нет	Да	Да
httpEquiv	Нет	Нет	Да	Да	Нет	Нет	Да	Да
id	Нет	Да	Да	Да	Нет	Да	Да	Да
image	Да	Да	Да	Да	Да	Да	Да	Да
indeterminate	Нет	Нет	Да	Да	Нет	Нет	Да	Да
index	Да	Да	Да	Да	Да	Да	Да	Да
innerHeight	Да	Да	Нет	Нет	Да	Да	Нет	Нет
innerHTML	Нет	Нет	Да	Да	Нет	Нет	Да	Да
innerText	Нет	Нет	Да	Да	Нет	Нет	Да	Да
innerWidth	Да	Да	Нет	Нет	Да	Да	Нет	Нет
isMap	Нет	Да	Да	Да	Нет	Да	Да	Да
isTextEdit	Нет	Нет	Да	Да	Нет	Нет	Да	Да
keyCode	Нет	Нет	Да	Да	Нет	Нет	Да	Да
lang	Нет	Да	Да	Да	Нет	Да	Да	Да
language	Да	Да	Да	Да	Да	Да	Да	Да
lastModified	Да	Да	Да	Да	Да	Да	Да	Да
layer	Да	Да	Нет	Нет	Да	Да	Нет	Нет
left	Да	Да	Да	Да	Да	Да	Да	Да
leftMargin	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
length	Да	Да	Да	Да	Да	Да	Да	Да
letterSpacing	Нет	Да	Да	Да	Нет	Да	Да	Да
lineHeight	Да	Да	Да	Да	Да	Да	Да	Да
link	Да	Да	Да	Да	Да	Да	Да	Да
linkColor	Да	Да	Да	Да	Да	Да	Да	Да
listStyle	Нет	Нет	Да	Да	Нет	Нет	Да	Да
listStyleImage	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
listStylePosition	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
listStyleType	Да	Да	Да	Да	Да	Да	Да	Да
location	Да	Да	Да	Да	Да	Да	Да	Да
locationbar	Да	Да	Нет	Нет	Да	Да	Нет	Нет
loop	Нет	Нет	Да	Да	Нет	Нет	Да	Да
lowsrc	Да	Да	Да	Да	Да	Да	Да	Да
map	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
margin	Да	Да	Да	Да	Да	Да	Да	Да
marginBottom	Да	Да	Да	Да	Да	Да	Да	Да
marginHeight	Нет	Да	Да	Да	Нет	Да	Да	Да
marginLeft	Да	Да	Да	Да	Да	Да	Да	Да
marginRight	Да	Да	Да	Да	Да	Да	Да	Да
marginTop	Да	Да	Да	Да	Да	Да	Да	Да
marginWidth	Нет	Да	Да	Да	Нет	Да	Да	Да

продолжение

Таблица П3.4 (*продолжение*)

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
maxLength	Нет	Да	Да	Да	Нет	Да	Да	Да
media	Нет	Да	Да	Да	Нет	Да	Да	Да
menubar	Да	Да	Нет	Нет	Да	Да	Нет	Нет
method	Да	Да	Да	Да	Да	Да	Да	Да
Methods	Нет	Нет	Да	Да	Нет	Нет	Да	Да
mimeTypes	Да		Да	Да	Да	Да	Да	Да
multiple	Нет	Нет	Да	Да	Нет	Нет	Да	Да
name	Да	Да	Да	Да	Да	Да	Да	Да
next	Да	Да	Нет	Нет	Да	Да	Нет	Нет
noHref	Нет	Да	Да	Да	Нет	Да	Да	Да
noResize	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
noShade	Нет	Да	Да	Да	Нет	Да	Да	Да
noWrap	Нет	Да	Да	Да	Нет	Да	Да	Да
object	Нет	Да	Да	Да	Нет	Да	Да	Да
offscreenBuffering	Нет	Нет	Да	Да	Нет	Нет	Да	Да
offsetHeight	Нет	Нет	Да	Да	Нет	Нет	Да	Да
offsetLeft	Нет	Нет	Да	Да	Нет	Нет	Да	Да
offsetParent	Нет	Нет	Да	Да	Нет	Нет	Да	Да
offsetTop	Нет	Нет	Да	Да	Нет	Нет	Да	Да
offsetWidth	Нет	Нет	Да	Да	Нет	Нет	Да	Да
offsetX	Нет	Нет	Да	Да	Нет	Нет	Да	Да
offsetY	Нет	Нет	Да	Да	Нет	Нет	Да	Да
online	Нет	Нет	Да	Да	Нет	Нет	Да	Да
opener	Да	Да	Да	Да	Да	Да	Да	Да
option	Да	Да	Да	Да	Да	Да	Да	Да
outerHeight	Да	Да	Нет	Нет	Да	Да	Нет	Нет
outerHTML	Нет	Нет	Да	Да	Нет	Нет	Да	Да
outerText	Нет	Нет	Да	Да	Нет	Нет	Да	Да
outerWidth	Да	Да	Нет	Нет	Да	Да	Нет	Нет
overflow	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
owningElement	Нет	Нет	Да	Да	Нет	Нет	Да	Да
padding	Да	Да	Да	Да	Да	Да	Да	Да
paddingBottom	Да	Да	Да	Да	Да	Да	Да	Да
padding Left	Да	Да	Да	Да	Да	Да	Да	Да
paddingRight	Да	Да	Да	Да	Да	Да	Да	Да
paddingTop	Да	Да	Да	Да	Да	Да	Да	Да
pageBreakAfter	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
pageBreakBefore	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
pageX	Да	Да	Нет	Нет	Да	Да	Нет	Нет
pageXOffset	Да	Да	Нет	Нет	Да	Да	Нет	Нет
pageY	Да	Да	Нет	Нет	Да	Да	Нет	Нет

