

Мобильная разработка

Пирская Любовь Владимировна,
к.т.н., доцент кафедры МОП ЭВМ
lpirskaya@sfedu.ru

RelativeLayout

- View размещаются относительно друг друга и родителя
- Основной Layout
- Проще чем LinearLayout
- Performance

RelativeLayout. Размещение относительно родителя

Важные атрибуты, которые связаны с родителем:

- **android:layout_alignParentBottom** - выравнивание относительно нижнего края родителя
- **android:layout_alignParentLeft** - выравнивание относительно левого края родителя
- **android:layout_alignParentRight** - выравнивание относительно правого края родителя
- **android:layout_alignParentTop** - выравнивание относительно верхнего края родителя
- **android:layout_centerInParent** - выравнивание по центру родителя по вертикали и горизонтали
- **android:layout_centerHorizontal** - выравнивание по центру родителя по горизонтали
- **android:layout_centerVertical** - выравнивание по центру родителя по вертикали

RelativeLayout. Размещение относительно родителя



`android:layout_alignParentLeft`



`android:layout_alignParentTop`



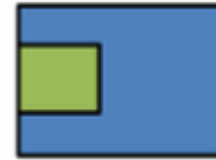
`android:layout_alignParentRight`



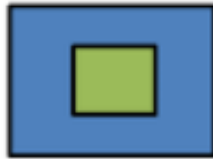
`android:layout_alignParentBottom`



`android:layout_centerHorizontal`



`android:layout_centerVertical`



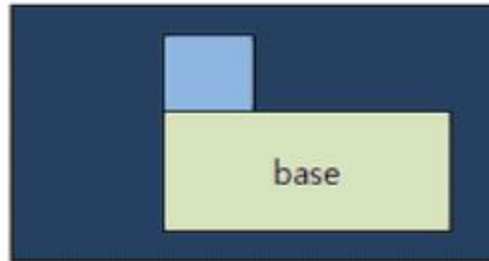
`android:layout_centerInParent`

RelativeLayout. Размещение относительно друг друга

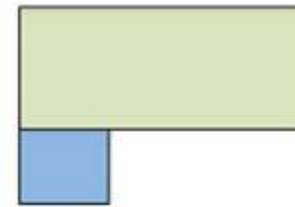
Атрибуты для размещения относительно друг друга:

- **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
- **android:layout_alignTop** - выравнивается по верхнему краю указанного компонента
- **android:layout_alignBottom** - выравнивается по нижнему краю указанного компонента
- **android:layout_toLeftOf** - правый край компонента размещается слева от указанного компонента
- **android:layout_toRightOf** - левый край компонент размещается справа от указанного компонента

RelativeLayout. Размещение относительно друг друга



`android:layout_above="base"`



`android:layout_below="base"`

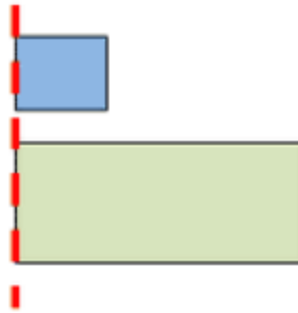


`android:layout_toLeftOf="base"`

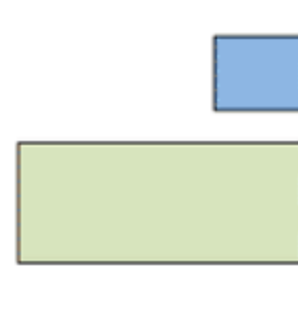


`android:layout_toRightOf="base"`

RelativeLayout. Размещение относительно друг друга



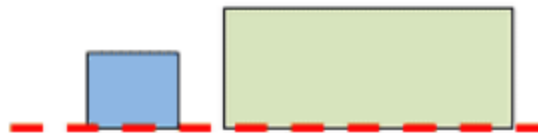
`android:layout_alignLeft="base"`



`android:layout_alignRight="base"`



`android:layout_alignTop="base"`



`android:layout_alignBottom="base"`

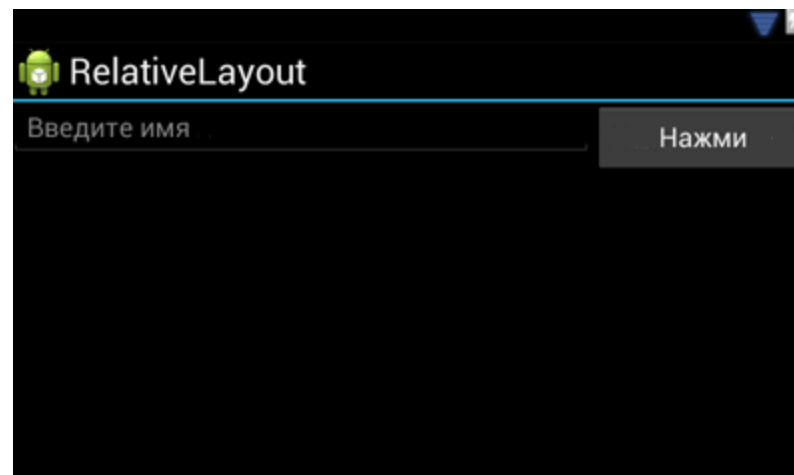
Пример

```
<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 имя" >
    </EditText>

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

</RelativeLayout>
```



Intent

Intent – намерение (*англ.*)

- Запустить другое приложение
- Открыть файл
- Перейти на другой экран
- Явные (explicit) и неявные (implicit)

Intent состоит из...

