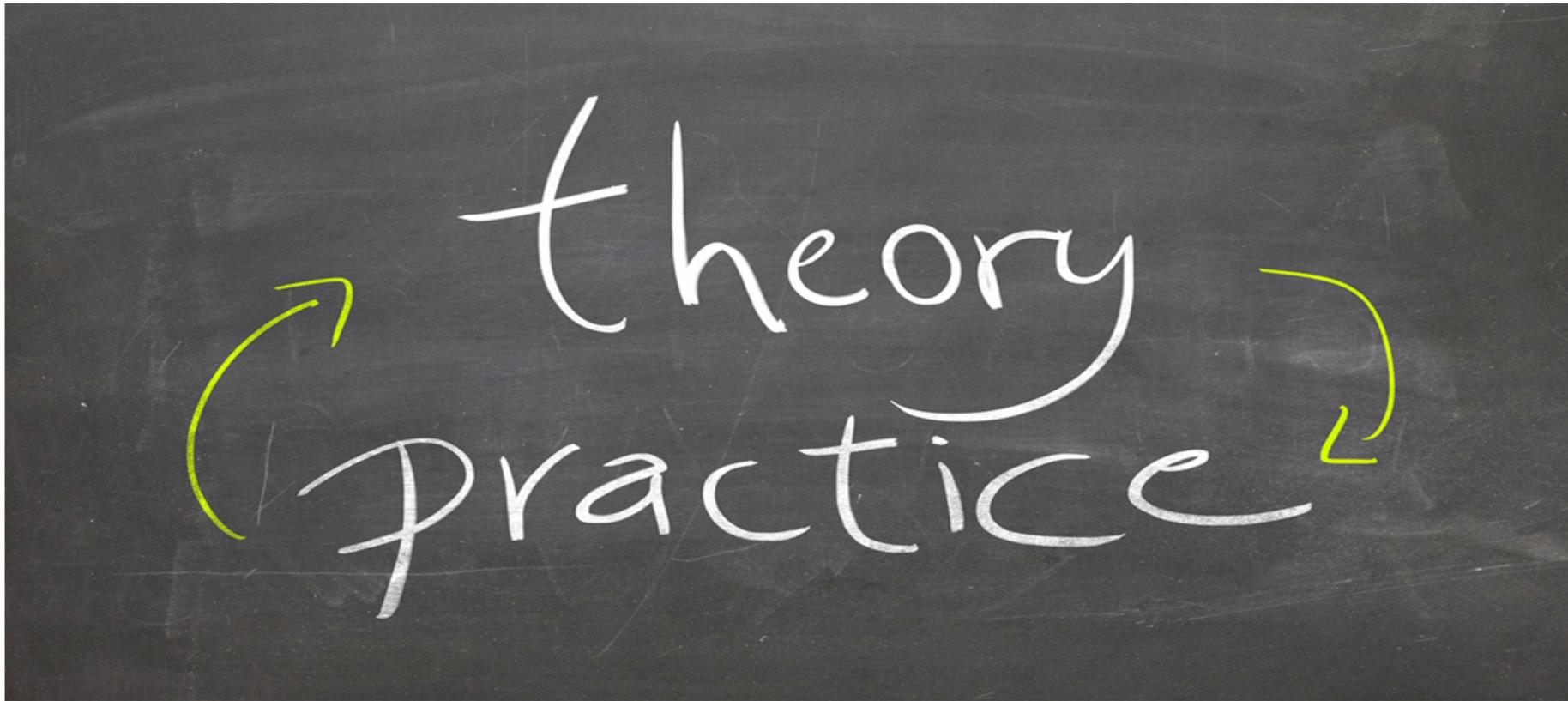


# Android Development

Juan Camilo Rada

# Lab methodology



# Setting up the environment

JDK 1.7

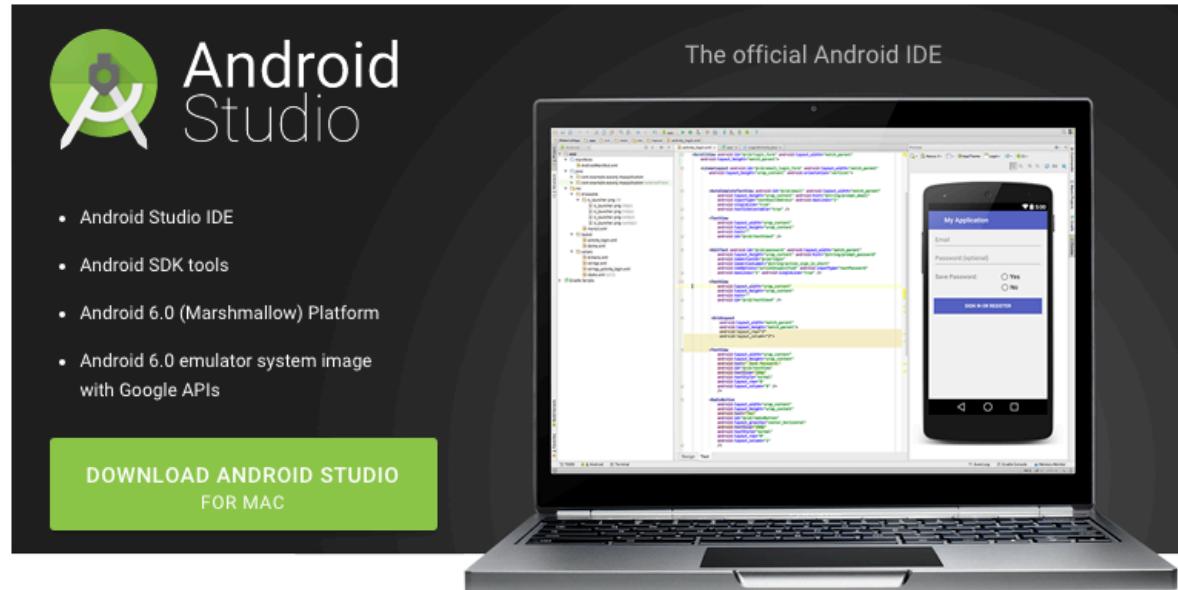
1. Go to oracle download [page](#)
2. Accept License
3. Install JDK