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
pageYOffset	Да	Да	Нет	Нет	Да	Да	Нет	Нет
palette	Нет	Нет	Да	Да	Нет	Нет	Да	Да
parent	Да	Да	Да	Да	Да	Да	Да	Да
parentElement	Нет	Да	Да	Да	Нет	Да	Да	Да
parentLayer	Да	Да	Нет	Нет	Да	Да	Нет	Нет
parentStyleSheet	Нет	Нет	Да	Да	Нет	Нет	Да	Да
parentTextEdit	Нет	Нет	Да	Да	Нет	Нет	Да	Да
parentWindow	Нет	Нет	Да	Да	Нет	Нет	Да	Да
password	Да	Да	Нет	Нет	Да	Да	Нет	Нет
pathname	Да	Да	Да	Да	Да	Да	Да	Да
personalbar	Да	Да	Нет	Нет	Да	Да	Нет	Нет
pixelDepth	Да	Да	Нет	Нет	Да	Да	Нет	Нет
pixelHeight	Нет	Нет	Да	Да	Нет	Нет	Да	Да
pixelLeft	Нет	Нет	Да	Да	Нет	Нет	Да	Да
pixelTop	Нет	Нет	Да	Да	Нет	Нет	Да	Да
pixelWidth	Нет	Нет	Да	Да	Нет	Нет	Да	Да
platform	Да	Да	Да	Да	Да	Да	Да	Да
plugins	Да	Да	Да	Да	Да	Да	Да	Да
pluginspace	Нет	Нет	Да	Да	Нет	Нет	Да	Да
port	Да	Да	Да	Да	Да	Да	Да	Да
posHeight	Нет	Нет	Да	Да	Нет	Нет	Да	Да
postion	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
posLeft	Нет	Нет	Да	Да	Нет	Нет	Да	Да
posTop	Нет	Нет	Да	Да	Нет	Нет	Да	Да
posWidth	Нет	Нет	Да	Да	Нет	Нет	Да	Да
previous	Да	Да	Нет	Нет	Да	Да	Нет	Нет
protocol	Да	Да	Да	Да	Да	Да	Да	Да
radio	Да	Да	Да	Да	Да	Да	Да	Да
readonly	Нет	Да	Да	Да	Нет	Да	Да	Да
readyState	Нет	Нет	Да	Да	Нет	Нет	Да	Да
reason	Нет	Нет	Да	Да	Нет	Нет	Да	Да
recordNumber	Нет	Нет	Да	Да	Нет	Нет	Да	Да
recordset	Нет	Нет	Да	Да	Нет	Нет	Да	Да
referrer	Да	Да	Да	Да	Да	Да	Да	Да
rel	Нет	Да	Да	Да	Нет	Да	Да	Да
reset	Да	Да	Нет	Нет	Да	Да	Нет	Нет
returnValue	Нет	Нет	Да	Да	Нет	Нет	Да	Да
rev	Нет	Да	Да	Да	Нет	Да	Да	Да
right	Да	Да	Нет	Нет	Да	Да	Нет	Нет
rightMargin	Нет	Нет	Да	Да	Нет	Нет	Да	Да
rowIndex	Нет	Да	Да	Да	Нет	Да	Да	Да
rows	Нет	Да	Да	Да	Нет	Да	Да	Да

продолжение

Таблица П3.4 *(продолжение)*

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
rowSpan	Нет	Да	Да	Да	Нет	Да	Да	Да
rules	Нет	Нет	Да	Да	Нет	Нет	Да	Да
screenX	Нет	Нет	Да	Да	Нет	Нет	Да	Да
screenY	Нет	Нет	Да	Да	Нет	Нет	Да	Да
scroll	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
scrollAmount	Нет	Нет	Да	Да	Нет	Нет	Да	Да
scrollbar	Да	Да	Нет	Нет	Да	Да	Нет	Нет
scrollDelay	Нет	Нет	Да	Да	Нет	Нет	Да	Да
scrollHeight	Нет	Нет	Да	Да	Нет	Нет	Да	Да
scrolling	Нет	Да	Да	Да	Нет	Да	Да	Да
scrollLeft	Нет	Нет	Да	Да	Нет	Нет	Да	Да
scrollTop	Нет	Нет	Да	Да	Нет	Нет	Да	Да
scrollWidth	Нет	Нет	Да	Да	Нет	Нет	Да	Да
search	Да	Да	Да	Да	Да	Да	Да	Да
sectionRowIndex	Нет	Нет	Да	Да	Нет	Нет	Да	Да
select	Да	Да	Да	Да	Да	Да	Да	Да
selected	Да	Да	Да	Да	Да	Да	Да	Да
selectedIndex	Да	Да	Да	Да	Да	Да	Да	Да
self	Да	Да	Да	Да	Да	Да	Да	Да
shape	Нет	Да	Да	Да	Нет	Да	Да	Да
shiftKey	Нет	Нет	Да	Да	Нет	Нет	Да	Да
siblingAbove	Да	Да	Нет	Нет	Да	Да	Нет	Нет
siblingBelow	Да	Да	Нет	Нет	Да	Да	Нет	Нет
sIze	Нет	Да	Да	Да	Нет	Да	Да	Да
sourceIndex	Нет	Нет	Да	Да	Нет	Нет	Да	Да
span	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
src	Да	Да	Да	Да	Да	Да	Да	Да
srcElement	Нет	Нет	Да	Да	Нет	Нет	Да	Да
srcFilter	Нет	Нет	Да	Да	Нет	Нет	Да	Да
start	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
status	Да	Да	Да	Да	Да	Да	Да	Да
statusbar	Да	Да	Нет	Нет	Да	Да	Нет	Нет
style	Нет	Да	Да	Да	Нет	Да	Да	Да
styleFloat	Нет	Нет	Да	Да	Нет	Нет	Да	Да
submit	Да	Да	Нет	Нет	Да	Да	Нет	Нет
systemLanguage	Нет	Нет	Да	Да	Нет	Нет	Да	Да
tabindex	Нет	Да	Да	Да	Нет	Да	Да	Да
tagName	Нет	Да	Да	Да	Нет	Да	Да	Да
target	Да	Да	Да	Да	Да	Да	Да	Да
text	Да	Да	Да	Да	Да	Да	Да	Да
textAlign	Да	Да	Да	Да	Да	Да	Да	Да

