



Android.Level1

Урок 2. Ресурсы в приложении Android



Урок 2. Ресурсы в Android

- Понятие ресурса приложения
- Типы ресурсов
- Работа с ресурсами в среде Eclipse
- Строки
- Размеры и цвета в Android
- Массивы строк
- Графика



Понятие ресурса

- **Ресурсы** – это один из основных компонентов Android приложения.
- **Ресурсы** – это объекты приложения, которые хранятся за пределами исходного кода (во внешних файлах).

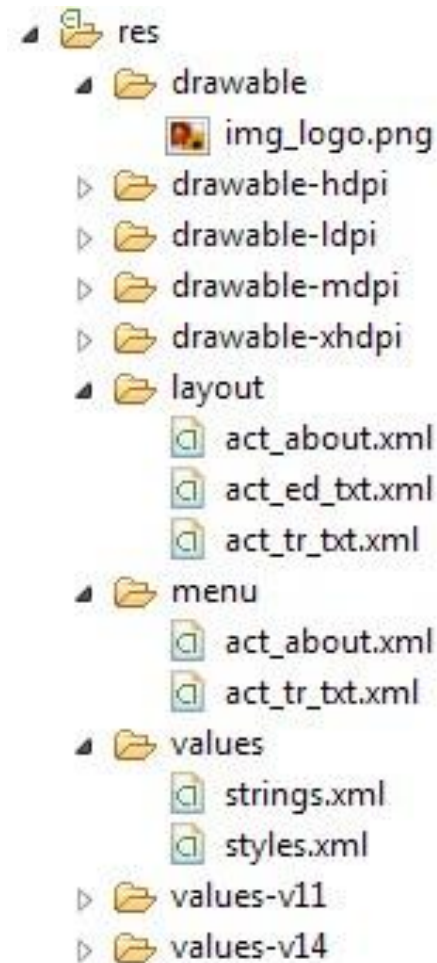


Типы ресурсов

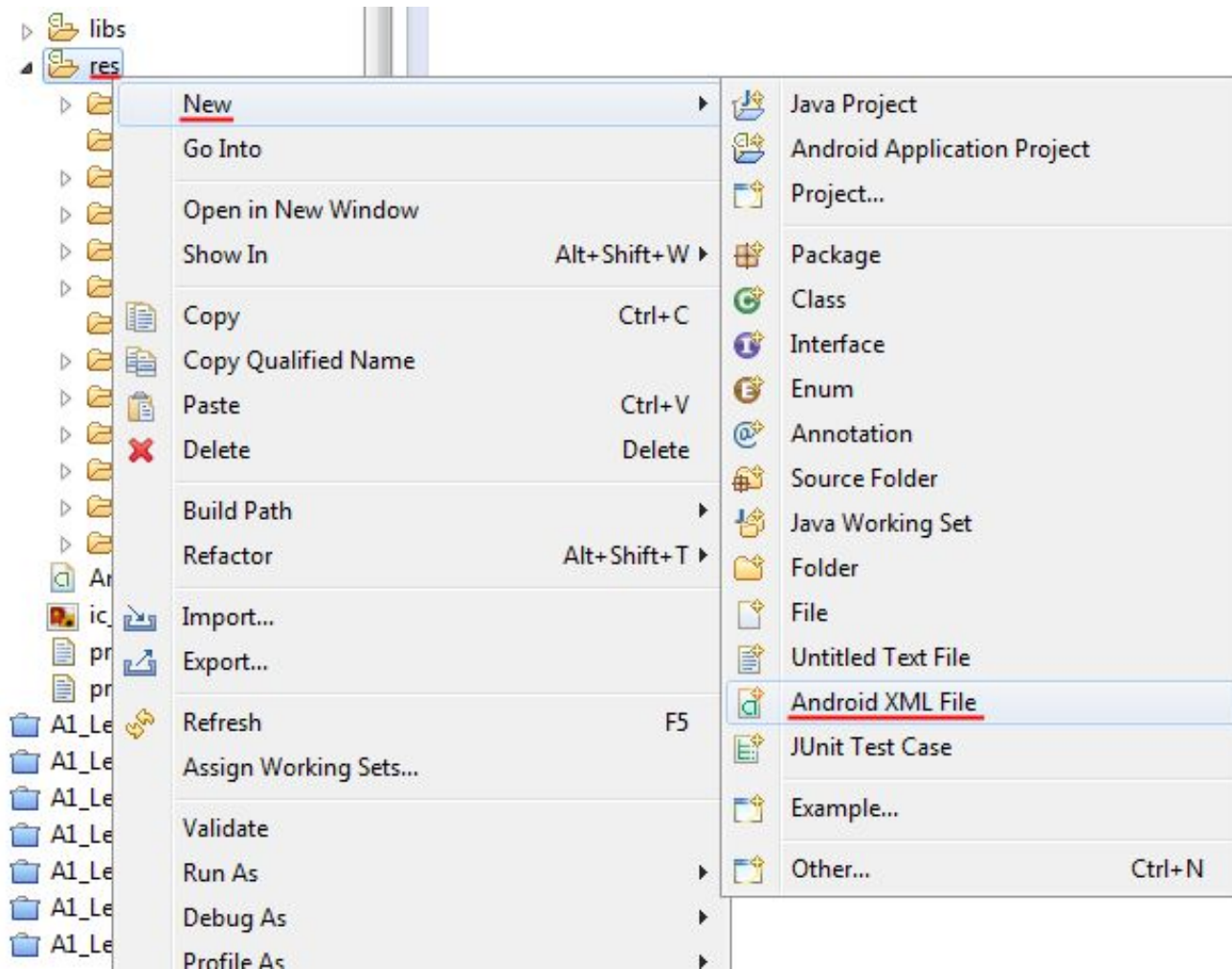
- Разметка (UI, меню – xml файлы)
- Строки и массивы строк
- Цвета
- Размеры
- Графика

