



Android

Sutula Oleksander

Why Android

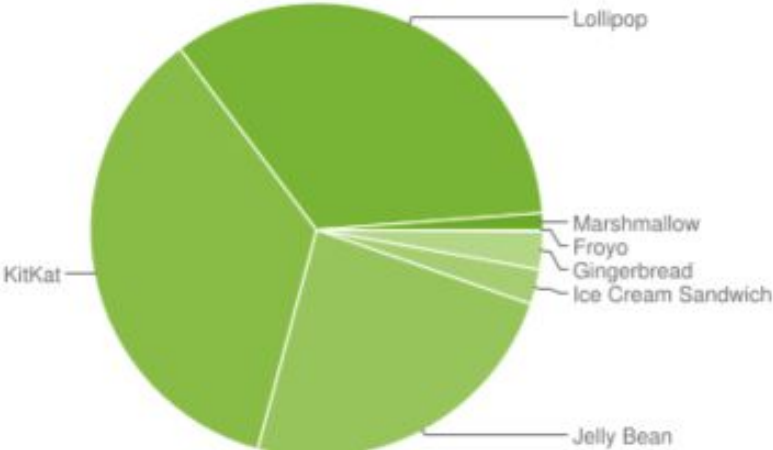
Because 82.8% devices use Android



Period	Android	iOS	Windows Phone	BlackBerry OS	Others
2015Q2	82.8%	13.9%	2.6%	0.3%	0.4%
2014Q2	84.8%	11.6%	2.5%	0.5%	0.7%
2013Q2	79.8%	12.9%	3.4%	2.8%	1.2%
2012Q2	69.3%	16.6%	3.1%	4.9%	6.1%

Source: IDC, Aug 2015

Platform versions



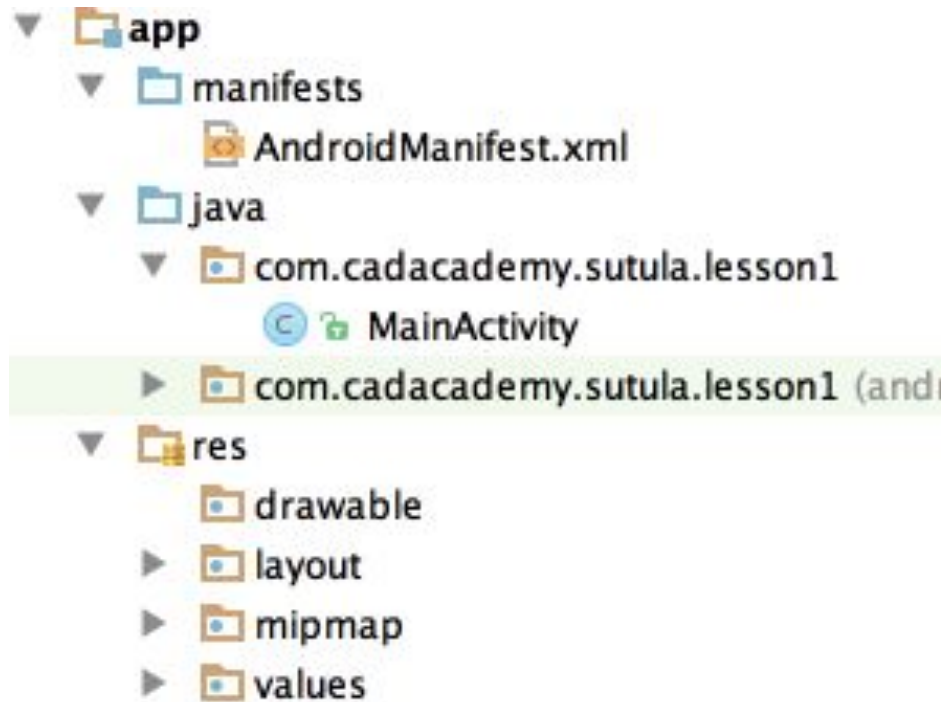
Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.5%
4.1.x	Jelly Bean	16	8.8%
4.2.x		17	11.7%
4.3		18	3.4%
4.4	KitKat	19	35.5%
5.0	Lollipop	21	17.0%
5.1		22	17.1%
6.0	Marshmallow	23	1.2%