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
textarea	Да	Да	Нет	Нет	Да	Да	Нет	Нет
textDecoration	Да	Да	Да	Да	Да	Да	Да	Да
textDecorationBlink	Нет	Нет	Да	Да	Нет	Нет	Да	Да
textDeooratfonUneThrough	Нет	Нет	Да	Да	Нет	Нет	Да	Да
textDecorationNone	Нет	Нет	Да	Да	Нет	Нет	Да	Да
textDecorationOverline	Нет	Нет	Да	Да	Нет	Нет	Да	Да
textDecorationUnderline	Нет	Нет	Да	Да	Нет	Нет	Да	Да
textIndent	Да	Да	Да	Да	Да	Да	Да	Да
textTransform	Да	Да	Да	Да	Да	Да	Да	Да
thead	Нет	Нет	Да	Да	Нет	Нет	Да	Да
tfoot	Нет	Нет	Да	Да	Нет	Нет	Да	Да
title	Да	Да	Да	Да	Да	Да	Да	Да
toElement	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
toolbar	Да	Да	Ош.	Ош.	Да	Да	Ош.	Ош.
top	Да	Да	Да	Да	Да	Да	Да	Да
topMargin	Нет	Нет	Да	Да	Нет	Нет	Да	Да
trueSpeed	Нет	Нет	Да	Да	Нет	Нет	Да	Да
type	Да	Да	Да	Да	Да	Да	Да	Да
updateInterval	Нет	Да	Да	Да	Нет	Да	Да	Да
units	Нет	Нет	Да	Да	Нет	Нет	Да	Да
URL	Нет	Нет	Да	Да	Нет	Нет	Да	Да
url	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
urn	Нет	Нет	Да	Да	Нет	Нет	Да	Да
useMap	Нет	Да	Да	Да	Нет	Да	Да	Да
userAgent	Да	Да	Да	Да	Да	Да	Да	Да
userLanguage	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
vAlign	Нет	Да	Да	Да	Нет	Да	Да	Да
value	Да	Да	Да	Да	Да	Да	Да	Да
verticalAlign	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
visibility	Да	Да	Да	Да	Да	Да	Да	Да
vLink	Нет	Да	Да	Да	Нет	Да	Да	Да
vlinkColor	Да	Да	Да	Да	Да	Да	Да	Да
volume	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
vspace	Да	Да	Да	Да	Да	Да	Да	Да
whitespace	Да	Да	Нет	Нет	Да	Да	Нет	Нет
width	Да	Да	Да	Да	Да	Да	Да	Да
wrap	Нет	Да	Да	Да	Нет	Да	Да	Да
x .	Нет	Да	Да	Да	Нет	Да	Да	Да
y	Нет	Да	Да	Да	Нет	Да	Да	Да
z-index	Да	Да	Да	Да	Да	Да	Да	Да

Методы DOM обеспечивают возможность взаимодействия объектов, составляющих HTML-документ, и их свойств.

Таблица П3.5. Методы

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
add	Ош.	Да	Да	Да	Ош.	Да	Да	Да
addChannel	Нет	Нет	Да	Да	Нет	Нет	Да	Да
addImport	Нет	Нет	Да	Да	Нет	Нет	Да	Да
addReadRequest	Нет	Нет	Да	Да	Нет	Нет	Да	Да
addRule	Нет	Нет	Да	Да	Нет	Нет	Да	Да
alert	Да	Да	Да	Да	Да	Да	Да	Да
assign	Ош.	Ош.	Да	Да	Ош.	Ош.	Да	Да
back	Да	Да	Да	Да	Да	Да	Да	Да
blur	Да	Да	Да	Да	Да	Да	Да	Да
captureEvents	Да	Да	Нет	Нет	Да	Да	Нет	Нет
clear	Да	Да	Да	Да	Да	Да	Да	Да
clearInterval	Да	Да	Да	Да	Да	Да	Да	Да
clearRequest	Нет	Нет	Да	Да	Нет	Нет	Да	Да
clearTImeout	Да	Да	Да	Да	Да	Да	Да	Да
click	Да	Да	Да	Да	Да	Да	Да	Да
close	Да	Да	Да	Да	Да	Да	Да	Да
collapse	Нет	Нет	Да	Да	Нет	Нет	Да	Да
compareEndPoints	Нет	Нет	Да	Да	Нет	Нет	Да	Да
confirm	Да	Да	Да	Да	Да	Да	Да	Да
contains	Да	Да	Да	Да	Да	Да	Да	Да
createCaption	Нет	Нет	Да	Да	Нет	Нет	Да	Да
createElement	Ош.	Да	Да	Да	Ош.	Да	Да	Да
createRange	Нет	Нет	Да	Да	Нет	Нет	Да	Да
createStyleSheet	Да	Да	Да	Да	Да	Да	Да	Да
createTextNode	Нет	Да	Нет	Нет	Нет	Да	Нет	Нет
createTextRange	Нет	Нет	Да	Да	Нет	Нет	Да	Да
createTHead	Нет	Нет	Да	Да	Нет	Нет	Да	Да
createTFoot	Нет	Нет	Да	Да	Нет	Нет	Да	Да
deleteCaption	Нет	Нет	Да	Да	Нет	Нет	Да	Да
deleteCell	Нет	Нет	Да	Да	Нет	Нет	Да	Да
deleteRow	Нет	Нет	Да	Да	Нет	Нет	Да	Да
deleteTFoot	Нет	Нет	Да	Да	Нет	Нет	Да	Да
deleteTHead	Нет	Нет	Да	Да	Нет	Нет	Да	Да
disableExternalCapture	Да	Да	Нет	Нет	Да	Да	Нет	Нет
doReadRequest	Нет	Нет	Да	Да	Нет	Нет	Да	Да
duplicate	Нет	Нет	Да	Да	Нет	Нет	Да	Да
elementFromPoint	Нет	Нет	Да	Да	Нет	Нет	Да	Да
empty	Ош.	Ош.	Да	Да	Ош.	Ош.	Да	Да