Работа с ресурсами в Eclipse

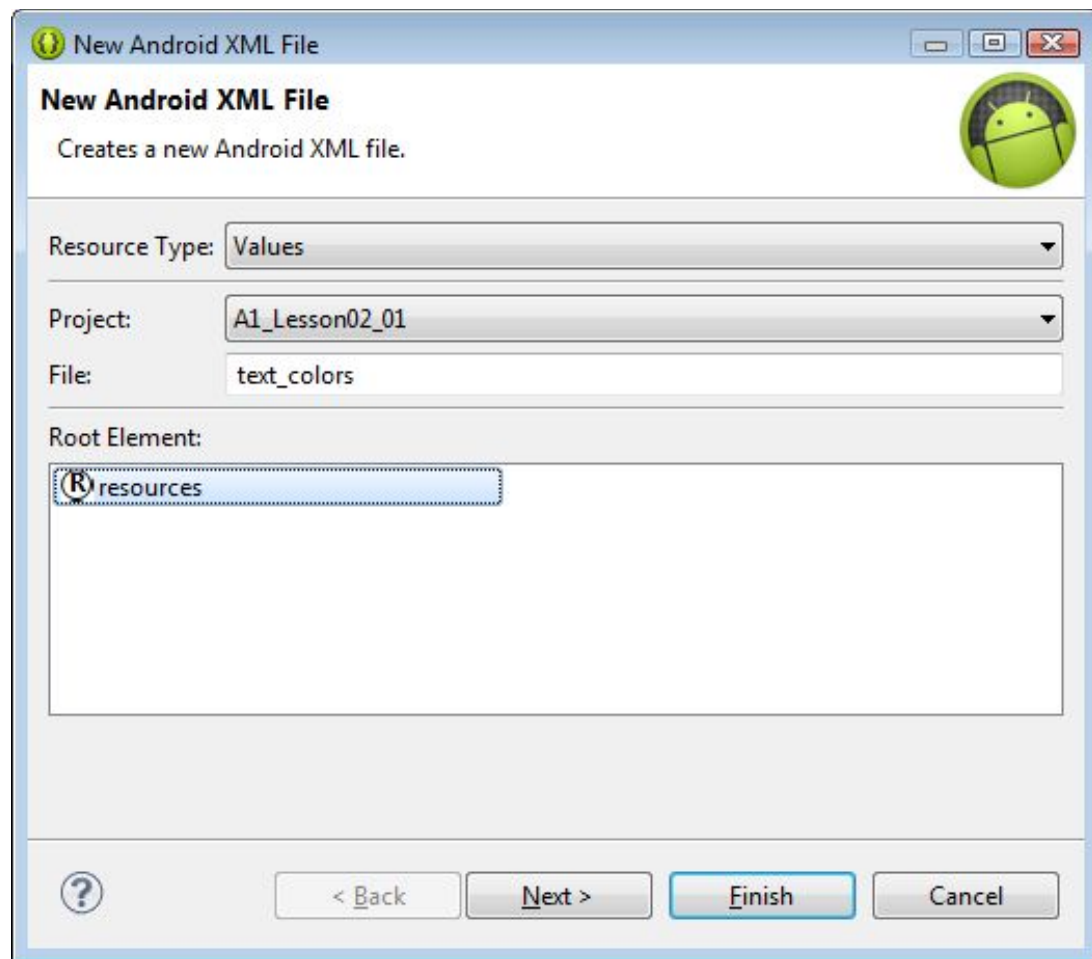
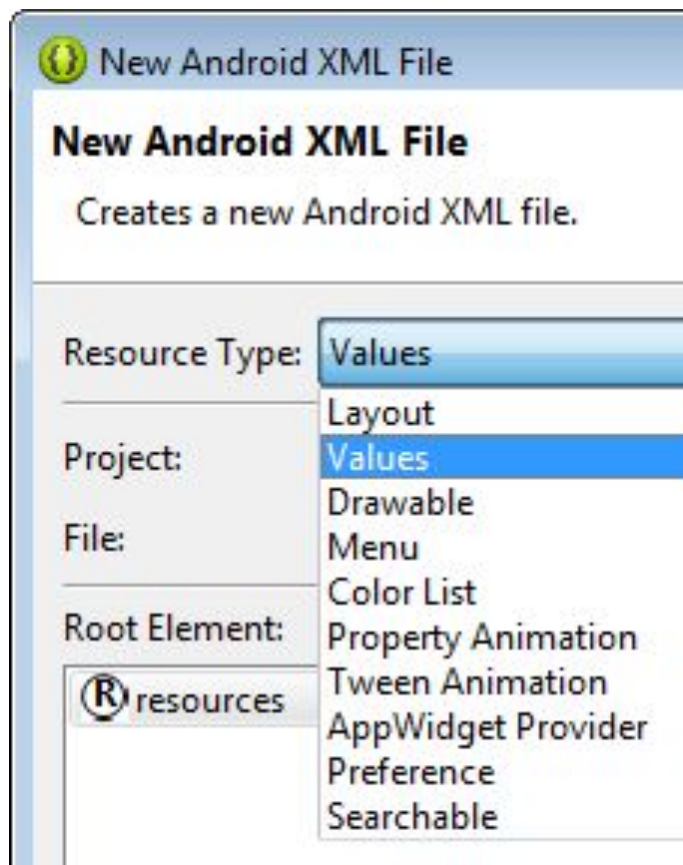
- *res*
- drawable-xxx
- layout-xxx
- menu-xxx
- values-xxx
- *xxx -квалификатор*