Screen sizes

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	2.4%						2.4%
Normal		5.1%	0.1%	41.5%	22.9%	14.8%	84.4%
Large	0.3%	5.0%	2.3%	0.6%	0.5%		8.7%
Xlarge		3.5%		0.3%	0.7%		4.5%
Total	2.7%	13.6%	2.4%	42.4%	24.1%	14.8%	

Структура проекта

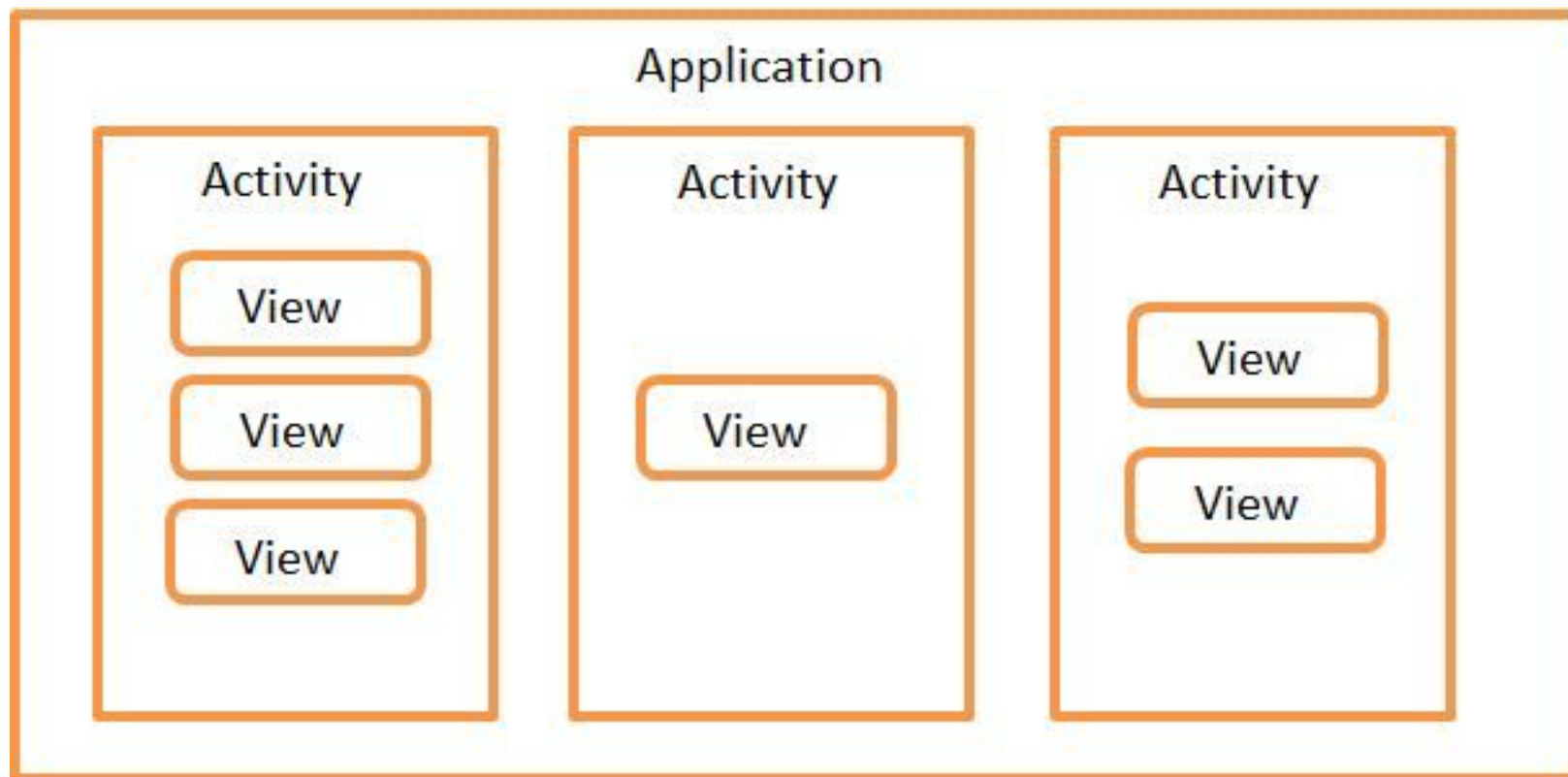


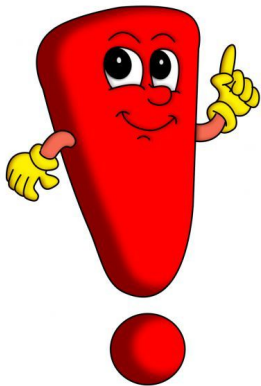
Файл **AndroidManifest.xml** – манифест или конфиг-файл приложения

В папке **java** и ее подпапках будет весь, написанный нами, код приложения

Папка **res** используется для файлов-ресурсов различного типа.

Приложение состоит из окон, называемых Activity. В конкретный момент времени обычно отображается одно Activity и занимает