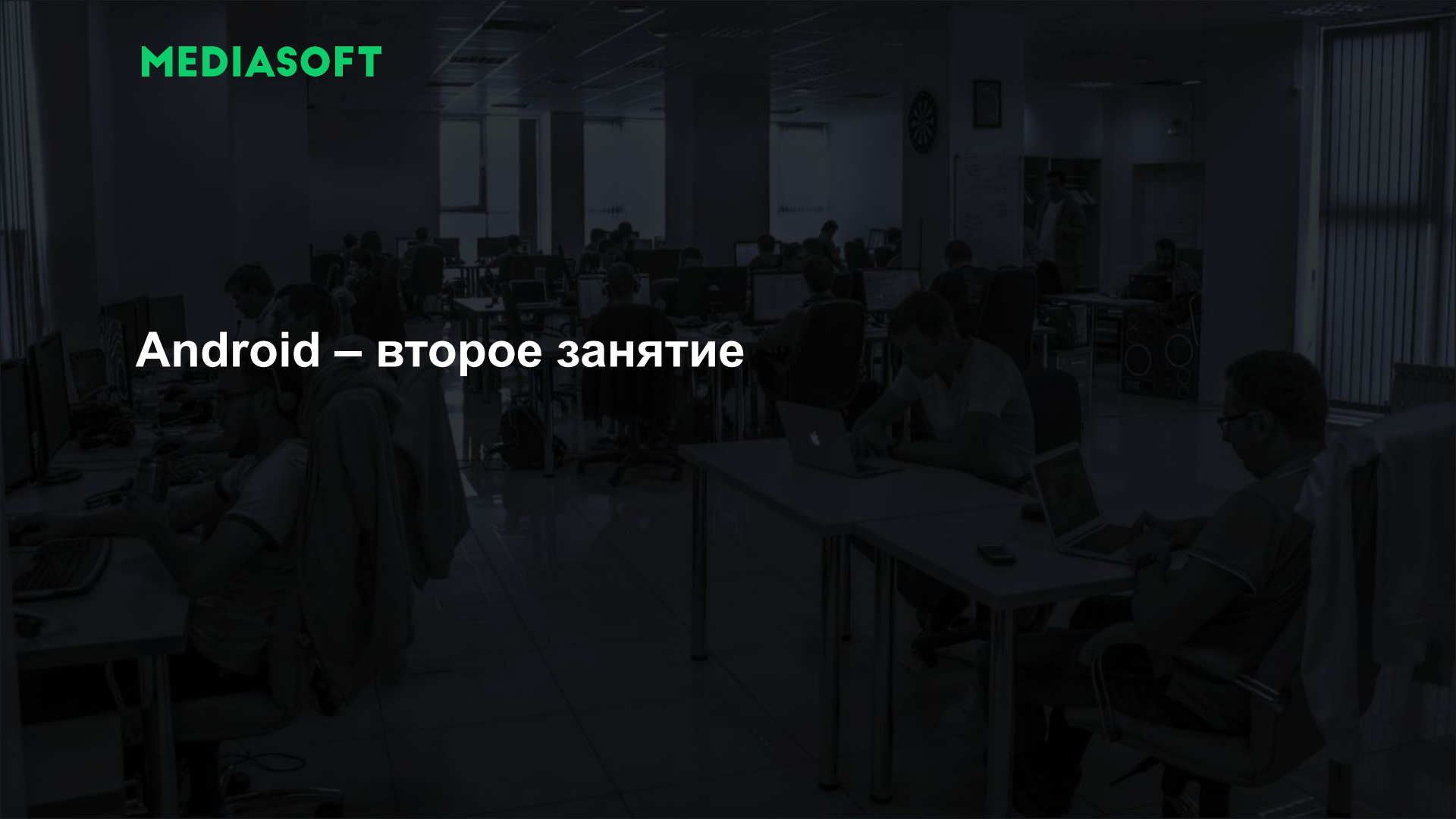


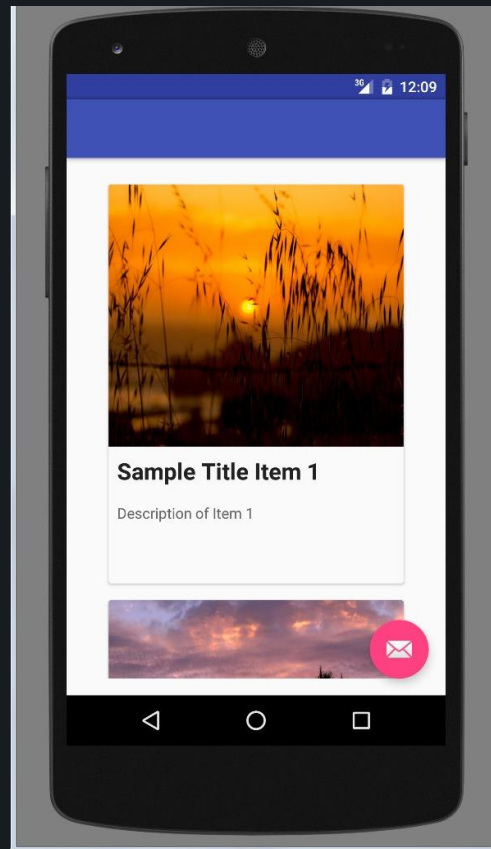
**MEDIASOFT**

# Android – второе занятие



## Разметка экрана