Создание ресурсов



Создание ресурсов



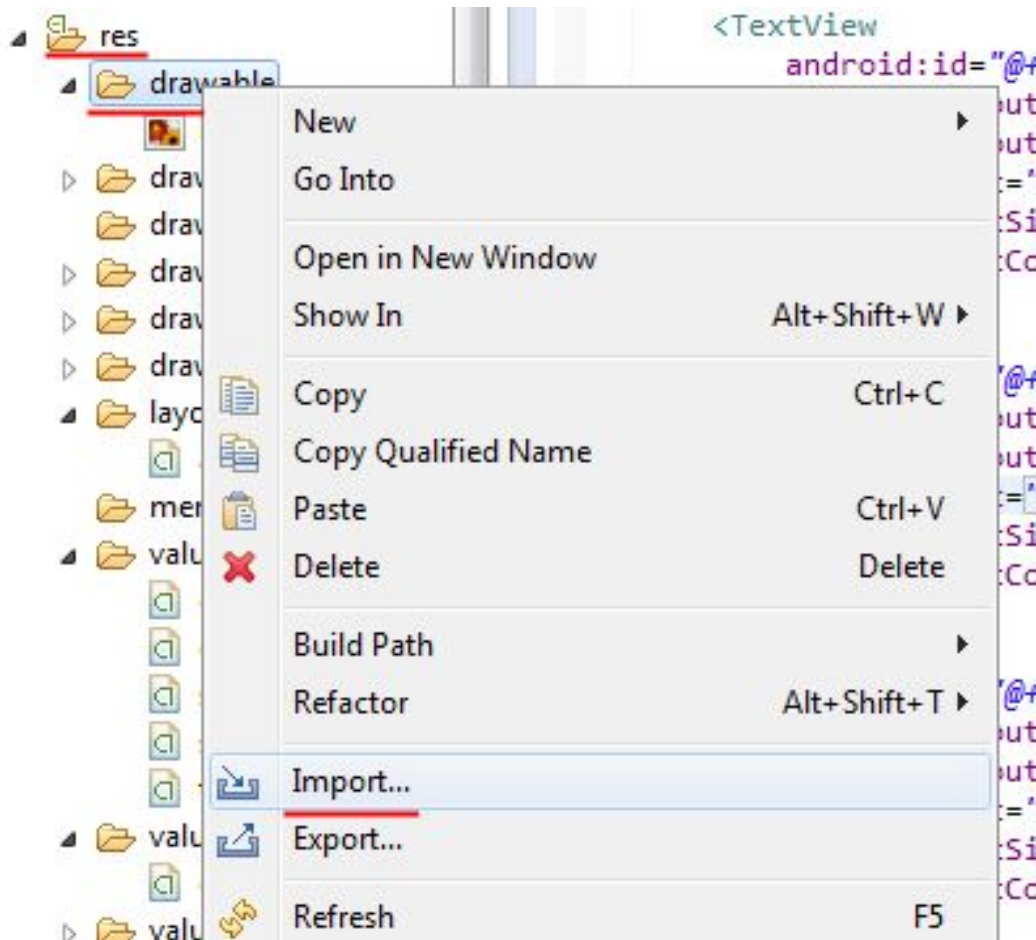
Создание ресурсов

- layout
 - activity_main.xml
- menu
- values
 - colors.xml
 - dimens.xml
 - strings.xml
 - styles.xml
 - text_colors.xml
- values-land