весь экран, а приложение переключается между ними. В качестве примера можно рассмотреть почтовое приложение. В нем одно Activity – список писем, другое – просмотр письма, третье – настройки ящика. При работе вы перемещаетесь по ним.





ACHTUNG

Содержимое Activity формируется из различных компонентов, называемых View. Самые распространенные View - это кнопка, поле ввода, чекбокс и т.д.

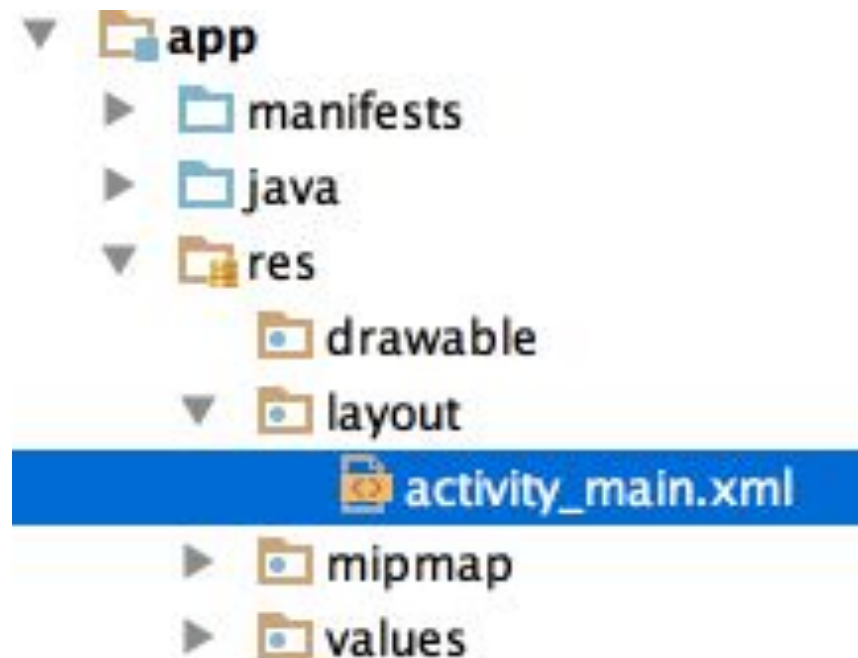
Необходимо заметить, что View обычно размещаются в ViewGroup. Самый распространенный пример ViewGroup – это Layout. Layout бывает различных типов и отвечает за то, как будут расположены его дочерние View (таблицей, строкой, столбцом ...)

