

## Л1. View елементи Android Basics

View це прямокутник на екрані, який відображає деякий зміст. Це може бути картинка, фрагмент тексту, кнопка або щось інше, що відображає мобільний додаток. Ці окремі View разом складають те що ми називаємо макет екрану Layout. Все що ви бачите та взаємодієте в вашому додатку називається призначеним для користувача інтерфейсом UI. Існує багато типів елементів View з різними назвами. View показує будь-якої текст називається TextView, View містить картинку називається ImageView, view з кнопкою називається buuton.



**TextView**

Morning Pizza

I'll be in your neighborhood doing errands this weekend if you want to meet up for brunch. Lemme know.

Show notifications

**ImageView**

**Button**

+

BUTTON

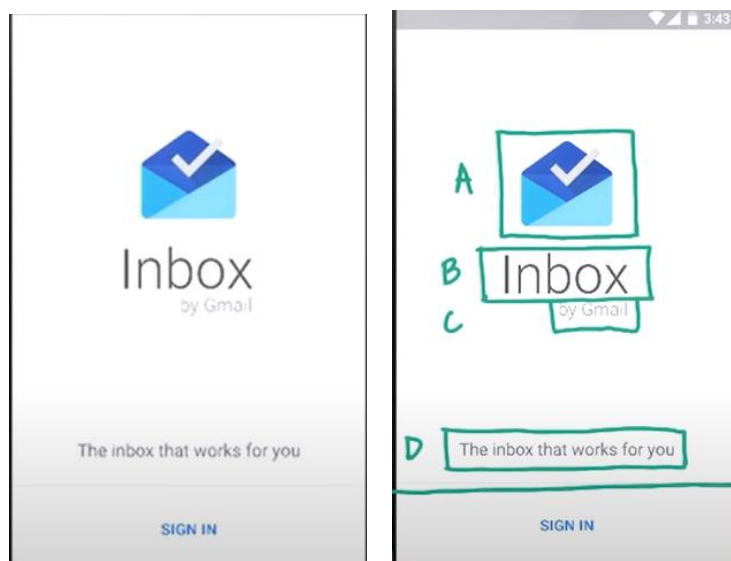
BUTTON

Кожен view це прямокутник на екрані, але де ж все прямокутники? Технічно вони невидимі але можна їх намалювати щоб побачити де знаходяться межі кожного прямокутника.



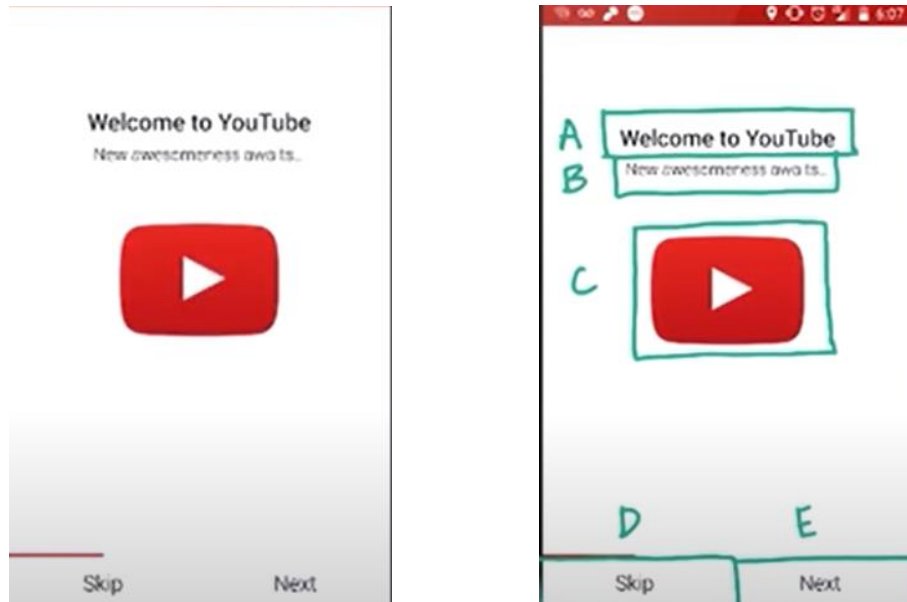
Кожний додаток може бути розділений на складові його окремі View. Іншими словами View це базовий блок який використовується для створення макета мобільного додатку. Форматом назв View є стиль CamelCase (між словами немає пропусків і перша буква кожного слова є заголовної).

