



Система ресурсів Android-додатків

Питання 3.1.

Що таке ресурси?

статичні біти інформації, що зберігаються окремо від вихідного Java-коду.



Ресурси розташовані в директорії `res/` Android-проекту

Містить всі іконки та інші зображення, рядки тексту для інтернаціоналізації та ін.

Ресурси відокремлені від коду, оскільки

- зберігаються в іншому форматі,
- використовуються в різних умовах, для яких мають відповідний опис.

Для інтернаціоналізації матимемо рядки тексту для різних мов.

Android буде обирати правильний ресурс з усіх кандидатів для конкретного випадку (рядки українською мовою, якщо локалізація - українська)

Посилання на ресурси

Посилатись на ресурси можна з різних місць проекту

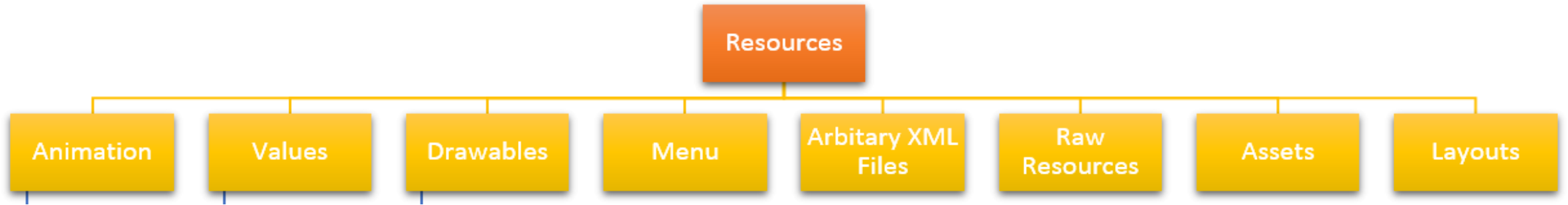


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.commonware.empublite" >
  <supports-screens
    android:largeScreens="true"
    android:normalScreens="true"
    android:smallScreens="false"
    android:xlargeScreens="true"/>
  <application
    android:allowBackup="false"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <activity
      android:name=".EmPubLiteActivity"
      android:label="@string/app_name" >
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```

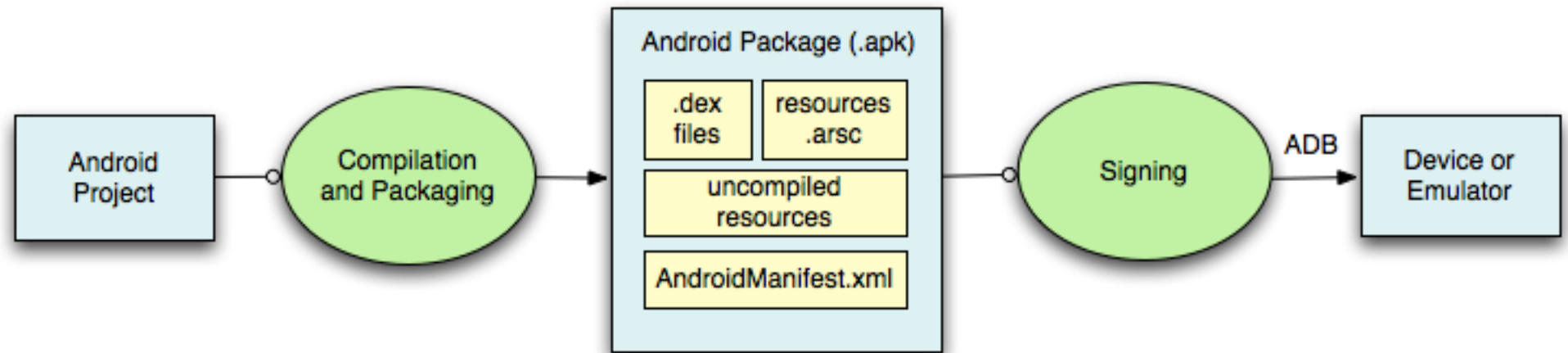
У прикладі рядковий ресурс `app_name` використовується в файлі маніфесту `AndroidManifest.xml`

Класифікація ресурсів

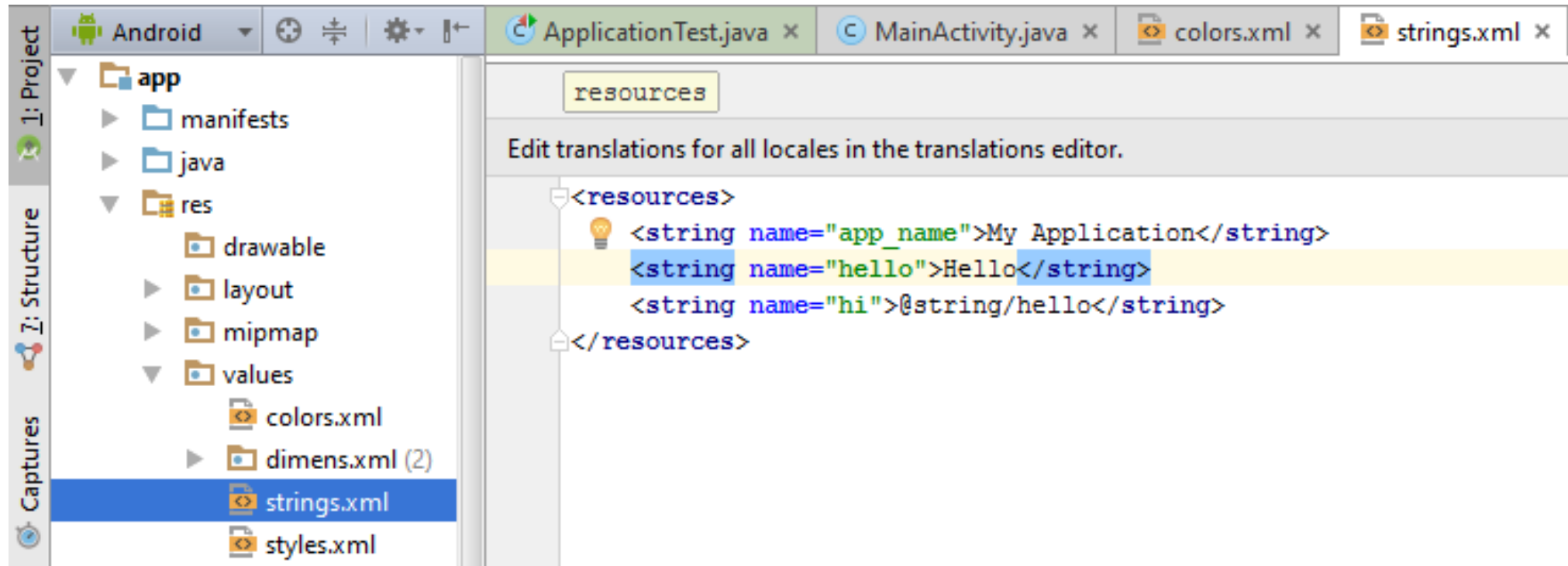
Розміщення ресурсів у каталогах проекту



MyProject/
res/
 drawable/
 graphic.png
 layout/
 main.xml
 info.xml
 mipmap/
 icon.png
 values/
 strings.xml



Рядкові ресурси



<TextView

```
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/hello" />
```

```
String str = getString(R.string.hello);
```

Інші прості ресурси.