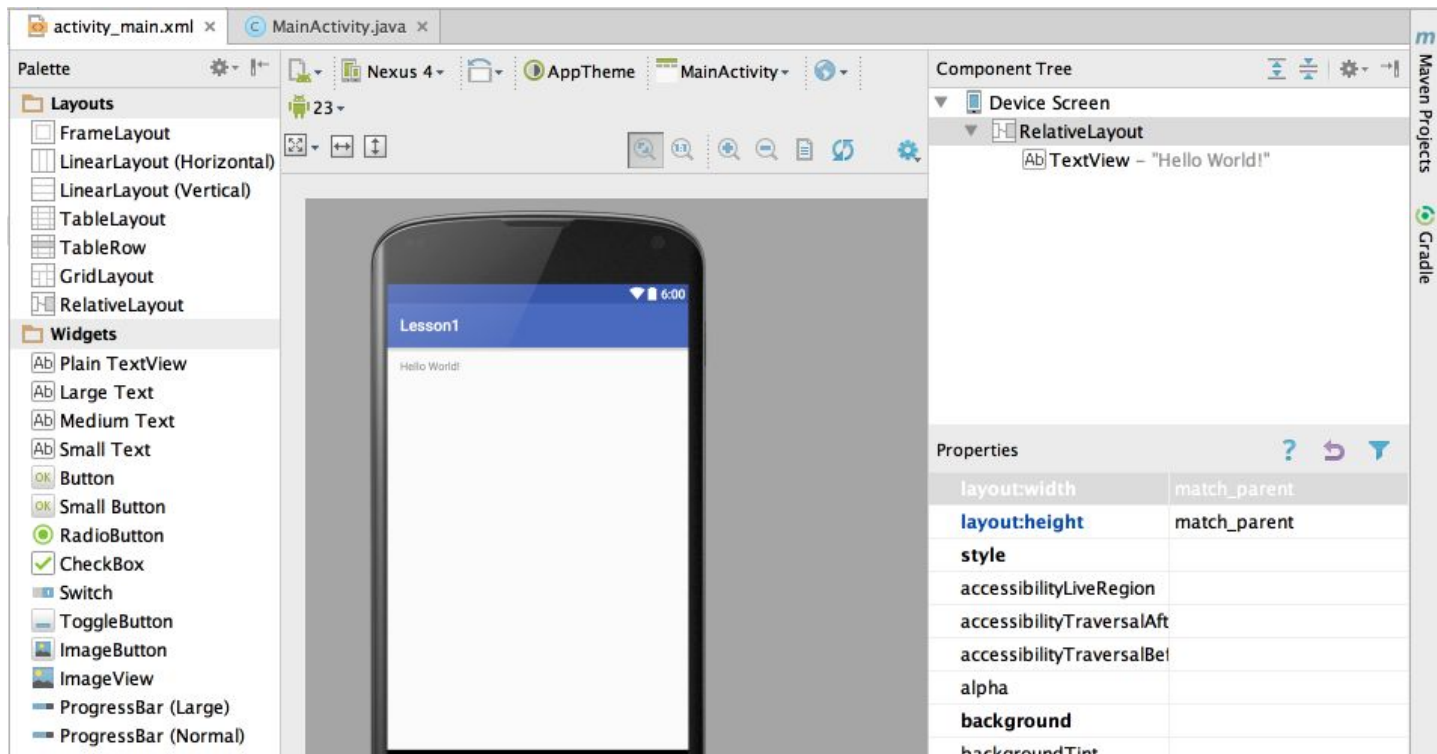


Это layout-файл.

В нем мы определяем набор и расположение элементов View, которые хотим видеть на экране.

