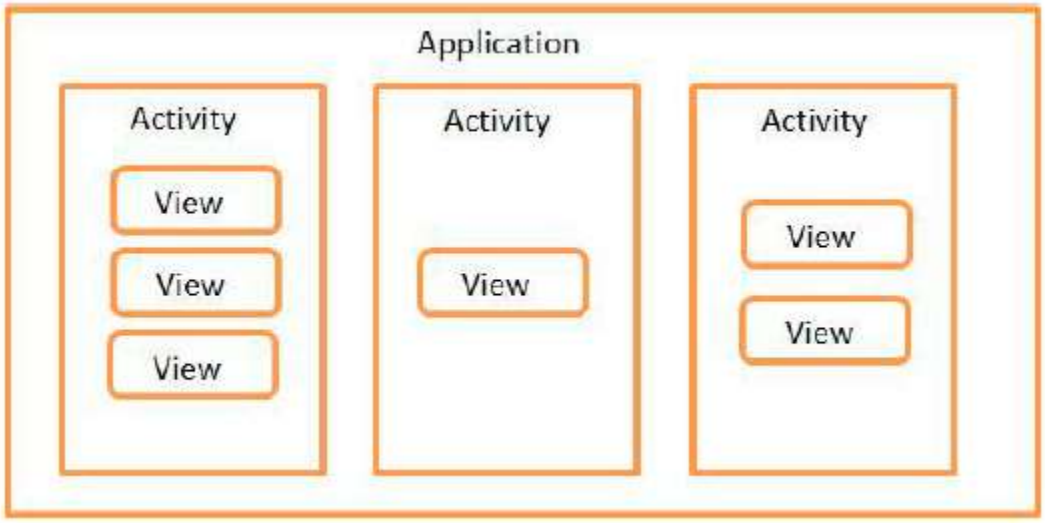
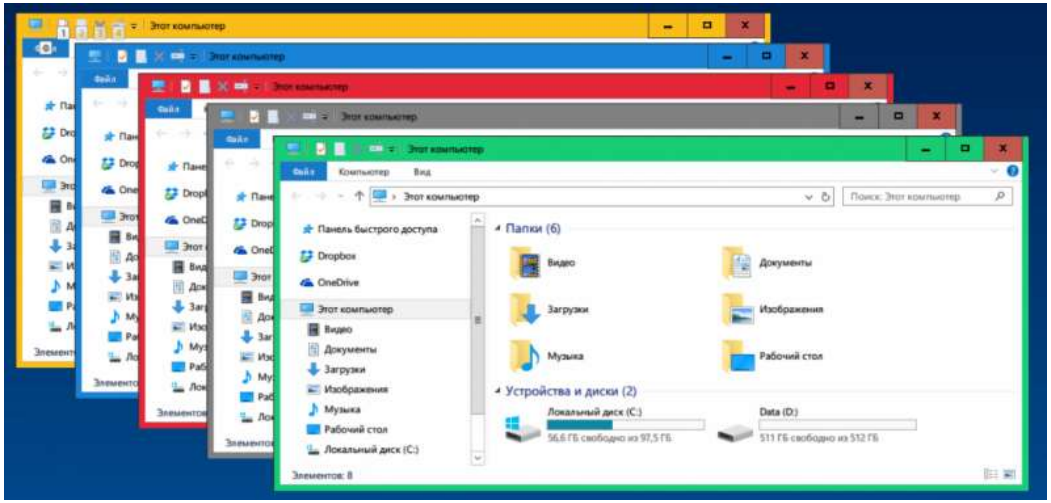
Android Development for Beginners

Лекция 2

Элементы экрана и их свойства

View





1 View (Button, checkbox, textfield)

2 ViewGroup (Layouts)

3 Activity

4 Application

Виды Layouts. Ключевые отличия и свойства.

Основные виды **Layout :**

LinearLayout – отображает View-элементы в виде одной строки (если он **Horizontal**) или одного столбца (если он **Vertical**).

Grid**Layout** – отображает элементы в виде таблицы, по строкам и столбцам.

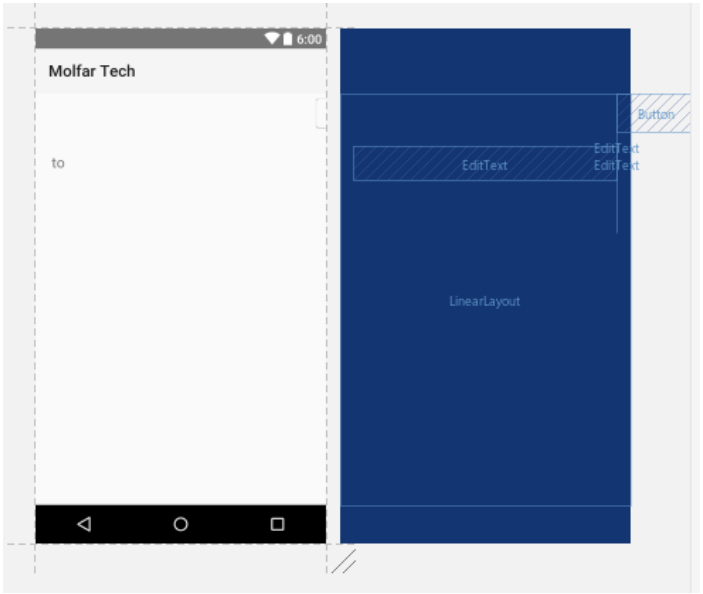
RelativeLayout – для каждого элемента настраивается его положение относительно других элементов.