String Array



Values

Colors

Strings

String Arrays

Plurals

Dimensions

Styles

String array – простий ресурс, на який посилаються за допомогою значення з атрибуту name.

Тому можна комбінувати string array з іншими простими ресурсами в одному XML-файлі в межах одного елементу <resources>.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="planets_array">
        <item>Mercury</item>
        <item>Venus</item>
        <item>Earth</item>
        <item>Mars</item>
    </string-array>
</resources>
```

<item>

A string, which can include styling tags. The value can be a reference to another string resource. Must be a child of a <string-array> element. Beware that you must escape apostrophes and quotation marks.

No attributes.

```
Resources res = getResources();
String[] planets = res.getStringArray(R.array.planets_array);
```

Інші прості ресурси

Quantity Strings (Plurals)



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <plurals name="plural_name">
    <item quantity=["zero" | "one" |
                    "two" | "few" | "many" | "other"]
      >text_string</item>
  </plurals>
</resources>
```

Різні мови мають різні правила для опису кількості елементів:

- В англійській окремо виділяється кількість 1. Пишемо "1 book", а в решті випадків "n books".
- В інших мовах правила утворення множини можуть бути складнішими.

Вибір рядка відбувається за граматичними правилами мови.

- В англійській випадок з нулем еквівалентний випадку з кількома одиницями ("zero books", "one book", "two books« і т. д.).
- У корейській мові навпаки, лише рядок other буде використовуватись.

Інші прості ресурси

Quantity Strings (Plurals)



res/values/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <plurals name="numberOfSongsAvailable">
        <item quantity="one">%d song found.</item>
        <item quantity="other">%d songs found.
            </item>
    </plurals>
</resources>
```

Розробнику рекомендується завжди постачати рядки для "one" та "other".

Java-код:

```
int count = getNumberOfSongsAvailable();
Resources res = getResources();
String songsFound = res.getQuantityString(R.plurals.numberOfSongsAvailable, count, count);
```

res/values-pl/strings.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <plurals name="numberOfSongsAvailable">
        <item quantity="one">Znaleziono %d
        piosenek. </item>
        <item quantity="few">Znaleziono %d
        piosenki. </item>
        <item quantity="other">Znaleziono %d
        piosenek. </item>
    </plurals>
</resources>
```


Ресурс кольору

Значення кольору визначається в XML за допомогою RGB-значення та альфа-каналу



Застосовувати ресурс-колір можна будь-де, де приймається hexadecimal color value.

#RGB

#ARGB

#RRGGBB

#AARRGGBB

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="color_name">
        hex_color
    </color>
</resources>
```

res/values/colors.xml:

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

Можна використовувати з drawable-ресурсом в XML (наприклад, android:drawable= "@color/green").

Вибір кольору для додатку

в матеріальному дизайні



Red	
500	#F44336
50	#FFEB3B
100	#FFCDD2
200	#EF9A9A
300	#E57373
400	#EF5350
500	#F44336
600	#E53935
700	#D32F2F
800	#C62828
900	#B71C1C



Google пропонує [палітри кольорів](#) (swatches) для оформлення додатку в матеріальному стилі для різних елементів інтерфейсу:

500 – основні (primary) кольори, підходять для домінуючих елементів інтерфейсу, зокрема тулбару

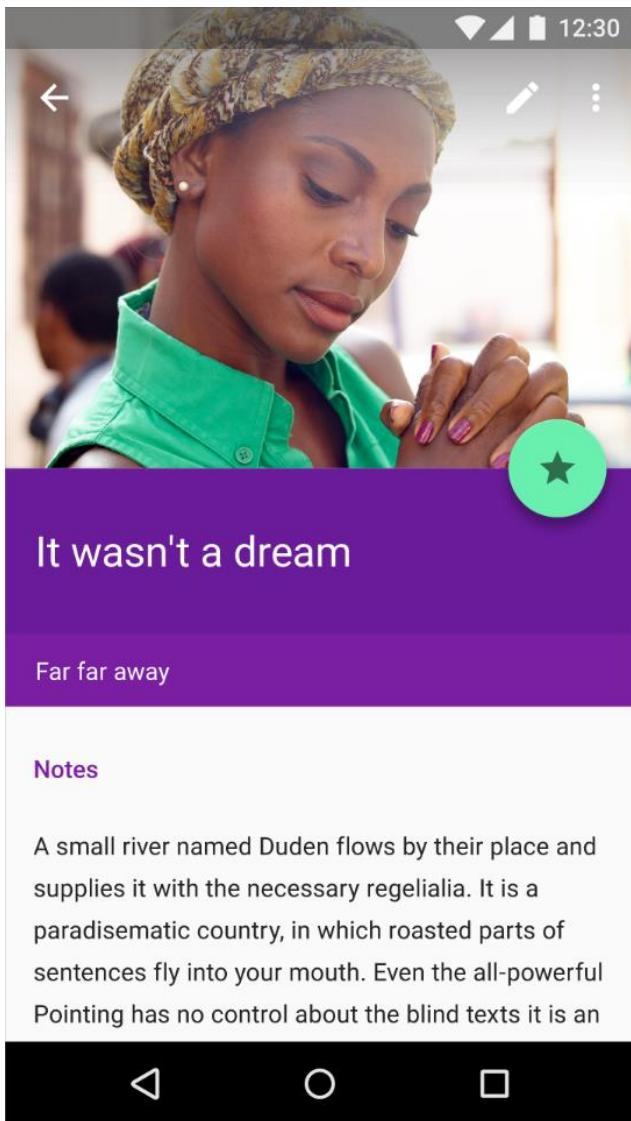
700 – темний основний колір, підходить для оформлення статус-бару

300 – виділяє додаткову інформацію

Акцентні кольори – яскравіші та більш насичені кольори для виділення кнопок, перемикачів, слайдерів тощо.

Система кольорів

у матеріальному дизайні



colors.xml in res/values