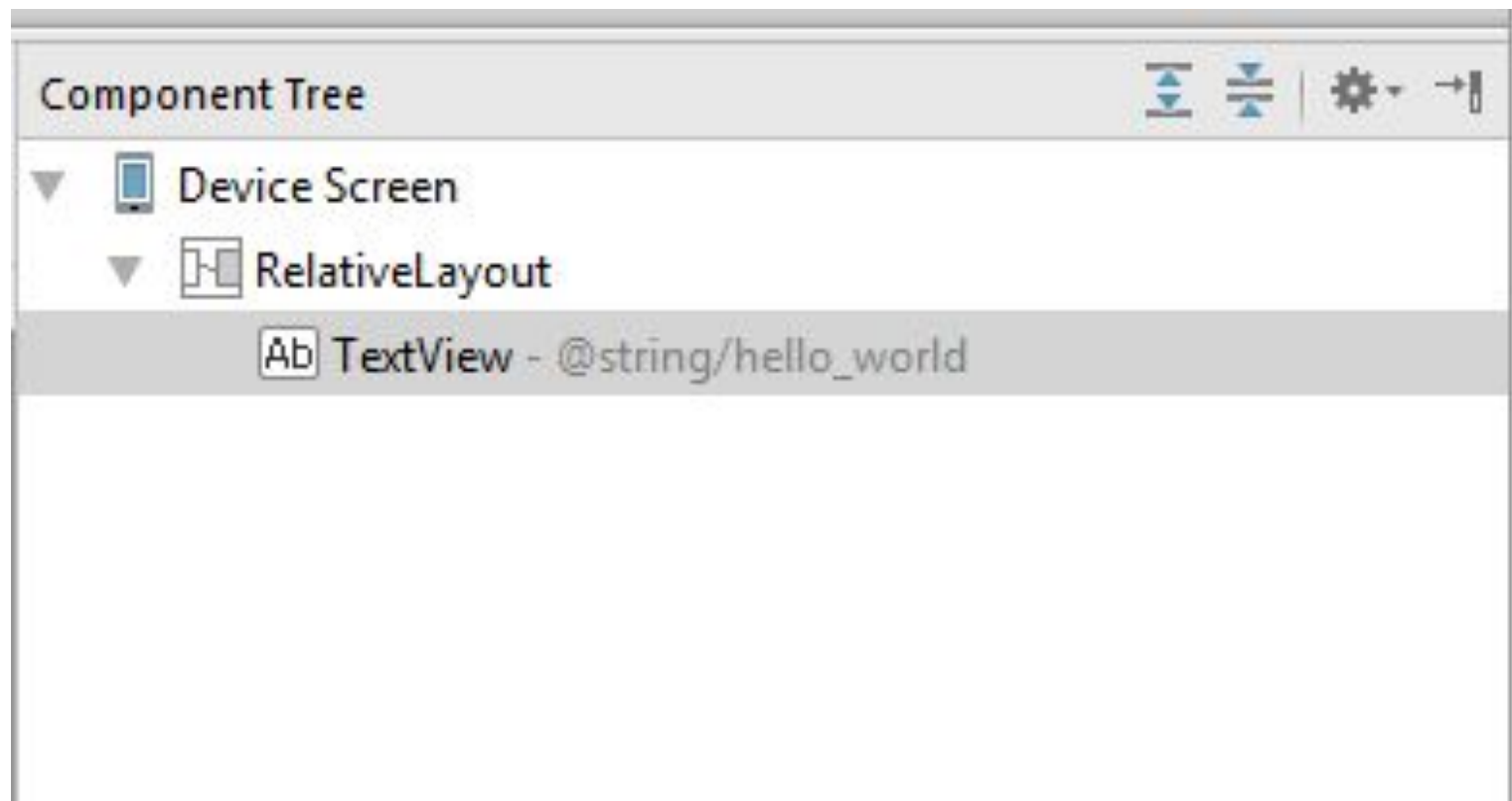
При запуске приложения, Activity читает этот файл и отображает нам то, что мы настроили