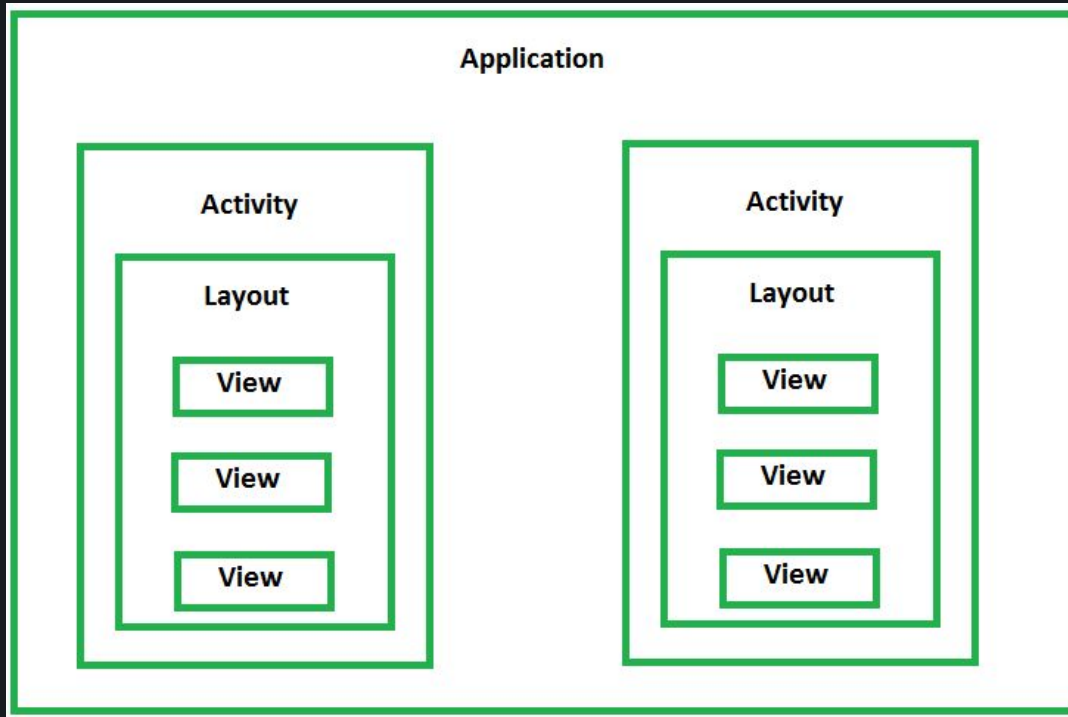
Расположение составных элементов (текста, заголовков, изображений, таблиц) на ~~странице документа~~ экране.