Розглянемо додаток від Google Inbox. Ви зможете визначити які різні типи view тут є.



	TextView	ImageView	Button
A			
B			
C			
D			
E			

Скріншот додатку YouTube при першому входженні до нього



	TextView	ImageView	Button
A			
B			
C			
D			
E			

## XML

Xml це розширювана мова розмітки, яка описує як буде виглядати додаток. Її можна використовувати для побудови макета, тобто як будуть розміщуватись елементи, які елементи будуть на екрані.

Якщо написати часть коду, то текст буде виглядати так:

```
<TextView
  android:text="Happy Birthday!"
  android:background="@android:color/darker_gray"
  android:layout_width="150dp"
  android:layout_height="75dp" />
```



Синтаксис це правило яке визначає правильно XML. Якщо ви напишіть невірно xml, то він навіть не буде відображений на телефоні. Це код для елемента xml під назвою TextView.

```
<TextView
    android:text="Happy Birthday!"
    android:background="@android:color/darker_gray"
    android:layout_width="150dp"
    android:layout_height="75dp" />
```

XML завжди починаємо з кутової дужки, що відкривається і назви. Це може бути TextView, ImageView, Button або щось інше. Далі йде список атрибутів по одному на рядок, прямий слеш і кутова дужка, що самозакривається.

```
<TextView
    android:text="Happy Birthday!"
    android:background="@android:color/darker_gray"
    android:layout_width="150dp"
    android:layout_height="75dp">
</TextView>
```

Замість тегу, що самозакривається можна окремо поставити тег, що відкриває та закриває. Він містить дужку, що відкриває, слеш, назву елемента розкладки, закриваючий тег. Приклад де це використовується.

```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:text="Happy Birthday"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:text="You're the best!"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

Всередині є 2 елемента TextView. Вони називаються молодшим елементами які знаходяться всередині старшого елемента LinearLayout.

Ви помітили що є багато додаткового тексту всередині тега.

```
<TextView
    android:text="Happy Birthday!"
    android:background="@android:color/darker_gray"
    android:layout_width="150dp"
    android:layout_height="75dp">
</TextView>
```

**Атрибути** це характеристики які визначають способи відображення на Android телефоні. Можна помітити, що назва атрибута з лівого боку, а праворуч стоїть значення. Важливо знати, що значення атрибутів треба запити в лапках. Це частина синтаксису xml.

```
<TextView  
  android:text="Happy Birthday!"  
  android:background="@android:color/darker_gray"  
  android:layout_width="150dp"  
  android:layout_height="75dp" />
```

↑  
Attribute  
Name

↑  
Attribute  
Value in  
"Quotations"

У всіх атрибутів є значення за замовчуванням

*TxtView*  
*Textview*  
*text view* X

