

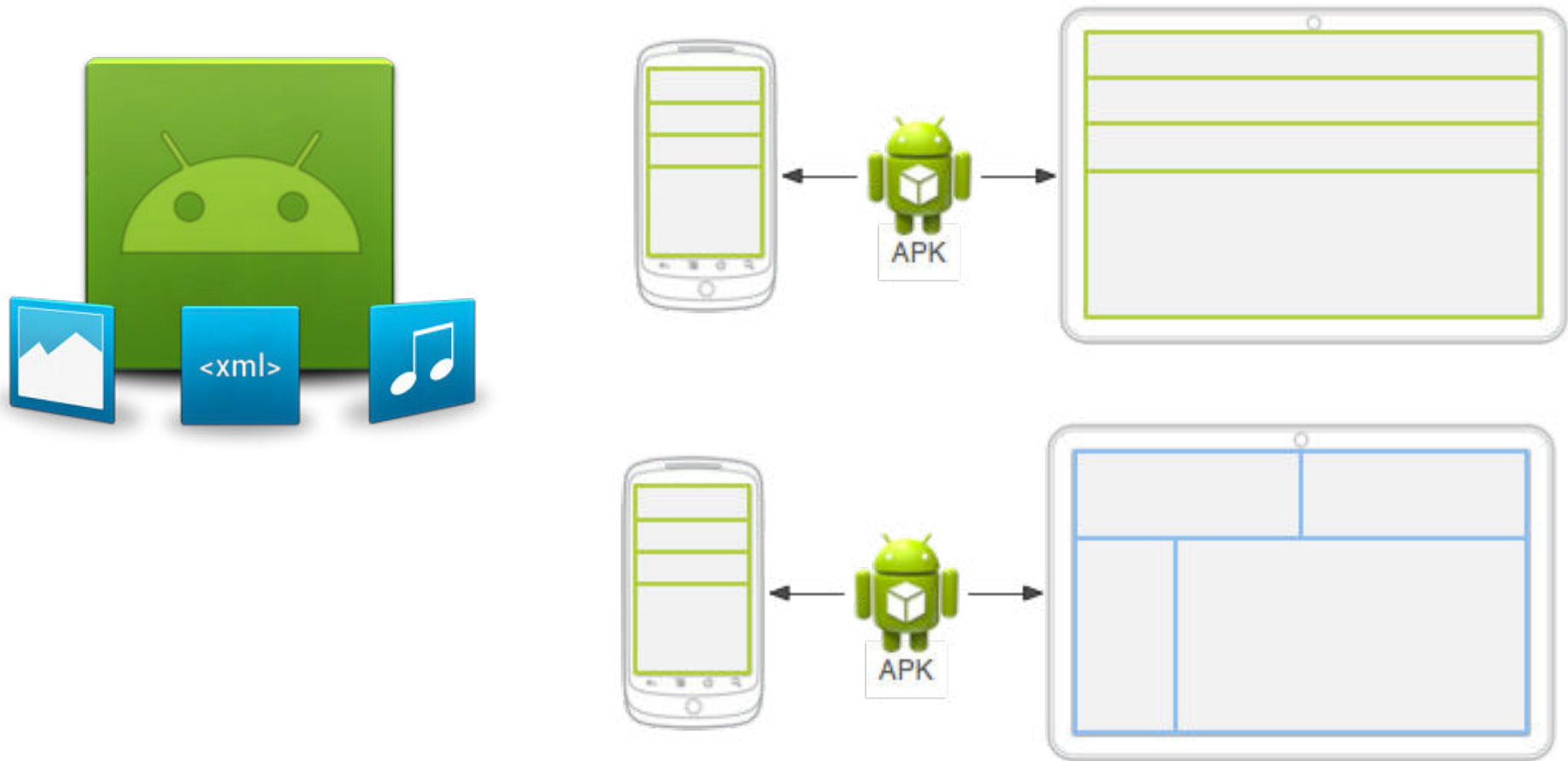
Risorse Android

Filosofia Android

- Esternalizzare le risorse dell'app per gestirle separatamente
 - immagini, stringhe, grafica, etc.
- Fornire risorse alternative per dispositivi diversi con configurazioni diverse
 - forma, dimensione schermo, densità pixel , lingua, etc