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
enableExternalCapture	Да	Да	Нет	Нет	Да	Да	Нет	Нет
eval	Да	Да	Нет	Нет	Да	Да	Нет	Нет
execCommand	Нет	Нет	Да	Да	Нет	Нет	Да	Да
execScript	Нет	Нет	Да	Да	Нет	Нет	Да	Да
expand	Ош.	Ош.	Да	Да	Ош.	Ош.	Да	Да
find	Да	Да	Нет	Нет	Да	Да	Нет	Нет
findText	Нет	Нет	Да	Да	Нет	Нет	Да	Да
focus	Да	Да	Да	Да	Да	Да	Да	Да
forward	Да	Да	Да	Да	Да	Да	Да	Да
getAttribute	Да	Да	Да	Да	Да	Да	Да	Да
getBookmark	Да	Да	Да	Да	Да	Да	Да	Да
go	Да	Да	Да	Да	Да	Да	Да	Да
handleEvent	Да	Да	Нет	Нет	Да	Да	Нет	Нет
home	Да	Да	Нет	Нет	Да	Да	Нет	Нет
inRange	Нет	Нет	Да	Да	Нет	Нет	Да	Да
insertAdjacentHTML	Нет	Нет	Да	Да	Нет	Нет	Да	Да
insertAdjacentText	Нет	Нет	Да	Да	Нет	Нет	Да	Да
insertCell	Нет	Нет	Да	Да	Нет	Нет	Да	Да
insertRow	Нет	Нет	Да	Да	Нет	Нет	Да	Да
isEqual	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
isSubscribed	Нет	Нет	Да	Да	Нет	Нет	Да	Да
item	Да	Да	Да	Да	Да	Да	Да	Да
javaEnabled	Да	Да	Да	Да	Да	Да	Да	Да
move	Нет	Ош.	Да	Да	Нет	Ош.	Да	Да
moveBy	Да	Да	Да	Да	Да	Да	Да	Да
moveEnd	Нет	Нет	Да	Да	Нет	Нет	Да	Да
moveStart	Нет	Нет	Да	Да	Нет	Нет	Да	Да
moveTo	Да	Да	Да	Да	Да	Да	Да	Да
moveToBookmark	Нет	Нет	Да	Да	Нет	Нет	Да	Да
moveToElementText	Нет	Нет	Да	Да	Нет	Нет	Да	Да
moveToPoint	Нет	Нет	Да	Да	Нет	Нет	Да	Да
navigate	Нет	Нет	Да	Да	Нет	Нет	Да	Да
nextPage	Да	Да	Да	Да	Да	Да	Да	Да
open	Да	Да	Да	Да	Да	Да	Да	Да
parentElement	Да	Да	Да	Да	Да	Да	Да	Да
pasteHTML	Нет	Нет	Да	Да	Нет	Нет	Да	Да
previousPage	Да	Да	Да	Да	Да	Да	Да	Да
print	Да	Да	Нет	Нет	Да	Да	Нет	Нет
prompt	Да	Да	Да	Да	Да	Да	Да	Да
queryCommandEnabled	Нет	Нет	Да	Да	Нет	Нет	Да	Да
queryCommandIndeterm	Нет	Нет	Да	Да	Нет	Нет	Да	Да

продолжение

Таблица П3.5 (*продолжение*)

Наименование	Win 95/98/NT				Macintosh			
	Nav4	Nav6	IE4	IE5	Nav4	Nav6	IE4	IE5
queryCommandState	Нет	Нет	Да	Да	Нет	Нет	Да	Да
queryCommandSupported	Нет	Нет	Да	Да	Нет	Нет	Да	Да
queryCommandValue	Нет	Нет	Да	Да	Нет	Нет	Да	Да
refresh	Нет	Да	Да	Да	Нет	Да	Да	Да
releaseEvents	Да	Да	Нет	Нет	Да	Да	Нет	Нет
reload	Нет	Да	Да	Да	Нет	Да	Да	Да
remove	Нет	Нет	Да	Да	Нет	Нет	Да	Да
removeAttribute	Нет	Нет	Да	Да	Нет	Нет	Да	Да
replace	Нет	Да	Да	Да	Нет	Да	Да	Да
reset	Нет	Да	Да	Да	Нет	Да	Да	Да
resizeBy	Да	Да	Да	Да	Да	Да	Да	Да
resizeTo	Да	Да	Да	Да	Да	Да	Да	Да
routeEvents	Да	Да	Нет	Нет	Да	Да	Нет	Нет
scroll	Да	Да	Да	Да	Да	Да	Да	Да
scroll By	Да	Да	Да	Да	Да	Да	Да	Да
scrollTo	Да	Да	Да	Да	Да	Да	Да	Да
scrollIntoView	Нет	Нет	Да	Да	Нет	Нет	Да	Да
select	Нет	Да	Да	Да	Нет	Да	Да	Да
setAttribute	Нет	Да	Да	Да	Нет	Да	Да	Да
setEndPoint	Нет	Да	Да	Да	Нет	Да	Да	Да
setInterval	Да	Да	Да	Да	Да	Да	Да	Да
SetTimeout	Да	Да	Да	Да	Да	Да	Да	Да
showHelp	Нет	Нет	Да	Да	Нет	Нет	Да	Да
showModalDialog	Да	Да	Да	Да	Да	Да	Да	Да
start	Нет	Да	Да	Да	Нет	Да	Да	Да
stop	Да	Да	Да	Да	Да	Да	Да	Да
submit	Нет	Да	Да	Да	Нет	Да	Да	Да
tags	Да	Да	Да	Да	Да	Да	Да	Да
taintEnabled	Да	Да	Да	Да	Да	Да	Да	Да
toString	Да	Да	Нет	Да	Да	Да	Нет	Да
valueOF	Да	Да	Нет	Да	Да	Да	Нет	Да
write	Да	Да	Да	Да	Да	Да	Да	Да
writeln	Да	Да	Да	Да	Да	Да	Да	Да
zOrder	Ош.	Да	Да	Да	Ош.	Да	Да	Да

Приложение 4. Совместимость JavaScript и JScript

Существует множество функций, объектов, свойств, методов и операторов языка JavaScript, которые работают как с двумя основными web-браузерами, так и с другими браузерами, такими как Opera. В данном приложении представлены именно эти элементы.

В табл. П4.1 перечислены функции, используемые в JScript. Они не воспринимаются Netscape Navigator.

Применяемые сокращения:

Ош. — существуют некоторые проблемы в реализации, и поэтому они не всегда работают при определенных условиях.

* — поддерживается только IE.

** — поддерживается только NS.

ПРИМЕЧАНИЕ

Netscape Navigator 6 поддерживает JavaScript 1.5, который на 100 % совместим с ECMAScript 3.

Таблица П4.1. Функции

Элемент	Netscape		Microsoft	
	Nav4.x	Nav6	IE4	IE5
ScriptEngine()	Нет	Нет	Да	Да
ScriptEngineBuildVersion()	Нет	Нет	Да	Да
ScriptEngineMajorVersion()	Нет	Нет	Да	Да
ScriptEngineMinorVersion()	Нет	Нет	Да	Да

Снова корпорация Microsoft ввела собственные коллекции, которые не поддерживаются компанией Netscape и другими разработчиками программного обеспечения. Представленные в табл. П4.2 коллекции позволяют обращаться ко всем дисковым устройствам, файлам и папкам.