```
<resources>

  <color name="color_primary">#3F51B5</color>
  <color name="color_primary_dark">#303F9F</color>
  <color name="color_accent">#FFC107</color>

</resources>
```

Primary — Purple

500 #9B26AF

700 #7A1EA1

800 #691A99

Secondary — Green

A200 #68EFAD

XML-макет застосовує колір

<TextView

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/color_primary"
    android:text="Hello"/>
```

Java-код отримує колір:

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

Робота з палітрами

сервіс [material design palette](https://material.io/colors/)



material design palette

PALETTES ICONS COLORS

MORE MATERIAL DESIGN >

RED PINK PURPLE DEEP PURPLE

MADE BY MATERIALUP

INDIGO BLUE LIGHT BLUE CYAN TEAL

GREEN LIGHT GREEN LIME YELLOW AMBER

ORANGE DEEP ORANGE BROWN GREY BLUE GREY

Palette preview

Full Palette colors below

Your Palette ▾

DOWNLOAD TWEET

DARK PRIMARY COLOR	PRIMARY COLOR	LIGHT PRIMARY COLOR	TEXT / ICONS
	#212121	#757575	#BDBDBD
ACCENT COLOR	PRIMARY TEXT	SECONDARY TEXT	DIVIDER COLOR

Робота з палітрами

сервіс [material design palette](https://materialpalette.com)

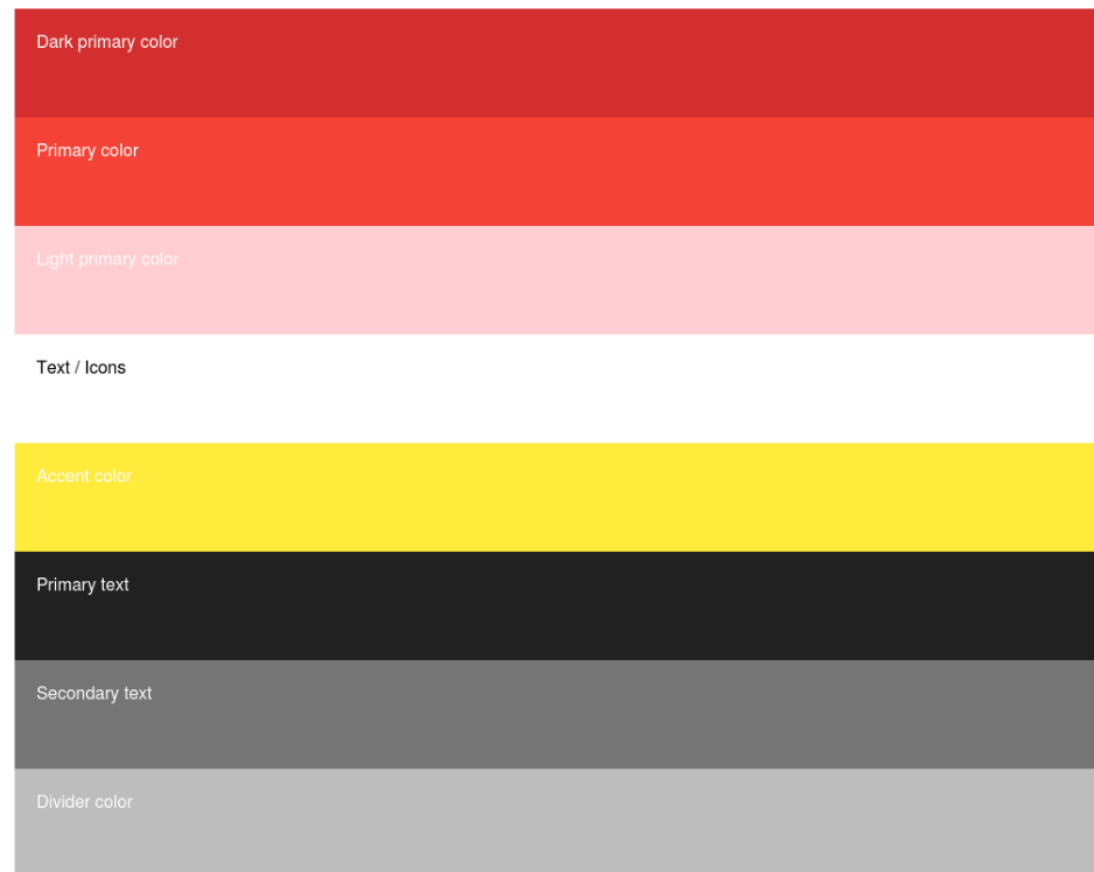


Згенерований xml-ресурс

```
<!-- Palette generated by Material Palette - materialpalette.com/red/yellow -->
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="primary">#F44336</color>
  <color name="primary_dark">#D32F2F</color>
  <color name="primary_light">#FFCDD2</color>
  <color name="accent">#FFEB3B</color>
  <color name="primary_text">#212121</color>
  <color name="secondary_text">#757575</color>
  <color name="icons">#FFFFFF</color>
  <color name="divider">#BDBDBD</color>
</resources>
```

Доступні варіанти генерації в CSS, SASS, LESS, SVG, XML, PNG, POLYMER

Palette generated by Material Palette - materialpalette.com/red/yellow



Інші прості ресурси



Назва ресурсу	Опис ресурсу
<u>Bool</u>	XML resource that carries a boolean value
<u>ID</u>	XML resource that provides a unique identifier for application resources and components.
<u>Integer</u> .	XML resource that carries an integer value
<u>Integer Array</u>	XML resource that provides an array of integers.
<u>Typed Array</u>	XML resource that provides a <u>TypedArray</u> (which you can use for an array of drawables)
Style	Управління стилями (далі в тексті)

Різні типи ресурсів: <https://developer.android.com/guide/topics/resources/available-resources.html>

Теми та стилі в Android

Стиль відноситься до представлення, а тема – до активності/додатку



Задача стилів – інкапсулювати набір значень атрибутів, що будуть використовуватись повторно, умовно або іншим чином з бажанням відокремити їх макетів інтерфейсу.

