В этом уроке вы узнаете о двух новых View — **TextView** и **Button**, а также научитесь базово взаимодействовать с View из Java-кода.

TextView

TextView отвечает за отображение текста.

Измените код activity\_main.xml, чтобы он выглядел следующим образом:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center"

android:orientation="vertical"

tools:context="com.test.lessons.MainActivity">

<TextView

android:id="@+id/hello\_tv"

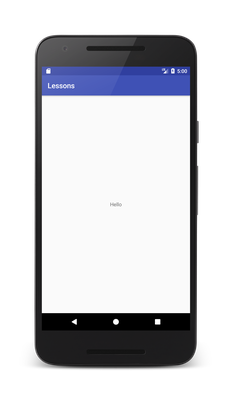
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello" />

</LinearLayout>

Запустите приложение:

[](https://skillberg.com/media/uploads/2017/11/17/eab986eeec674bb0a3325f59cce95d1d.png)TextView

Кстати, у каждого View есть атрибут gravity. Он работает так же, как и layout\_gravity, но изменяет не положение View относительно контейнера, а положение контента внутри этого View.

В данном случае, как вы могли заметить, я поместил TextView по центру, задав атрибут gravity у LinearLayout.

Вернемся к TextView. Первое, на что стоит обратить внимание — это значения в layout\_width и layout\_height. Там выставлен wrap\_content, таким образом, TextView занимает ровно столько места, сколько занимает текст внутри него.

Атрибут android:text

Самый важный атрибут TextView — это android:text. Как нетрудно догадаться, с помощью него задается текст внутри TextView.

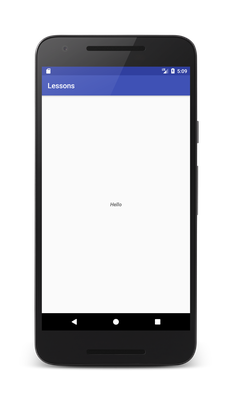
Его можно задавать просто текстом, а можно ссылкой на строковый ресурс (об этом попозже).

Атрибут android:textStyle

Это атрибут задает стиль текста и принимает следующие значения:

* normal — обычный текст. Это — значение по умолчанию.
* bold — жирный текст
* italic — курсив

bold и italic можно использовать по отдельности, а можно и комбинировать:

[](https://skillberg.com/media/uploads/2017/11/17/fd70c9163ca743b5b6f2390af7bf521d.png)Атрибут android:textStyle у TextView

Атрибут android:textColor

Этим атрибутом задается цвет текста. Может принимать следующие значения:

* Ссылка на ресурс
* Значение в формате #RGB, #ARGB, #AARRGGBB, #RRGGBB

Например, давайте сделаем текст красным:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center"

android:orientation="vertical"

tools:context="com.test.lessons.MainActivity">

<TextView

android:id="@+id/hello\_tv"

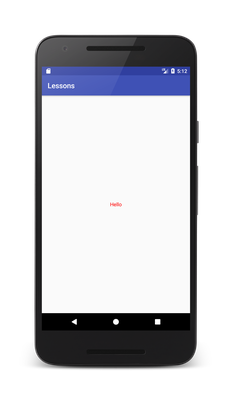
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello"

android:textColor="#ff0000" />

</LinearLayout>

[](https://skillberg.com/media/uploads/2017/11/17/490652193e40438b86d434b72487f07b.png)Атрибут android:textColor у TextView

Атрибут android:textSize

Задает размер текста. Задается в sp. Что это такое — вы узнаете в одном из следующих уроков. Давайте увеличим текст:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center"

android:orientation="vertical"

tools:context="com.test.lessons.MainActivity">

<TextView

android:id="@+id/hello\_tv"

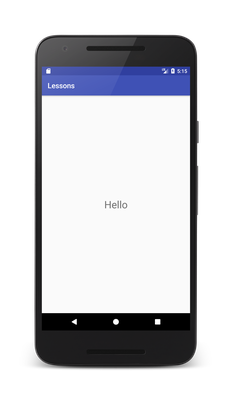
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello"

android:textSize="28sp" />

</LinearLayout>

[](https://skillberg.com/media/uploads/2017/11/17/3eaba2732e654bccb6c84006c9d42823.png)Атрибут android:textSize у TextView

А теперь поговорим о кнопках.

Button

За кнопки в Android отвечает, как ни странно, Button.

На самом деле, Button — это наследник TextView с некоторыми изменениями, и поэтому она обладает всеми теми же атрибутами, про которые мы говорили чуть выше.

Давайте добавим под TextView кнопку:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center"

android:orientation="vertical"

tools:context="com.test.lessons.MainActivity">

<TextView

android:id="@+id/hello\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello"

android:textSize="28sp" />

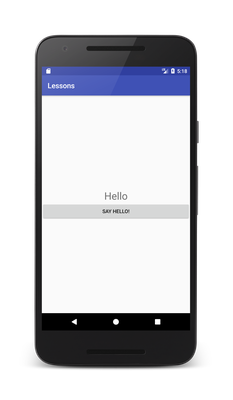
<Button

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Say hello!" />

</LinearLayout>

[](https://skillberg.com/media/uploads/2017/11/17/98f6b78499c946d6b42611891040cb50.png)Кнопка в Android

Кнопка отображается и даже визуально откликается на нажатия, но ничего не происходит. А все потому что нам нужно заставить её выполнять какие-то действия.