## Варианты создания UI в Android

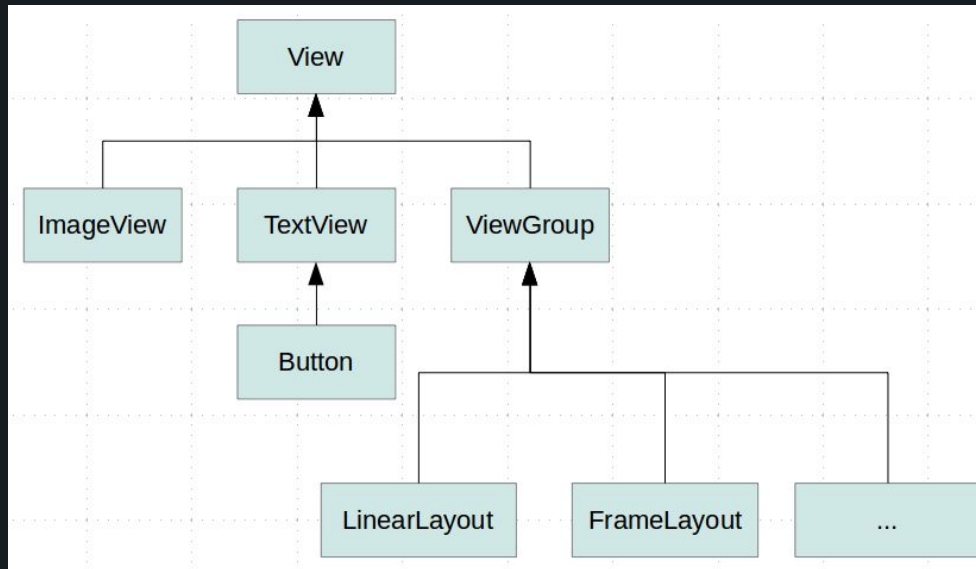
- Создать элементы управления программно в коде(java)
- Объявить элементы интерфейса в XML
- Сочетание обоих способов





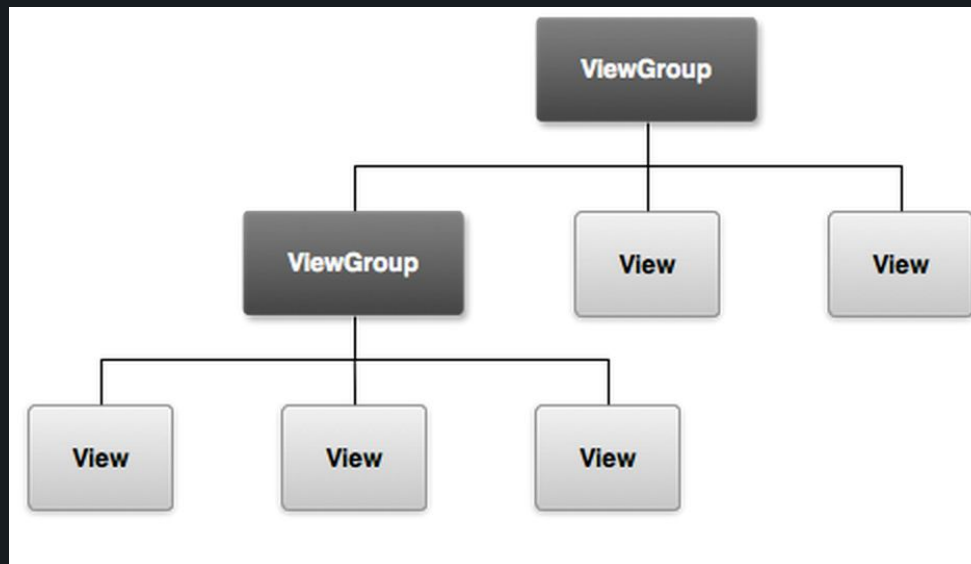
## Компоненты экрана

- Каждый компонент экрана является наследником *View*.
- Все компоненты экрана можно разделить на две группы: **widgets**, **view group**.