AbsoluteLayout – для каждого элемента указывается явная позиция на экране в системе координат (x,y)

LinearLayout

- LinearLayout:
  - Horizontal (Выравнивает все дочерние элементы верикально) (android:orientation="horizontal")
  - Vertical (Выравнивает все дочерние элементы горизонтально) (android:orientation="vertical")
  - Weight, индивидуальный вес дочернего элемента (android:layout_weight) (важность элемента)


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```

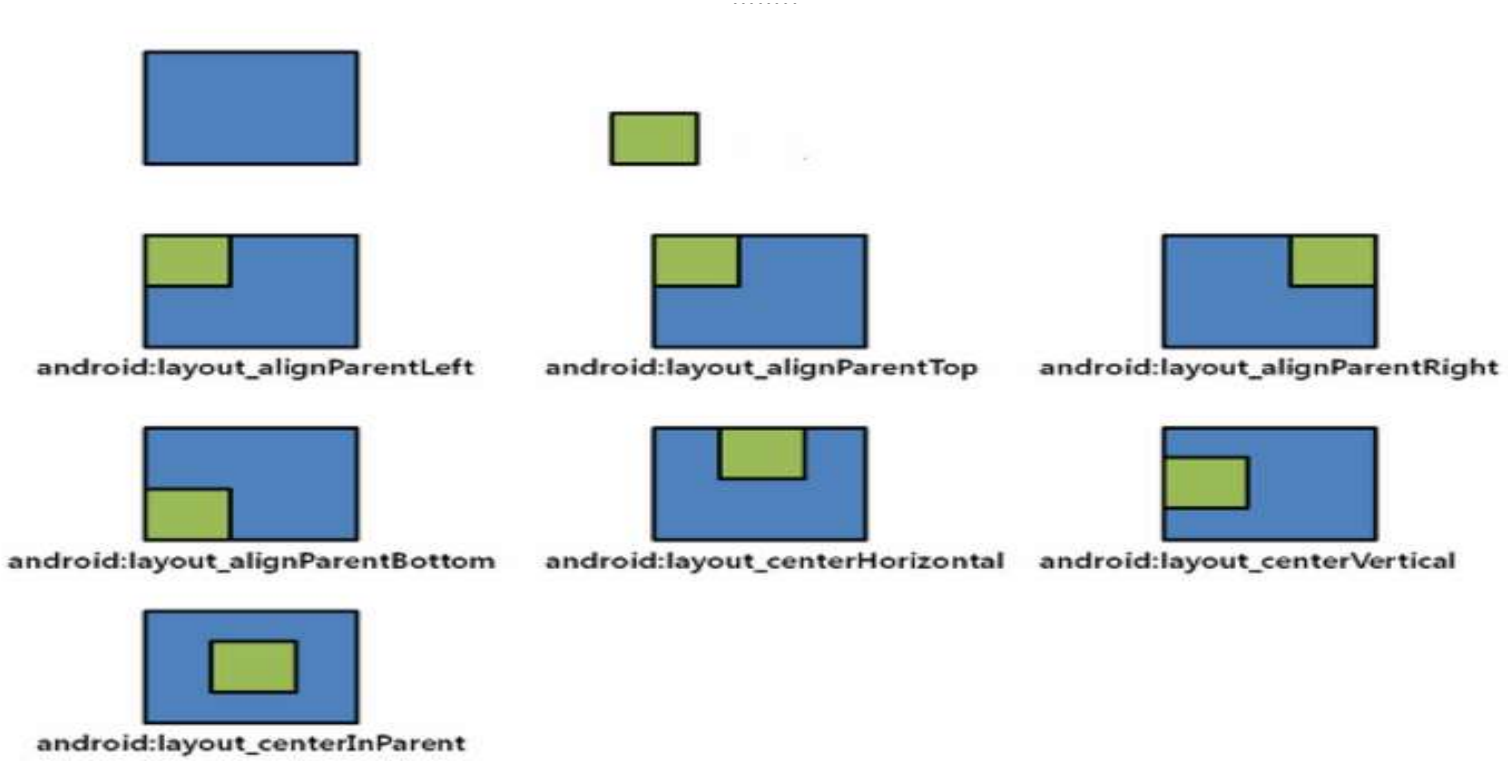


RelativeLayout

ReiativeLayout (относительная разметка) позволяет расположить каждый View-элемент может быть расположен определенным образом относительно указанного View-элемента