Обработка нажатия

Давайте сформулируем задачу: при нажатии на кнопку текст в TextView должен измениться на "Hi!".

Соответственно, что надо сделать? Надо каким-то образом получить событие нажатия на кнопку, и установить текст в TextView.

Определенно, делать это надо при помощи Java-кода.

В Java для обработки каких-либо событий как правило используют интерфейсы или (очень редко) абстрактные классы. Android — не исключение. Мы хотим обработать событие нажатия, и в этом нам поможет интерфейс OnClickListener.

Для начала нам нужно его реализовать в MainActivity. Для этого можно использовать два способа:

* Реализовать интерфейс в нашей Activity.
* Создать объект, реализующий интерфейс.

Пример первого способа:

package com.test.lessons;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

@Override

public void onClick(View view) {

}

}

Однако, лично мне такой способ кажется не очень хорошим, и я предпочту создать отдельный объект:

package com.test.lessons;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

private final View.OnClickListener onClickListener = new View.OnClickListener() {

@Override

public void onClick(View view) {

}

};

}

**Обратите внимание**: помимо View.OnClickListener существует так жеDialogInterface.OnClickListener. Он нам не подходит — нужен именно View.OnClickListener. Новички допускают эту ошибку очень часто!

Значит, обработчик мы сделали. Когда произойдет клик, вызовется метод onClick(). Однако, работать он, само собой не будет, потому что мы его никуда не присвоили.

Использование findViewById

Помните, я рассказывал про атрибут android:id? Сейчас он пригодится нам еще раз.

В разметке присвоим идентификаторы TextView и Button:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center"

android:orientation="vertical"

tools:context="com.test.lessons.MainActivity">

<TextView

android:id="@+id/hello\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello"

android:textSize="28sp" />

<Button

android:id="@+id/hello\_btn"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Say hello!" />

</LinearLayout>

Теперь мы можем обращаться к ним из кода.

Открываем MainActivity. В классе находим метод onCreate(). Он вызывается при создании Activity.

В нем есть строчка setContentView(R.layout.activity\_main); — в ней мы говорим, что нужно использовать именно этот файл разметки. Сразу после вызова этого метода мы уже можем обращаться к View, которые находятся в Activity.

Для того, чтобы получить View из разметки для взаимодействия в Java-коде, используется метод findViewById(). Давайте найдем оба наших View:

package com.test.lessons;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

TextView textView = findViewById(R.id.hello\_tv);

Button button = findViewById(R.id.hello\_btn);

}

private final View.OnClickListener onClickListener = new View.OnClickListener() {

@Override

public void onClick(View view) {

}

};

}

Отлично, теперь у нас есть оба View, и нужно установить onClickListener на кнопку, чтобы она вызывала Listener при каждом клике.

Сделать это можно методом setOnClickListener (он, кстати, есть у любых View, а не только у кнопок):

package com.test.lessons;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

TextView textView = findViewById(R.id.hello\_tv);

Button button = findViewById(R.id.hello\_btn);

button.setOnClickListener(onClickListener);

}

private final View.OnClickListener onClickListener = new View.OnClickListener() {

@Override

public void onClick(View view) {

}

};

}

Остался последний шаг: установить текст в textView.

Из Java мы можем установить текст вызовом метода setText("text"), однако, если мы так и напишем в onClick(), то получим ошибку — ведь textView существует только в пределах метода onCreate()!

Давайте сделаем его членом класса, чтобы он был доступен, и допишем наконец код:

package com.test.lessons;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

private TextView textView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

textView = findViewById(R.id.hello\_tv);

Button button = findViewById(R.id.hello\_btn);

button.setOnClickListener(onClickListener);

}

private final View.OnClickListener onClickListener = new View.OnClickListener() {

@Override

public void onClick(View view) {

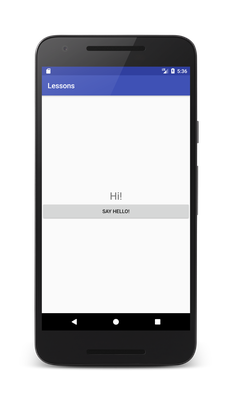
textView.setText("Hi!");

}

};

}

Запускаем, нажимаем на кнопку:

[](https://skillberg.com/media/uploads/2017/11/17/e904eee7ce4c405e9cf36e5b85ffa3bf.png)OnClickListener на кнопке

Мы справились!

А в следующем уроке мы научимся получать информацию из поля для ввода текста.

ПРИМЕРЫ КОДА

[ИСХОДНЫЙ КОД УРОКА](https://github.com/Skillberg/LessonsAndroid/tree/lesson7)