Risorse alternative: Concetto

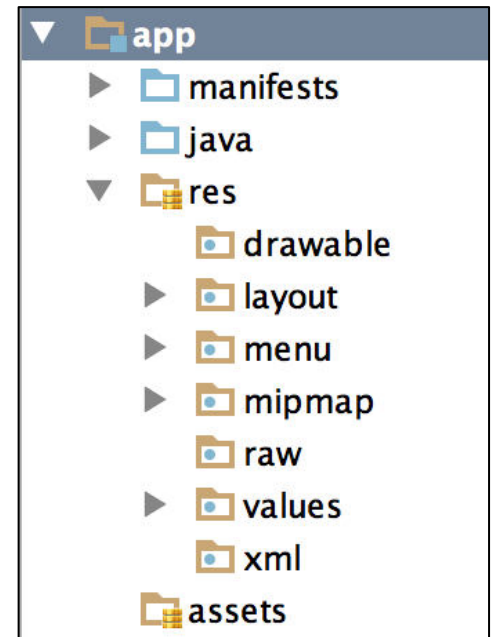


posso specificare layout differenti per schermi differenti
aggiungendo un file xml

Risorse per tipo

- Le risorse sono contenute nella cartella res
- Ogni risorsa va messa in una cartella specifica a seconda del tipo
- La cartella res contiene sottocartelle il cui nome è definito
 - [Tabella 1](#)

```
MyProject/  
  src/  
    MainActivity.java  
  res/  
    drawable/  
      graphic.png  
    layout/  
      main.xml  
      info.xml  
    mipmap/  
      icon.png  
    values/  
      strings.xml
```



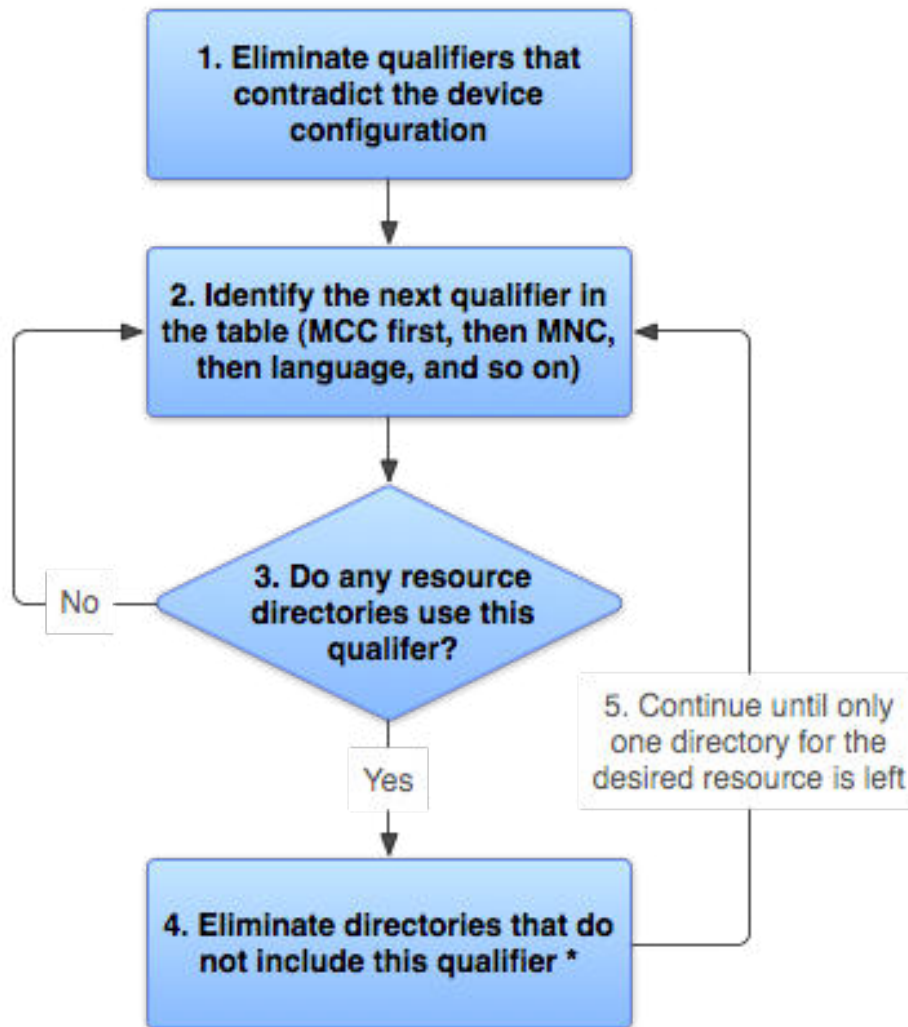
Risorse Alternative

- Per fornire risorse alternative si creano delle sottocartelle di res con nomi definiti
- Schema dei nomi
 - **<resources_name>-<config_qualifier>**
 - resources_name: nome della cartella da tabella
 - config_qualifier: configurazione specifica come sequenza ordinata di qualificatori
- Qualificatori
 - lingua, densità, orientamento, etc.
 - [Tabella 2](#)