- Действие – action (ACTION_VIEW, ACTION_EDIT и т.д.)
- Категория – category (CATEGORY_LAUNCHER, CATEGORY_BROWSABLE)
- Данные – data
- Дополнительные параметры - extras

«ЯВНЫЙ» Intent

```
private void notifyThreat(String appName, String threatName) {  
    Intent intent = new Intent(mContext, ThreatInfoActivity.class);  
    intent.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);  
    intent.putExtra(ThreatInfoActivity.EXTRA_APP_NAME, appName);  
    intent.putExtra(ThreatInfoActivity.EXTRA_THREAT_NAME, threatName);  
    mContext.startActivity(intent);  
}
```

```
@Override  
protected void onResume() {  
    super.onResume();  
  
    String appName = getIntent().getStringExtra(EXTRA_APP_NAME);  
    ((TextView) findViewById(R.id.threat_app_name)).setText(appName);  
  
    String threatName = getIntent().getStringExtra(EXTRA_THREAT_NAME);  
    ((TextView) findViewById(R.id.threat_name)).setText(threatName);  
}
```

Запуск другого приложения

```
public static void openBrowser(Context context, Uri uri) {  
    Intent intent = new Intent(Intent.ACTION_VIEW);  
    intent.setData(uri);  
    intent.addCategory(Intent.CATEGORY_BROWSABLE);  
    if (intent.resolveActivity(context.getPackageManager()) != null) {  
        context.startActivity(intent);  
    }  
}
```

```
public static void openChrome(Context context) {  
    PackageManager manager = context.getPackageManager();  
    Intent chromeIntent = manager.getLaunchIntentForPackage("com.android.chrome");  
    if (chromeIntent != null) {  
        context.startActivity(chromeIntent);  
    }  
}
```

Запуск сервиса

```
public class BootReceiver extends BroadcastReceiver {  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Intent serviceIntent = new Intent(context, KMSDaemon.class);  
        context.startService(serviceIntent);  
    }  
}
```

Запуск сервиса

```
public class BootReceiver extends BroadcastReceiver {  
  
    @Override  
    public void onReceive(Context context, Intent intent) {  
        Intent serviceIntent = new Intent(context, KMSDaemon.class);  
        context.startService(serviceIntent);  
    }  
}
```


Tasks и Back Stack

В одном таске может быть несколько Activity одного класса.

- Запуск Activity A
- Запуск Activity B
- Запуск ещё одной Activity A

```
<activity
    android:name=".activities.MainActivity"
    android:label="@string/app_name"
    android:launchMode="standard" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
```

Activity A

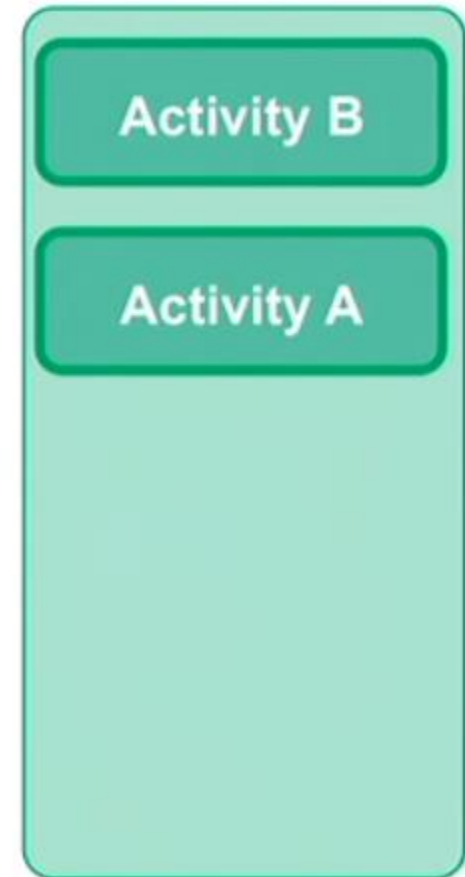
Activity B

Activity A

Tasks и Back Stack

launchMode="singleTop"

- Start Activity A
- Start Activity B
- Start Activity B
- Start Activity B

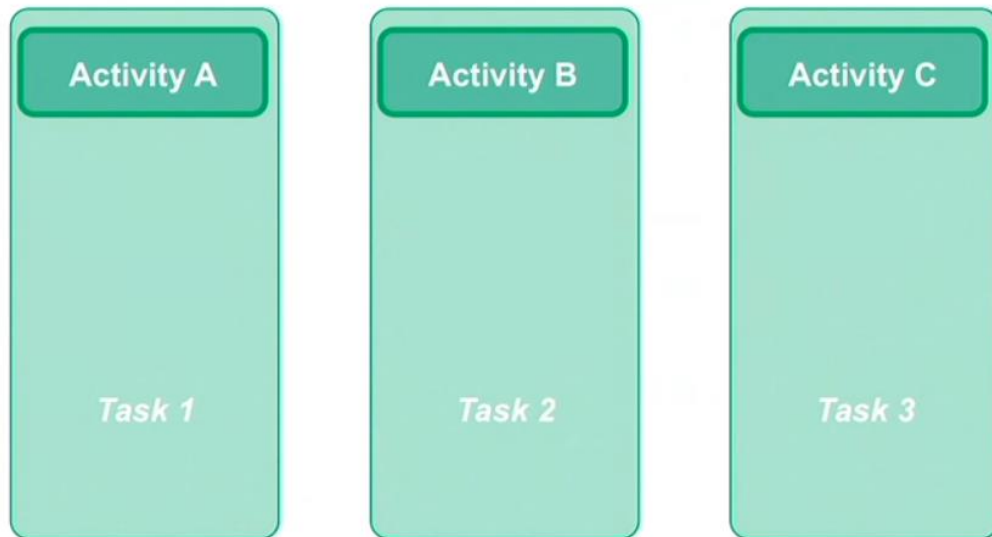


onNewIntent (Intent intent)

Tasks и Back Stack

launchMode="singleInstance"

- Start Activity A (singleInstance)
- Start Activity B
- Start Activity C с флагом NEW_TASK
- Start Activity A из Activity C



Что такое ресурс?