```
<TextView  
  android:text="Happy Birthday!"  
  android:background="@android:color/darker_gray"  
  android:layout_width="150dp"  
  android:layout_height="75dp" />
```

Назва View елемента має бути записана в стилі CamelCase. Назви атрибутів починаються зі слова android з двокрапкою, далі знак рівності та значення потрібно взяти у лапки.

## Android studio. XML візуалізатор

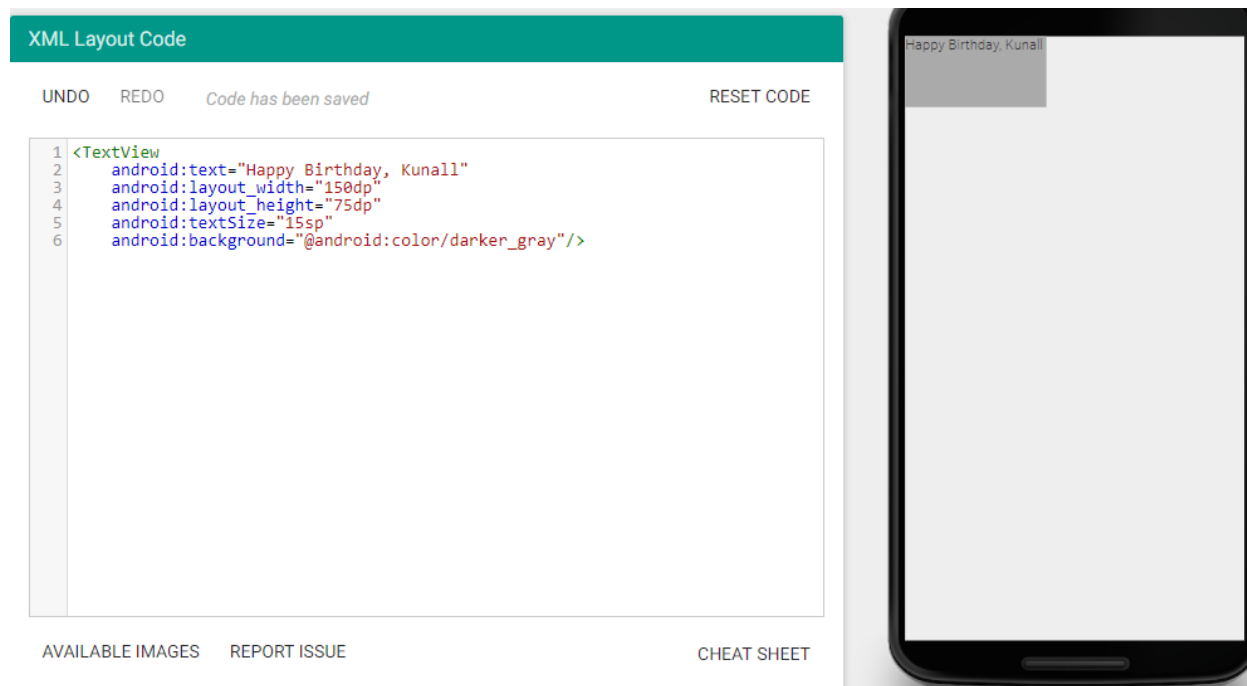
Вікно Android studio з XML кодом розмітки представлено на рис.



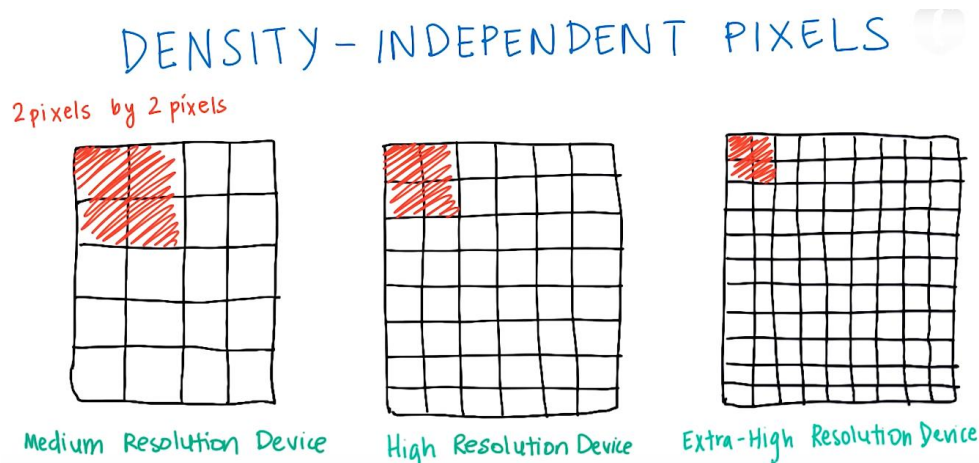
XML візуалізатор – це щось на зразок Android Studio. Дозволяє за кодом XML



бачити відразу результат але в ньому немає всього функціоналу Android Studio (<https://labs.udacity.com/android-visualizer/#/android/text-v>)

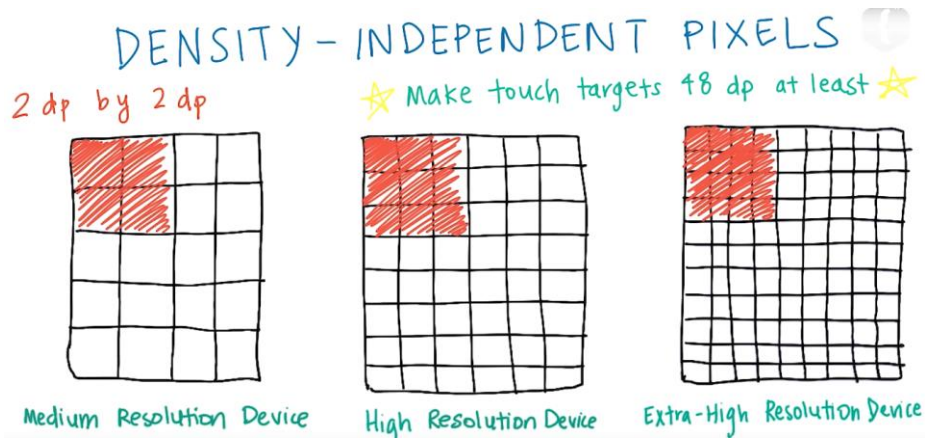