Nomi sottocartelle

- I qualificatori si concatenano con un trattino
 - drawable-en-rUS-land
 - US-English devices in landscape orientation
- Devono comparire nell'ordine della tabella
 - Wrong: drawable-hdpi-port/
 - Correct: drawable-port-hdpi/
- Le cartelle non si possono annidare
 - Wrong: res/drawable/drawable-en/.
- I valori sono case-insensitive.
- Potete fornire un solo valore per qualificatore in una cartella

Scelta della risorsa



* If the qualifier is screen density, the system selects the "best match" and the process is done

La selezione avviene confrontando le risorse fornite con le caratteristiche del device

Eccezione: la screen density non è mai eliminata, si sceglie il più vicino

Scelta della risorsa

- **Risorse fornite**
 - drawable/
 - drawable-en/
 - drawable-fr-rCA/
 - drawable-en-port/
 - drawable-en-notouch-12key/
 - drawable-port-ldpi/
 - drawable-port-notouch-12key/
- **Configurazione device**
 - Locale = **en-GB**
 - Screen orientation = **port**
 - Screen pixel density = **hdpi**
 - Touchscreen type = **notouch**
 - Primary text input method = **12key**

Eliminare contraddizioni

- **Risorse fornite**
 - drawable/
 - drawable-en/
 - ~~drawable-fr-rCA/~~
 - drawable-en-port/
 - drawable-en-notouch-12key/
 - drawable-port-ldpi/
 - drawable-port-notouch-12key/
- **Configurazione device**
 - Locale = **en-GB**
 - Screen orientation = **port**
 - Screen pixel density = **hdpi**
 - Touchscreen type = **notouch**
 - Primary text input method = **12key**

Scorrimento ed eliminazioni

- MMC, **Language**, Layout, SW, W, H, size, aspect, **orient.**, etc.

MMC non fa niente ma Language elimina chi non lo ha

- **Risorse fornite**

~~drawable/~~

~~drawable-en/~~

drawable-en-port/

~~drawable-en-notouch-12key/~~

~~drawable-port-ldpi/~~

~~drawable-port-notouch-12key/~~

- **Configurazione device**

Locale = **en-GB**

Screen orientation = **port**

Screen pixel density = **hdpi**

Touchscreen type = **notouch**

Primary text input method = **12key**

Risorse di tipo valore e file

- Risorse di tipo valore

```
<resources>
  <string name="app_name">HelloWorldPJDM</string>

  <string name="hello_world">Hello world!</string>
  <string name="action_settings">Settings</string>
</resources>
```

- File come risorse
 - eg. l'icona per lanciare la activity principale

Risorse di tipo valore

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

```
<resources>
  <string name="app_name">My Application Test PJDM</string>
  <string name="hello_world">Hello world!</string>
  <string name="action_settings">Settings</string>
</resources>
```

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

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

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello world!" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView" />

</RelativeLayout>
```

- Esempi - ResourceLayout // ResourceColor

Identificatori per le risorse

- Ad ogni risorsa viene associato un “nome”
- Sintassi per identificatori di risorse
 - @[package:][+]type/name
- Esempi
 - @string/app_name
 - @+id/myButton
 - @id/myButton
- Gli ID sono risorse particolari per assegnare identificativi unici ai componenti grafici

Esempi

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="opaque_red">#f00</color>
    <string name="hello">Hello!</string>
</resources>
```

```
<?xml version="1.0" encoding="utf-8"?>
<EditText xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:textColor="@color/opaque_red"
    android:text="@string/hello" />
```

```
<?xml version="1.0" encoding="utf-8"?>
<EditText xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:textColor="@android:color/secondary_text_dark"
    android:text="@string/hello" />
```

```
<?xml version="1.0" encoding="utf-8"?>
<bitmap xmlns:android="http://schemas.android.com/apk/res/android"
    android:src="@drawable/other_drawable" />
```

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 **generated**

Identificatori in java

`getResources().getString(R.string.hello);`

- id risorse tipo file
 - R.<cartella>.<file name>
 - es. R.xml.libri
- id risorse tipo valore
 - R.<tipo>.<nome>
 - es. R.string.app_title