android:layout_alignParentBottom - выравнивание относительно нижнего края родителя

android:layout_alignParentLeft - выравнивание относительно левого края родителя



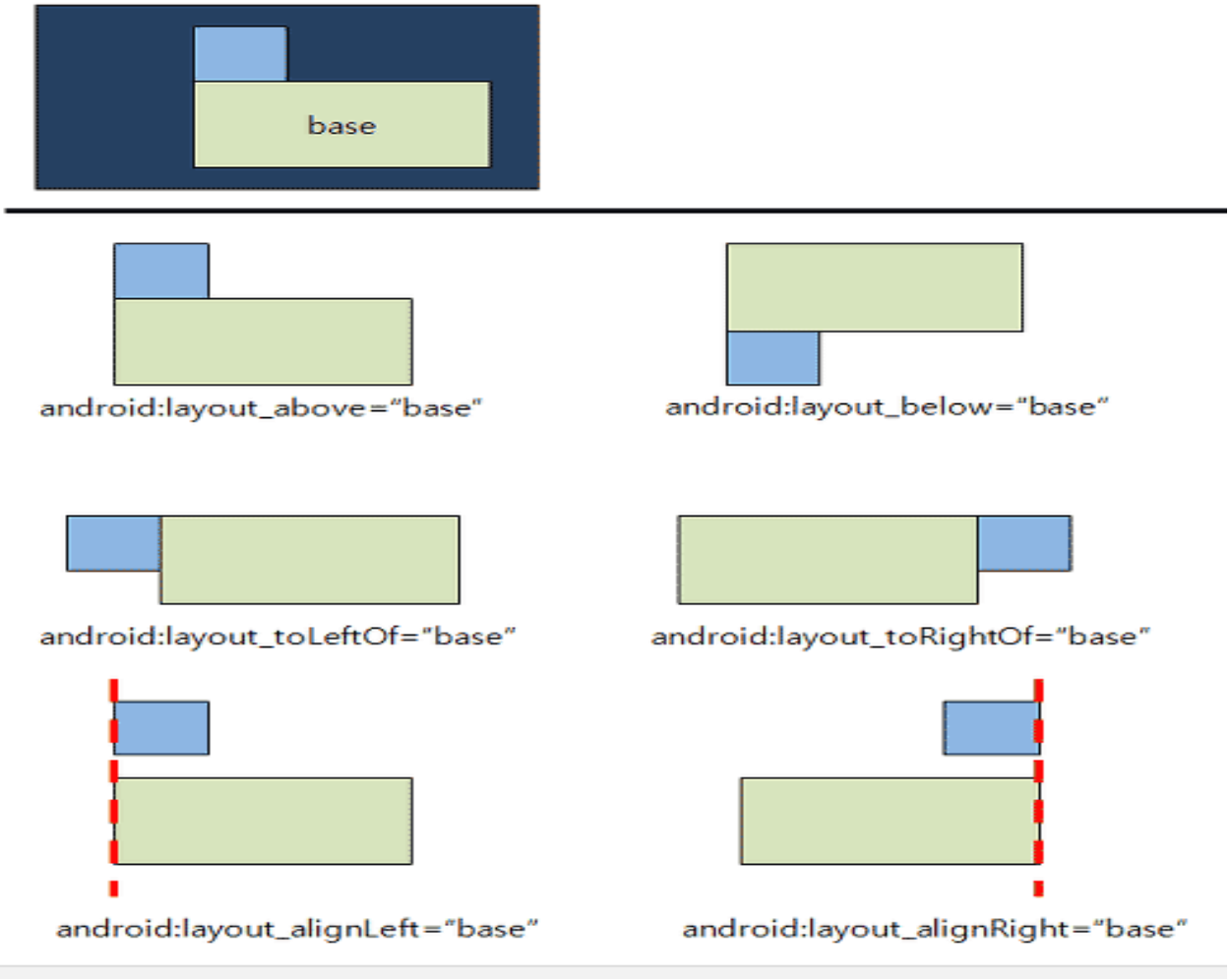
RelativeLayout

android:layout_above - *размещается над указанным компонентом*

android:layout_below - *размещается под указанным компонентом*

android:layout_alignLeft - *выравнивается по левому краю указанного компонента*

android:layout_alignRight - *выравнивается по правому краю указанного компонента*



RelativeLayout

Чтобы компоненты выдерживали расстояние другу другу, используются атрибуты, добавляющие пространство между ними.