Java SE Development Kit 7u79		
You must accept the <a href="#">Oracle Binary Code License Agreement</a> for Java SE to download this software.		
<input type="radio"/> Accept License Agreement	<input checked="" type="radio"/> Decline License Agreement	
Product / File Description	File Size	Download
Linux x86	130.4 MB	<a href="#">jdk-7u79-linux-i586.rpm</a>
Linux x86	147.6 MB	<a href="#">jdk-7u79-linux-i586.tar.gz</a>
Linux x64	131.69 MB	<a href="#">jdk-7u79-linux-x64.rpm</a>
Linux x64	146.4 MB	<a href="#">jdk-7u79-linux-x64.tar.gz</a>
Mac OS X x64	196.89 MB	<a href="#">jdk-7u79-macosx-x64.dmg</a>
Solaris x86 (SVR4 package)	140.79 MB	<a href="#">jdk-7u79-solaris-i586.tar.Z</a>
Solaris x86	96.66 MB	<a href="#">jdk-7u79-solaris-i586.tar.gz</a>
Solaris x64 (SVR4 package)	24.67 MB	<a href="#">jdk-7u79-solaris-x64.tar.Z</a>
Solaris x64	16.38 MB	<a href="#">jdk-7u79-solaris-x64.tar.gz</a>
Solaris SPARC (SVR4 package)	140 MB	<a href="#">jdk-7u79-solaris-sparc.tar.Z</a>
Solaris SPARC	99.4 MB	<a href="#">jdk-7u79-solaris-sparc.tar.gz</a>
Solaris SPARC 64-bit (SVR4 package)	24 MB	<a href="#">jdk-7u79-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	18.4 MB	<a href="#">jdk-7u79-solaris-sparcv9.tar.gz</a>
Windows x86	138.31 MB	<a href="#">jdk-7u79-windows-i586.exe</a>
Windows x64	140.06 MB	<a href="#">jdk-7u79-windows-x64.exe</a>