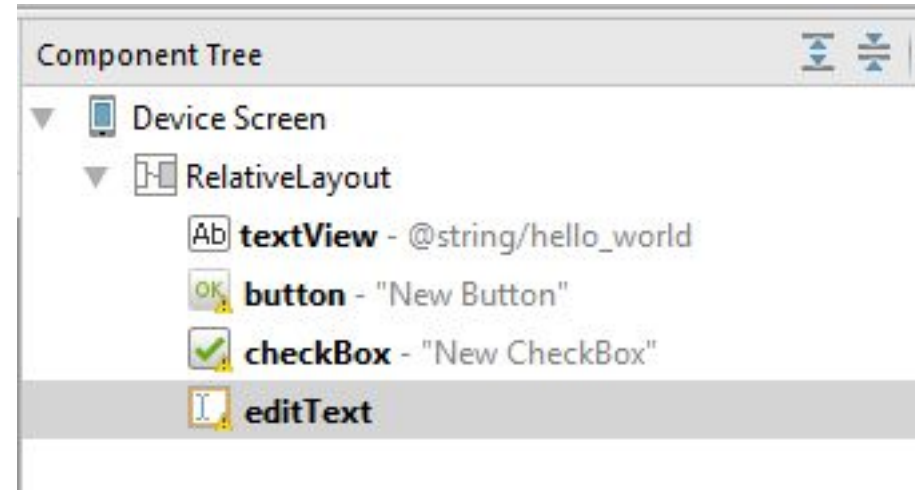
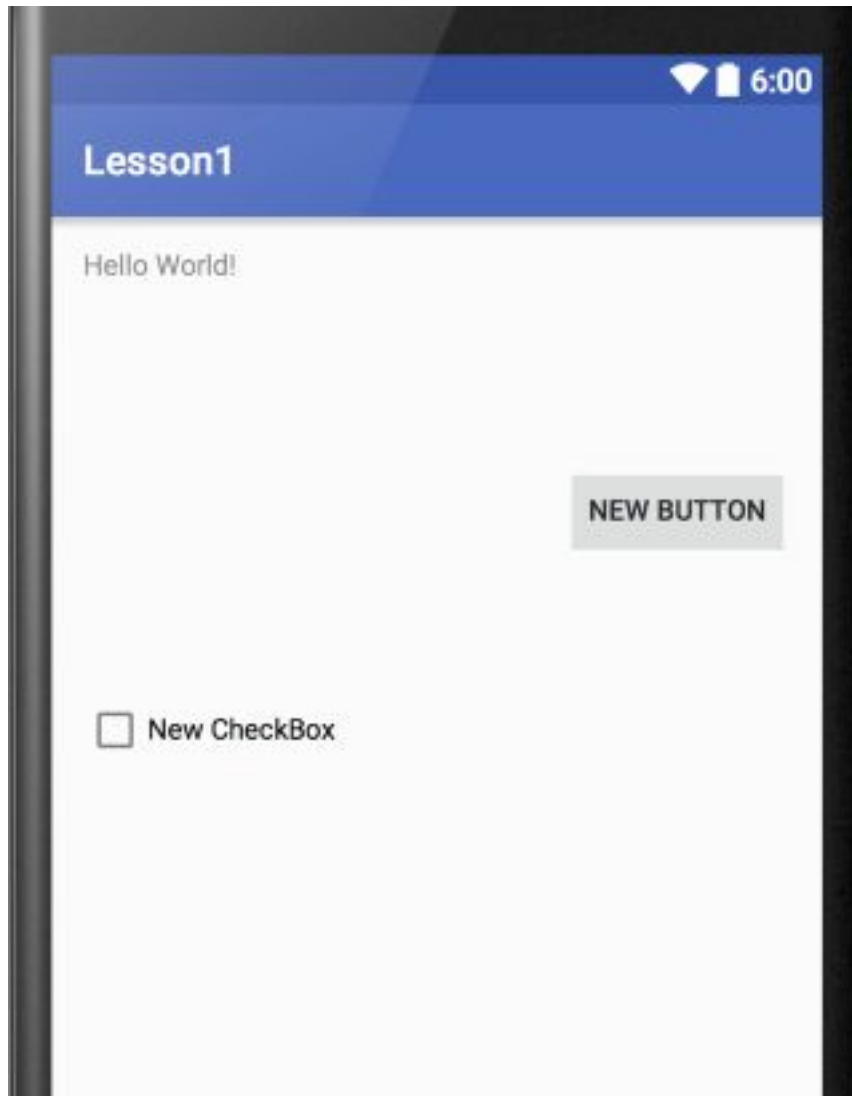
Слева видим список View, разделенный на группы. Здесь отображены все View-элементы, которые вы можете использовать в своих приложениях.