Таблица П4.2. Коллекции

Элемент	При использовании:	Netscape		Microsoft	
		Nav4.x	Nav6	IE4	IE5
Drives	Count, Item	Нет	Нет	Да	Да
Files	Count, Item	Нет	Нет	Да	Да
Folders	AddFolders(),Count, Item	Нет	Нет	Да	Да

Объекты позволяют обращаться к определенным частям информации, хранимым в таких элементах, как массивы или строчные переменные. В таблице П4.3 представлены объекты, поддерживаемые web-браузерами Netscape и Microsoft.

Таблица П4.3. Объекты

Элемент	**При использовании:**	**Netscape**		**Microsoft**	
		Nav4.x	**Nav6**	**IE4**	**IE5**
Array	concat(), join(), reverse(), slice(), sort(), constructor**, length, prototype, toString, valueOf**	Да	Да	Да	Да
Boolean	constructor*, prototype, toString, valueOf*	Нет	Да	Да	Да
Date	getDate(), getDay(), getFullYear(), getHours(), getMilliseconds()*, getMinutes(), getMonth(), getSeconds(), getTime(), getTimeZoneOffset(), getUTCDate(), getUTCDay(), getUTCFullYear()*, getUTCHours(), getUTCMilliSeconds()*, getUTCMinutes(), getUTCMonth(), getUTCSeconds(), getVarDate(), getYear(), setDate(), setFullYear(), setHours(), setMilliseconds()*, setMinutes(), setMonth(), setSeconds(), setTime(), setUTCDate(), setUTCFullYear()*, setUTCHours(), setUTCMilliSeconds()*, setUTCMinutes(), setUTCMonth(), setUTCSeconds(), toGMTString(), toLocaleString(), toUTCString()*, toString, valueOf*, parse, UTC, constructor*, prototype	Да	Да	Да	Да
Dictionary	Add(), Exists(), Items(), Keys(), Remove(), RemoveAll(), Count, Item, Key	Нет	Нет	Да	Да
Drive	AvailableSpace, DriveLetter, DriveType, FileSystem, FreeSpace, IsReady, Path, RootFolder, SerialNumber, ShareName, TotalSize, VolumeName	Нет	Нет	Да	Да
Enumerator	atEnd(), item(), moveFirst(), moveNext()	Нет	Нет	Да	Да
Error	constructor, name, message, toString()	Да	Да	Да	Да
File	Copy(), Delete(), Move(), OpenAsTextStream(), Attributes, DateCreated, DateLastAccessed, DateLastModified, Drive, Name, ParentFolder, Path, ShortName, ShortPath, Size, Type	Нет	Нет	Да	Да
FileSystemObject	BuildPath(), CopyFile(), CreateFolder(), CreateTextFile(), DeleteFile(), DeleteFolder(), DriveExists(), FileExists(), FolderExists(), GetBaseName(), GetDrive(), GetDriveName(), GetExtensionName(), GetFile(), GetFileName(), GetFolder(),GetParentFolderName(), GetSpecialFolder(), GetTempName(), MoveFile(), MoveFolder(), OpenTextFile(), Drives	Да	Нет	Нет	Да

Элемент	При использовании:	Netscape		Microsoft	
		Nav4.x	Nav6	IE4	IE5
Folder	Copy(), CreateTextFile(), Delete(), Move(), OpenAsTextStream(), Attributes, DateCreated, DateLastAccessed, DateLastModified, Drive, IsRootFolder, Name, ParentFolder, path, ShortName, ShortPath, Size, SubFolders	Нет	Нет	Да	Да
Global	boolean()**, escape(), eval(), isFinite()*, isNaN(), number()**, parseFloat(), parseInt(), string()**, unescape()	Да	Да	Да	Да
Function	toString, valueOf*, arguments, caller, constructor*, prototype	Да	Да	Да	Да
Math	abs(), acos(), asin(), atan(), atan2(),ceil(), cos(), exp(), floorQ, log(), max(), min(), pow(), random(), round(), sin(), sqrt(), tan(), E, LN2, LN10, LOG2E, LOG10E, PI, SQRT1_2, SQRT2	Да	Да	Да	Да
Number	toString, valueOf*, MAX VALUE, MIN VALUE, NaN, NEGATIVE_INFINITY, POSITIVE_INFINITY, constructor*, prototype	Да	Да	Да	Да
Object	toString, valueOf*, prototype, constructor*	Да	Да	Да	Да
RegExp	$1..$9, index, input, lastIndex, lastMatch, lastParen, leftContext, multiline, rightContext	Да	Да	Да	Да
RegularExpression	compile(), exec(), test(), global, ignoreCase, lastIndex, source, str.Match()**, str.Replace()**, str.Search()**, str.Split()**	Да	Да	Да	Да
String	anchor(), big(), blink(), bold(), charAt(), charCodeAt(), concat(), fixed(), fontcolor(), fontsize(), fromCharCode(), indexOf(), italics(), lastIndexOf(), link(), match(), replace(), search(), slice(), small(), split(), sub(), substr(), substring()**, sup(), toLowerCase(), toUpperCase(), toString, valueOf*, constructor*, length, prototype	Да	Да	Да	Да
TextStream	Close(), Read(), ReadAll(), ReadLine(), Skip(), SkipLine(), Write(), WriteBlankLines(), WriteLine(), AtEndOfLine, AtEndOfStream, Column, Line	Нет	Нет	Да	Да
VBArray	dimensions(), getItem(), lbound(), toArray(), ubound()	Нет	Нет	Да	Да

Осуществлять взаимодействие с объектами и свойствами позволяют методы. Все методы, поддерживаемые Netscape и/или Microsoft, представлены в табл. П4.4.

Таблица П4.4. Методы

Элемент	Применяется к объектам/коллекциям:	Netscape		Microsoft	
		Nav4.x	Nav6	IE4	IE5
abs(number)	Math	Да	Да	Да	Да
acos(number)	Math	Да	Да	Да	Да
Add(key, item)	Dictionary	Нет	Нет	Да	Да
AddFolders(foldername)	Folders	Нет	Нет	Да	Да
anchor(string)	String	Да	Да	Да	Да
asin(number)	Math	Да	Да	Да	Да