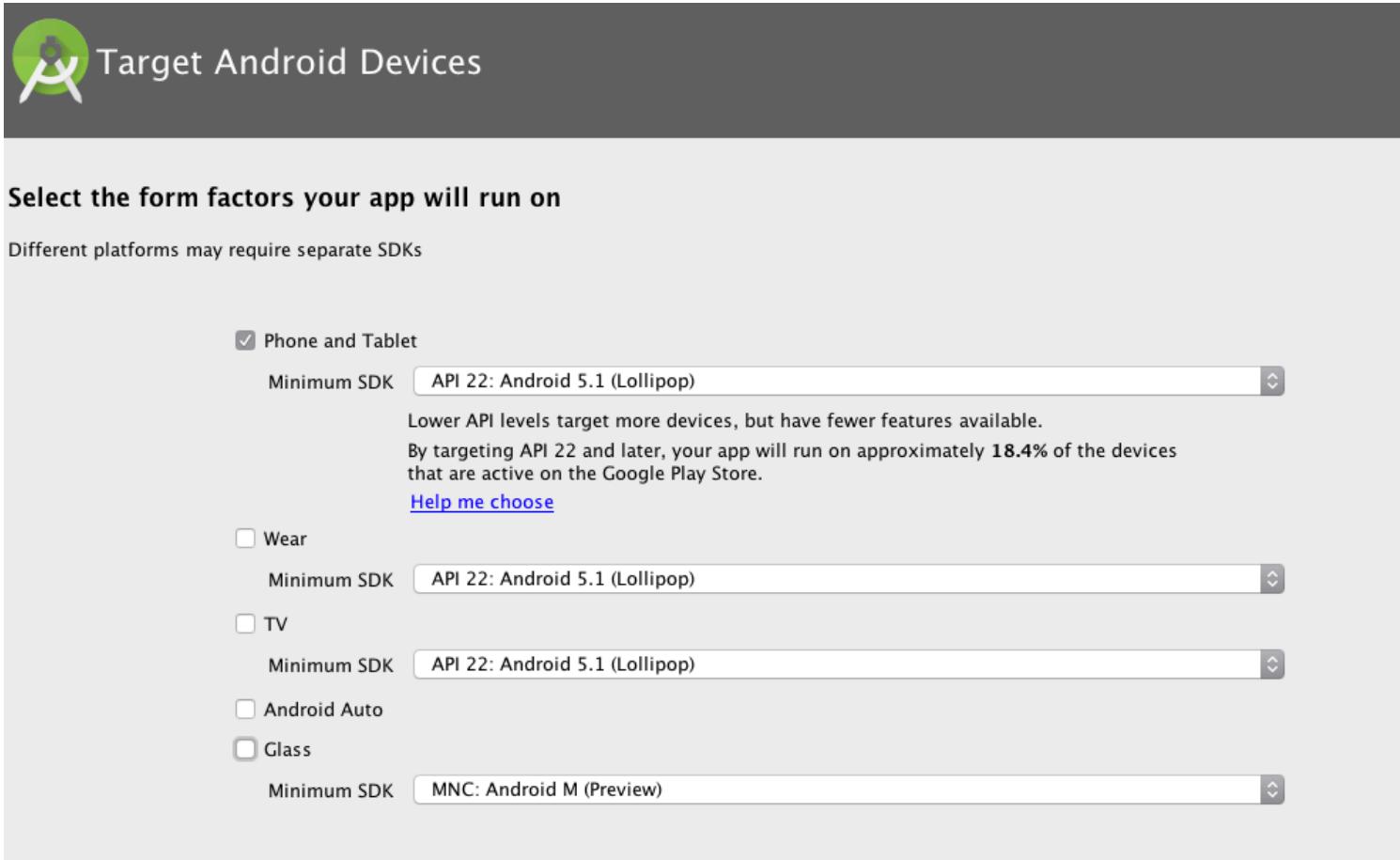
# Setting up the environment

## Android Studio

1. Go to [Android Studio Page](#)
2. Download version for your OS



# Creating new Android Application



# API Version vs Operative System

Platform Version	API Level	VERSION_CODE	Notes
Android 6.0	23	M	API Changes
Android 5.1	22	LOLLIPOP_MR1	Platform Highlights
Android 5.0	21	LOLLIPOP	
Android 4.4W	20	KITKAT_WATCH	KitKat for Wearables Only
Android 4.4	19	KITKAT	Platform Highlights
Android 4.3	18	JELLY_BEAN_MR2	Platform Highlights
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	Platform Highlights
Android 4.1, 4.1.1	16	JELLY_BEAN	Platform Highlights
Android 4.0.3, 4.0.4	15	ICE_CREAM SANDWICH_MR1	Platform Highlights
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM SANDWICH	
Android 3.2	13	HONEYCOMB_MR2	
Android 3.1.x	12	HONEYCOMB_MR1	Platform Highlights
Android 3.0.x	11	HONEYCOMB	Platform Highlights
Android 2.3.4	10	GINGERBREAD_MR1	Platform Highlights
Android 2.3.3			
Android 2.3.2	9	GINGERBREAD	
Android 2.3.1			
Android 2.3			
Android 2.2.x	8	FROYO	Platform Highlights
Android 2.1.x	7	ECLAIR_MR1	Platform Highlights
Android 2.0.1	6	ECLAIR_0_1	
Android 2.0	5	ECLAIR	
Android 1.6	4	DONUT	Platform Highlights
Android 1.5	3	CUPCAKE	Platform Highlights
Android 1.1	2	BASE_1_1	
Android 1.0	1	BASE	