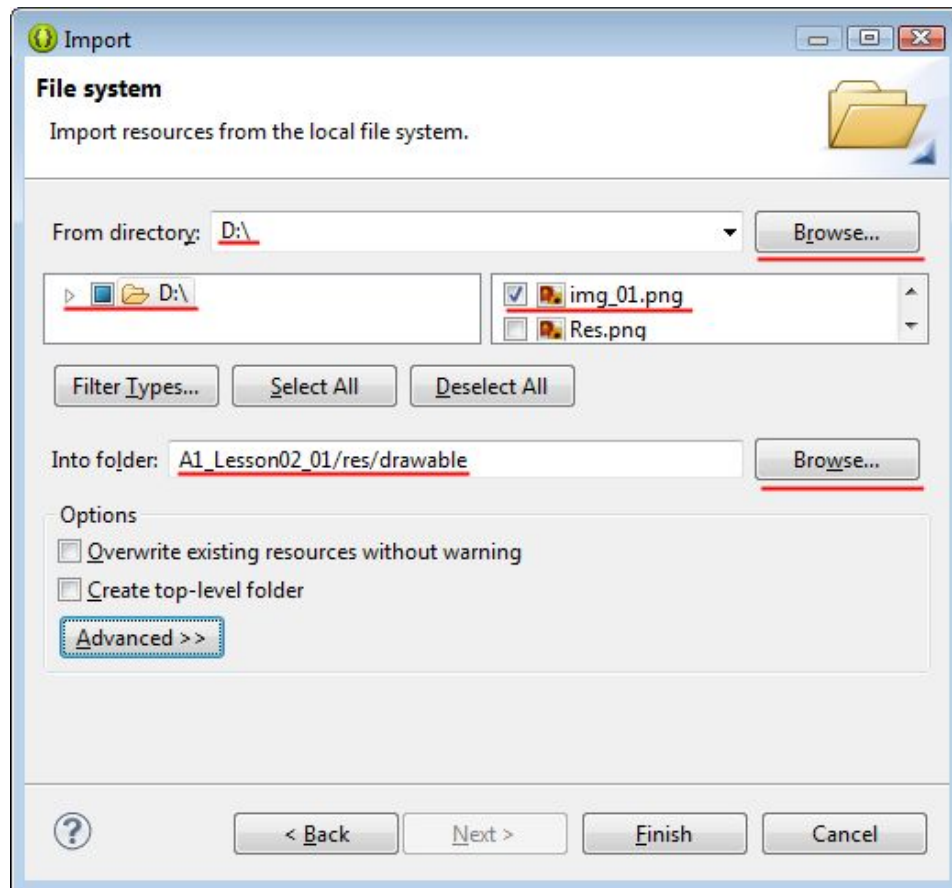
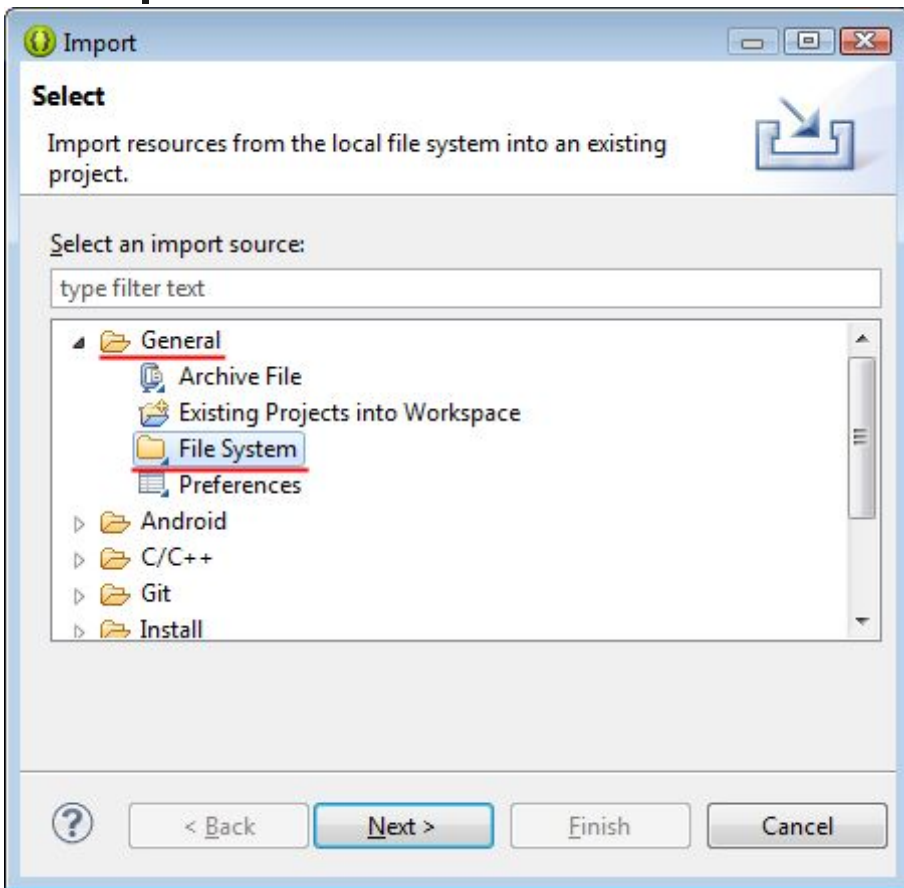
```
text_colors.xml ⌵  
|?xml version="1.0" encoding="utf-8"?>  
<resources>  
  
</resources>
```

```
text_colors.xml ⌵  
|?xml version="1.0" encoding="utf-8"?>  
<resources>  
  
    <color name="text_color_r">#FFFF0000</color>  
    <color name="text_color_g">#FF00FF00</color>  
    <color name="text_color_b">#FF0000FF</color>  
  
</resources>
```