Обратим внимание на белый экран. Мы видим, что на экране сейчас присутствует элемент с текстом **Hello world!** Чтобы узнать, что это за View нажмите на этот текст. Справа во вкладке **Component Tree** вы видите все элементы, которые описаны в этом layout-файле.



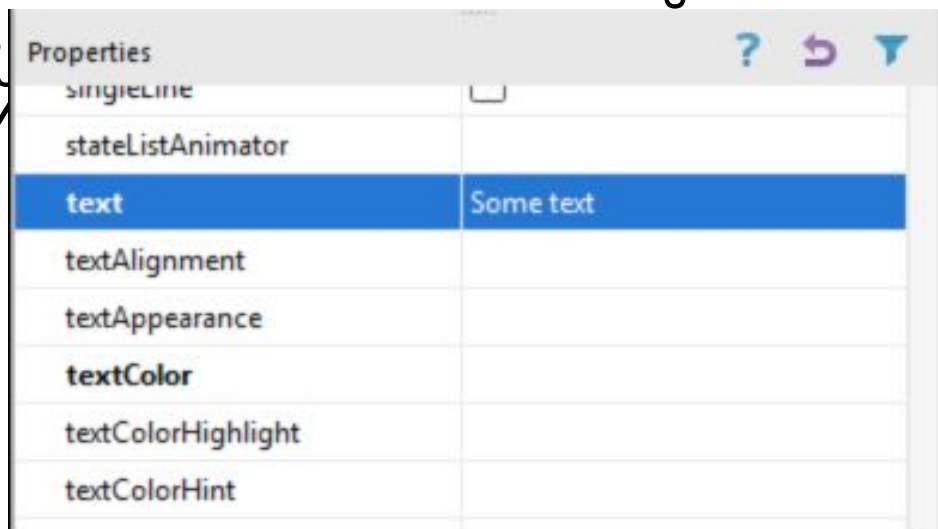
Видим, что выделенный нами элемент – это **TextView**. Это элемент, который умеет отображать текст. Обратите внимание, что он вложен в элемент **RelativeLayout** – это один из видов **ViewGroup**, про которые писалось выше.

Добавим еще элементов на экран, пусть это будут **Button** и **CheckBox**. Для этого просто найдите в списке слева и перетащите на экран вашего будущего приложения. Также можно перетащить их на **RelativeLayout** во вкладке **Component Tree**, результат будет почти тот же. Кроме **Button** и **CheckBox**, добавим еще на экран **Plain Text** из группы **Text Fields**.



Теперь давайте изменим надписи на компонентах нашего экрана. Во вкладке Component Tree жмем на `textView`. Теперь нам нужна вкладка **Properties**. Она отображает свойства выделенного в Component Tree или на экране View-элемента. Располагается она обычно сразу под Component Tree.

Найдем во вкладке Properties свойство `text`. Сейчас там стоит ссылка на текстовую константу. Где создаются эти константы мы рассмотрим в следующих уроках, напишем сюда свой



Ресурси для вивчення

- <http://www.startandroid.ru>
- <http://stackoverflow.com>
- **Android Cookbook**