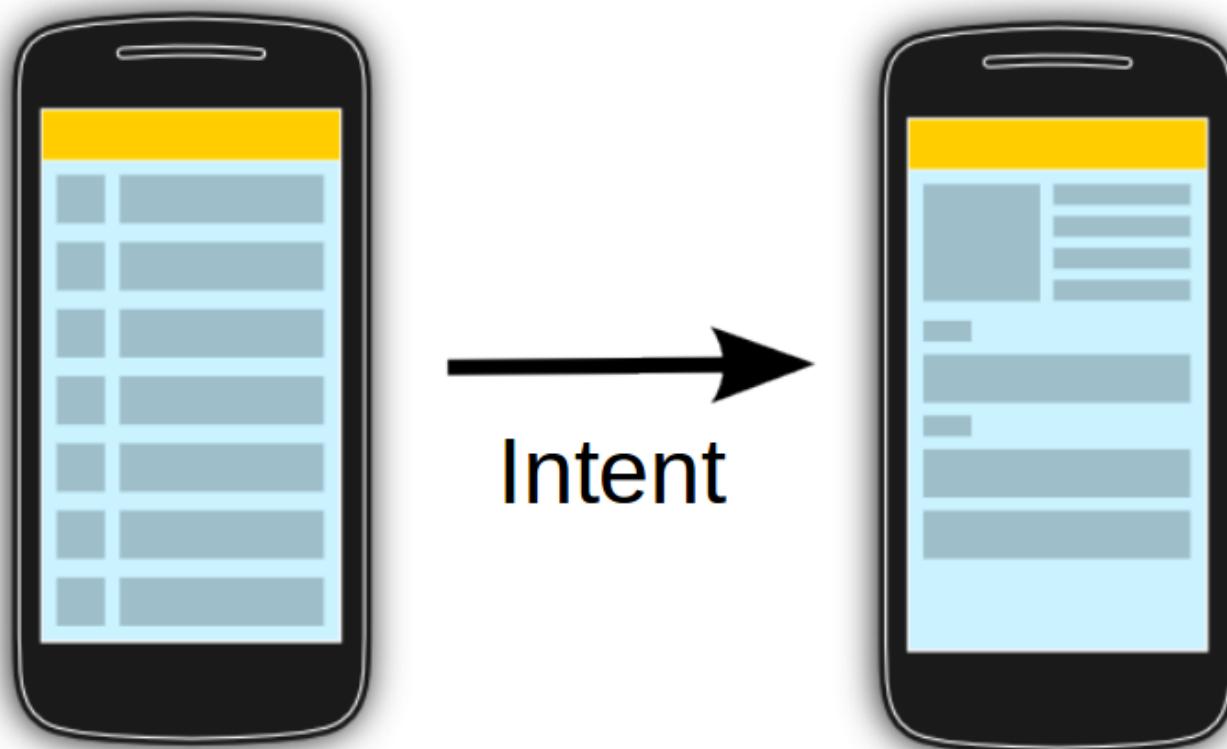
# Android Application



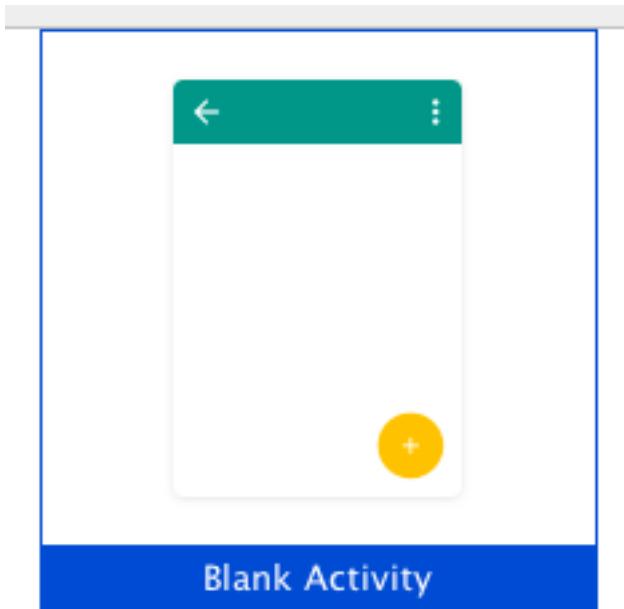
# Web Page



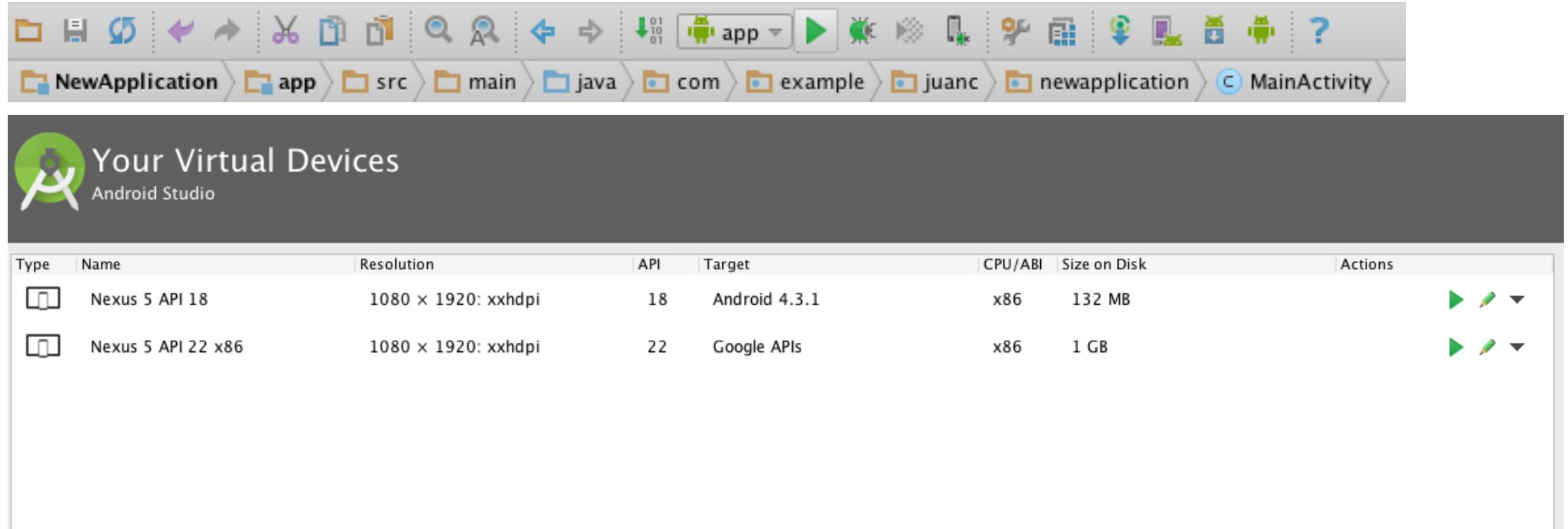
# Android App



# Android First App (API 22 – Android 5.1)



# Running application

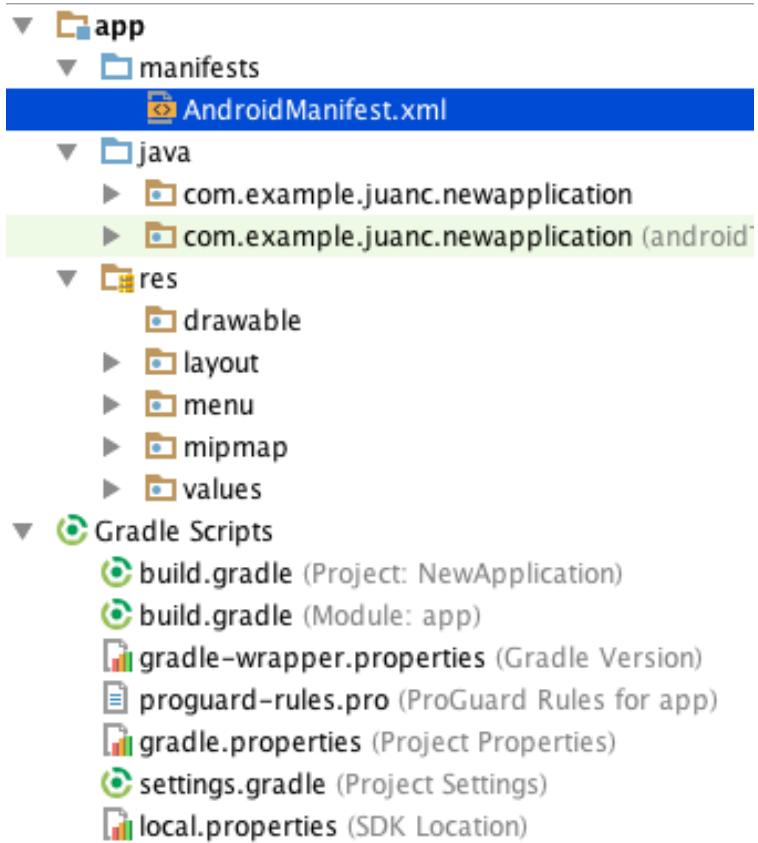


# Lab

- Create a basic project
- Run the application



# Project Structure



# Android Manifest

## 1. Components

- **Activity**
- **Intents**
- Broadcast Receiver
- Services
- Content providers

## 2. Permissions, Package

# Android Manifest

1. Launcher Intent
2. Label and resources

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.juanc.newapplication">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="NewApplication"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="NewApplication"
            android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

# gradle

- gradlew (gradle wrapper)
- build.gradle
  - Task definition
  - Definitions
  - Repositories
- gradle.properties



# Resources

- Drawables - Mipmap (\*)
- Layout
- Values
- Menu



# Drawables

- *ldpi* (low) ~120dpi
- *mdpi* (medium) ~160dpi
- *hdpi* (high) ~240dpi
- *xhdpi* (extra-high) ~320dpi
- *xxhdpi* (extra-extra-high) ~480dpi
- *xxxhdpi* (extra-extra-extra-high) ~640dpi



# Mipmap

“Different home screen launcher apps on different devices show app launcher icons at various resolutions. When app resource optimization techniques remove resources for unused screen densities, launcher icons can wind up looking fuzzy because the launcher app has to upscale a lower-resolution icon for display. To avoid these display issues, apps should use the mipmap/resource folders for launcher icons. The Android system preserves these resources regardless of density stripping, and ensures that launcher apps can pick icons with the best resolution for display”

# Layout

The screenshot shows the Android Studio interface with two main panes: the XML editor on the left and the Layout Editor on the right.