Атрибут `text` визначає текст, який буде виводитись. Спробуємо поміняти `android:layout_width="150dp"`, `android:layout_height="75dp"` на `300 dp` та `500 dp`.



Одиниця вимірювання `dp`, як сантиметр або міліметр, з допомогою якої визначається розмір `View` елемента в `android`, а також відстань між `View` елементами.

Рекомендується робити усі кнопки розміром мінімум `48 dp` у висоту та ширину (`48dp` — це приблизно `9 мм` фізичного розміру), а не `48 пікселів`, інакше кнопка буде дуже маленькою на пристрої з дуже високою роздільною здатністю, так і всім буде легко користуватися вашим додатком.

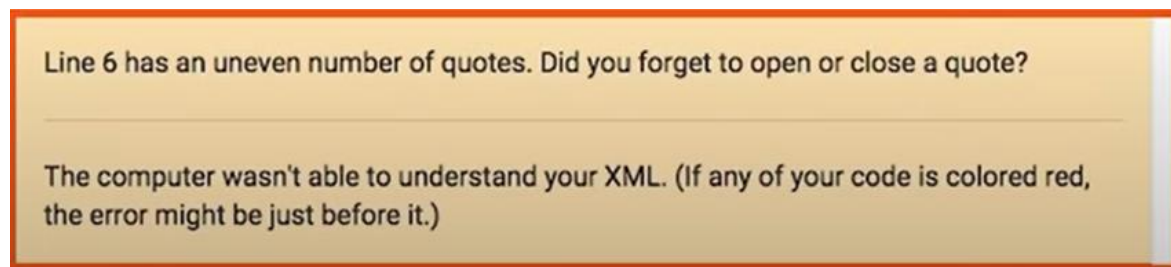


Помилки в XML підкреслені червоним або виділені червоним текстом. Читаємо повідомлення про помилку та номер рядка.

