

Risorse Android

Struttura delle risorse

- Android fornisce una architettura per externalizzare alcune informazioni
 - es. un parametro può essere inserito in un file di configurazione
- Le risorse in Android sono contenute in file organizzate in cartelle/file precise
- Le due cartelle principali sono res e asset
 - res conterrà le risorse da “compilare”+
 - asset conterrà le altre risorse

Risorse nella cartella res

- le risorse sono dichiarate nei file

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, HelloWorld2Activity!</string>
    <string name="app_name">HelloWorld2</string>
</resources>
```

- le risorse possono essere dei file
 - eg. l'icona per lanciare la activity principale

La classe R

- In compilazione ad ogni risorsa viene associata una costante di tipo int

```
package ppl.pdm.helloworld2;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class layout {
        public static final int main=0x7f030000;
    }
    public static final class string {
        public static final int app_name=0x7f040001;
        public static final int hello=0x7f040000;
    }
}
```

- la classe R è contenuta nella cartella gen

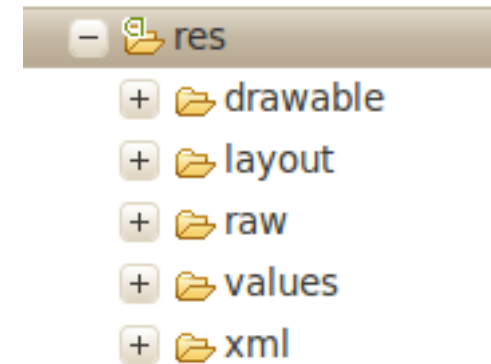
Risorse che descrivono valori

- Sono contenute in file xml nella cartella res/values
- Sono dichiarate in fra tag specifici
 - <string>
 - <string-array>
 - <integer-array>
 - <color>
 - <drawable>
 - <style>
 - <dimension>

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <string name="app_title">Button Counter</string>
    <string name="button_label">Press</string>
    <string name="output_label">Numero Click:</string>
    <string-array name="myArray">
        <item>Valore 1</item>
        <item>Valore 2</item>
        <item>Valore 3</item>
    </string-array>
    <integer-array name="primeArray">
        <item>2</item>
        <item>3</item>
        <item>5</item>
        <item>7</item>
        <item>11</item>
    </integer-array>
</resources>
```

File come risorse

- Alcuni file sono visti come risorse
 - es. le immagini, i file xml
- I file vanno inseriti in cartelle specifiche
 - Le immagini/icone in res/drawable
 - I file xml generici in res/xml
 - raw contiene file qualunque
- per accedere alla risorsa uso la costante
 - R.<cartella>.<file name>
 - eg. R.xml.libri



Risorse di tipo Layout

- Ogni activity ha almeno un layout
 - sono contenuti nella cartella layout
- Serve a posizionare degli “elementi grafici” sullo schermo
 - le view

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
</LinearLayout>
```

- Esempi - ResourceLayout // ResourceColor

Identificatori per le risorse

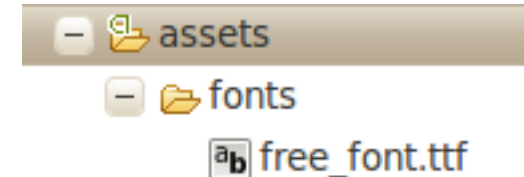
- Ad ogni risorsa viene associato un “nome”
- Ad ogni componente viene associato un id che si usa nel codice per farvi riferimento
 - android:id
- Sintassi per identificatori di risorse
 - @[package:][+]type/name
- Esempi
 - @string/app_name
 - @+id/myButton
 - @id/myButton

Risorse Dimension

- Sono dei valori numerici che permettono di specificare delle unità di misura
- Specificano la dimensione degli oggetti grafici
 - altezza e larghezza, dimensione testo, etc.
- Le unità di misura sono:
 - px -pixel
 - in - (inch) pollici
 - mm - millimetri
 - pt - punti (1/72 pollici)
 - dp - “pixel” indipendenti dalla densità di punti del display
 - dp=px se il display è 160 dpi (dots per inch)

Assets

- Sono nella cartella assets
- Si gestiscono con un AssetsManager



```
public class AssetsTestActivity extends Activity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
        // Per ottenere il riferimento all'Asset utilizziamo l'oggetto  
        // di tipo AssetManager a cui accediamo attraverso le Resources  
        AssetManager assetManager = getResources().getAssets();  
        // Otteniamo il riferimento all'oggetto relativo ai font  
        Typeface typeface = Typeface.createFromAsset(assetManager, "fonts/free_font.ttf");  
        // Otteniamo il riferimento alla TextView  
        TextView textView=(TextView)findViewById(R.id.output);  
        // Assegnamo il font alla textView  
        textView.setTypeface(typeface);  
    }  
}
```

Risorse XML

- Sono nella cartella res/xml
- Si caricano con il metodo della classe resources
 - XmlResourceParser getXML(int id)
- XmlResourceParser è un parser di xml
 - un software che “capisce” la sintassi xml
 - analizza un flusso di caratteri (o byte) e ne estrae il significato

Risorse Raw

- Sono contenute nella cartella res/raw
- Tipicamente sono accessibili con un `InputStream`

```
InputStream in = getResources().openRawResource(R.raw.books)
```
- Tipicamente sono suoni, immagini, file binari, etc.

Risorse Drawable

- sono elementi grafici che possono essere visualizzati sullo schermo
 - Es. immagini, forme, sfumature di colore

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval" >
    <gradient
        android:angle="45"
        android:centerColor="@color/green"
        android:endColor="@color/blue"
        android:startColor="@color/red" />
    </shape>
```

I18N - L10N

- Internazionalizzazione e Localizzazione
- Struttura dei nomi delle cartelle in cui sono le risorse
- Esempio
 - `res/values-<mmc mnc>-<lingua>-<classe schermo>-<tipo schermo>-<orientation>-<densità px>-<tipo touch>-<dispositiva>-<tipo input>-<prim Input no key>-<screen dim>-<sdk>-<minor ver>`