**XML Editor (content\_main.xml):**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingViewBehavior"
    tools:context="com.example.juanc.newapplication.MainActivity"
    tools:showIn="@layout/activity_main">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

</RelativeLayout>
```

**Layout Editor:**

- Palette:** Shows the available layouts and widgets. Under "Layouts", "RelativeLayout" is selected. Under "Widgets", various UI components like Plain TextView, Large Text, Medium Text, Small Text, Button, Small Button, Radio Button, Check Box, Switch, Toggle Button, Image Button, Image View, Progress Bar (Large), Progress Bar (Normal), Progress Bar (Small), Progress Bar (Horizontal), and Seek Bar are listed.
- Preview:** Displays a mobile device screen with a blue header bar containing a red circular icon with a white envelope symbol. The main content area displays the text "Hello World!". The bottom of the screen shows a dark grey navigation bar with three icons.
- Toolbar:** Includes standard Android Studio tools like Undo, Redo, Find, Replace, and Save.

# Values – Applications Constants

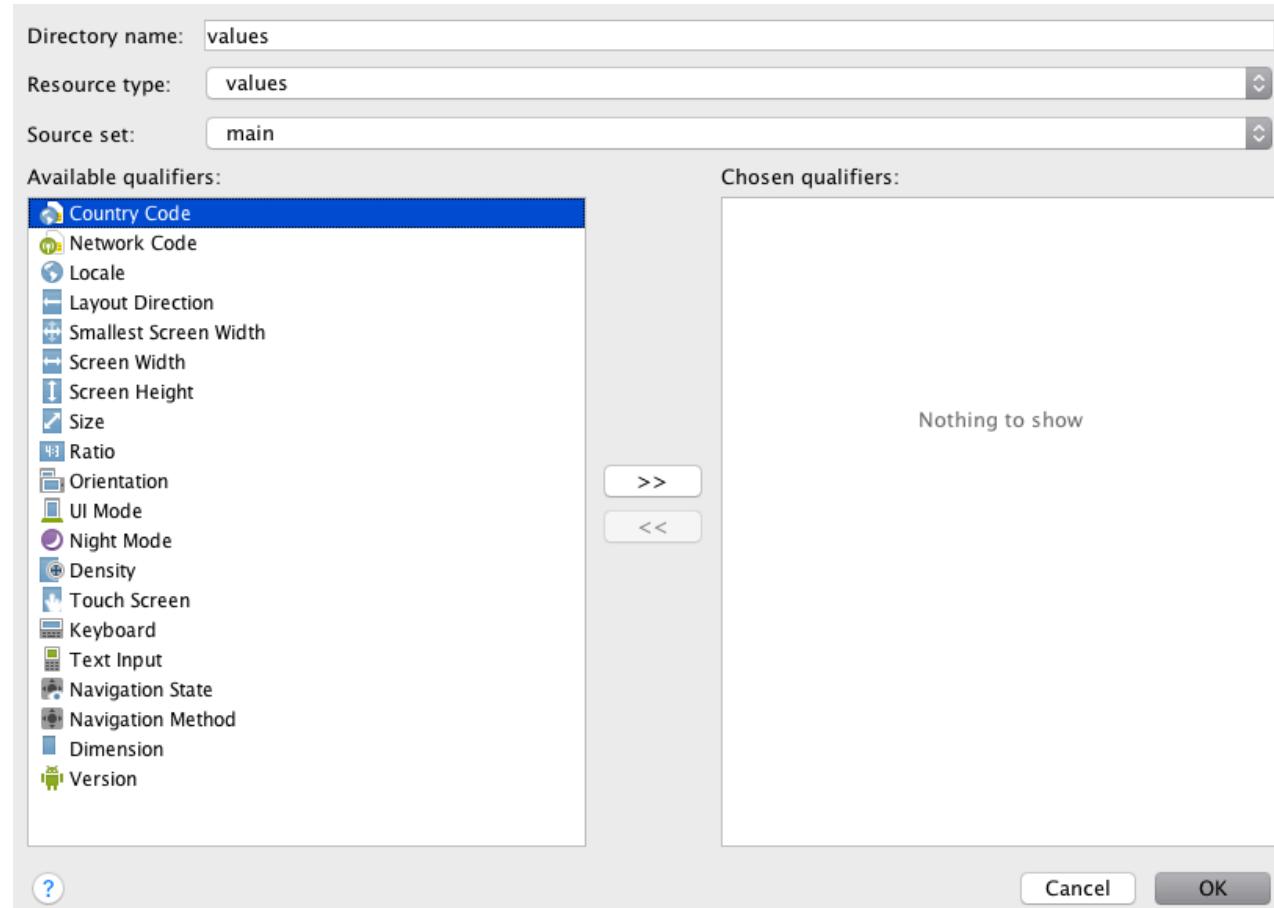
```
    android:text="@string/text_view_title"