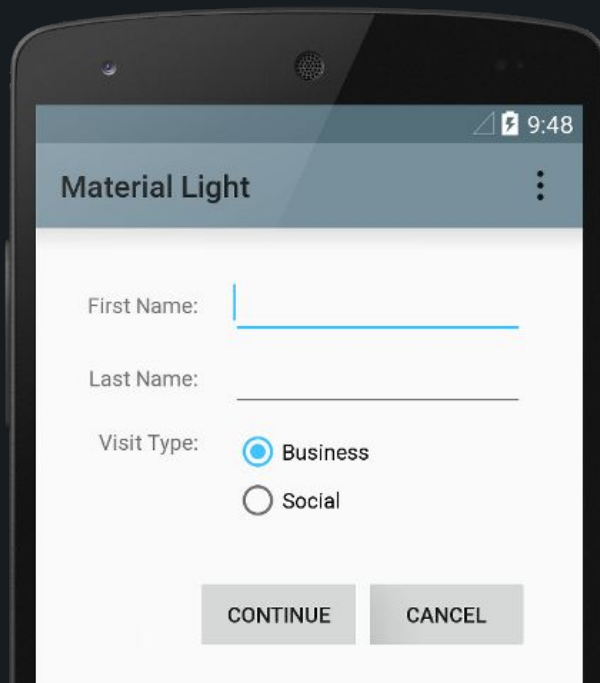
## Компоненты экрана

- **Widget** – компонент для взаимодействия с пользователем
- **ViewGroup** – контейнер размещающий и измеряющий вложенные View элементы.
- Конечную структуру верстки можно представить в виде дерева.



## Примеры Widget-ов

- TextView, EditText, Button, ImageView, RadioButton и др.
- <https://developer.android.com/guide/topics/ui/controls.html?hl=ru>
- 3-я глава  
<https://metanit.com/java/android/3.1.php>



## View

- Любой компонент экрана имеет минимальный набор параметров.
- Например: `layout_width`, `layout_height`, `visibility`, `id`, `background` и др.
- Полный список - <https://developer.android.com/reference/android/view/View.html>





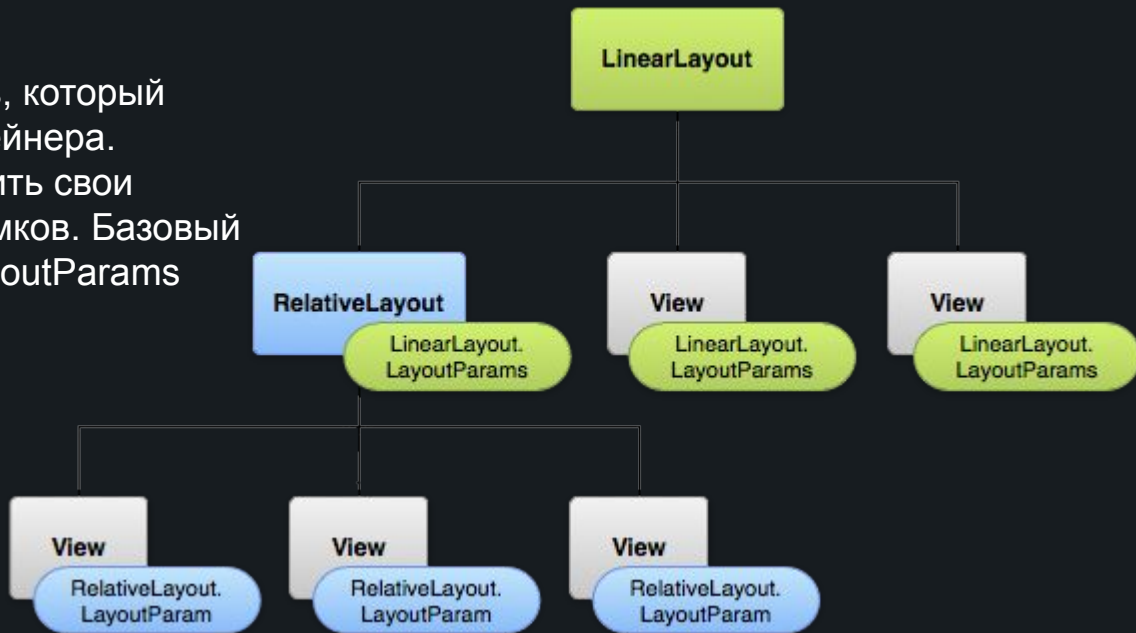
## layout\_width, layout\_height

- Длина и ширина View элемента. Определяется не самим элементом а контейнером в котором лежит.
- Принимает значение: MATCH\_PARENT(по размеру предка), WRAP\_CONTENT(по размеру контента), Конкретное числовое значение.
- Задается не для самого элемента, а с помощью объекта LayoutParams



## LayoutParams

Специальный контейнер атрибутов, который передается каждому потомку контейнера. Каждый контейнер может определить свои нестандартные атрибуты для потомков. Базовый набор реализован в `ViewGroup.LayoutParams` (`layout_width`, `layout_height`).