<https://developer.android.com/design/downloads/index.html>

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
  <style name="text">
```

```
    <item name="android:padding">4dip</item>
```

```
    <item name="android:textAppearance">?android:attr/textAppearanceLarge</item>
```

```
    <item name="android:textColor">#000000</item>
```

```
  </style>
```

```
  <style name="layout">
```

```
    <item name="android:background">#CoCoCo</item>
```

```
  </style>
```

```
</resources>
```

styles.xml

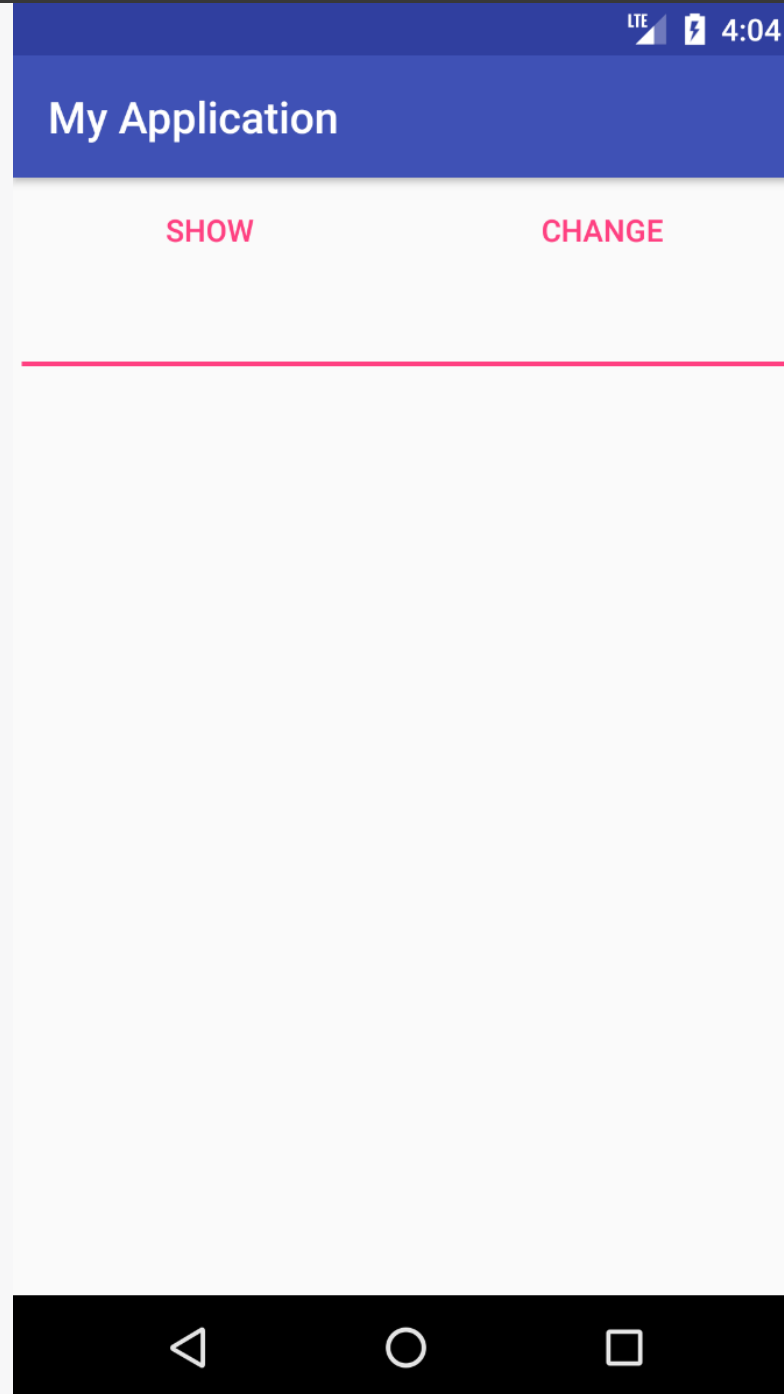
You assign the style attribute to entries in your layout file via the style="@style/text" attribute.

Стилi i теми пiдтримують наслiдування за допомогою атрибуту parent у тегу style.

```

<LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <LinearLayout style="?android:attr/buttonBarStyle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >
        <Button android:id="@+id/Button01"
            style="?android:attr/buttonBarButtonStyle"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Show" />
        <Button android:id="@+id/Button02"
            style="?android:attr/buttonBarButtonStyle"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Change" />
    </LinearLayout>
    <EditText android:id="@+id/myView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10" >
        <requestFocus />
    </EditText>
</LinearLayout>

```



Android зберігає перелік стандартних атрибутів для стилізації у файлі [R.attr](#).

Елементи стилю

Де застосовується стиль?



- Атрибут `style` може застосовуватись до віджета/контейнера, щоб діяти відповідно лише на цей віджет/контейнер.
- Проте стиль автоматично не застосовується до дочірніх представлень.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    style="@style/bigred">
    <Button
        android:id="@+id/button"
        android:text=""
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```

Отриманий UI не матиме тесту Button великими червоними літерами, незважаючи на атрибут `style`.

Стиль впливає на контейнер, але не на його вміст.

Непрацюючі стилі ігноруються. Хоч у `LinearLayout` немає атрибутів `android:textSize` чи `android:textColor`, помилок чи виключень не трапиться.

При стилізації віджета або контейнера у стилі можна застосовувати доступні для них атрибути.

Елементи стилю

Наслідування стилю



Припустимо, потрібно унаслідувати стиль від Theme.Holo: додаємо атрибут parent.

Якщо його не задати, your attribute definitions змішаються зі стилем, який за замовчуванням застосовується до віджета чи контейнера.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <style name="activated" parent="android:Theme.Holo">
    <item name="android:background">
      ?android:attr/activatedBackgroundIndicator
    </item>
  </style>
</resources>
```


