



## Лекция #12. Текстовые поля

Текстовые поля в **Android** представлены двумя классами:

- `TextView`;
- `EditText`.

Виджет **TextView** предназначен для отображения текста без возможности редактирования его пользователем. Виджет поддерживает многострочное отображение, форматирование и автоматический перенос слов и символов.

Если необходимо редактирование текста, используется виджет **EditText**. Классы **TextView** и **EditText** имеют множество атрибутов и методов, наследуемых от класса **View**, который рассмотрим позже.

У элемента **TextView** есть многочисленные методы и XML-атрибуты для работы с текстом. Виджет **TextView** записывается с помощью xml-файла следующим образом:

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"

    далее другие атрибуты

/>
```

здесь:

- **android:text** - здесь указывается текст
- **android:layout\_width** – ширина текста
- **android:layout\_height** – высота текста
- **android:id** – уникальный идентификатор виджета **TextView**.

Знак **+** в записи **android:id="@+id/textView1"** означает, что если для элемента **TextView** не создан **id** со значением **textView1**, то его следует определить.

Вот основные XML-атрибуты, отображающие свойства элемента **TextView**:

1. Для задания текста в файле разметки используется атрибут **android:text**
2. Размер текста

При установке размера текста используются несколько единиц измерения:

- **px (pixels)** — пиксели;
- **dp (density-independent pixels)** — независимые от плотности пиксели.  
Это абстрактная единица измерения, основанная на физической плотности экрана;
- **sp (scale-independent pixels)** — независимые от масштабирования пиксели;
- **in (inches)** — дюймы, базируются на физических размерах экрана;
- **pt (points)** — 1/72 дюйма, базируются на физических размерах экрана;
- **mm (millimeters)** — миллиметры, также базируются на физических размерах экрана.

#### Рекомендации

- при установке размера текста используются единицы измерения **sp**, которые наиболее корректно отображают шрифты, например:  
**android:textSize="48sp"**
- во всем остальном используются единицы измерения **dp**  
**android:layout\_width="50dp"**

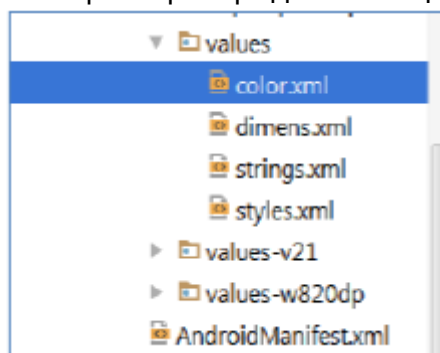
3. Стиль текста **android:textstyle**

Используются константы:

- normal;
- bold;
- italic.

4. Цвет текста **android:textcolor**

Вот пример определения цвета в XML файле.



В папке values создаем файл color.xml

В файле **color.xml** используем тег **<color> </color>** записываем параметры, определяющие различные цвета: красный, зеленый, голубой, черный, белый, желтый, зеленый.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="red">#f00</color>
  <color name="green">#0f0</color>
  <color name="blue">#00f</color>
  <color name="back">#000</color>
  <color name="white">#fff</color>
  <color name="yellow">#ffff00</color>
  <color name="_green">#ff00ff00</color>
</resources>
```

Задать текст для **TextView** можно в трех местах:

1. в программном коде
2. непосредственно в XML файле
3. в строковых ресурсах – рекомендуемый способ

### Задание текста программном коде

В файле разметки должен быть размещен элемент **TextView**, например, с **id=@+id/textView1**, далее в программном коде текст задается методом **setText()**:

```
val text: TextView = findViewById(R.id.textView);
text.setText("Hello world");
```

### Задание текста в строковых ресурсах

Файл, содержащий строковые ресурсы, по умолчанию имеет имя **string.xml**, находится в паке **res/values/string.xml**. Вот его содержание.

```
<resources>
  <string name="app_name">MyTextView</string>
</resources>
```

Файл состоит из строк, каждая строка использует тег **<string>**, в котором каждой строке приваривается уникальное имя, с помощью которого можно получать доступ к текстовой строке.

Для задания нового строкового ресурса необходимо задать уникальное имя ресурса и строку, используя тег **<string>**. Например, зададим текст **"Hello, Android!"** и имя ресурса **name="hello\_android"**.

```
<resources>
  <string name="app_name">MyTextView</string>
  <string name="hello_android"> Hello, Android!</string>
</resources>
```

Далее, в файле разметки **activity\_main.xml**, указываем ссылку на строковый ресурс

```
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    android:text="@string/hello_android"
```

---