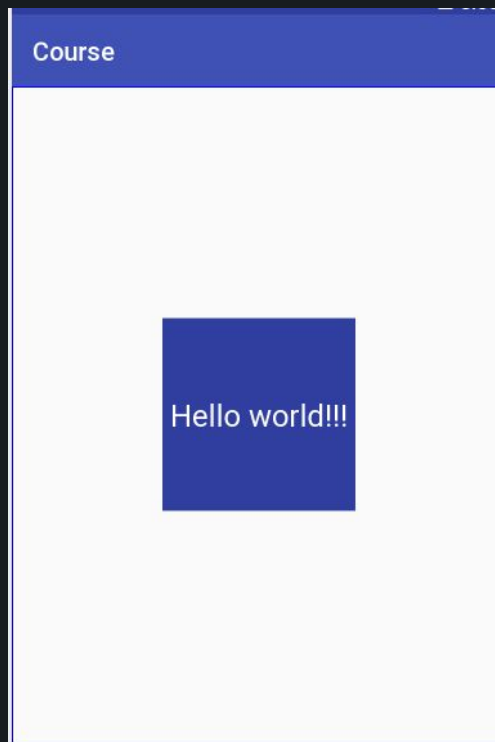
## ViewGroup

- LinearLayout, RelativeLayout, FrameLayout
- Более полный список 3-я глава. <https://metanit.com/java/android/3.2.php>



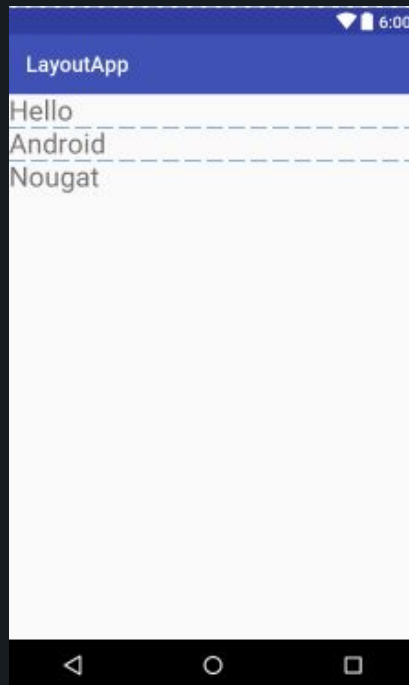
## FrameLayout

- Контейнер FrameLayout предназначен для вывода на экран одного помещенного в него визуального элемента. Если же мы поместим несколько элементов, то они будут накладываться друг на друга.



## LinearLayout

Упорядочивает все дочерние элементы в одном направлении: по горизонтали или по вертикали. Все элементы расположены один за другим. Направление разметки указывается с помощью атрибута `android:orientation`.

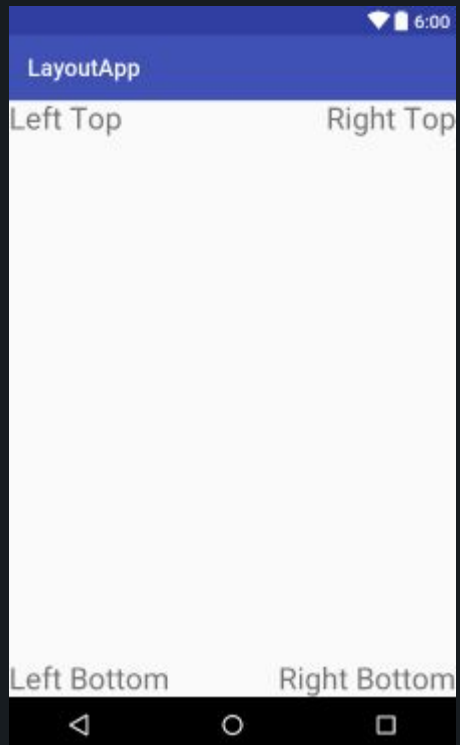


## RelativeLayout

Располагает дочерние элементы относительно позиции других дочерних элементов или относительно себя с помощью выставления правил. Например: `layout_toLeftOf`, `layout_alignParentBottom`