- **Значение** (например, заголовков экрана)
- **Файл** (например, музыкальный файл, файл описывающий компоновку экрана и т.д.)

Ресурсы можно модифицировать или предоставлять альтернативы без перемкомпиляции исходного кода

Первые шаги в Android Studio



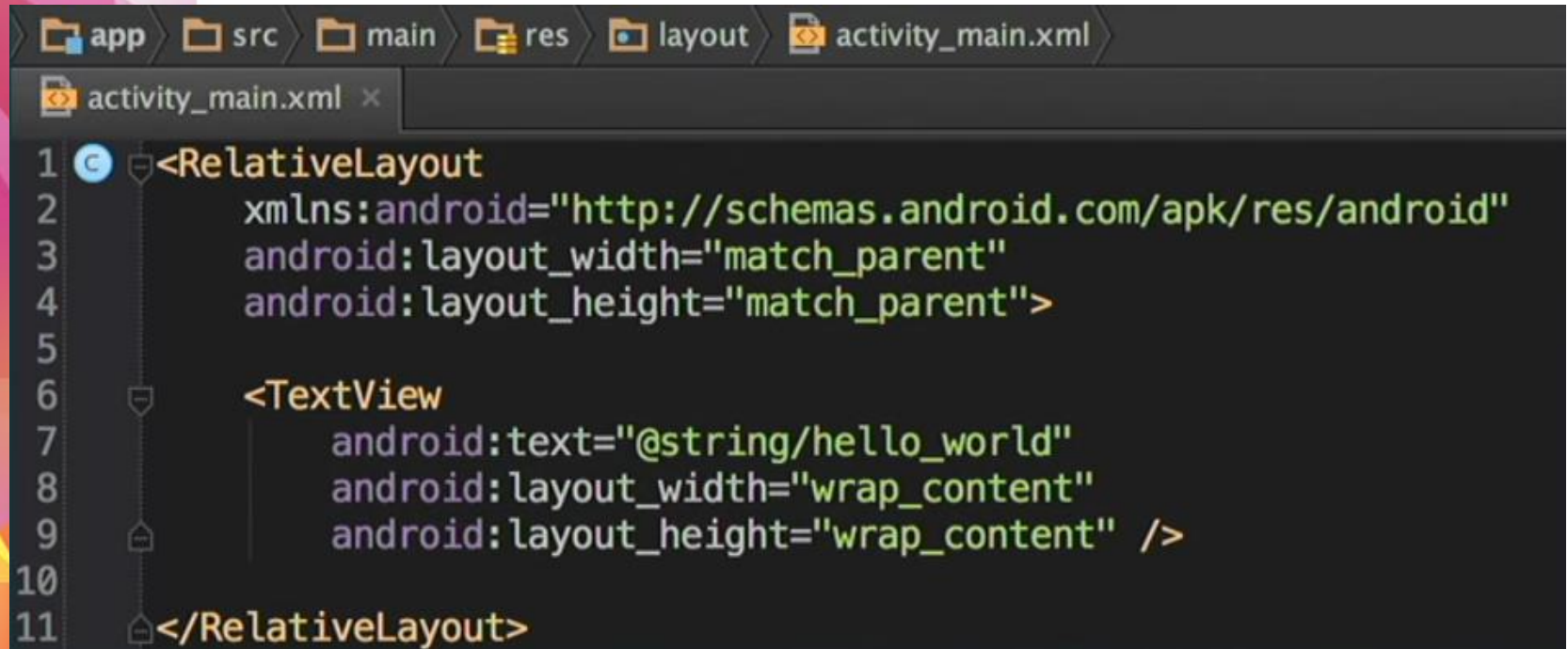
1. “Start a new Android Studio project”
2. На экране “Add an Activity to Mobile” выберите “Blank Activity”

Строковые ресурсы

```
app > src > main > res > values > strings.xml
strings.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3
4      <string name="app_name">Resources Demo</string>
5      <string name="hello_world">Hello world!</string>
6      <string name="action_settings">Settings</string>
7
8  </resources>
```

```
app > src > main > res > values-ru > strings.xml
ru/strings.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3
4      <string name="app_name">Пример использования ресурсов</string>
5      <string name="hello_world">Здравствуй Мир!</string>
6      <string name="action_settings">Настройки</string>
7
8  </resources>
```