}<resources>
    <string name="app_name">NewApplication</string>
    <string name="action_settings">Settings</string>
    <string name="text_view_title">Hola Mundo!</string>
}</resources>
```



KEEP  
CALM  
AND  
HARD  
CODE

# Values – Resource Directory

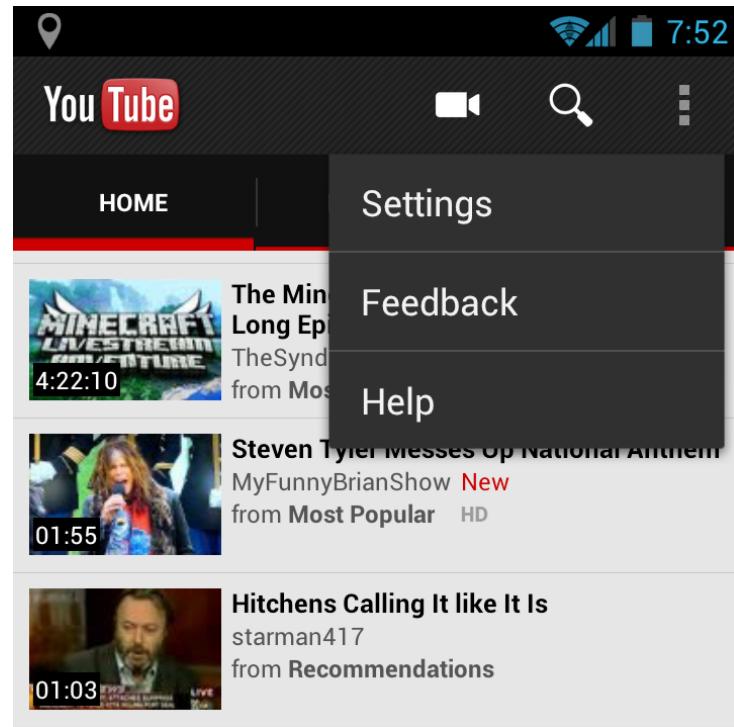


# Menu

Options menu and app bar

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/file"
          android:title="@string/file" >
        <!-- "file" submenu -->
        <menu>
            <item android:id="@+id/create_new"
                  android:title="@string/create_new" />
            <item android:id="@+id/open"
                  android:title="@string/open" />
        </menu>
    </item>
</menu>
```

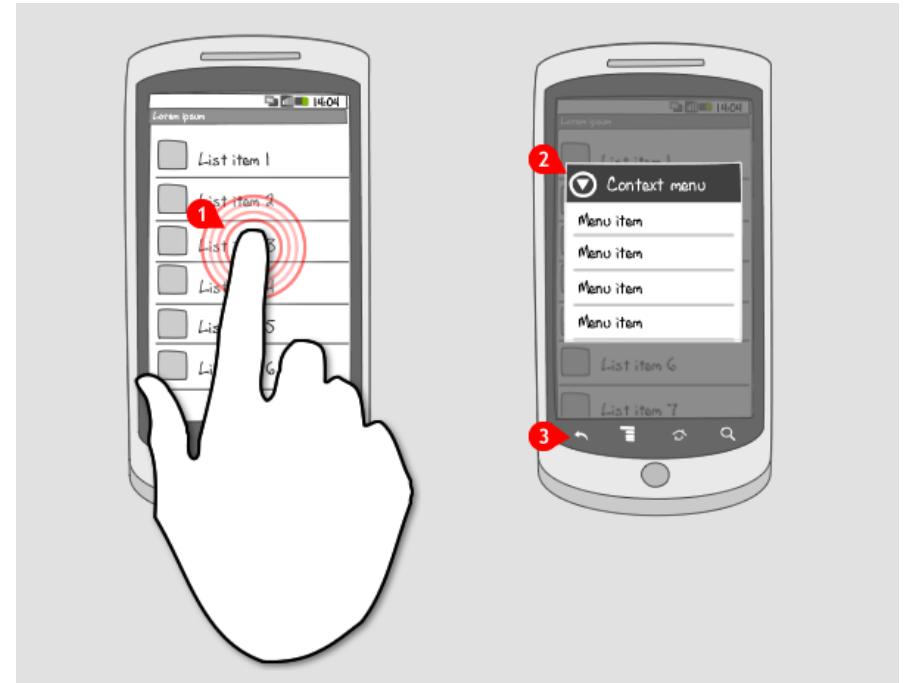
```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.game_menu, menu);
    return true;
}
```



# Menu

Context menu and contextual action mode

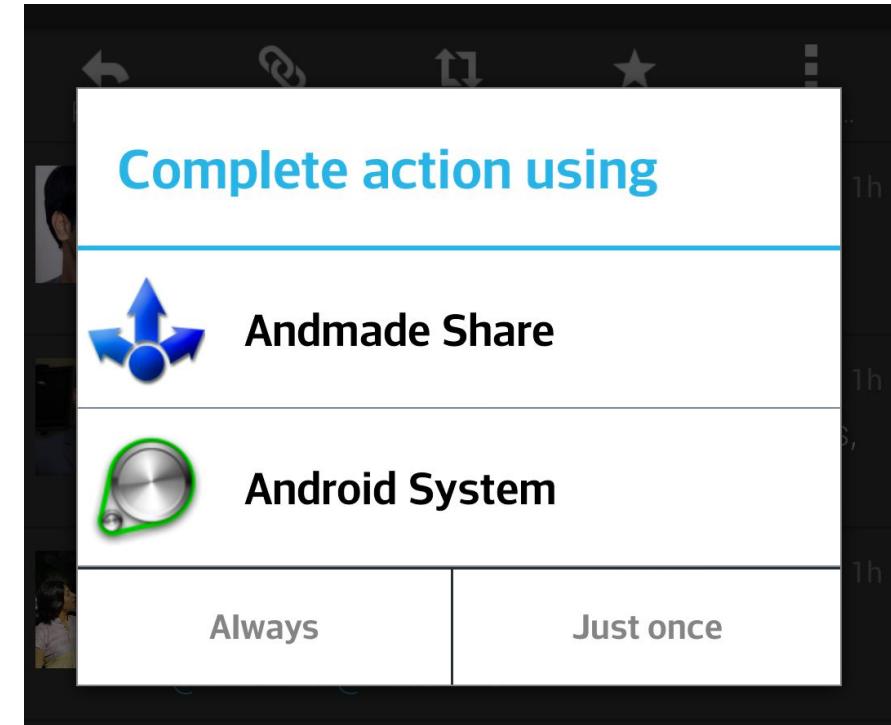
1. registerForContextMenu(view) on Create
2. onCreateContextMenu
3. onContextItemSelected



# Menu - Popup menu

```
<ImageButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/ic_overflow_holo_dark"  
    android:contentDescription="@string/descr_overflow_button"  
    android:onClick="showPopup" />
```

```
public void showPopup(View v) {  
    PopupMenu popup = new PopupMenu(this, v);  
    MenuInflater inflater = popup.getMenuInflater();  
    inflater.inflate(R.menu.actions, popup.getMenu());  
    popup.show();  
}
```