```

<TextView
  android:layout_wid="100dp"
  android:layout_height="50dp"
  android:background="@android:color/darker_gray|
  android:text="@string/hello_world" />

```



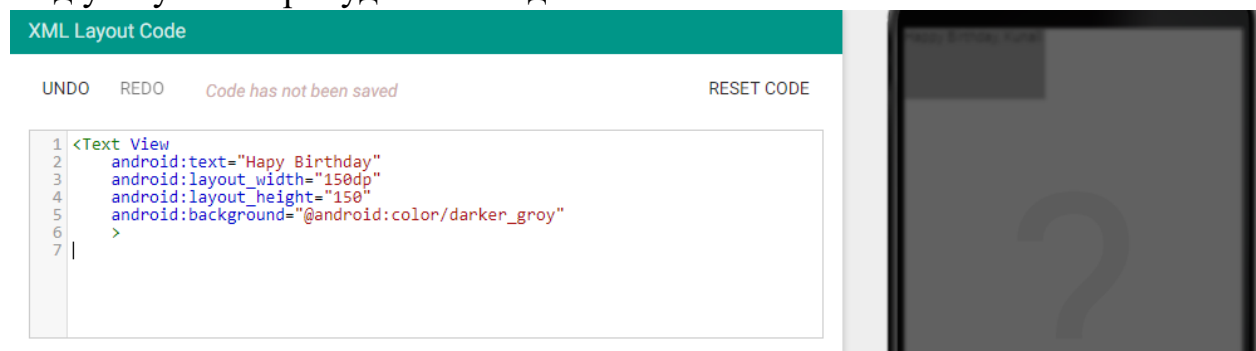
Знайти та виправимо помилки в коді