Елементи стилю

Можливі значення



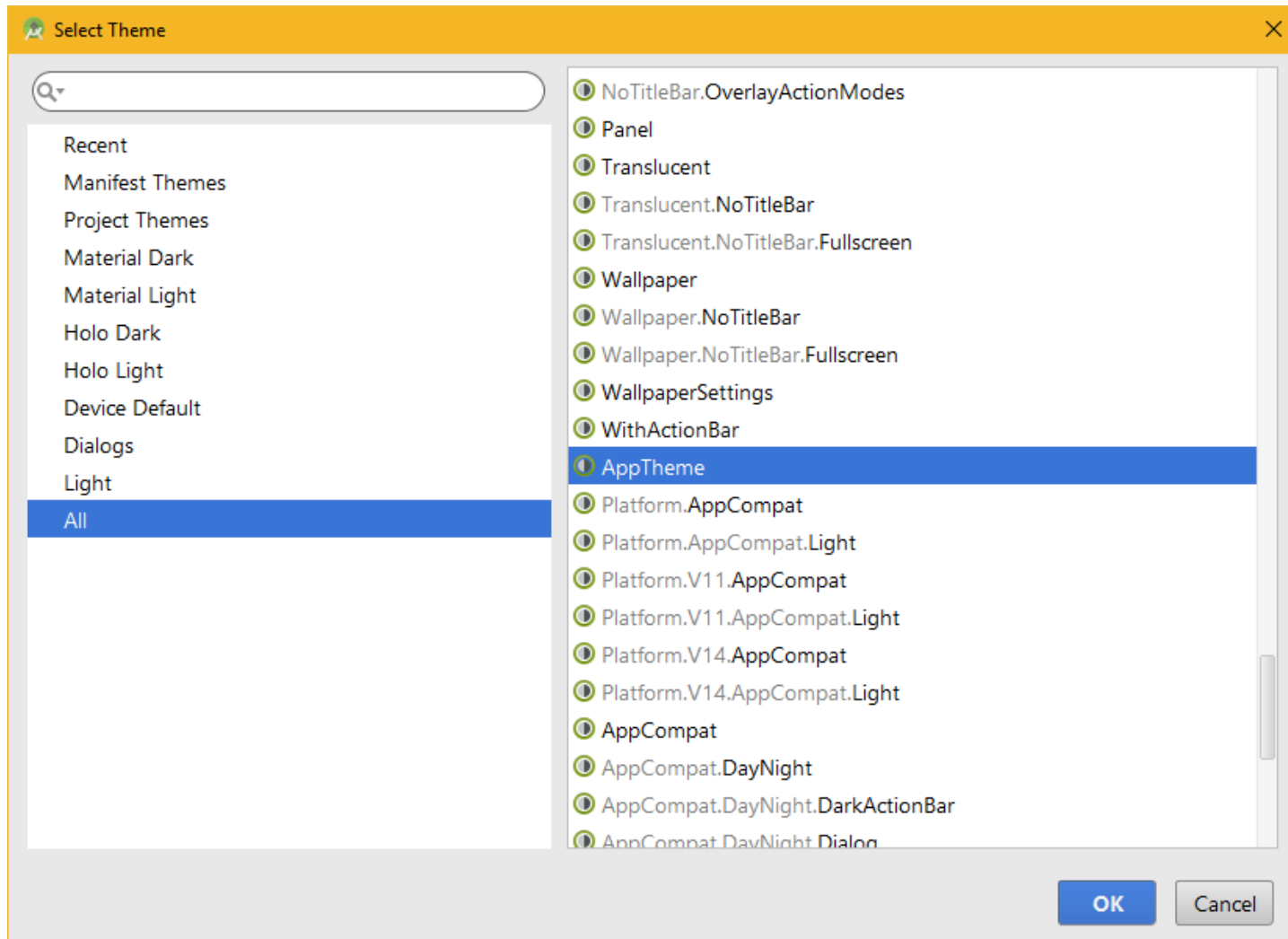
Зазвичай присвоюються деякі константи на зразок `zosp` або `#FFFFFF0000`.

Інколи потрібно застосувати значення іншого атрибуту (від батьківської теми).
Застосовується синтаксис `?android:attr/назва_атрибуту`.

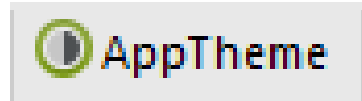
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <style name="activated" parent="android:Theme.Holo">
    <item name="android:background">?android:attr/activatedBackgroundIndicator</item>
  </style>
</resources>
```

Внутрішня побудова стилів мало задокументована, Google рекомендує йти до їх первинного коду.

Теми оформлення



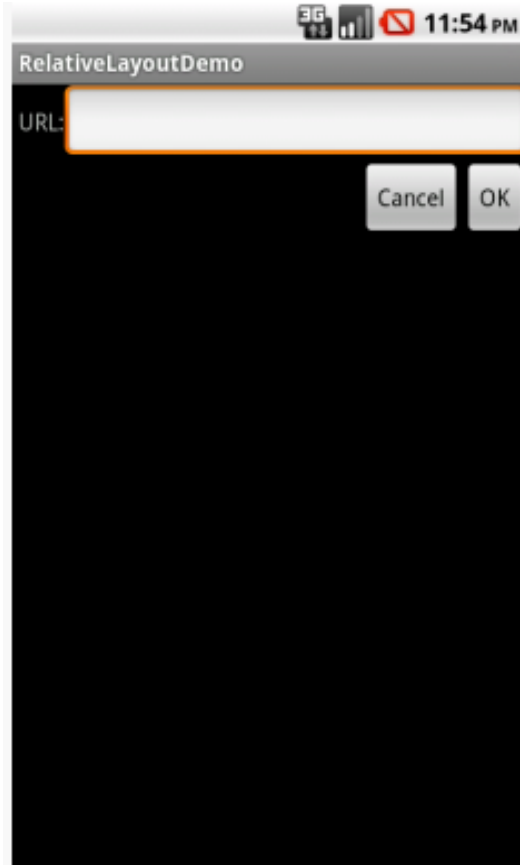
Тема – це колекція стилів, що забезпечує професійний вигляд додатку.



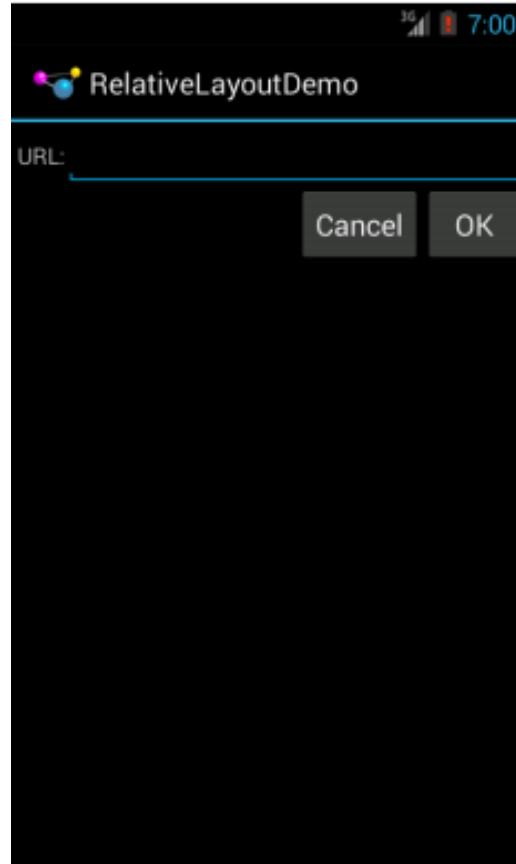
Теми інтерфейсу



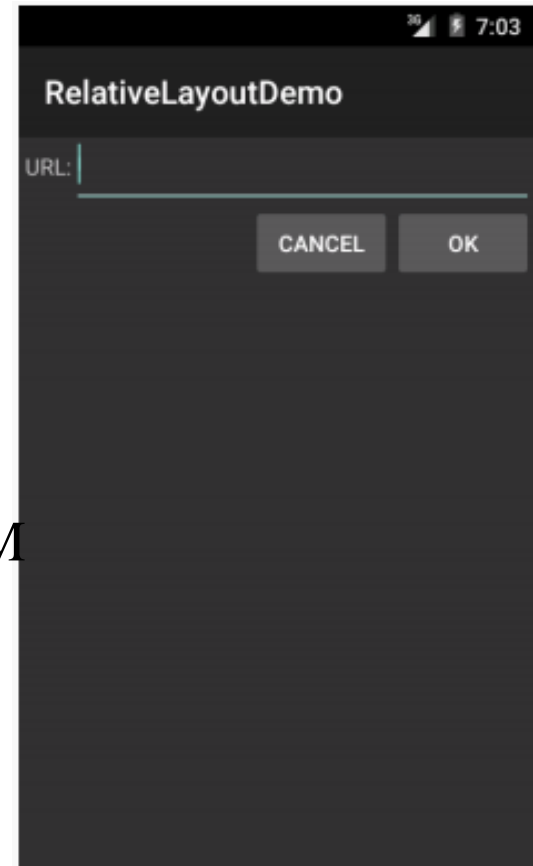
Theme



Theme.
Holo



Theme.M
aterial



Стандартні теми в різних версіях Android

Теми – це стилі, що застосовуються до активності або додатку

за допомогою атрибуту `android:theme` у відповідному елементі `<activity>` або `<application>`.



Якщо тема, що застосовується, - Ваша власна, пошліться на неї: `@style/...`

Якщо тема комплектна для Android, зазвичай використовують префікс `@android:style/`, зокрема, `@android:style/Theme.Holo.Dialog` або `@android:style/Theme.Holo.Light`.

Наприклад, тема `@android:style/Theme.Holo.NoActionBar.Fullscreen`:

```
<!-- Variant of the default (dark) theme that has no title bar and fills the entire screen -->
<style name="Theme.Holo.NoActionBar.Fullscreen">
    <item name="android:windowFullscreen">true</item>
    <item name="android:windowContentOverlay">@null</item>
</style>
```

Theme Editor в Android Studio

you may get a preview of that custom theme



Edit all themes in the project in the theme editor. [Open editor](#) [Hide notification](#)

```
<resources>

<!-- Base application theme. -->
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
</style>

</resources>
```

activity_main.xml × styles.xml × Theme Editor × MainActivity.java ×

Nexus 4 24

Language

ImageButton Toolbar

App bar

NORMAL

DISABLED

Raised button

NORMAL

Theme

AppTheme

Theme parent default

AppCompat Light [Then

colorPrimary default

@color/colorPri...

colorPrimaryDark default

@color/colorPri...

colorAccent default

@color/colorAcc...

android:colorBackgroundAPI

@android:color/b...

android:colorForegroundAPI

Теми матеріального дизайну



Default Material Themes

Material theme (API 21+)

`android:style/Theme.Material` (dark version)

`android:style/Theme.Material.Light` (light version)

`android:style/Theme.Material.Light.DarkActionBar`

`android:style/Theme.Material.NoActionBar` (No Action Bar)

Themes with Backward Compatibility

AppCompat (API 7+)

`Theme.AppCompat`

`Theme.AppCompat.Light`

`Theme.AppCompat.Light.DarkActionBar`

`Theme.AppCompat.NoActionBar`

For `MinSDKVersion >= 21`

Resource Directory →

`drawable` (For API 21 and above)

`values` (For API 21 and above)

`layout` (For API 21 and above)

Backward Compatibility : `MinSDKVersion < 21`

Resource Directory →

`drawable` (For below API 21)

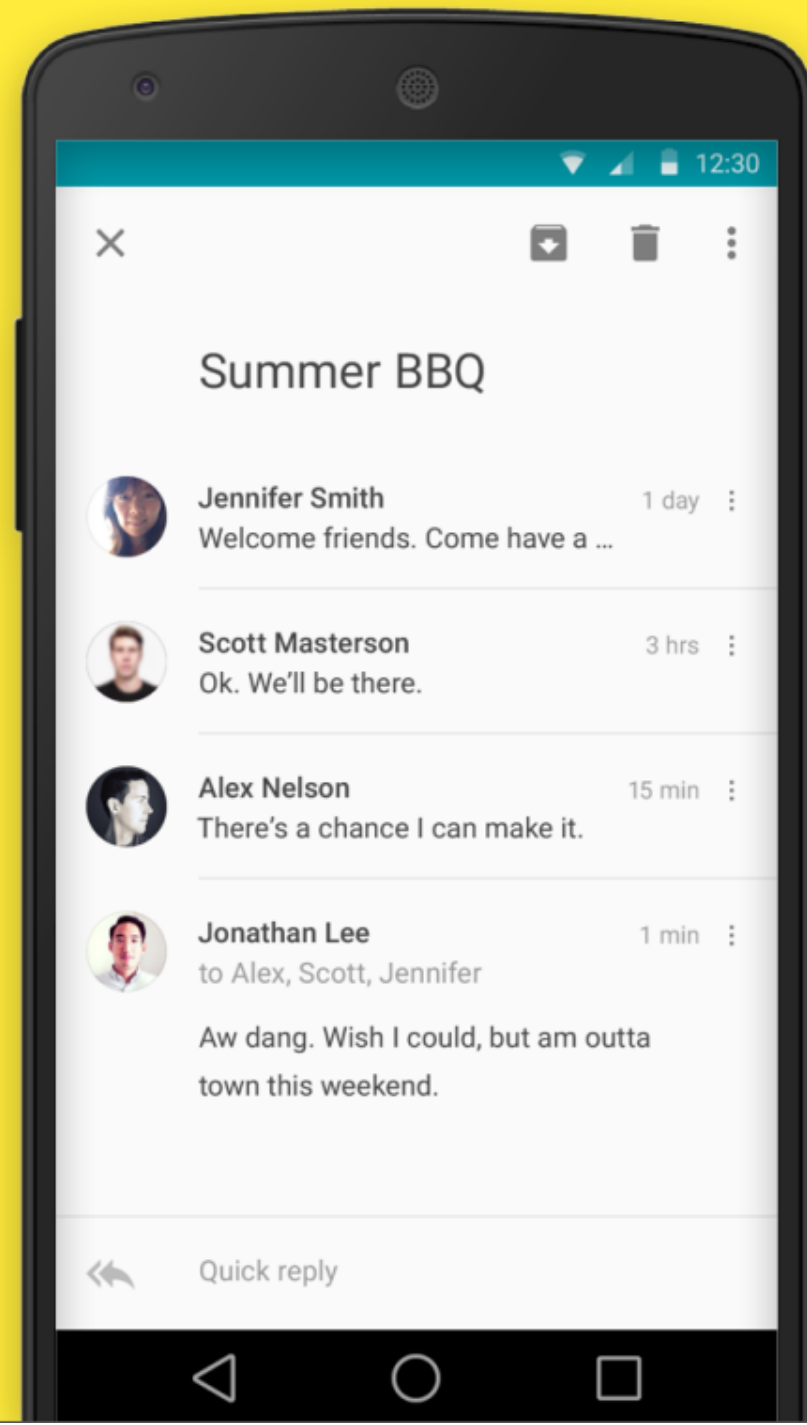
`drawable-v21` (For API 21 and above)

`values` (For below API 21)

`values-v21` (For API 21 and above)

`layout` and `layout-v21`

2 Print-like design



Підібрана типографіка



Typographic scale

Display 4

Light 112sp

Display 3

Regular 56sp

Display 2

Regular 45sp

Display 1

Regular 34sp

Headline

Regular 24sp

Title

Medium 20sp

Subhead

Regular 16sp (Device), Regular 15sp (Desktop)

Body 2

Medium 14sp (Device), Medium 13sp (Desktop)

Body 1

Regular 14sp (Device), Regular 13sp (Desktop)

Caption

Regular 12sp

Menu

Medium 14sp (Device), Medium 13sp (Desktop)

Button

MEDIUM (ALL CAPS) 14sp

Робота з шрифтами

Android SDK постачається з набором стандартних шрифтів



Стандартні шрифти можна стилізувати за допомогою кількох XML-атрибутів.

- **android:fontFamily** : використовується для заміни шрифту за замовчуванням для додатку. У комплекті йде sans-serif шрифт. android:fontFamily вимагає API 16+.
- **android:typeface** : використовується замість fontFamily до API 15. За замовчуванням ти цього атрибуту normal або sans. Доступні значення:
- **android:textStyle** : може набувати значень **bold**, *italic*, normal та комбінувати їх, наприклад, android:textStyle="bold|italic".