Esempi

```
ImageView imageView = (ImageView) findViewById(R.id.myimageview);
imageView.setImageResource(R.drawable.myimage);
```

```
// Load a background for the current screen from a drawable resource
getWindow().setBackgroundDrawableResource(R.drawable.my_background_image) ;

// Set the Activity title by getting a string from the Resources object, because
// this method requires a CharSequence rather than a resource ID
getWindow().setTitle(getResources().getText(R.string.main_title));

// Load a custom layout for the current screen
setContentView(R.layout.main_screen);

// Set a slide in animation by getting an Animation from the Resources object
mFlipper.setInAnimation(AnimationUtils.loadAnimation(this,
    R.anim.hyperspace_in));

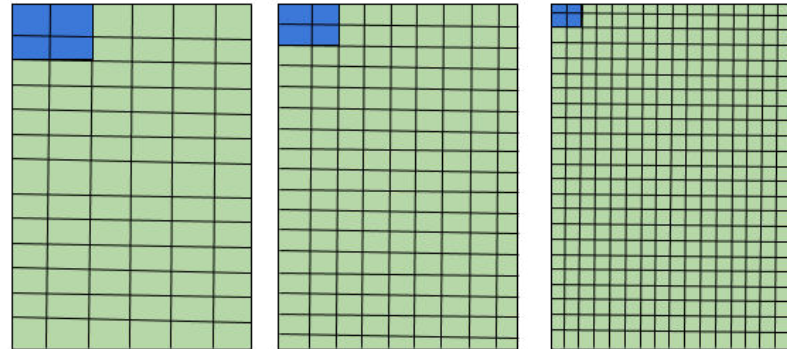
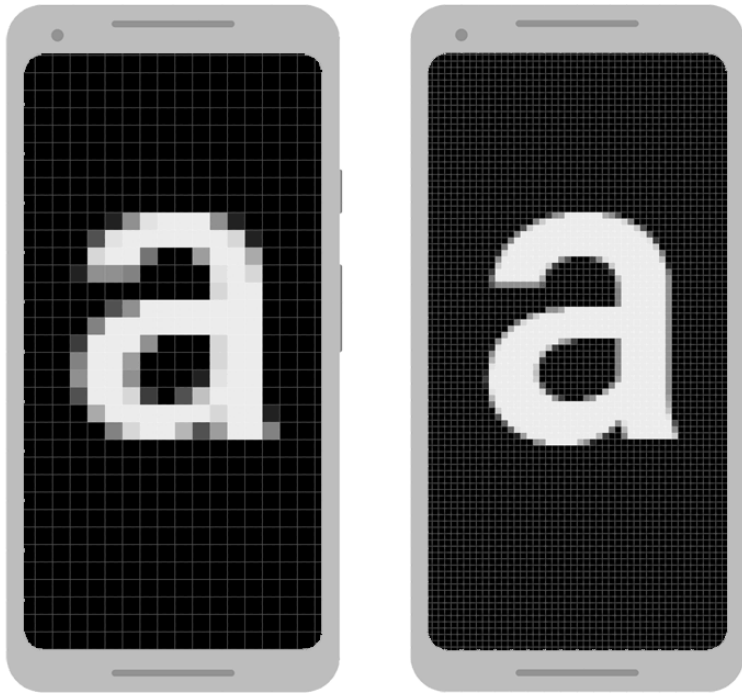
// Set the text on a TextView object using a resource ID
TextView msgTextView = (TextView) findViewById(R.id.msg);
msgTextView.setText(R.string.hello_message);
```

Risorse Dimension

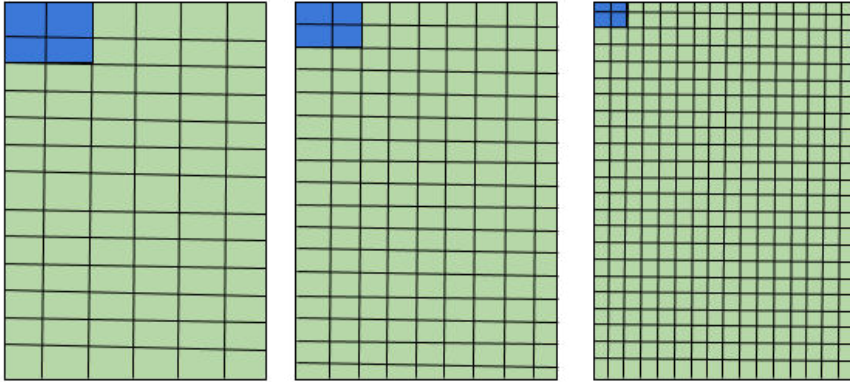
Unit	Description	Units Per Physical Inch	Density Independent?	Same Physical Size On Every Screen?
px	Pixels	Varies	No	No
in	Inches	1	Yes	Yes
mm	Millimeters	25.4	Yes	Yes
pt	Points	72	Yes	Yes
dp	Density Independent Pixels	~160	Yes	No
sp	Scale Independent Pixels	~160	Yes	No