```

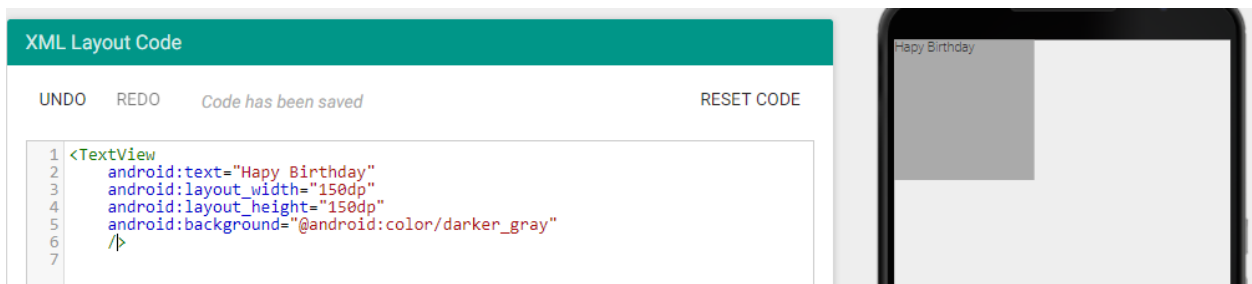
<Text View
  android:text="Hapy Birthday"
  android:layout_width="150dp"
  android:layout_height="150"
  android:background="@android:color/darker_groy"
>

```

Код у візуалізаторі буде мати вид



Після виправлення помилок коду макет додатку буде мати вигляд



### Параметр Wrap\_content



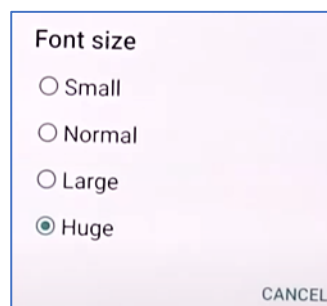
Wrap\_content вказує, що розмір елемента TextView підлаштовується під зміст тексту.

### Параметр textSize

Даний параметр задає розмір шрифту в текстовому полі



Параметр тексту у цьому фрагменті коду встановлений на 45 sp. Це одиниця виміру схожа на dp. З її допомогою додатки виглядають однаково на різних пристроях з різним дозволом.



Використовується тільки для шрифтів.