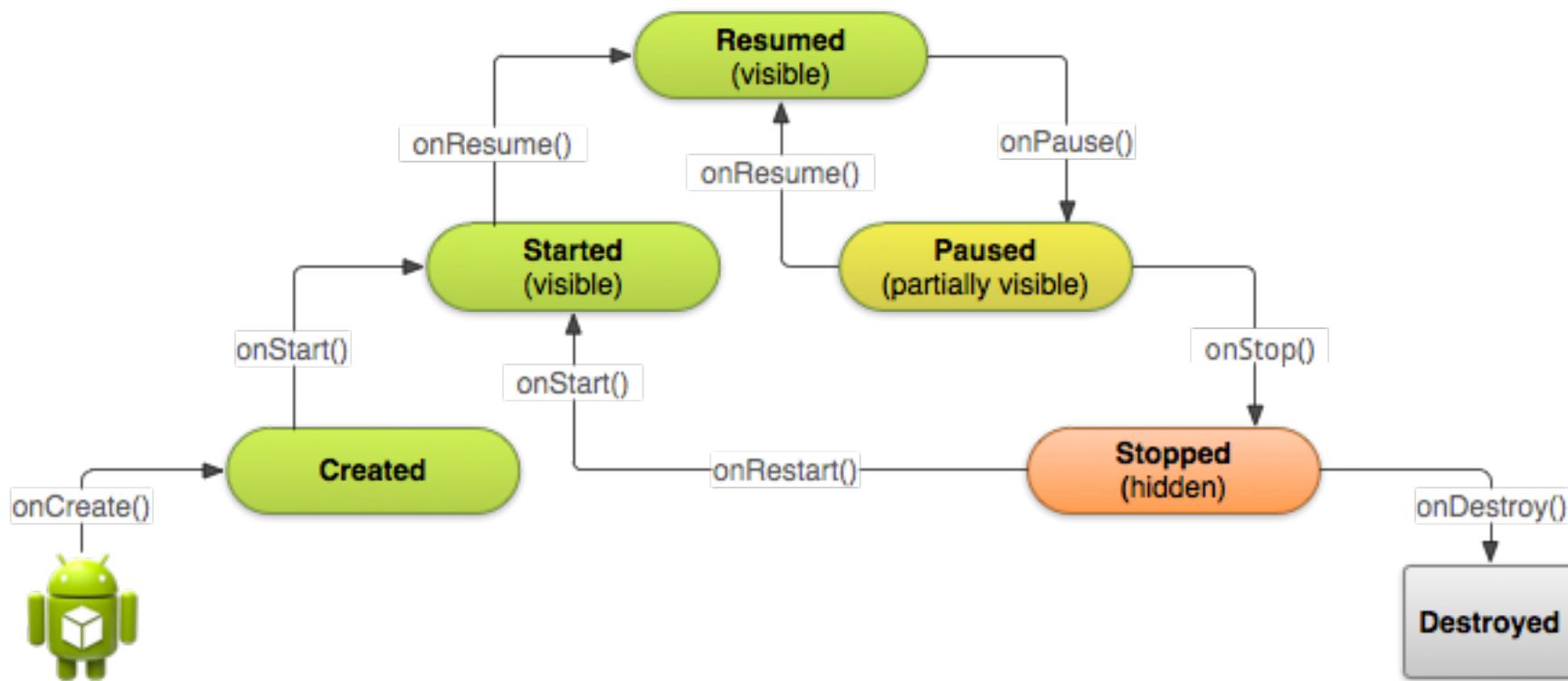


# Lab

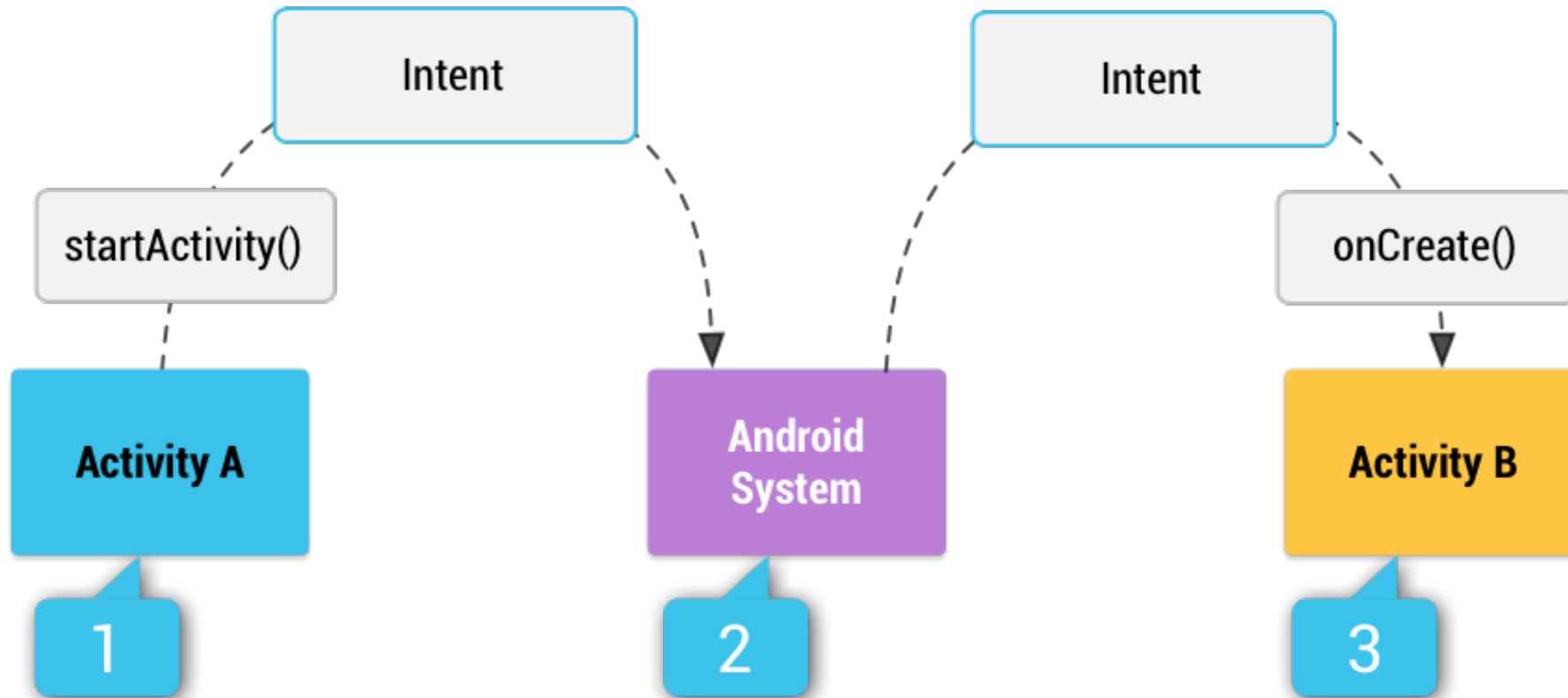
1. Change Hello world to be a text constant.
2. Change Message if screen orientation is landscape.
3. Create one menu (any type) with an action to display Toasts message.