продолжение

Таблица П4.4 *(продолжение)*

Элемент	Применяется к объектам/коллекциям:	Netscape		Microsoft	
		Nav4.x	Nav6	IE4	IE5
atan(number)	Math	Да	Да	Да	Да
atan2(x,y)	Math	Да	Да	Да	Да
atEnd()	Enumerator	Нет	Нет	Да	Да
big()	String	да	Да	Да	Да
blink()	String	Да	Да	Да	Да
bold()	String	Да	Да	Да	Да
buildpath(path, name)	FileSystemObject	Нет	Нет	Да	Да
ceil(number)	Math	Да	Да	Да	Да
charAt(index)	String	Да	Да	Да	Да
charCodeAt(index)	String	Да	Да	Да	Да
close()	TextStream	Нет	Да	Да	Да
compile(pattern)	Regular Expression	Да	Да	Да	Да
concat(array2)	Array, String	Да	Да	Да	Да
copy(destination, [overright])	File, Folder	Нет	Нет	Да	Да
CopyFile(source, destination, [overright])	FileSystemObject	Нет	Нет	Да	Да
cos(number)	Math	Да	Да	Да	Да
CreateFolder(name)	FileSystemObject	Нет	Нет	Да	Да
CreateTextFile(filename [, overwrite[, unicode]])	File	Да	Да	Да	Да
Delete(force)	File, Folder	Нет	Да	Да	Да
DeleteFile(filespec, [force])	FileSystemObject	Нет	Нет	Да	Да
DeleteFolder(folderspec, [force])	FileSystemObject	Нет	Нет	Да	Да
dimensions()	VBArray	Нет	Нет	Да	Да
DriveExists(drivespec)	FileSystemObject	Нет	Нет	Да	Да
escape(charstring)	Global	Да	Да	Да	Да
eval(codestring)	Global	Да	Да	Да	Да
exec(string)	Regular Expression	Да	Да	Да	Да
Exists(key)	Dictionary	Нет	Нет	Да	Да
exp(number)	Math	Да	Да	Да	Да
FileExists(filespec)	FileSystemObject	Нет	Нет	Да	Да
fixed()	String	Да	Да	Да	Да
floor(number)	Math	Да	Да	Да	Да
FolderExists(folderspecs)	FileSystemObject	Нет	Нет	Да	Да
fontcolor(color)	String	Да	Да	Да	Да
fontsize(intSize)	String	Да	Да	Да	Да
fromCharCode(code1, code2, ..., coden)	String	Да	Да	Да	Да
GetAbsolutePathName (pathspec)	FileSystemObject	Нет	Нет	Да	Да
GetBaseName(path)	FileSystemObject	Нет	Нет	Да	Да
getDate()	Date	Да	Да	Да	Да
getDay()	Date	Да	Да	Да	Да
getDrive(drivespec)	FileSystemObject	Нет	Нет	Да	Да

Элемент	Применяется к объектам/коллекциям:	Netscape		Microsoft	
		Nav4.x	Nav6	IE4	IE5
getDriveName(path)	FileSystemObject	Нет	Нет	Да	Да
getExtensionName(path)	FileSystemObject	Нет	Нет	Да	Да
getFile(filespec)	FileSystemObject	Нет	Нет	Да	Да
getFileName(pathspec)	FileSystemObject	Нет	Нет	Да	Да
GetFoldergetFolder(folderspec)	FileSystemObject	Нет	Нет	Да	Да
getFullYear	Date	Да	Да	Да	Да
getHours()	Date	Да	Да	Да	Да
getItem(dimension1[, dimension2, ...], dimension)	VBArray	Нет	Нет	Да	Да
getMilliseconds()	Date	Нет	Да	Да	Да
getMinutes()	Date	Да	Да	Да	Да
getMonth	Date	Да	Да	Да	Да
GetParentFolderName(path)	FileSystemObject	Нет	Нет	Да	Да
getSeconds()	Date	Да	Да	Да	Да
GetSpecialFolder(path)	FileSystemObject	Нет	Нет	Да	Да
GetTempName (path)	FileSystemObject	Нет	Нет	Да	Да
getTime()	Date	Да	Да	Да	Да
getTimezoneOffset()	Date	Да	Да	Да	Да
getUTCDate()	Date	Да	Да	Да	Да
getUTCDay()	Date	Да	Да	Да	Да
getUTCFullYear()	Date	Да	Да	Да	Да
getUtCHours()	Date	Да	Да	Да	Да
getUTCMilliseconds()	Date	Нет	Да	Да	Да
getUTCMinutes()	Date	Да	Да	Да	Да
getUTCMonth()	Date	Да	Да	Да	Да
getUTCSeconds()	Date	Да	Да	Да	Да
getVarDate()	Date	Нет	Нет	Да	Да
getYear()	Date	Да	Да	Да	Да
indexOf(string, [i,j])	String	Да	Да	Да	Да
isFinite()	Global	Нет	Да	Да	Да
isNaN(expression)	Global	Да	Да	Да	Да
italics()	String	Да	Да	Да	Да
item	Enumerator	Нет	Да	Да	Да
Items	Dictionary	Нет	Нет	Да	Да
join(char)	Array	Да	Да	Да	Да
Keys	Dictionary	Нет	Нет	Да	Да
lastIndexOf(string, [i,j])	String	Да	Да	Да	Да
lbound	VBArray	Нет	Нет	Да	Да
link(url)	String	Да	Да	Да	Да
log(number)	Math	Да	Да	Да	Да
match(regexp)	String	Да	Да	Да	Да
max(num1, num2)	Math	Да	Да	Да	Да
min(numl, num2)	Math	Да	Да	Да	Да
Move	File, Folder	Нет	Нет	Да	Да
MoveFile	FileSystemObject	Нет	Нет	Да	Да
moveFirst	Enumerator	Нет	Нет	Да	Да