## Специфікації material design



[Material design](#) - концепція дизайну розроблена google щоб допомогти розробникам створювати узгоджені і красиві додатки для різних пристроїв і платформ, інтернет і мобільних елефонів. Специфікація відображений кращий досвід дизайну та представляє керівництво по макету для вашого додатку, стилю, зображенням, анімації і багато чому іншому. У [довіднику](#) вказані рекомендації до розмірів тексту.

### Колір шрифту в текстовому полі

Змінемо колір фону на чорний, білий, а потім спробуйте встановити синій. Отримаємо помилку, тому що синій колір не підтримується. В Android обмежена кількість кольорів але інші кольори можна вказати у [шістнадцятковому представленні](#) (hex-кодом). [Рекомендації material design з палітри кольорів](#).

## ImageView

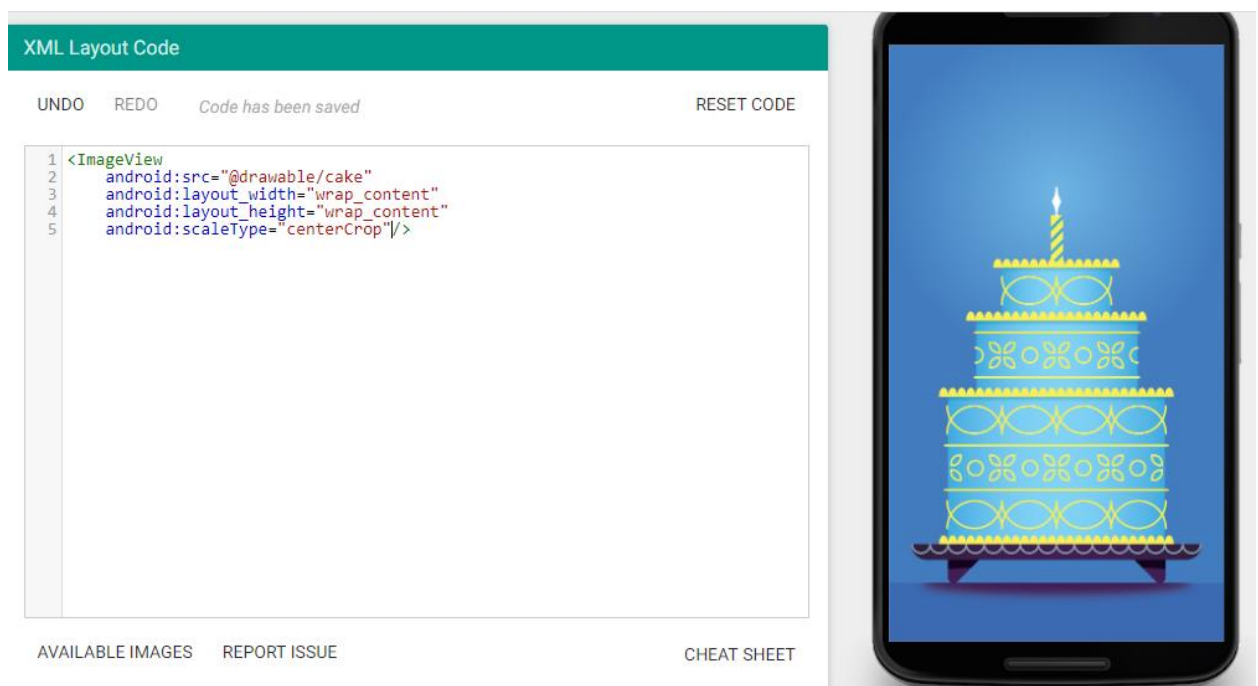
Розглянемо елемент xml ImageView. Цей ImageView буде показувати на екрані зображення.

XML Layout Code

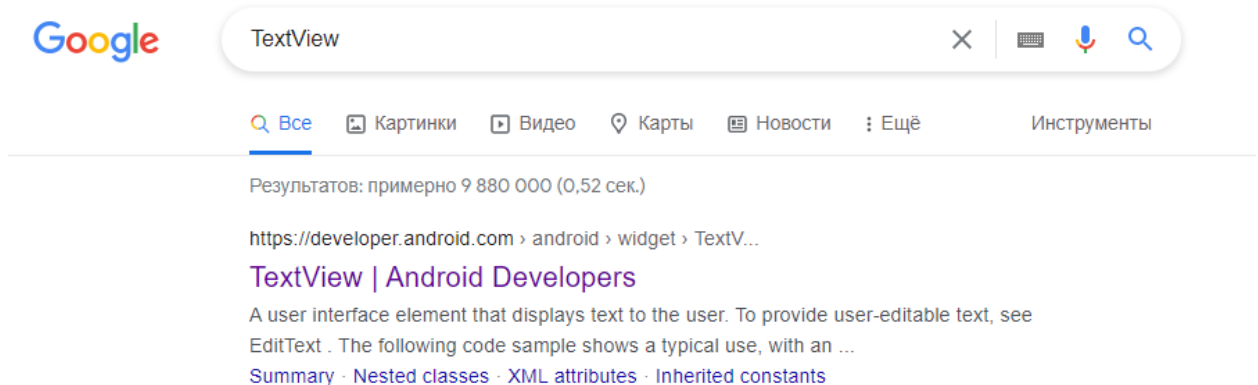
UNDO REDO Code has been saved RESET CODE