Ресурсы компоновки



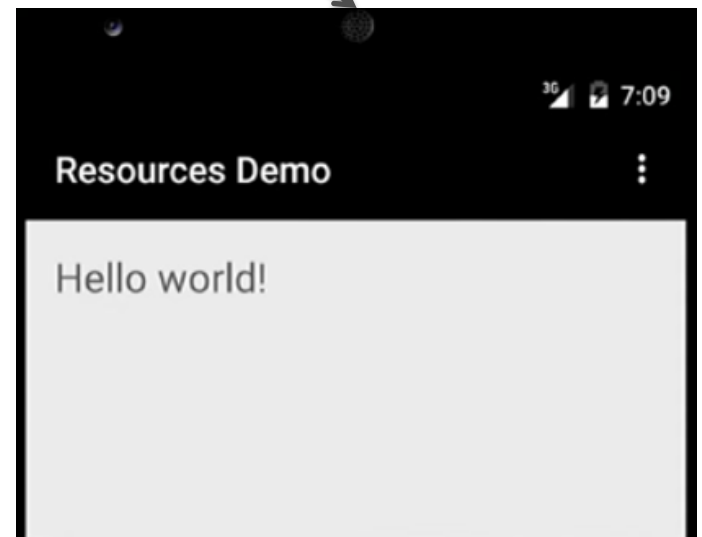
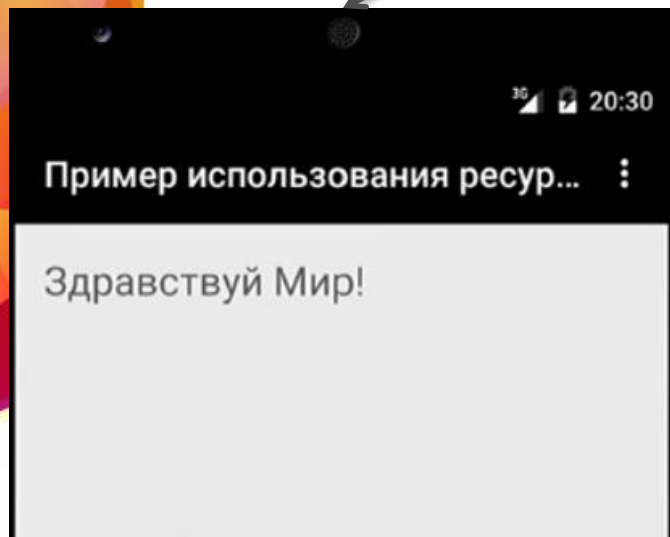
```
1 <RelativeLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <TextView
7         android:text="@string/hello_world"
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content" />
10
11 </RelativeLayout>
```

А что в Java?

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
}
```

RU

default



А что внутри?

```
1 package com.example.resapp;
2
3 public final class R {
4
5     public static final class layout {
6         public static final int activity_main=0x7f030017;
7     }
8
9     public static final class string {
10         public static final int action_settings=0x7f0a0010;
11         public static final int app_name=0x7f0a0011;
12         public static final int hello_world=0x7f0a0012;
13     }
14 }
```

Обращение к ресурсам приложения

- Через XML

```
@string/hello_world
```

- Через код Java

```
R.string.hello_world
```

Синтаксис ссылок

Java

`[package.]R.{type}.{name}`

- R.drawable
- R.id
- R.layout
- R.string
- R.attr
- R.plural
- R.array
- ...

XML

`@[package:]{type}\{name}`

- drawable
- id
- layout
- string
- attr
- plurals
- string-array
- ...

Синтаксис ссылок

```
app > src > main > res > layout > activity_main.xml
activity_main.xml x
1  <RelativeLayout
2      xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent">
5
6      <ImageView
7          android:src="@drawable/im"
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content" />
10
11 </RelativeLayout>
```



res/drawable/im.png

Синтаксис ссылок через Java

```
app > src > main > res > layout > activity_main.xml
activity_main.xml x
1  <RelativeLayout
2      xmlns:android="http://schemas.android.com/apk/res/android"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent">
5
6      <ImageView
7          android:id="@+id/imageWithin"
8          android:layout_width="wrap_content"
9          android:layout_height="wrap_content" />
10
11 </RelativeLayout>
```



res/drawable/im.png

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    ImageView im = (ImageView) findViewById(R.id.imageWithin);
    im.setImageDrawable(getResources().getDrawable(R.drawable.im));
}
```

Id

`@+id/your_best_id` – создает новый id

`@id/another_best_id` – использует уже созданный

Обзор структуры каталога ресурсов

```
MyProject/  
  java/com/example/MainActivity.java  
  res/  
    animator/  
      *.xml  
    anim/  
      *.xml  
    color/  
      *.xml  
    drawable/  
      *.{png|9.png|jpg|gif}  
      *.xml  
    layout/  
      *.xml  
    menu/  
      *.xml  
    raw/  
      *.*  
    values/  
      *.xml  
    xml/  
      *.xml  
  assets/  
    *.*/*.*
```

Основные типы ресурсов

```
MyProject/  
  java/com/example/MainActivity.java  
  res/  
    drawable/  
      myimage.png - изображение  
    layout/  
      activity_main.xml - компоновка пользовательского интерфейса  
    values/  
      colors.xml - цвета  
      dims.xml - размерности  
      strings.xml - строки  
      styles.xml - стили  
  assets/  
    *.*/*.* - низкоуровневые ресурсы
```

Изображения

Расположение файла: `res/drawable/{drawable_name}.{png|jpg|gif|9.png|xml}`

Ссылка в Java: `[package.]R.drawable.{drawable_name}`

Ссылка в XML: `@[package:]drawable/{drawable_name}`

XML:

```
<Button
    android:id="@+id/helloButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello_button"
    android:background="@drawable/btn_background" />
```

```
<ImageView
    android:src="@drawable/im"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

Java:

```
Drawable background = getResources().getDrawable(R.drawable.btn_background);

button.setBackground(background);
/* или проще */
button.setBackgroundResource(R.drawable.btn_background);
```


КОМПОНОВКИ

Расположение файла: `res/layout/{layout_name}.xml`

Ссылка в Java: `[package.]R.layout.{layout_name}`

Ссылка в XML: `@[package:]layout/{layout_name}`

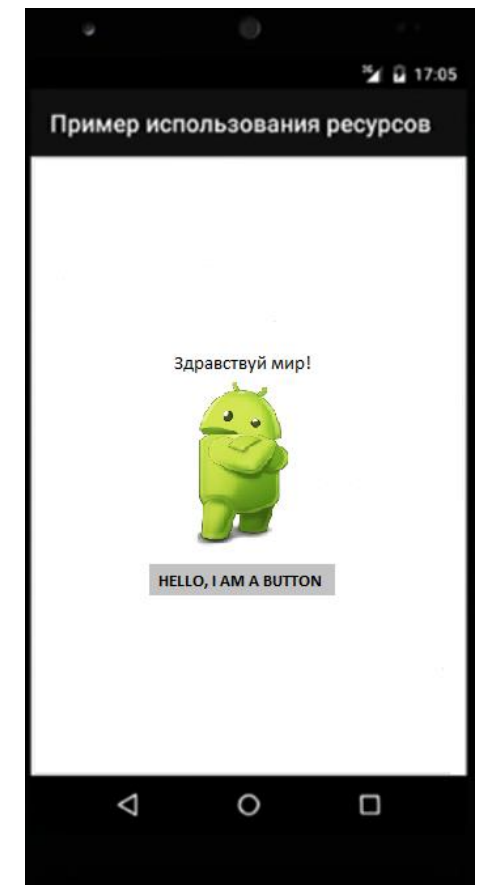
Структура в XML:

```
<!--  
  <ViewGroup  
    [ViewGroup-specific attributes] >  
    <View  
      [View-specific attributes] />  
  </ViewGroup >  
-->
```

```
1 <RelativeLayout  
2     xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="match_parent">  
5  
6     <ImageView  
7         android:id="@+id/imageWithin"  
8         android:layout_width="wrap_content"  
9         android:layout_height="wrap_content" />  
10  
11 </RelativeLayout>
```


КОМПОНОВКИ

```
app src main res layout activity_main.xml
activity_main.xml x
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:orientation="vertical"
6   android:gravity="center">
7
8   <TextView
9     android:id="@+id/helloText"
10    android:layout_width="wrap_content"
11    android:layout_height="wrap_content"
12    android:text="@string/hello_world"/>
13
14   <ImageView
15     android:id="@+id/imageWithim"
16     android:layout_width="wrap_content"
17     android:layout_height="wrap_content"
18     android:src="@drawable/im"/>
19
20   <Button
21     android:id="@+id/helloButton"
22     android:layout_width="wrap_content"
23     android:layout_height="wrap_content"
24     android:text="@string/hello_button" />
25
26 </LinearLayout>
```

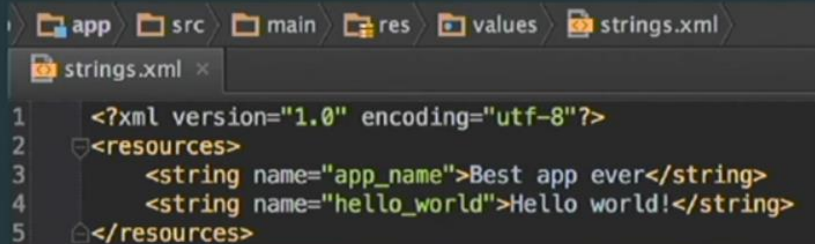


Строки

Расположение файла: `res/values/{strings}.xml`

Ссылка в Java: `[package.]R.string.{string_name}`

Ссылка в XML: `@[package:]string/{string_name}`



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <string name="app_name">Best app ever</string>
4   <string name="hello_world">Hello world!</string>
5 </resources>
```

XML:

```
<TextView
    android:text="@string/hello_world"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

Java:

```
String helloWorld = getString(R.string.hello_world);
```

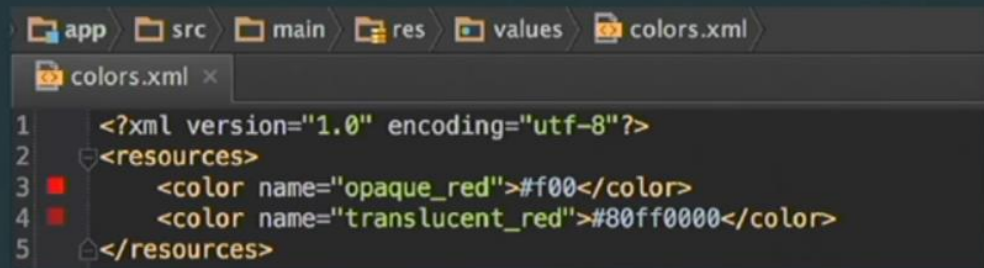
Цвета

Расположение файла: `res/values/{colors}.xml`

Ссылка в Java: `[package.]R.color.{color_name}`

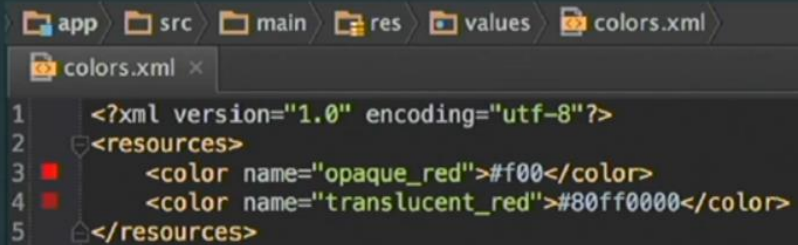
Ссылка в XML: `@[package:]color/{color_name}`

- #RGB
- #ARGB
- #RRGGBB
- #AARRGGBB



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <color name="opaque_red">#f00</color>
4   <color name="translucent_red">#80ff0000</color>
5 </resources>
```