Полный список правил -

<https://metanit.com/java/android/3.3.php>



## Размеры

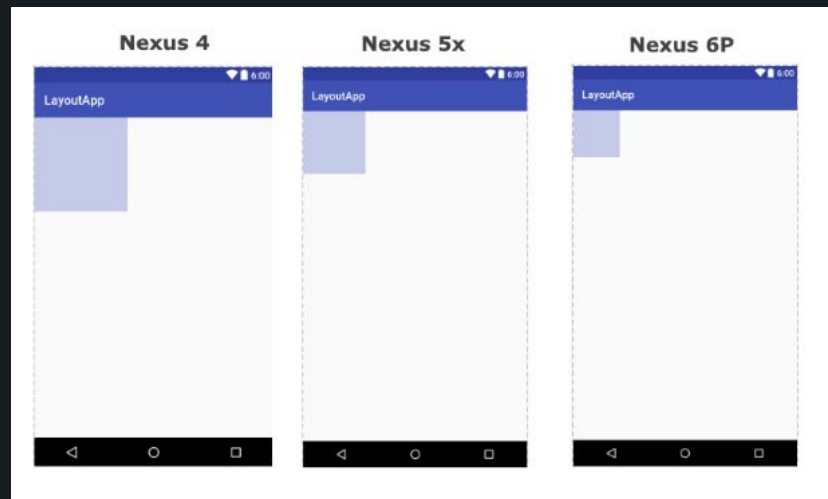
```
<ImageView  
    android:layout_width="150dp"  
    android:layout_height="150dp"  
    android:layout_gravity="center"  
    android:background="@color/colorPrimaryDark" />
```

- px: пиксели текущего экрана.
- dp: (device-independent pixels) независимые от плотности экрана пиксели.
- sp: (scale-independent pixels) независимые от масштабирования пиксели
- pt: 1/72 дюйма, базируются на физических размерах экрана
- mm: миллиметры
- in: дюймы



## dp (device-independent pixels)

Абстрактная единица измерения, основанная на физической плотности экрана с разрешением 160 dpi (точек на дюйм). В этом случае  $1dp = 1px$ . Если размер экрана больше или меньше, чем 160dpi, количество пикселей, которые применяются для отрисовки 1dp соответственно увеличивается или уменьшается. Например, на экране с 240 dpi  $1dp=1,5px$ , а на экране с 320dpi  $1dp=2px$ . Общая формула для получения количества физических пикселей из dp:  $px = dp * (dpi / 160)$





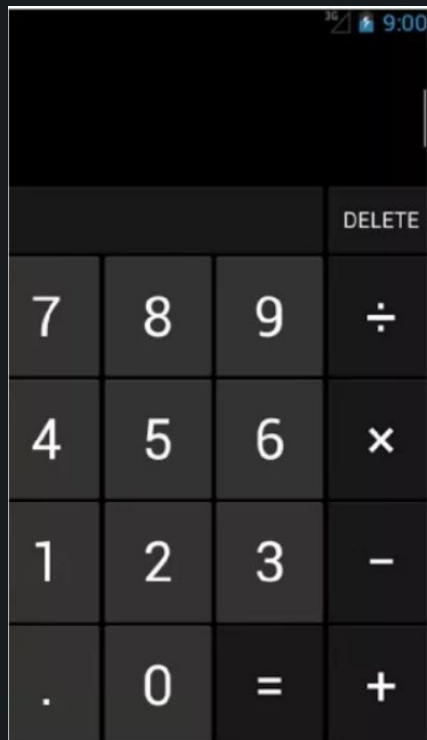
## sp: (scale-independent pixels)

Независимые от масштабирования пиксели. Допускают настройку размеров, производимую пользователем. Рекомендуются для работы со шрифтами. Аналог em в web-верстке



# Домашнее задание

- Сверстать калькулятор



**СПАСИБО ЗА ВНИМАНИЕ**

**MEDIASOFT**

**Сергей Полуэктов**  
директор