android:layout_marginTop

android:layout_marginBottom

android:layout_marginLeft

android:layout_marginRight

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/button"
        android:hint="Введите ТЕКСТ..." >
    </EditText>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:text="Клик Ми!" >
    </Button>

</RelativeLayout>
```

AbsoluteLayout

Обеспечивает абсолютное позиционирование элементов на экране.

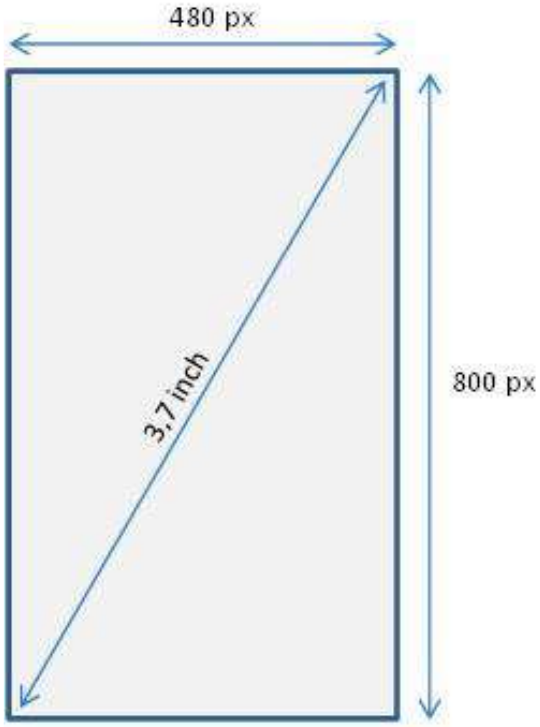
(Вы указываете координаты для левого верхнего угла компонента.)