продолжение

Таблица П4.4 *(продолжение)*

Элемент	Применяется к объектам/коллекциям:	Netscape		Microsoft	
		Nav4.x	Nav6	IE4	IE5
MoveFolder	FileSystemObject	Нет	Нет	Да	Да
moveNext	Enumerator	Нет	Нет	Да	Да
OpenAsTextStream	File, Folder	Нет	Нет	Да	Да
OpenTextFile	FileSystemObject	Нет	Нет	Да	Да
parse(datestring)	Date	Да	Да	Да	Да
parseFloat(string)	Global	Да	Да	Да	Да
parseInt(string)	Global	Да	Да	Да	Да
pow(num, power)	Math	Да	Да	Да	Да
random()	Math	Да	Да	Да	Да
Read	TextStream	Нет	Нет	Да	Да
ReadAll	Textstream	Нет	Нет	Да	Да
ReadLine	TextStream	Нет	Нет	Да	Да
Remove	Dictionary	Нет	Нет	Да	Да
RemoveAll	Dictionary	Нет	Нет	Да	Да
replace(regexp, string)	String	Да	Да	Да	Да
reverse()	Array	Да	Да	Да	Да
round(number)	Math	Да	Да	Да	Да
search(string, string)	String	Да	Да	Да	Да
setDate()	Date	Да	Да	Да	Да
setFullYear()	Date	Да	Да	Да	Да
setHours()	Date	Да	Да	Да	Да
setMilliseconds()	Date	Нет	Да	Да	Да
setMinutes()	Date	Да	Да	Да	Да
setMonth()	Date	Да	Да	Да	Да
setSeconds()	Date	Да	Да	Да	Да
setTime()	Date	Да	Да	Да	Да
setUTCDate()	Date	Да	Да	Да	Да
setUTCFullYear()	Date	Да	Да	Да	Да
setUTCHours()	Date	Да	Да	Да	Да
setUTCMilliseconds()	Date	Нет	Да	Да	Да
setUTCMinutes()	Date	Да	Да	Да	Да
setUTCMonth()	Date	Да	Да	Да	Да
setUTCSeconds()	Date	Да	Да	Да	Да
setYear()	Date	Да	Да	Да	Да
sin(number)	Math	Да	Да	Да	Да
Skip	TextStream	Нет	Нет	Да	Да
SkipLine	TextStream	Нет	Нет	Да	Да
slice(i,j)	Array, String	Да	Да	Да	Да
small()	String	Да	Да	Да	Да
sort(comp.function)	Array	Да	Да	Да	Да
split(char)	String	Да	Да	Да	Да
sqrt(number)	Math	Да	Да	Да	Да
strike()	String	Да	Да	Да	Да
sub()	String	Да	Да	Да	Да
substr(start, length)	String	Да	Да	Да	Да

Элемент	Применяется к объектам/коллекциям:	Netscape		Microsoft	
		Nav4.x	Nav6	IE4	IE5
substring(intA, intB)	String	Да	Да	Да	Да
sup()	String	Да	Да	Да	Да
tan(number)	Math	Да	Да	Да	Да
test(string)	Regular Expression	Да	Да	Да	Да
toArray	VBArray	Нет	Нет	Да	Да
toGMTString()	Date	Да	Да	Да	Да
toLocaleString()	Date	Да	Да	Да	Да
toLowerCase()	String	Да	Да	Да	Да
toString()	Array, Boolean, Date, Function, Number, Object, String	Да	Да	Да	Да
toUpperCase()	String	Да	Да	Да	Да
toUTCString()	Date	Нет	Да	Да	Да
ubound	VBArray	Нет	Нет	Да	Да
unescape(string)	Global	Да	Да	Да	Да
UTC()	Date	Да	Да	Да	Да
valueOf*	Array, Boolean, Date, Function, Number, Object, String	Нет	Да	Да	Да
Write(string)	TextStream	Нет	Да	Да	Да
WriteBlankLines	TextStream	Нет	Нет	Да	Да
WriteLine(string)	TextStream	Нет	Нет	Да	Да

Операторы сценария позволяют осуществлять математические действия с информацией. Можно складывать числа или объединять строки. Можно осуществлять логическое сравнение или находить большее из двух чисел. Все операторы, поддерживаемые Netscape или Microsoft, представлены в табл. П4.5.

Таблица П4.5. Операторы

Элемент	Netscape		Microsoft	
	Nav4.5	Nav6	IE4	IE5
Сложение (+)	Да	Да	Да	Да
Увеличение на единицу (++}	Да	Да	Да	Да
Присвоение (=)	Да	Да	Да	Да
Побитное AND (&)	Да	Да	Да	Да
Побитное AND на единицу (&=)	Да	Да	Да	Да
Побитный сдвиг влево (<<)	Да	Да	Да	Да
Побитное NOT (~)	Да	Да	Да	Да
Побитное OR (|)	Да	Да	Да	Да
Побитное OR с присвоением результата левому операнду (|=)	Да	Да	Да	Да
Побитный сдвиг вправо (>>)	Да	Да	Да	Да
Побитное XOR (^)	Да	Да	Да	Да
Побитное XOR с присвоением результата левому операнду (^=)	Да	Да	Да	Да
Запятая (,)	Да	Да	Да	Да
Сравнения	Нет	Да	Да	Да
Сложные присвоения	Нет	Да	Да	Да

продолжение

Таблица П4.5 *(продолжение)*

Элемент	Netscape		Microsoft	
	Nav4.5	Nav6	IE4	IE5
Удаление (delete)	Да	Да	Да	Да
Вычитание единицы (--)	Да	Да	Да	Да
Деление (/)	Да	Да	Да	Да
Деление с присвоением результата левому операнду (/=)	Да	Да	Да	Да
Равно (==)	Да	Да	Да	Да
Больше (>)	Да	Да	Да	Да
Больше или равно (>=)	Да	Да	Да	Да
Оператор равенства (===)	Нет	Да	Да	Да
Оператор увеличения на единицу (++)	Да	Да	Да	Да
Оператор неравенства (!=)	Да	Да	Да	Да
Меньше (<)	Да	Да	Да	Да
Меньше или равно (<=)	Да	Да	Да	Да
Логическое AND (&&)	Да	Да	Да	Да
Логическое NOT (!)	Да	Да	Да	Да
Логическое OR (||)	Да	Да	Да	Да
Деление по модулю (%)	Да	Да	Да	Да
Деление по модулю с присвоением результата левому операнду (%=)	Да	Да	Да	Да
Умножение (*)	Да	Да	Да	Да
Умножение с присвоением результата левому операнду (*=)	Да	Да	Да	Да
Отрицание (-val)	Да	Да	Да	Да
Оператор new	Да	Да	Да	Да
Отрицание равенства (!==)	Нет	Да	Да	Да
Вычитание (-)	Да	Да	Да	Да
Вычитание с присвоением результата левому операнду (-=)	Да	Да	Да	Да
Оператор this	Да	Да	Да	Да
Оператор typeof	Да	Да	Да	Да
Унарный минус (-)	Да	Да	Да	Да
Сдвиг вправо без учета знака (>>>)	Да	Да	Да	Да
Оператор void	Да	Да	Да	Да

Наибольшую часть языка JavaScript составляют свойства. Представленные в табл. П4.6 свойства позволяют определить любой объект.