Импорт внешних файлов



Импорт внешних файлов





Доступ к UI элементам из кода

```
public class MainActivity extends Activity {

    /* Fields for store links to UI components */
    private TextView tvText0 = null;
    private TextView tvText1 = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        /* Invoke a parent method */
        super.onCreate(savedInstanceState);

        /* Load UI from resources */
        setContentView(R.layout.activity_main);

        /* Load UI components */
        tvText0 = (TextView) this.findViewById(R.id.tvText0);
        tvText1 = (TextView) this.findViewById(R.id.tvText1);
    }
}
```



Доступ к ресурсам из кода

- Класс R
- Подклассы
 - color
 - dimen
 - drawable
 - id
 - layout
 - menu
 - string

```
public final class R {  
    public static final class attr {  
    }  
    public static final class color {  
        /** Text attributes  
        */  
        public static final int simple_text_0_color=0x7f040005;  
        /** Text attributes  
        Text attributes  
        */  
        public static final int simple_text_1_color=0x7f040000;  
        public static final int simple_text_2_color=0x7f040001;  
        public static final int simple_text_3_color=0x7f040002;  
        public static final int simple_text_4_color=0x7f040003;  
        public static final int simple_text_5_color=0x7f040004;  
        public static final int text_color_b=0x7f040008;●  
        public static final int text_color_g=0x7f040007;●  
        public static final int text_color_r=0x7f040006;●  
    }  
    public static final class dimen {  
        /** Default screen margins, per the Android Design guidelines.
```



Доступ к ресурсам из кода

```
/* Load string resource */
String str1 = this.getString(R.string.app_name);
String str2 = this.getResources().getString(R.string.app_name);

/* Using strings */
tvText0.setText(str1);
tvText1.setText(str2);

/* Load color resource */
int color_r = this.getResources().getColor(R.color.text_color_r);
int color_b = this.getResources().getColor(R.color.text_color_b);

/* Using colors */
tvText0.setTextColor(color_r);
tvText1.setTextColor(color_b);
```



Доступ к ресурсам из XML

свойство = "@тип_ресурса/имя_ресурса"

```
<TextView
    android:id="@+id/tvText0"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/simple_text_0"
    android:textSize="@dimen/simple_text_0_size"
    android:textColor="@color/simple_text_0_color"/>
```

```
<TextView
    android:id="@+id/tvText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/simple_text_1"
    android:textSize="@dimen/simple_text_0_size"
    android:textColor="@color/simple_text_1_color" />
```



Типы ресурсов – Строки

- Файлы: *res/values-xxx/strings.xml*
- Доступ из кода через класс R.string
- Доступ из XML:
`android:text = "@string/res_name"`
- Пример определения:

```
<!-- Name of this application -->  
<string name="app_name">A1_L02_01</string>
```

```
<!-- Simple text -->  
<string name="simple_text_0">Simple text 0</string>  
<string name="simple_text_1">Simple text 1</string>  
<string name="simple_text_2">Simple text 2</string>  
<string name="simple_text_3">Simple text 3</string>  
<string name="simple_text_4">Simple text 4</string>  
<string name="simple_text_5">Simple text 5</string>
```



Задание 1. Ресурсы

- Создать приложение AboutMe (ФИО, контакты, профессиональный навыки).
- Добавить в ресурсы приложения сведения о себе:
 - Фамилия, Имя, Отчество
 - E-Mail



Типы ресурсов – Цвет

- Задается в RGB (Red Green Blue) формате + прозрачность (FF – полностью не прозрачный, 00 – полностью прозрачный), в шестнадцатеричном виде.
- Примеры определения:
 - #RGB (#F00 – красный цвет)
 - #ARGB (#8F00 – прозрачный красный цвет)
 - #RRGGBB (#00FF00 – зеленый цвет)
 - #AARRGGBB (#AA00FF00 – прозрачный зеленый цвет)



Типы ресурсов – Цвет

- Файлы
 - *res/values-xxx/strings.xml*
 - *res/values-xxx/colors.xml*
- Доступ из кода через класс R.color
- Доступ из XML:
`android:textColor="@color/txt_color"`