```
<AbsoluteLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 android:layout_width="match_parent"
 android:layout_height="match_parent">
<Button
 android:id="@+id/button1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_x="42dp"
 android:layout_y="62dp"
 android:text="Button">
</Button>
<TextView
 android:id="@+id/textView1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_x="142dp"
 android:layout_y="131dp"
 android:text="TextView">
</TextView>
<CheckBox
 android:id="@+id/checkBox1"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_x="55dp"
 android:layout_y="212dp"
 android:text="CheckBox">
</CheckBox>
```

Layout параметры для View-элементов

- **Диагонал**ь – это расстояние между противоположными углами экрана, обычно измеряется в дюймах.
- **Разрешение** – кол-во точек по горизонтали и вертикали, которое экран способен отобразить, измеряется в пикселах.

Пример Диагональ = 3,7 дюйма, разрешение = 800x480 пикселов.



dpi (dot per inch). Кол-во пикселов в одном дюйме

dpi равняется: $c_2 = a_2 + b_2$, где c – кол-во пикселей по диагонали, т.е. вмещаемое в 3,7 дюйма. a и b – стороны экрана.

c = 3,7 * dpi $(3,7 * dpi)_2 = 480_2 + 800_2$ $dpi_2 = 870400 / 13,69 = 63579$ dpi = 252. Т.е. в одном дюйме экрана помещается ряд из 252 пикселов

Layout параметры для View-элементов

АБСОЛЮТНЫЕ ЗНАЧЕНИЯ:

В Android разроботке используются следующие единицы измерения ЕИ):

dp или **dip** - Density-independent Pixels. Абстрактная ЕИ, позволяющая приложениям выглядеть одинаково на различных экранах и разрешениях.

sp - Scale-independent Pixels. То же, что и dp, только используется для размеров шрифта в View элементах

pt - 1/72 дюйма, определяется по физическому размеру экрана. Эта ЕИ из типографии.

px – пиксел, не рекомендуется использовать т.к. на разных экранах приложение будет выглядеть по-разному.

mm – миллиметр, определяется по физическому размеру экрана

in – дюйм, определяется по физическому размеру экрана

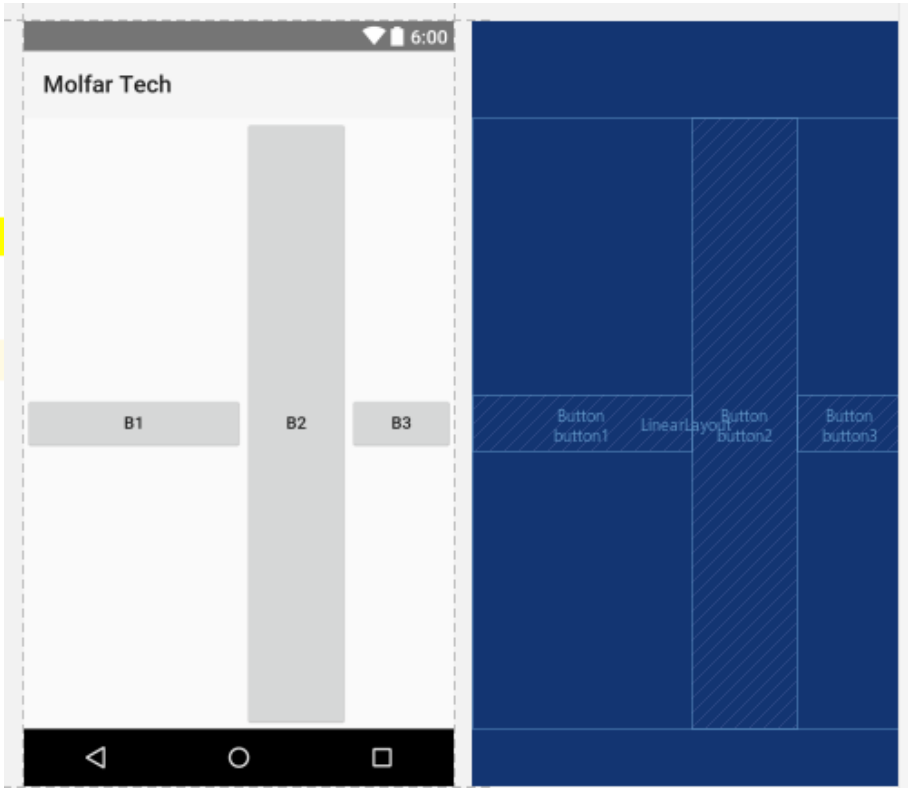
Подробней о различиях и соотношениях между этими ЕИ вы можете прочесть в этом материале сайта.

Layout параметры для View-элементов

match_parent (fill_parent) – означает, что элемент займет всю доступную ему в родительском элементе ширину/высоту.

wrap_content – ширна/высота элемента будет определяться его содержимым

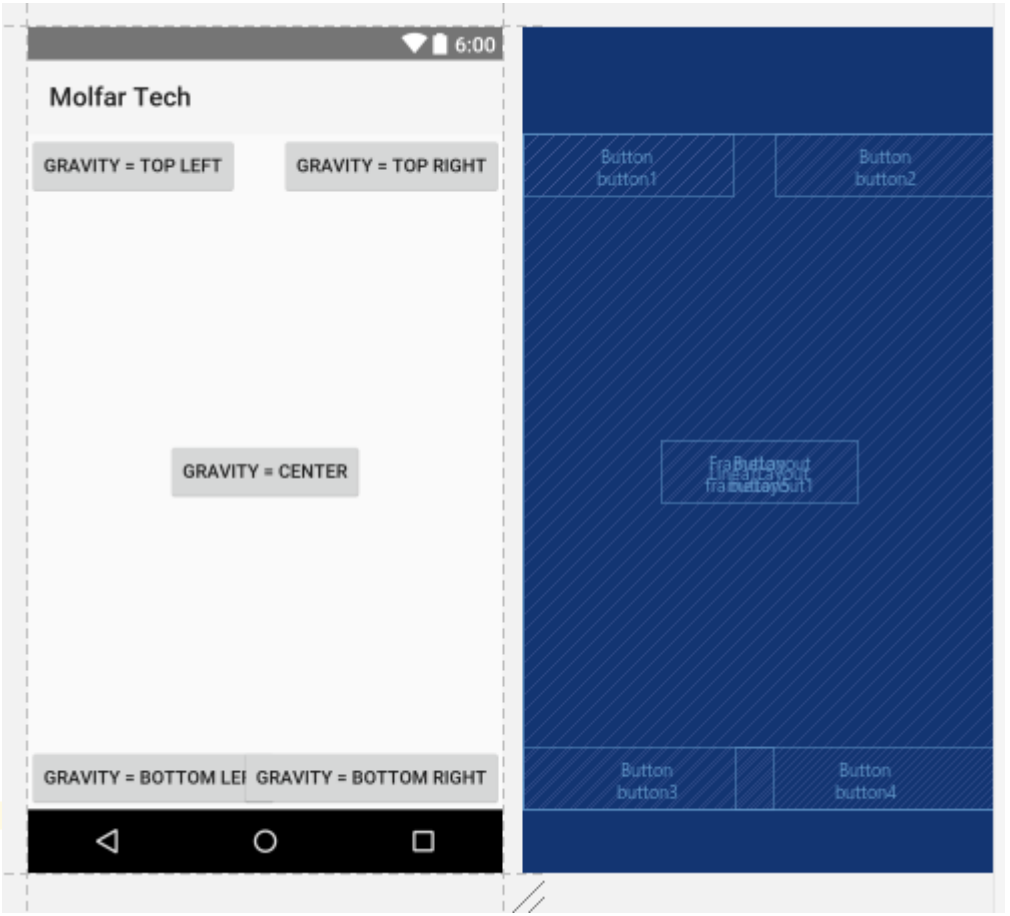
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal">
    <Button android:id="@+id/button1"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="B1" android:layout_weight="1">
    </Button> <Button android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:text="B2" android:layout_weight="0">
</Button> <Button android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:id="@+id/button3"
    android:text="B3"
    android:layout_weight="0">
</Button> </LinearLayout>
```



Layout параметры для View-элементов

layout_gravity аналогичен выравниванию из Word или Excel .