<https://developer.android.com/guide/topics/resources/more-resources#Dimension>

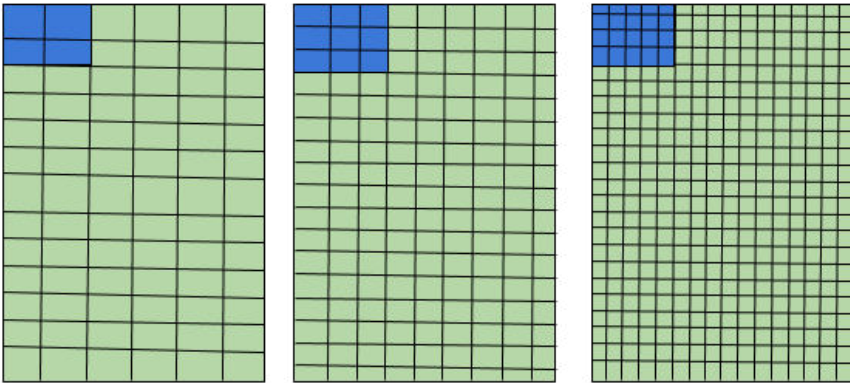
Screen density



dp: density-independent pixels



2 px

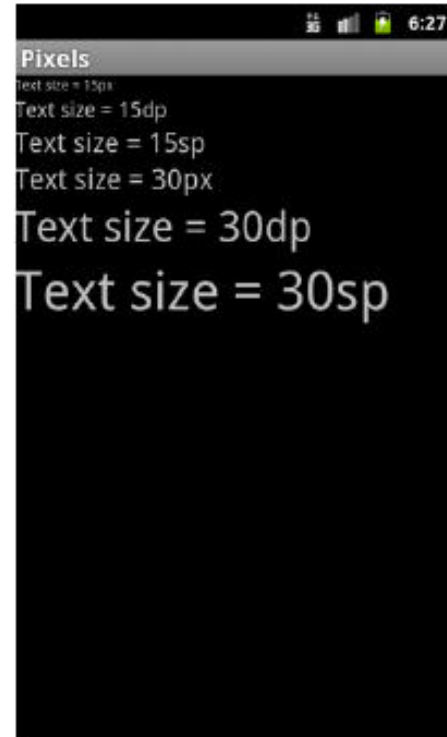
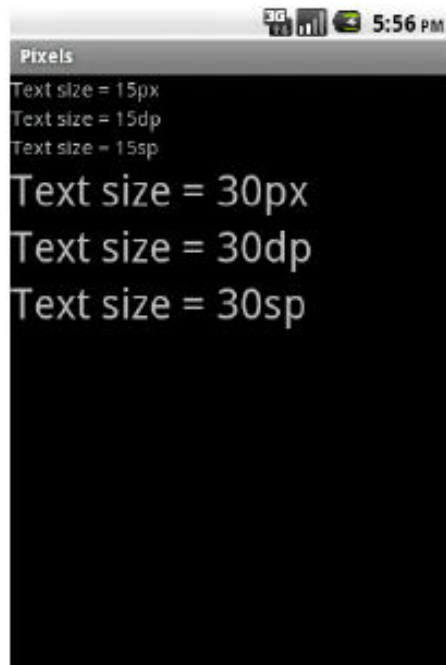


2 dp

$$px = dp * (dpi / 160)$$

sp: Scalable Pixels

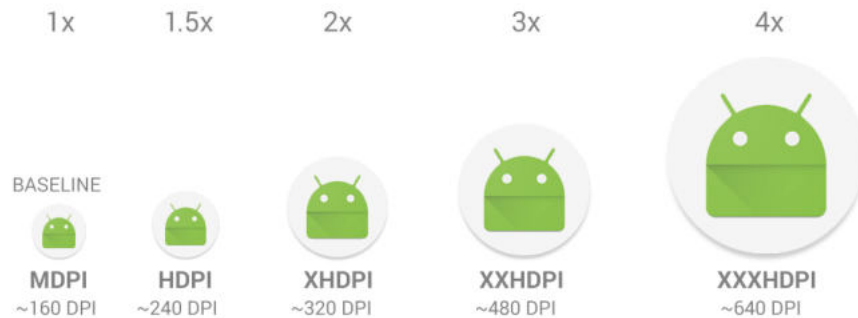
- scalable like the dp unit,
- it is also scaled by the user's font size preference.