Типы ресурсов – Цвет

- ▷ layout
- ▷ menu
- ▲ values
 - colors.xml
 - dimens.xml
 - strings.xml
 - styles.xml
 - text_colors.xml
- ▲ values-land
 - colors.xml
- ▷ values-sw600dp

```
<resources>
```

```
  <color name="text_color_r">#FFFF0000</color>  
  <color name="text_color_g">#FF00FF00</color>  
  <color name="text_color_b">#FF0000FF</color>
```

```
</resources>
```

```
/* Using color resource */  
tvText0.setTextColor(this.getResources().getColor(  
    R.color.text_color_g));
```



Задание 2. Ресурсы

- Добавить в ресурсы приложения несколько значений цветов:
 - Белый
 - Черный
 - Красный
 - Зеленый
 - Синий
 - Желтый



Типы ресурсов – Размеры

- ***px*** (*pixels*) – размер в пикселях
- ***dip/dp*** (*device/density-independent*) – независимые от устройства пиксели
- ***sp*** (*scale pixels*) – масштабированные пиксели (то же что и *dp*, только более точные), полезны при работе с текстом.
- ***pt*** (*points*) – размер в пунктах (задание размера текста). 1 пункт = 1/72 дюйма.
- ***in*** (*inches*) – размер в дюймах.
- ***mm*** (*millimeters*) – размер в миллиметрах.



Типы ресурсов. Размеры.

- Файлы:
 - *res/values-xxx/strings.xml*
 - *res/values-xxx/dimens.xml*
- Доступ из кода через класс R.dimen
- Доступ из XML:
`android:textSize = "@dimen/res_name"`
- Пример определения:

```
<resources>  
  
  <!-- Default screen margins, per the Android Design guidelines. -->  
  <dimen name="activity_horizontal_margin">16dp</dimen>  
  <dimen name="activity_vertical_margin">16dp</dimen>  
  <dimen name="text_size">10pt</dimen>  
  
</resources>
```



Задание 3. Ресурсы

- Добавить в ресурсы приложения несколько значений размеров текста (в пунктах):
 - Header Level 1 (заголовок первого уровня)
 - Header Level 2 (заголовок второго уровня)
 - Header Level 3 (заголовок третьего уровня)
 - Simple Text (простой текст)



Типы ресурсов – Массивы

- Файлы: *res/values-xxx/strings.xml*
- Доступ из кода через класс `R.array`
- Пример определения:

```
<!-- Array of items -->
<string-array name="my_str_array">
    <item>Item 00</item>
    <item>Item 01</item>
    <item>Item 02</item>
    <item>Item 03</item>
    <item>Item 04</item>
    <item>Item 05</item>
    <item>Item 06</item>
    <item>Item 07</item>
    <item>Item 08</item>
    <item>Item 09</item>
</string-array>
```




Типы ресурсов – Массивы

```
/* Work with array */
String[] array = this.getResources().getStringArray(
    R.array.my_str_array);

/* Convert array to string */
StringBuilder sBuilder = new StringBuilder();

for (int i = 0; i < array.length; i++) {
    sBuilder.append(array[i] + "\n");
}

tvText1.setText(sBuilder.toString());
```



Задание 3. Ресурсы

- Добавить в ресурсы приложения сведения о знаниях и навыках в сфере IT (в виде массива).



Типы ресурсов - Графика

- Папки: *res/drawable-xxx/*
- Доступ из кода через класс `R.drawable`
- Перед началом использования необходимо импортировать файлы изображений в проект
- Пример использования: (след. слайд)