```
casual  
monospace  
cursive  
sans-serif  
@android:  
sans-serif-condensed  
sans-serif-smallcaps  
serif  
serif-monospace π
```

```
monospace  
normal  
sans  
serif π
```

Завантаження власного шрифту

Android підтримує формати .otf (OpenType) та .ttf (TrueType).



- 1) Завантажте шрифти, які бажаєте використовувати (.ttf або .otf).
- 2) Створіть папку для шрифтів у папці Assets Вашого проекту та перенесіть туди відповідні файли.
- 3) Орієнтуйтесь на код, щоб застосувати шрифти:

```
String fontPath = "fonts/JaneAust.ttf";
```

```
// завантаження Font Face
```

```
Typeface m_typeFace = Typeface.createFromAsset(getAssets(), fontPath);
```

```
// застосування шрифту
```

```
m_tvFont.setTypeface(m_typeFace);
```


Автомасштабування (autosizing) тексту

починаючи з Android O (можливі 2 способи)



Granularity

- TextView масштабуватиме текст горизонтально та вертикально на обране значення.
- Для реалізації потрібно додати атрибути до TextView:
 - autoSizeMinTextSize.
 - autoSizeMaxTextSize.
 - autoSizeStepGranularity: за замовчуванням 1px.

<TextView

```
android:text="This is some auto-scaling text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:autoSizeText="uniform"
android:autoSizeMinTextSize="10sp"
android:autoSizeMaxTextSize="100sp"
android:autoSizeStepGranularity="2sp"/>
```

Preset Sizes

- Визначається масив усіх можливих розмірів тексту.
- Android O вибирає доречний розмір на основі розміру TextView на основі кількості тексту для показу, розмірів TextView та поточної конфігурації екрана.

Можна відкрити файл res/values/arrays.xml та визначити потрібні розміри.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
```

```
<array
```

```
  name="autosize">
```

```
    <item>10sp</item>
```

```
    <item>15sp</item>
```

```
    <item>25sp</item>
```

```
    <item>30sp</item>
```

```
    <item>100sp</item>
```

```
    <item>150sp</item>
```

```
  </array>
```

```
</resources>
```

<TextView

```
android:text="This is some auto-scaling text"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:autoSizeText="uniform"
android:autoSizePresetSizes="@array/autosize"
/>
```