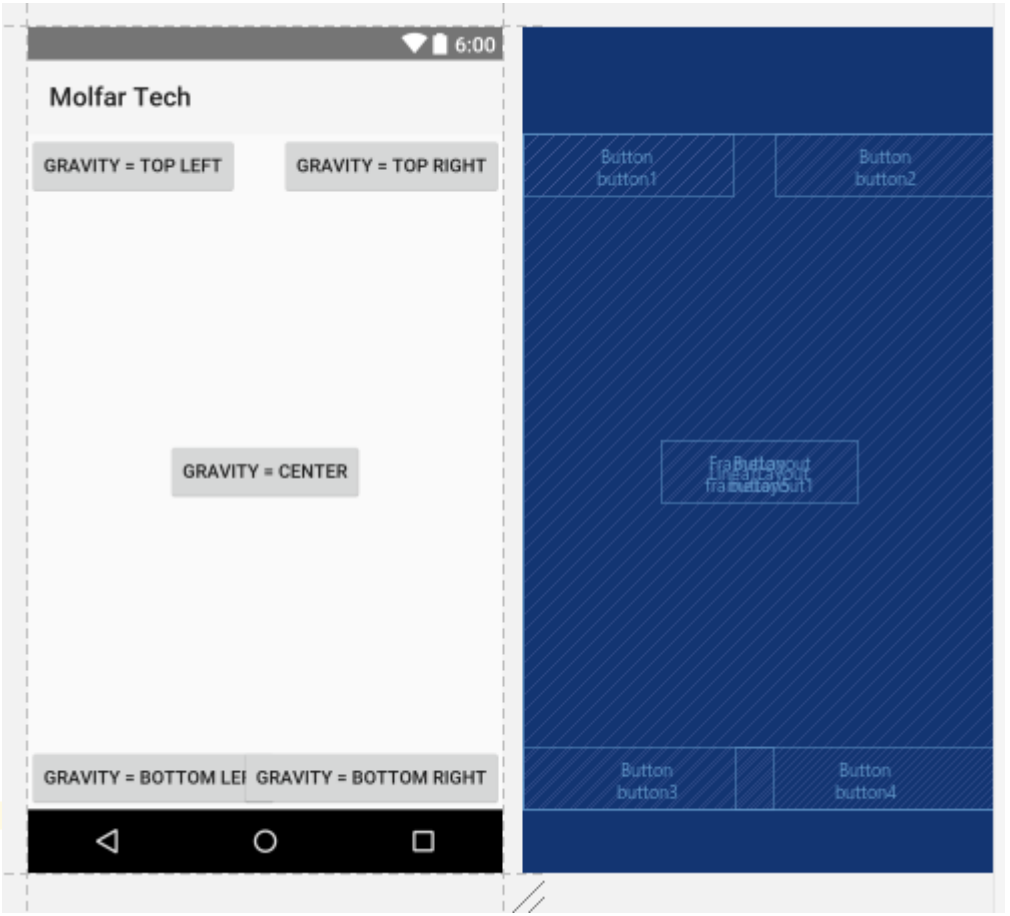
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <FrameLayout android:id="@+id/frameLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <Button android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
    android:layout_gravity="top|left"
    android:text="gravity = top left">
</Button>
    <Button android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="top|right"
        android:text="gravity = top right">
    </Button> <Button android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|left"
        android:text="gravity = bottom left">
    </Button> <Button android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|right"
        android:text="gravity = bottom right">
    </Button> <Button android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="gravity = center">
    </Button>
</FrameLayout>
</LinearLayout>
```



Layout параметры для View-элементов

layout_gravity аналогичен выравниванию из Word или Excel .

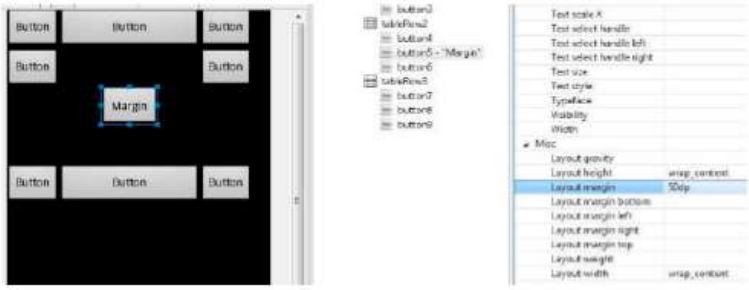
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <FrameLayout android:id="@+id/frameLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <Button android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
    android:layout_gravity="top|left"
    android:text="gravity = top left">
</Button>
    <Button android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="top|right"
        android:text="gravity = top right">
    </Button> <Button android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|left"
        android:text="gravity = bottom left">
    </Button> <Button android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|right"
        android:text="gravity = bottom right">
    </Button> <Button android:id="@+id/button5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="gravity = center">
    </Button>
</FrameLayout>
</LinearLayout>
```