Цвета



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <color name="opaque_red">#f00</color>
4   <color name="translucent_red">#80ff0000</color>
5 </resources>
```

XML:

```
<TextView
    android:text="@string/hello_world"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/opaque_red"/>
```

Java:

```
int red = getResources().getColor(R.color.opaque_red);
```

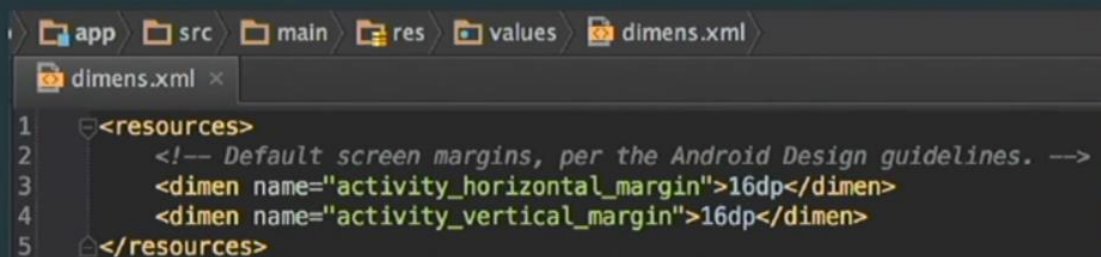
Размерности

Расположение файла: `res/values/{dimens}.xml`

Ссылка в Java: `[package.]R.dimen.{dimension_name}`

Ссылка в XML: `@[package:]dimen/{dimension_name}`

- px - пиксели
- in - дюймы
- mm - миллиметры
- pt - пункты
- **dp** - пиксели, не зависящие от разрешения
- **sp** - пиксели не зависящие от масштаба



The screenshot shows an IDE interface with a breadcrumb trail at the top: `app > src > main > res > values > dimens.xml`. Below the trail, the file `dimens.xml` is open. The XML content is as follows:

```
1 <resources>
2   <!-- Default screen margins, per the Android Design guidelines. -->
3   <dimen name="activity_horizontal_margin">16dp</dimen>
4   <dimen name="activity_vertical_margin">16dp</dimen>
5 </resources>
```

Размерности

```
app > src > main > res > values > dims.xml
dims.xml x
1 <resources>
2   <dimen name="hello_padding">16dp</dimen>
3   <dimen name="hello_text_size">14sp</dimen>
4 </resources>
```

XML:

```
<TextView
    android:text="@string/hello_world"
    android:padding="@dimen/hello_padding"
    android:textSize="@dimen/hello_text_size"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

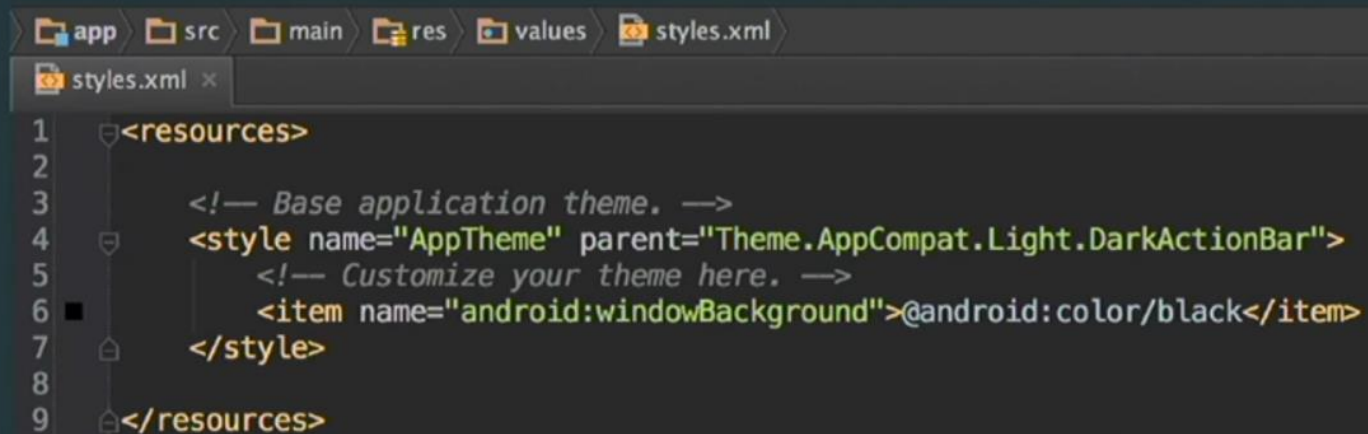
Java:

```
float textSize = getResources().getDimension(R.dimen.hello_text_size);
```


Стили

Расположение файла: `/res/values/{styles}.xml`

Ссылка в XML: `@[package:]style/{style_name}`



The screenshot shows an IDE window with the file path `app > src > main > res > values > styles.xml`. The file is named `styles.xml`. The code inside the file is as follows:

```
1 <resources>
2
3     <!-- Base application theme. -->
4     <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
5         <!-- Customize your theme here. -->
6         <item name="android:windowBackground">@android:color/black</item>
7     </style>
8
9 </resources>
```