Шрифти в XML

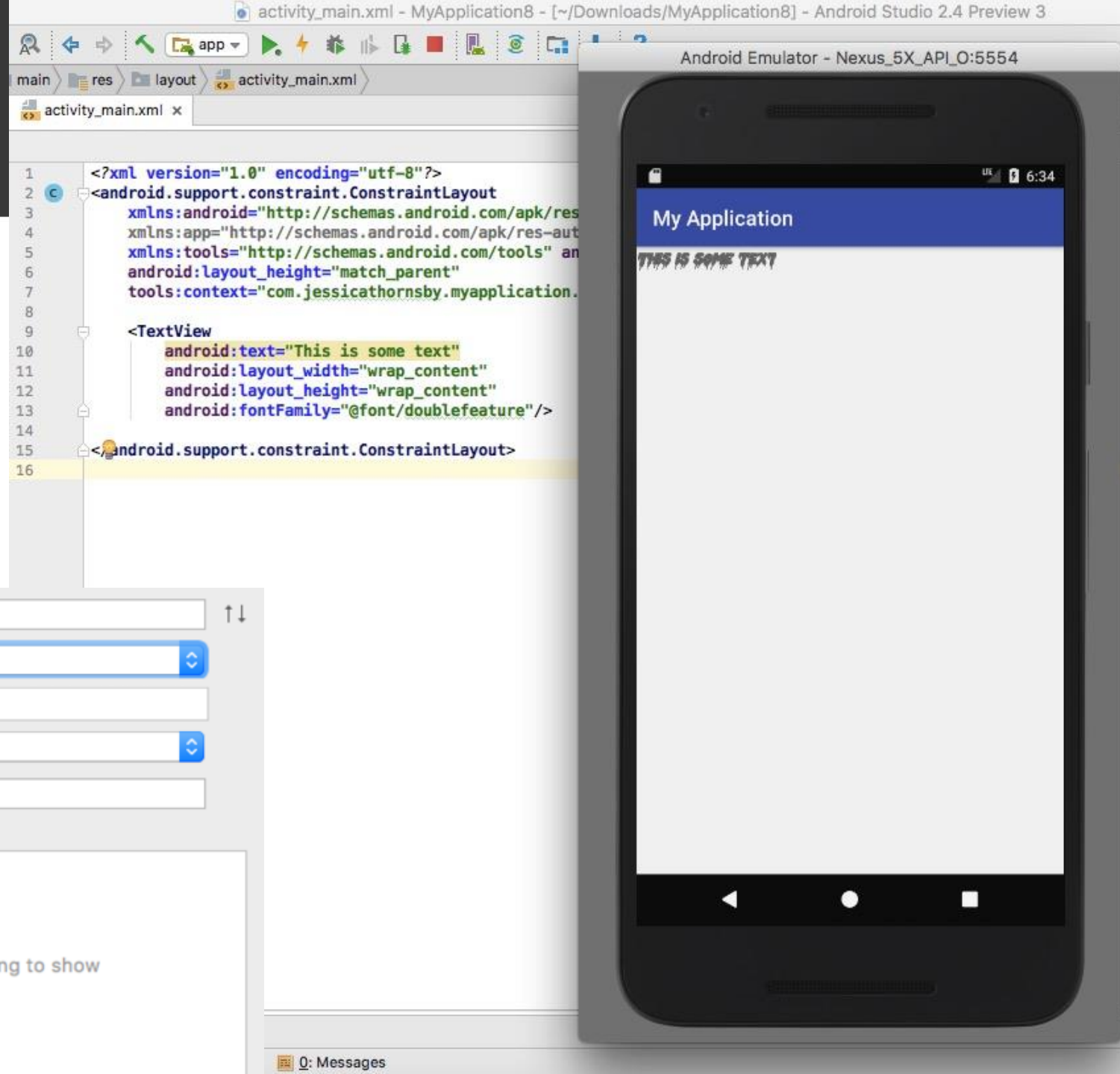
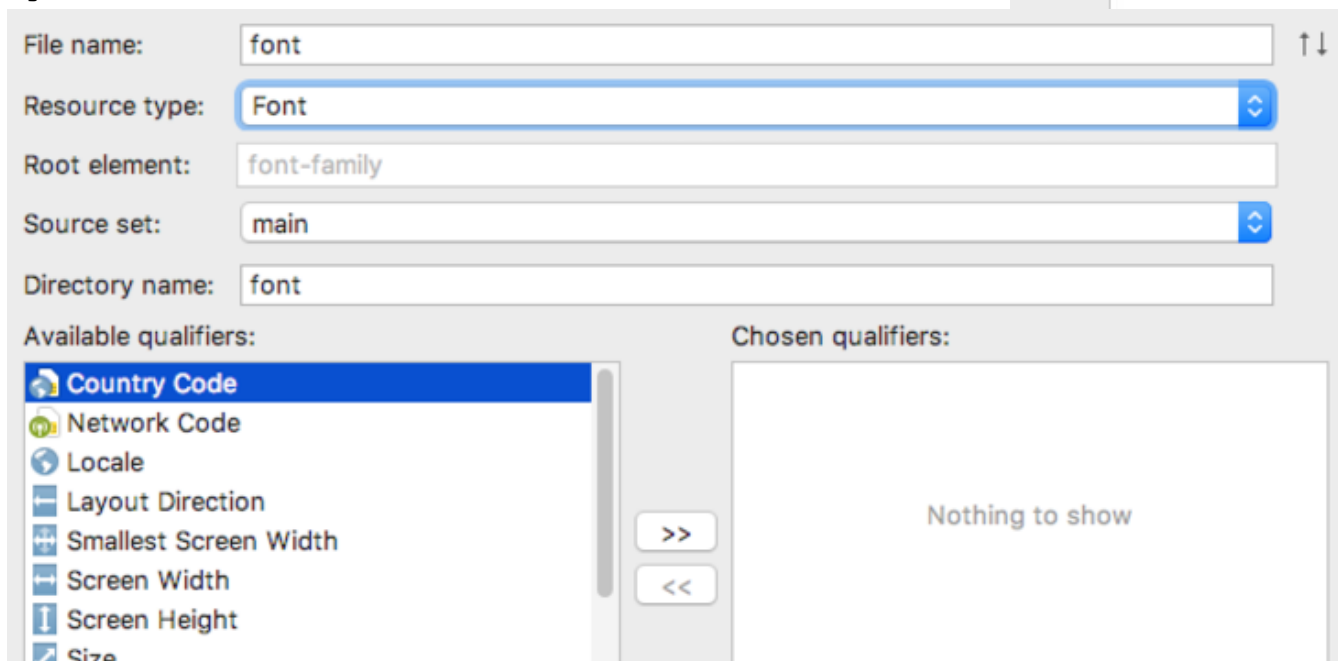
з Android O

Маючи файл із шрифтом, його потрібно
одесь зберегти:

ПКМ на папці app/res та обираємо New >
Android resource directory.

Обираємо font з випадаючого меню, вводимо
назву шрифту по назві файлу. ОК.

Перетягуємо файл(и) шрифтів у нову папку
у res/font.



Шрифти в XML

(з Android O)



Шрифти можна додавати до стилів, створених у додатку:

```
<style name="headlineFont" parent="@android:style/TextAppearance.Small">
  <item name="android:fontFamily">@font/doublefeature</item>
</style>
```

Програмно можна отримувати кастомний шрифт за допомогою методу `getFont(int)`, наприклад:

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        LinearLayout linearLayout = new LinearLayout(this);
        TextView ProgrammaticallyTextView = new TextView(this);
        ProgrammaticallyTextView.setText("This is some text");

        //Reference your custom font//

        Typeface typeface = getResources().getFont(R.font.doublefeature);
        ProgrammaticallyTextView.setTypeface(typeface);
        linearLayout.addView(ProgrammaticallyTextView);

        this.setContentView(linearLayout, new LinearLayout.LayoutParams(
            LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT));
    }
}
```

Створення сімейства шрифтів

з Android O



- 1) Переконайтесь, що всі файли шрифтів є в папці проекту `res/font`.
- 2) ПКМ на папці `res/font`, оберіть `New > Font resource file`. Назвіть файл. ОК.
- 3) Відкрийте XML-файл та визначте всі різні версії шрифту разом зі стилями та атрибутами ваги (`weight`), наприклад:

Потім можна посилатись на будь-який шрифт сім'ї за допомогою атрибуту `android:fontFamily`.

Наприклад:

```
android:fontFamily="@font/doublefeature_bold"
```

```
<?xml version="1.0" encoding="utf-8"?>
http://schemas.android.com/apk/res/android">
  <font
    android:fontStyle="normal"
    android:fontWeight="400"
    android:font="@font/doublefeature_regular" />

  <font
    android:fontStyle="italic"
    android:fontWeight="400"
    android:font="@font/doublefeature_italic" />

  <font
    android:fontStyle="bold"
    android:fontWeight="400"
    android:font="@font/doublefeature_bold" />

</font-family>
```