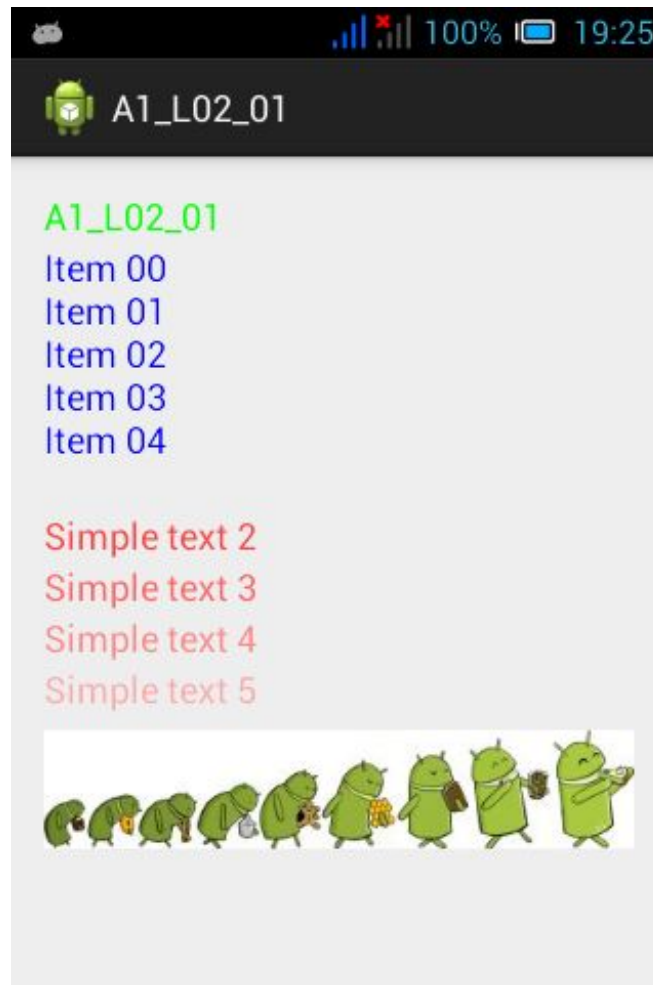
Типы ресурсов - Графика

- res
 - drawable
 - img_01.png
 - drawable-hdpi
 - ic_launcher.png
 - drawable-ldpi
 - drawable-mdpi
 - ic_launcher.png
 - drawable-xhdpi
 - ic_launcher.png
 - drawable-xxhdpi
 - ic_launcher.png
 - layout
 - activity_main.xml
 - menu

```
<ImageView
    android:id="@+id/ivImage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/img_01"
    android:contentDescription="@string/app_name"/>
```

```
/* Load image resources */
ivImage.setImageDrawable(this.getResources().getDrawable(
    R.drawable.img_01));
```

Приложение





Задание 4. Ресурсы

- Добавить в ресурсы приложения свою фотографию.

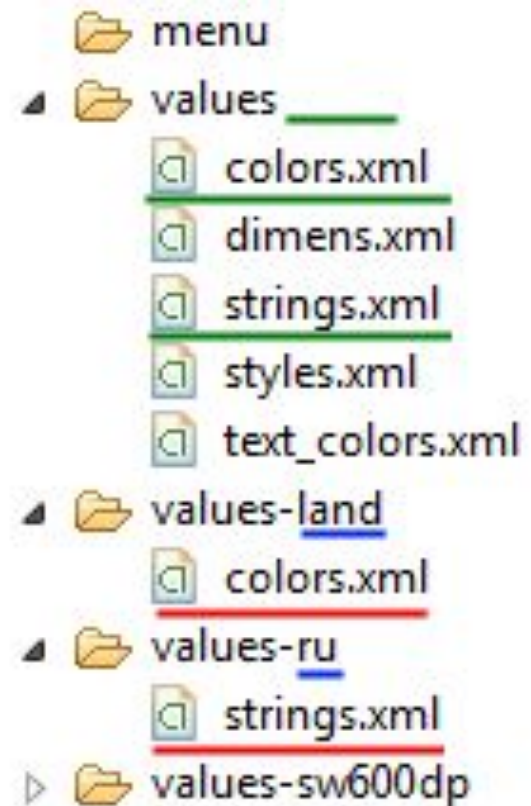


Квалификаторы ресурсов

- Квалификаторы позволяют создавать альтернативные ресурсы с одинаковыми именами и типами, но с разными значениями
- Это дает возможность автоматически использовать различные значения ресурсов в зависимости от типа и состояния устройства (размер экрана, ориентация, язык системы и тд.)

Примеры квалификаторов

- Язык: en, ru, fr
- Ориентация экрана: port, land
- Папка без квалификатора – ресурсы, используемые по умолчанию