Таблица П4.6. Свойства

Элемент	Применяется к **объекту/коллекции/методу**	Netscape		Microsoft	
		Nav4.5	Nav6	IE4	IE5
Свойства $1...$9	RegExp	Да	Да	Да	Да
Аргументы	Function	Да	Да	Да	Да
AtEndOfLine	TextStream	Нет	Нет	Да	Да
AtEndOfStream	TextStream	Нет	Нет	Да	Да
Attributes	File, Folder	Нет	Нет	Да	Да

Элемент	Применяется к объекту/коллекции/методу	Netscape		Microsoft	
		Nav4.5	Nav6	IE4	IE5
AvailableSpace	Drive	Нет	Нет	Да	Да
caller	Function	Да	Да	Да	Да
Column	TextStream	Нет	Нет	Да	Да
CompareMode		Нет	Нет	Да	Да
Переменные условной	компиляции	Нет	Нет	Да	Да
constructor	Array, Boolean, Date, Function, Number, Object, String	Нет	Да	Да	Да
Count	Dictionary, Drives, Files, Folders,	Нет	Нет	Да	Да
DateCreated	File, Folder	Нет	Нет	Да	Да
DateLastAccessed	File, Folder	Нет	Нет	Да	Да
DateLastModified	File, Folder	Нет	Нет	Да	Да
Drive	File, Folder	Нет	Нет	Да	Да
DriveLetter	Drive	Нет	Нет	Да	Да
Drives	FileSystemObject	Нет	Нет	Да	Да
DriveType	Drive	Нет	Нет	Да	Да
E	Math	Да	Да	Да	Да
Files		Нет	Нет	Да	Да
FileSystem	Drive	Нет	Нет	Да	Да
FreeSpace	Drive	Нет	Нет	Да	Да
global	Regular Expression	Да	Да	Да	Да
ignoreCase	Regular Expression	Да	Да	Да	Да
index	RegExp	Да	Да	Да	Да
Infinity	Global	Да	Да	Да	Да
input	RegExp	Да	Да	Да	Да
IsReady	Drive	Нет	Нет	Да	Да
IsRootFolder	Folder	Нет	Нет	Да	Да
Item	Dictionary, Drives, Files, Folders,	Нет	Нет	Да	Да
Key	Dictionary	Нет	Нет	Да	Да
lastIndex	RegExp, Regular Expression	Да	Да	Да	Да
lastMatch	RegExp	Да	Да	Да	Да
lastParen	RegExp	Да	Да	Да	Да
leftContext	RegExp	Да	Да	Да	Да
length	Array, Function, String	Да	Да	Да	Да
Line	TextStream	Нет	Нет	Да	Да
LN2	Math	Да	Да	Да	Да
LN10	Math	Да	Да	Да	Да
LOG2E	Math	Да	Да	Да	Да
LOG10E	Math	Да	Да	Да	Да
MAX_VALUE	Number	Да	Да	Да	Да
MIN_VALUE	Number	Да	Да	Да	Да
multiline	RegExp	Да	Да	Да	Да
Name	File, Folder	Нет	Нет	Да	Да
NaN	Global, Number	Да	Да	Да	Да
NEGATIVE_INFINITY	Number	Да	Да	Да	Да
ParentFolder	File, Folder	Нет	Нет	Да	Да
Path	Drive, File, Folder	Нет	Нет	Да	Да
PI	Math	Нет	Да	Да	Да

продолжение

Таблица П4.6 (*продолжение*)

Элемент	Применяется к объекту/коллекции/методу	Netscape		Microsoft	
		Nav4.5	Nav6	IE4	IE5
POSITIVE_INFINITY	Number	Да	Да	Да	Да
prototype	Array, Boolean, Date, Function, Number, Object, String	Да	Да	Да	Да
rightContext	RegExp	Да	Да	Да	Да
RootFolder	Drive	Нет	Нет	Да	Да
SerialNumber	Drive	Нет	Нет	Да	Да
ShareName	Drive,	Нет	Нет	Да	Да
ShortName	File, Folder	Нет	Нет	Да	Да
ShortPath	File, Folder	Нет	Нет	Да	Да
Size	File, Folder	Нет	Нет	Да	Да
source	Regular Expression	Нет	Нет	Да	Да
SQRT1_2	Math	Да	Да	Да	Да
SQRT2	Math	Да	Да	Да	Да
SubFolders	Folder	Нет	Нет	Да	Да
TotalSize	Drive	Нет	Нет	Да	Да
Type	File	Нет	Нет	Да	Да
VolumeName	Drive	Нет	Нет	Да	Да

JavaScript поддерживает множество специальных операторов, обеспечивающих выполнение циклов, прерываний выполнения программы и т. д. Эти операторы представлены в табл. П4.7.

Таблица П4.7. Специальные операторы

Элемент	Netscape		Microsoft	
	Nav4.5	Nav6	IE4	IE5
break	Да	Да	Да	Да
@cc_on	Да	Да	Да	Да
Comment s	Да	Да	Да	Да
continue	Да	Да	Да	Да
do...while	Да	Да	Да	Да
for	Да	Да	Да	Да
for...in	Да	Да	Да	Да
function	Да	Да	Да	Да
@if	Да	Да	Да	Да
if...else	Да	Да	Да	Да
Labeled	Да	Да	Да	Да
return	Да	Да	Да	Да
@set	Да	Да	Да	Да
switch	Да	Да	Да	Да
this	Да	Да	Да	Да
throw	Нет	Да	Нет	Нет
try	Нет	Да	Нет	Нет
var	Да	Да	Да	Да
with	Да	Да	Да	Да
while	Да	Да	Да	Да

Алфавитный указатель

Х. Вильямсон

БИБЛИОТЕКА ПРОГРАММИСТА

Универсальный Dynamic HTML

Эта книга адресована всем, кто разрабатывает web-страницы назло препятствиям, заложенным в существующее программное обеспечение.

На ее страницах вы найдете всесторонний анализ программной поддержки, необходимой для размещения динамического сайта в Интернете.

Все web-разработчики знают, насколько сложно сделать страницу, одинаково выглядящую при четырех разрешениях монитора и во всех существующих браузерах.

С помощью автора вы освоите тонкости DHTML, и разработанный вами web-сайт будет радовать вас, его хозяина и посетителей.

- объектная модель документа
- реализация каскадных таблиц стиля
- обзор языков сценария
- определение настроек среды пользователя
- работа со слоями
- планирование и разработка общедоступного сайта
- реализация взаимодействия страницы с пользователем

и многое **другое...**

Уровень пользователя:
опытный/эксперт

Серия:
библиотека программиста

ISBN 5-318-00368-0