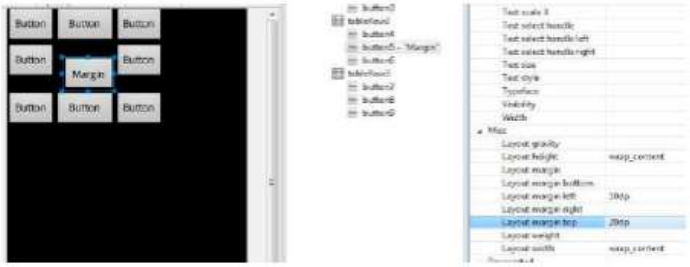
Layout параметры для View-элементов

Layout margin
Параметры margin полностью аналогичны **margin** из **html**. Это отступ. Он может быть со всех сторон сразу, либо только с необходимых сторон.

margin = 50 dp
Вокруг кнопки со всех сторон образовался отступ = 50 dp.



margin left = 10 dp
margin top = 20 dp
Отступ слева и сверху.



Layout параметры для View-элементов

Стили

В Android как и в HTML - CSS, есть стили. Стили позволяют вам группировать атрибуты элементов (кнопок, таблиц, параграфов и т.д.). Далее вы просто применяете к элементам стили, и элемент рисуется с учетом всех атрибутов стиля. И нет необходимости повторять несколько раз один и тот же код для элементов, которые должны выглядеть одинаково. Особенно это удобно в случае изменения атрибутов. Вы просто меняете один раз стиль и все элементы с этим стилем меняются.

Программное разметка View-элементов

Для динамического создания относительной разметки, есть возможность делать это програмно в коде:

```
public class ChooseDeviceActivity extends Activity {

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // setContentView(R.layout.activity_main);

        EditText editText = new EditText(this);
        RelativeLayout.LayoutParams params = new RelativeLayout.LayoutParams(LayoutParams.MATCH_PARENT,
                LayoutParams.WRAP_CONTENT);
        params.addRule(RelativeLayout.ALIGN_PARENT_LEFT);
        // use same id as defined when adding the button
        params.addRule(RelativeLayout.LEFT_OF, 1001);
        editText.setLayoutParams(params);
        editText.setHint("Введите имя кота....");

        Button button = new Button(this);
        RelativeLayout.LayoutParams params2 = new RelativeLayout.LayoutParams(LayoutParams.WRAP_CONTENT,
                LayoutParams.WRAP_CONTENT);
        params2.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);
        button.setLayoutParams(params2);
        button.setText("Нажми нежно!");
        // give the button an id that we know
        button.setId(1001);
        RelativeLayout layout = new RelativeLayout(this);
        layout.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
        layout.addView(editText);
        layout.addView(button);
        setContentView(layout);
    }
```

Файл манифеста AndroidManifest.xml

Файл манифеста **AndroidManifest.xml** предоставляет основную информацию о программе системе

Назначение файла

•объявляет имя Java-пакета приложения, который служит уникальным идентификатором;
•описывает компоненты приложения — деятельности, службы, приемники широковещательных намерений и контент-провайдеры, что позволяет вызывать классы, которые реализуют каждый из компонентов, и объявляет их намерения;
•содержит список необходимых разрешений для обращения к защищенным частям API и взаимодействия с другими приложениями;
•объявляет разрешения, которые сторонние приложения обязаны иметь для взаимодействия с компонентами данного приложения;
•объявляет минимальный уровень API Android, необходимый для работы приложения;
•перечисляет связанные библиотеки;