Локализация приложения

res/values/strings.xml

```
<!-- Name of this application -->
<string name="app_name">A1_L02_01</string>

<!-- Background color for all Activities -->
<color name="bg_color">#55CCCC</color>

<!-- Simple text -->
<string name="simple_text_0">Simple text 0</string>
<string name="simple_text_1">Simple text 1</string>
<string name="simple_text_2">Simple text 2</string>
<string name="simple_text_3">Simple text 3</string>
<string name="simple_text_4">Simple text 4</string>
<string name="simple_text_5">Simple text 5</string>

<!-- Text attributes -->
<color name="simple_text_0_color">#FFFF0000</color>
<dimen name="simple_text_0_size">8pt</dimen>

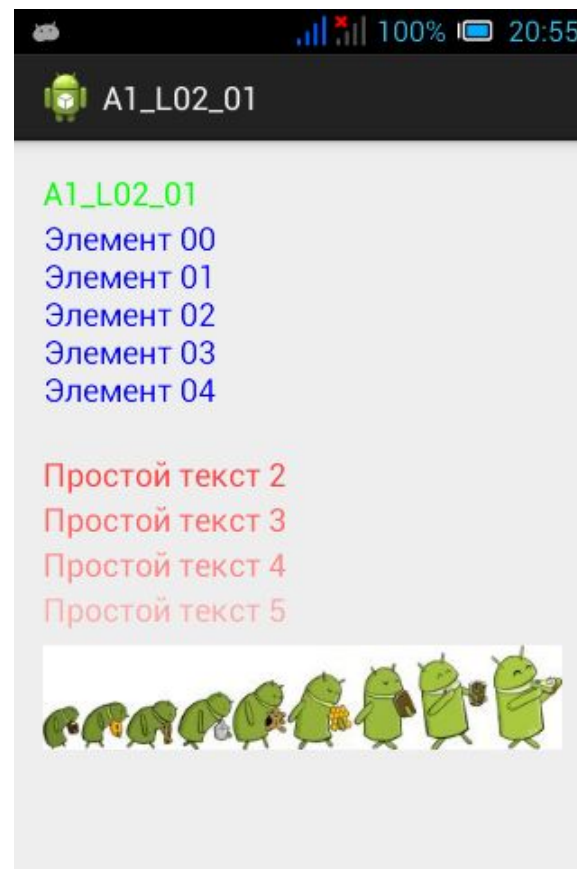
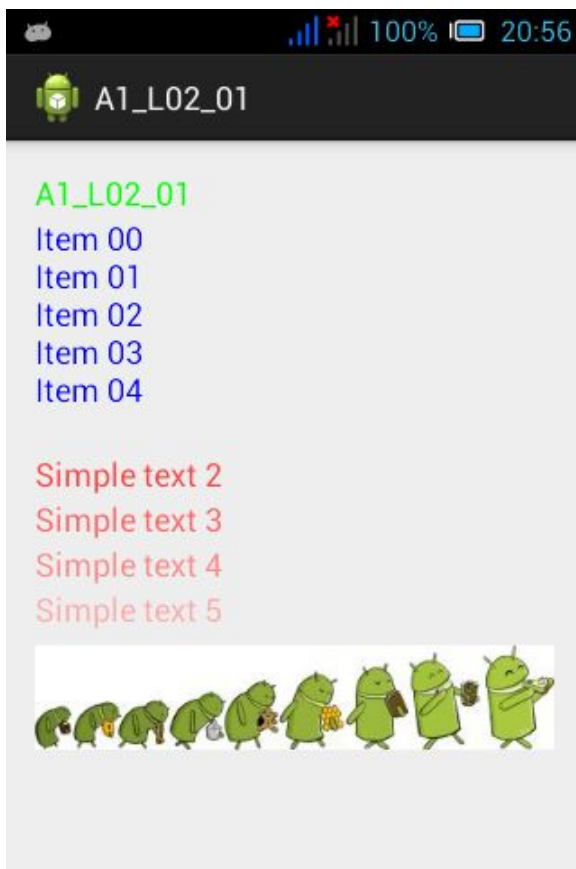
<!-- Array of items -->
<string-array name="my_str_array">
    <item>Item 00</item>
    <item>Item 01</item>
    <item>Item 02</item>
    <item>Item 03</item>
    <item>Item 04</item>
</string-array>
```

res/values-ru/strings.xml

```
<!-- Simple text -->
<string name="simple_text_0">Простой текст 0</string>
<string name="simple_text_1">Простой текст 1</string>
<string name="simple_text_2">Простой текст 2</string>
<string name="simple_text_3">Простой текст 3</string>
<string name="simple_text_4">Простой текст 4</string>
<string name="simple_text_5">Простой текст 5</string>

<!-- Array of items -->
<string-array name="my_str_array">
    <item>Элемент 00</item>
    <item>Элемент 01</item>
    <item>Элемент 02</item>
    <item>Элемент 03</item>
    <item>Элемент 04</item>
</string-array>
```

Локализация приложения





Задание 5. Работа с ресурсами

- Необходимо реализовать пример использования разных ресурсов (values) для различных ориентаций экрана (portrait and landscape orientations)
- Добавить в ресурсы приложения строку "Портретная ориентация" с именем my_txt.
- Добавить в ресурсы приложения значение цвета "FFFF1111" с именем txt_color.
- В папке ресурсов создать папку для хранения ресурсов-значений (values), которые будут использоваться только при альбомной ориентации экрана (values-land).
- Скопировать файл strings.xml из папки values, в папку values-land.
- В файле values-land/strings.xml изменить: строку с именем my_txt на "Альбомная ориентация"; цвет с именем txt_color на "FF11FF11".



Доступ к ресурсам по имени

```
/* Get link to application resources */
Resources res = this.getResources();

/* Resource ID */
int resID = 0;

/* Resource Name */
String resName = "ic_launcher";

/* Resource Type */
String resType = "drawable";

/* Resource Package (in this case this is application Package) */
String resPackage = this.getPackageName();

/* Get resource id by name */
resID = res.getIdentifier(resName, resType, resPackage);

/* Using resource id */
if (resID > 0) {
    ivImage.setImageDrawable(res.getDrawable(resID));
}
```



Домашнее задание

- Выполнить все задания урока
- Выбрать темы итоговой работы



Класс StringArray

- Работа с динамическим массивом строк
- Загрузка строк из ресурсов
- Чтение/запись в поток