```
1 <ImageView
2   android:src="@drawable/cake"
3   android:layout_width="wrap_content"
4   android:layout_height="wrap_content"
5   android:scaleType="center"/>
```

Властивість `android:src = "@drawable/cake"` вказує в якому файлі треба шукати джерело зображення. Символ `@` використовується щоб вказати що посилаємося на ресурс в додатку Android, а `drawable` цей тип ресурсу. Це графіка яка буде показано на екрані в Android. `Cake` це назва файлу із зображенням, розширення файлу не вказується. Також у `xml` є властивість висоти та ширини для упакованого вмісту, тому представлення зображення буде мати розмір картини `Cake`. Властивість `android:scaleType="center"` вказує пристрою як масштабувати зображення. Властивість не змінює розмір картини, а тільки розташовує по центру, тому бачимо центральну частину торта. Властивість `android:scaleType="centerCrop"` масштабує зображення в межах розміру екрану (обрізає) пристрою та розташовує по центру. Таке зображення називають – картинка без полей.



## Документація по TextView



developers Platform Android Studio Google Play Jetpack Docs Більше Search English

DOCUMENTATION

Overview Guides **Reference** Samples Design & Quality

Android API Reference Overview

Android Platform Packages API level REL Class Index Package Index android android.accessibilityservice android.accounts android.animation

Android Developers > Docs > Reference Was this helpful? Added in API level 1

# TextView

public class TextView extends View implements ViewTreeObserver.OnPreDrawListener

java.lang.Object
 

- android.view.View
  - android.widget.TextView

 Known direct subclasses

Summary
 

- Nested classes
- XML attributes
- Inherited XML attributes
- Constants
- Inherited constants
- Inherited fields
- Public

developers Platform Android Studio Google Play Jetpack Docs Більше Search

Android API Reference Overview

Android Platform Packages API level REL Class Index Package Index android android.accessibilityservice android.accounts android.animation android.annotation android.app android.app.admin

<code>android:textColorLink</code>	Reference to a drawable that will be drawn under the insertion cursor.
<code>android:textCursorDrawable</code>	Reference to a drawable that will be drawn under the insertion cursor.
<code>android:textFontWeight</code>	Weight for the font used in the TextView.
<code>android:textIsSelectable</code>	Indicates that the content of a non-editable text can be selected.
<code>android:textScaleX</code>	Sets the horizontal scaling factor for the text.
<code>android:textSelectHandle</code>	Reference to a drawable that will be used to display a text selection anchor for positioning the cursor within text.
<code>android:textSelectHandleLeft</code>	Reference to a drawable that will be used to display a text selection anchor on the left side of a selection region.
<code>android:textSelectHandleRight</code>	Reference to a drawable that will be used to display a text selection anchor on the right side of a selection region.
<code>android:textSize</code>	Size of the text.

Пошукаємо атрибут, який робить усі букви заглавними

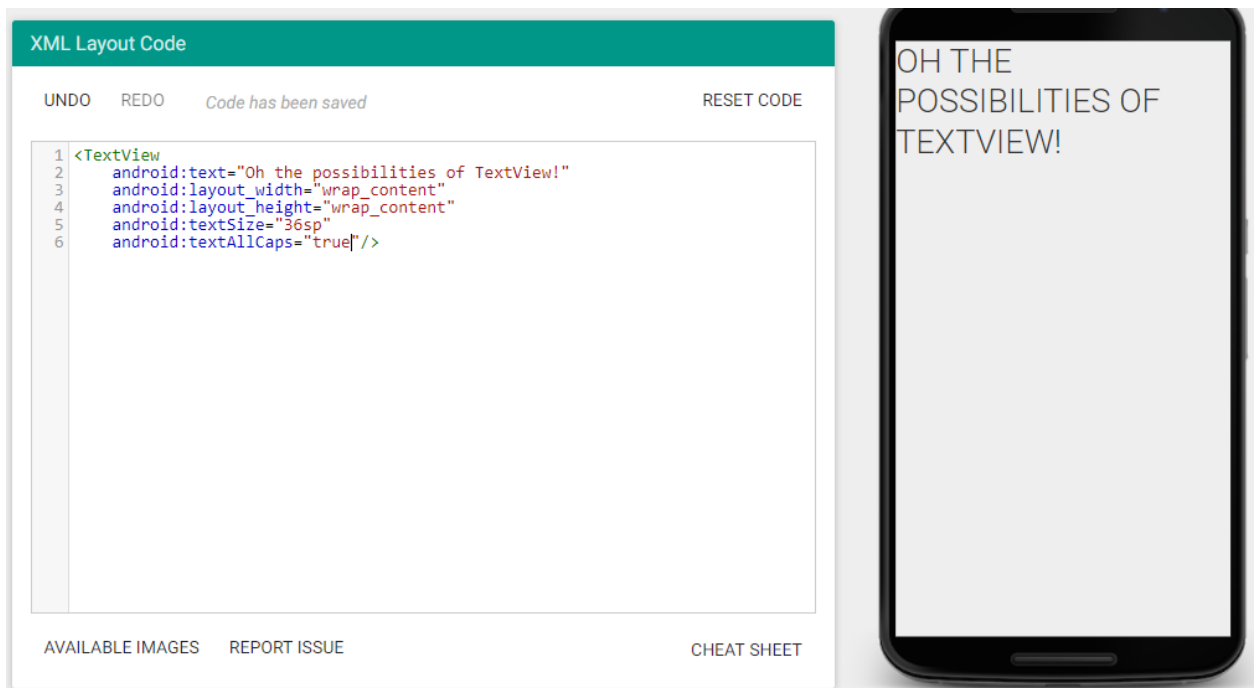
Входящие http://ai2.appinvent... Онлайн переводч... Replit - PossibleSc... Online C++ caps 5 з 28

developers Platform Android Studio Google Play Jetpack Docs Більше

Android API Reference Overview

Android Platform Packages API level REL Class Index Package Index android android.accessibilityservice

<code>android:shadowRadius</code>	Blur radius of the text shadow.
<code>android:singleLine</code>	Constrains the text to a single horizontally letting it wrap onto multiple lines, and advancing to a new line when you press the enter key.
<code>android:text</code>	Text to display.
<code>android:textAllCaps</code>	Present the text in ALL CAPS.
<code>android:textAppearance</code>	Base text color, typeface, size, and style.
<code>android:textColor</code>	Text color.



1. XML візуалізатор. URL: <https://labs.udacity.com/android-visualizer>
2. Шпаргалка з найбільш поширених View-елементів. URL: <https://cutt.ly/wEkm3KO>
3. Material design typography. URL: <https://cutt.ly/QEkQjF>
4. Material design color. URL: <https://material.io/design/color/the-color-system.html>
5. Шістнадцяткова таблиця кольорів. URL: [https://www.w3schools.com/colors/colors\\_hex.asp](https://www.w3schools.com/colors/colors_hex.asp)
6. Material design. Color Tool. URL: <https://cutt.ly/WEkQ2E2>
7. Документація по TextView. URL: <https://developer.android.com/reference/android/widget/TextView>