Density Buckets

Density Bucket	Screen Density	Physical Size	Pixel Size
ldpi	120 dpi	0.5 x 0.5 in	0.5 in * 120 dpi = 60x60 px
mdpi	160 dpi	0.5 x 0.5 in	0.5 in * 160 dpi = 80x80 px
hdpi	240 dpi	0.5 x 0.5 in	0.5 in * 240 dpi = 120x120 px
xhdpi	320 dpi	0.5 x 0.5 in	0.5 in * 320 dpi = 160x160 px
xxhdpi	480 dpi	0.5 x 0.5 in	0.5 in * 480 dpi = 240x240 px
xxxhdpi	640 dpi	0.5 x 0.5 in	0.5 in * 640 dpi = 320x320 px

Risorse Alternative Bitmap



```
res/  
  drawable-xxxhdpi/  
    awesome-image.png  
  drawable-xxhdpi/  
    awesome-image.png  
  drawable-xhdpi/  
    awesome-image.png  
  drawable-hdpi/  
    awesome-image.png  
  drawable-mdpi/  
    awesome-image.png
```


style mess

Style

Theme

TextAppearance

ThemeOverlay

Default Style?

Material Theming?

...and they all use the style tag!

<https://www.youtube.com/watch?v=oSSi50Qv9DE>

Risorsa Style

- Key-value store
 - Le chiavi (name) sono agli **attributi**

```
<style name="MyButton" parent="Widget.MaterialComponents.Button">  
  <item name="android:textSize">19dp</item>  
  <item name="android:layout_margin">8dp</item>  
  <item name="android:padding">8dp</item>  
  <item name="android:textColor">@color/black</item>  
  <item name="android:backgroundTint">?attr/colorAccent</item>  
</style>
```

Attributo Style

- Si applicano ad una sola view

```
<Button
    android:id="@+id/btLay"
    style="@style/MyButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Layout" />
```

```
<style name="MyButton" parent="Widget.MaterialComponents.Button">
    <item name="android:textSize">19dp</item>
    <item name="android:layout_margin">8dp</item>
    <item name="android:padding">8dp</item>
    <item name="android:textColor">@color/black</item>
    <item name="android:backgroundTint">?attr/colorAccent</item>
</style>
```

Attributo TextAppearance

- Specifica lo stile di una TextView o sua sottoclasse
 - Sempre mediante uno style

```
<EditText
    ...
    android:textAppearance="@style/RaspberryTextAppearance"
/>
```

```
<style name="RaspberryTextAppearance">
    <item name="android:textColor">@color/raspberry</item>
    <item name="android:fontFamily">@font/cera_pro</item>
</style>
```

Attributo Theme

- Applica uno stile ad un application o activity

```
<application
    android:theme="@style/AppTheme"
    ...
/>
```

```
<style name="AppTheme"
    parent="Theme.MaterialComponents.Light.NoActionBar">
    <item name="colorPrimary">@color/white</item>
    <item name="colorPrimaryVariant">@color/midnight</item>
    <item name="colorSecondary">@color/blinkist_green</item>
    <item name="colorControlNormal">@color/midnight</item>
    <item name="cardViewStyle">@style/RoundCardViewStyle</item>
    ...
</style>
```

Attributi del tema

```
<selector>
  <item android:color="?attr/colorSecondary"
  android:state_enabled="true"/>
  <item android:alpha="0.6" android:color="?attr/colorSecondary"/>
</selector>
```

```
<style name="AppTheme"
parent="Theme.MaterialComponents.Light.NoActionBar">
  <item name="colorPrimary">@color/white</item>
  <item name="colorPrimaryVariant">@color/midnight</item>
  <item name="colorSecondary">@color/blinkist_green</item>
  <item name="colorControlNormal">@color/midnight</item>
  <item name="cardViewStyle">@style/RoundCardViewStyle</item>
  ...
</style>
```