Стили

```
app > src > main > res > values > styles.xml
styles.xml x
1  <resources>
2
3  <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
4      <item name="android:windowBackground">@android:color/black</item>
5  </style>
6
7  <style name="BigRedText" parent="@style/BigText">
8      <item name="android:textColor">@color/custom_red</item>
9  </style>
10
11 <style name="BigText">
12     <item name="android:textSize">@dimen/large_text_size</item>
13     <item name="android:textColor">@android:color/white</item>
14 </style>
15
16 </resources>
```

XML:

```
<TextView
    android:text="@string/hello_world"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="@style/BigRedText" />
```

Asserts

Расположение файла: `/assets/*.*/*.*`

Доступ в Java: через `AssetManager`

```
public String readTestFile() throws IOException {
    AssetManager assetManager = getAssets();
    InputStream is = null;
    try {
        is = assetManager.open("test.txt");
        String text = convertStreamToString(is);

        return text;
    } finally {
        if (is != null) {
            is.close();
        }
    }
}
```

Альтернативные ресурсы

Множество альтернативных ресурсов:

```
/res/values/strings.xml
```

```
/res/values-ru/strings.xml
```

```
/res/values-fr-notnight/strings.xml
```

```
...
```


```
/res/drawable[-qualifiers]/strings.xml
```

Один идентификатор:

```
R.string.hello_world
```

Модификаторы ресурсов

Приоритет для Android

- 
1. Мобильный код страны и код оператора
 2. Язык и регион
 3. Layout direction
 4. Размер экрана
 5. Тип экрана
 6. Ориентация экрана
 7. Тип стыковки и UI mode
 8. Ночь или день
 9. Разрешение экрана (плотность пикселей)
 10. Тип сенсорного экрана
 11. Вид и доступность клавиатуры
 12. Клавиши навигации скрыты или видны
 13. Тип навигационного устройства
 14. Уровень API

Мобильный код страны и код оператора

`mcc{country_code}` – мобильный код страны
`mnc{network_code}` – мобильный код сети (опционально)

Пример:

`mcc250` – Россия

`mcc250-mnc01` – Россия и МТС

Язык и регион

`{lang_code}` – код страны (ISO 639-1)

`r{region_code}` – код региона (опционально, ISO 3166-1-alpha-2)

Пример:

`ru` – Русский язык

`fr_rCA` – Французский язык, Канада

Layout direction

`ltr`

– слева на право (используется по-умолчанию)

`rtl`

– справа на лево (для Арабских стран)

Layout direction

`ltr`

– слева на право (используется по-умолчанию)

`rtl`

– справа на лево (для Арабских стран)

Размер экрана

- `sw{shortest_width}dp` – наименьшая ширина, не зависит от ориентации экрана
- `w{available_width}dp` – доступная ширина
- `h{available_height}dp` – доступная высота

Пример:

`sw600dp` - 7" планшет

`sw720dp` - 10" планшет

Доступно с API 13 (Android 3.2)

Размер экрана

- `sw{shortest_width}dp` – наименьшая ширина, не зависит от ориентации экрана
- `w{available_width}dp` – доступная ширина
- `h{available_height}dp` – доступная высота

Пример:

`sw600dp` - 7" планшет

`sw720dp` - 10" планшет

Доступно с API 13 (Android 3.2)

Тип экрана

`long` – WQVGA, WVGA, FWVGA

`notlong` – QVGA, HVGA, VGA

Не зависят от ориентации экрана

Ориентация экрана

`port` – портретная

`land` – ландшафтная

UI mode

car

– стыковка с мультимедиа-системой автомобиля

desk

– стыковка с настольным доком

television

– стыковка с TV (с API 13, Android 3.2)

appliance

– устройство не имеет экрана

watch

– устройство имеет экран и носится
на руке (часы, с API 20, Android 4.4W)



Ночь или день

night - НОЧЬ



notnight - день



Разрешение экрана

	Плотность пикселей на дюйм:	Соотношение dp к px:
<code>ldpi</code>	- 120dpi,	1 dp = 3/4 px
<code>mdpi</code>	- 160dpi,	1 dp = 1 px
<code>hdpi</code>	- 240dpi,	1 dp = 1.5 px
<code>xdpi</code>	- 320dpi,	1 dp = 2 px
<code>xxdpi</code>	- 480dpi (с API 16, Android 4.1)	1 dp = 3 px
<code>xxxdpi</code>	- 640dpi (с API 18, Android 4.3)	1 dp = 4 px
<code>nodpi</code>	- "как есть"	
<code>tvdpi</code>	- 1.33 * mdpi	1 dp = 1.33 px

Тип сенсорного экрана

- | | |
|----------------------|---|
| <code>notouch</code> | – без сенсорного экрана |
| <code>finger</code> | – сенсорный экран |
| <code>stylus</code> | – стилус (deprecated с API 16, Android 4.1) |

Вид и доступность клавиатуры

Зависят от того, скрыта или показана клавиатура:

`keysexposed` – раскрыта программная или аппаратная клавиатура

`keyshidden` – аппаратная клавиатура скрыта, программная отсутствует

Не зависят:

`keysoft` – доступна программная клавиатура

`nokeys` – отсутствует аппаратная клавиатура

`qwerty` – доступна аппаратная qwerty клавиатура

`12key` – доступна аппаратная 12key клавиатура

Видимость клавиш навигации

`navexposed` – видны

`navhidden` – скрыты

Тип навигационного устройства

<code>nonav</code>	- только сенсорный экран
<code>dpad</code>	- direction pad (d-pad)
<code>trackball</code>	- trackball
<code>wheel</code>	- колесо