Файл манифеста AndroidManifest.xml

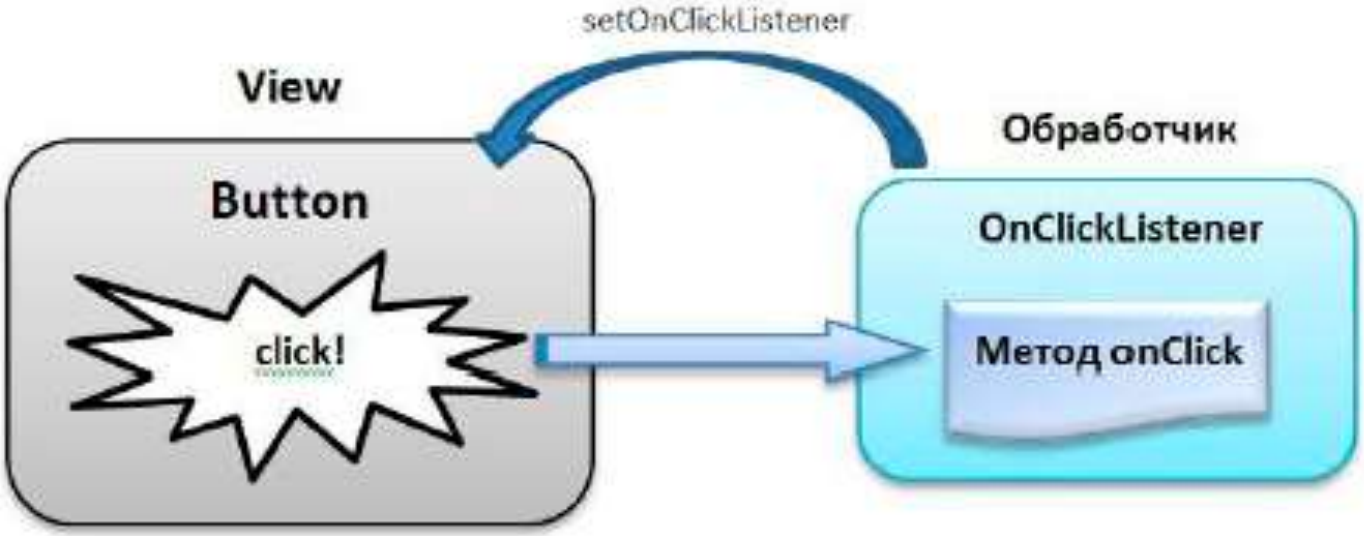
Файл манифеста инкапсулирует всю архитектуру Android-приложения

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.molfar.nxtmolfar"
    android:versionCode="5"
    android:versionName="1.4"
    android:installLocation="auto">
    <supports-screens android:resizeable="true"
        android:smallScreens="true"
        android:normalScreens="true"
        android:largeScreens="true"
        android:anyDensity="true" />
    <application
        android:icon="@drawable/icon" android:label="Molfar Tech">
        <activity
            android:name=".NXTRemoteControl"
            android:label="Molfar Tech" android:configChanges="keyboardHidden|orientation"
            android:launchMode="singleTask">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.front.autofocus" android:required="false"/>

    <uses-sdk android:minSdkVersion="7" android:targetSdkVersion="16"></uses-sdk>
</manifest>
```

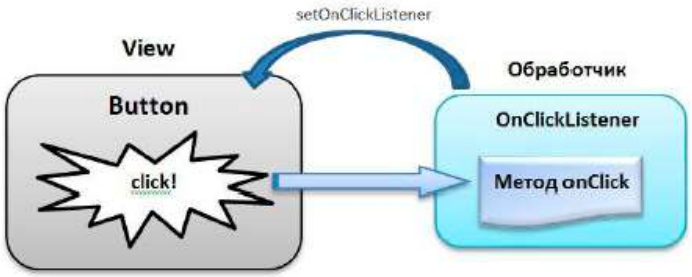
Обработчики событий



Соответственно для реализации обработчика необходимо выполнить следующие шаги:

- создаем обработчик
- заполняем метод onClick
- присваиваем обработчик кнопке

Обработчики событий




```
private void setupUI() {
    setContentView(R.layout.auto);
    updateMenu(R.id.menuitem_auto);
   // найдем View-элементы
    buttonStart = (Button) findViewById(R.id.b_start);
    buttonStop = (Button) findViewById(R.id.b_stop);
    mConnectButton = (Button) findViewById(R.id.connect_button);
    mDisconnectButton = (Button) findViewById(R.id.dissconnect_button);
    buttonStop.setBackgroundColor(0);
    buttonStart.setBackgroundColor(0);
    buttonStart.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            isEnableAuto = true;
            buttonStart.setTextColor(Color.GREEN);
            buttonStop.setTextColor(Color.WHITE);
        }
    });
    buttonStop.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            isEnableAuto = false;
            mNXTTalker.motors((byte) 0, (byte) 0, mRegulateSpeed, mSynchronizeMotors);
            buttonStop.setTextColor(Color.RED);
            buttonStart.setTextColor(Color.WHITE);
        }
```

```
<Button
    android:text="Start"
    android:id="@+id/b_start"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/b_start"
    android:layout_gravity="right"
    android:textStyle="bold"
    android:textSize="20sp"
    android:layout_weight="1" />
```

Задание и полезные ссылки

Модифицировать созданное приложение:

- Практически потренироваться в работе с View компонентами
- Изучить структуру AndroidManifest.
- Ознакомиться с процессом создания стилей.

Полезные ссылки:

https://developer.android.com/guide/index.html

http://javarush.ru/

http://startandroid.ru/

http://developer.alexanderklimov.ru/android/