Уровень API

`v{android_api_ver}` – версия API Android

Пример:

`v7` - Android 2.1

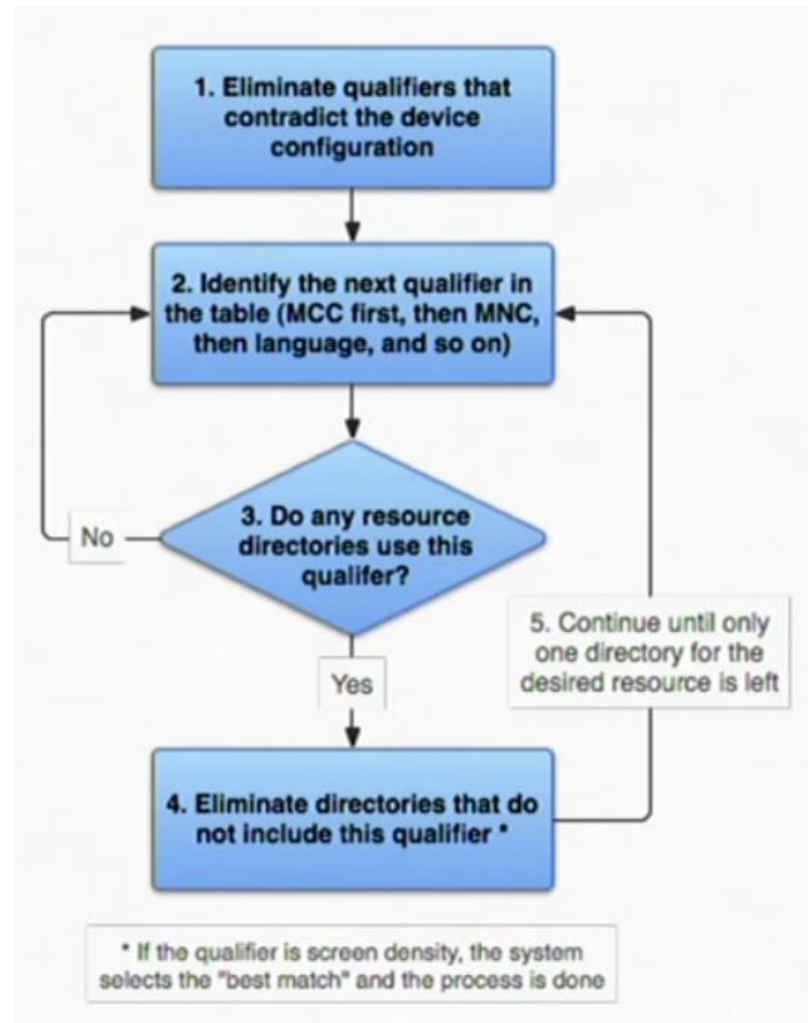
`v14` - Android 4

`v21` - Android 5

Список версий:

<http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>

Алгоритм выбора подходящего ресурса



Пример поиска

Дано: RU, портретный режим, hdpi

0. Доступные ресурсы:

`/res/drawable/cat.png`

`/res/drawable-port/cat.png`

`/res/drawable-ru/cat.png`

`/res/drawable-land-hdpi/cat.png`

`/res/drawable-xdpi/cat.png`

1. Убираются противоречия (land):

`/res/drawable/cat.png`

`/res/drawable-port/cat.png`

`/res/drawable-ru/cat.png`

~~`/res/drawable-land-hdpi/cat.png`~~

`/res/drawable-xdpi/cat.png`

2-5. Выбирается квалификатор с максимальным приоритетом (ru):

~~`/res/drawable/cat.png`~~

~~`/res/drawable-port/cat.png`~~

`/res/drawable-ru/cat.png`

~~`/res/drawable-land-hdpi/cat.png`~~

~~`/res/drawable-xdpi/cat.png`~~

Исключение: квалификаторы screen density остаются на этом шаге

Изменение конфигурации в Runtime

Активность разрушается и создается вновь при изменении конфигурации, если данный тип изменения не указан в «configChanges» (особые случаи: «screenSize» и «smallestScreenSize»)

```
<activity
    android:configChanges=[ "mcc",
                           "mnc",
                           "locale",
                           "touchscreen",
                           "keyboard",
                           "keyboardHidden",
                           "navigation",
                           "screenLayout",
                           "fontScale",
                           "uiMode",
                           "orientation",
                           "screenSize",
                           "smallestScreenSize",
                           "layoutDirection" ] />
```


Собственный обработчик изменения конфигурации

AndroidManifest.xml:

```
<activity
    android:name=".MainActivity"
    android:label="@string/app_name"
    android:configChanges="orientation|screenSize">
```

Активность:

```
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);

    // Проверяем ориентацию экрана
    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
        Toast.makeText(this, "landscape", Toast.LENGTH_SHORT).show();
    } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {
        Toast.makeText(this, "portrait", Toast.LENGTH_SHORT).show();
    }
}
```

```
int orientation = getResources().getConfiguration().orientation;
```

Собственный обработчик изменения конфигурации

AndroidManifest.xml:

```
<activity
    android:name=".MainActivity"
    android:label="@string/app_name"
    android:configChanges="orientation|screenSize">
```

Активность:

```
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);

    // Проверяем ориентацию экрана
    if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
        Toast.makeText(this, "landscape", Toast.LENGTH_SHORT).show();
    } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {
        Toast.makeText(this, "portrait", Toast.LENGTH_SHORT).show();
    }
}
```

```
int orientation = getResources().getConfiguration().orientation;
```