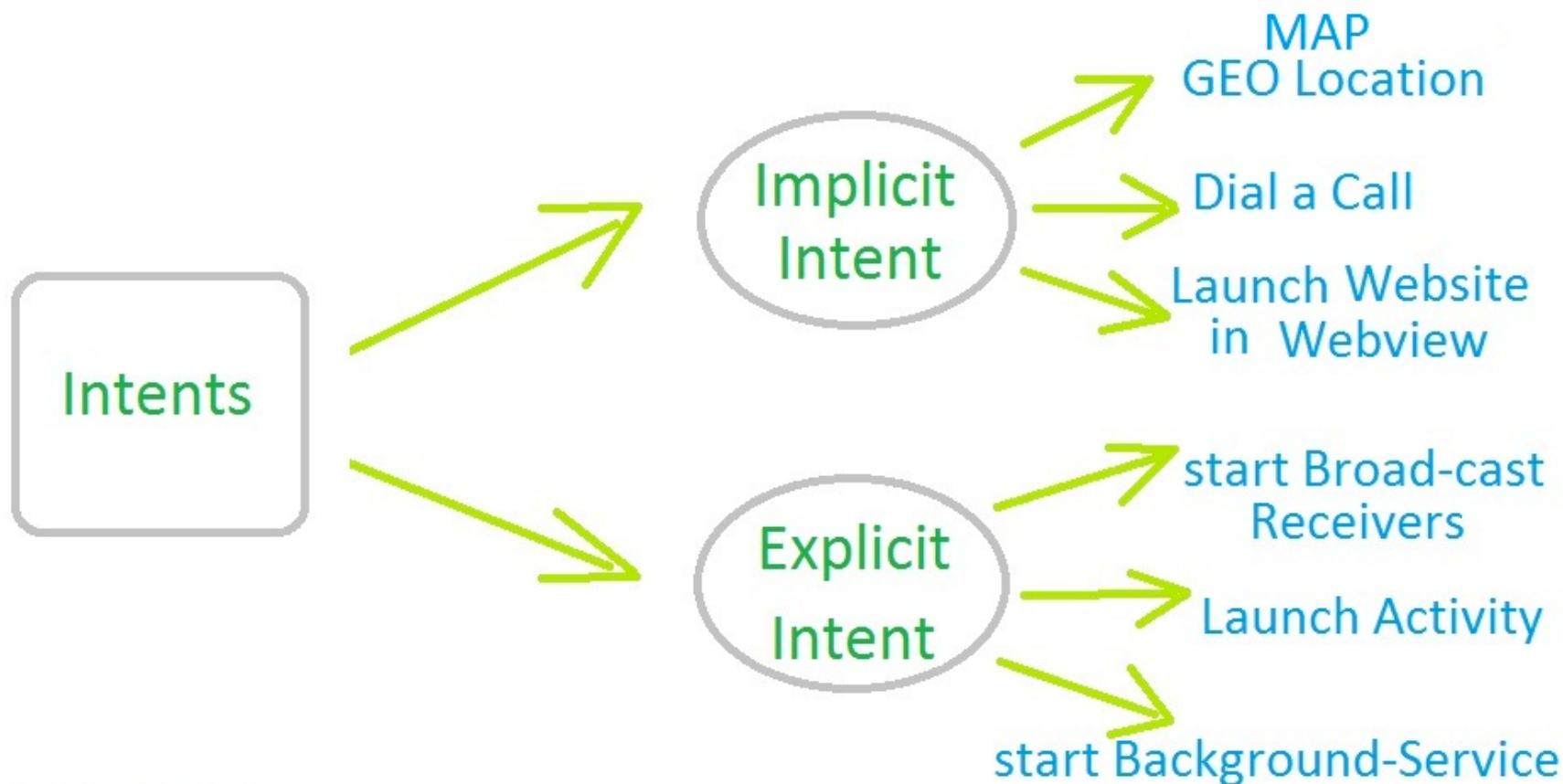
# Activity Life Cycle



# Intent



# Intent



# Intent

**Action** : the general action to be performed, such as ACTION VIEW, ACTION EDIT, ACTION MAIN, etc.

Data : the data to operate on, such as a person record in the contacts database, expressed as a Uri.

**Category:** Gives additional information about the action to execute. For example, CATEGORY LAUNCHER means it should appear in the Launcher as a top-level application.

**Type:** Specifies an explicit type (a MIME type) of the intent data

**Component:** Specifies an explicit name of a component class to use for the intent

**Extras:** **Extra information**

# Lab

Add three buttons handlers to Activity

1. Open a web site (Implicit)
2. Open a second Activity (Explicit)
3. Add login messages to Activity events



# Lab Hints

## *Button Handler action*

```
<Button  
    android:layout_height="wrap_content"  
    android:layout_width="wrap_content"  
    android:text="Action_Button"  
    android:onClick="action" />
```



```
public void action(View view)  
{  
    // Do something really cool  
}
```

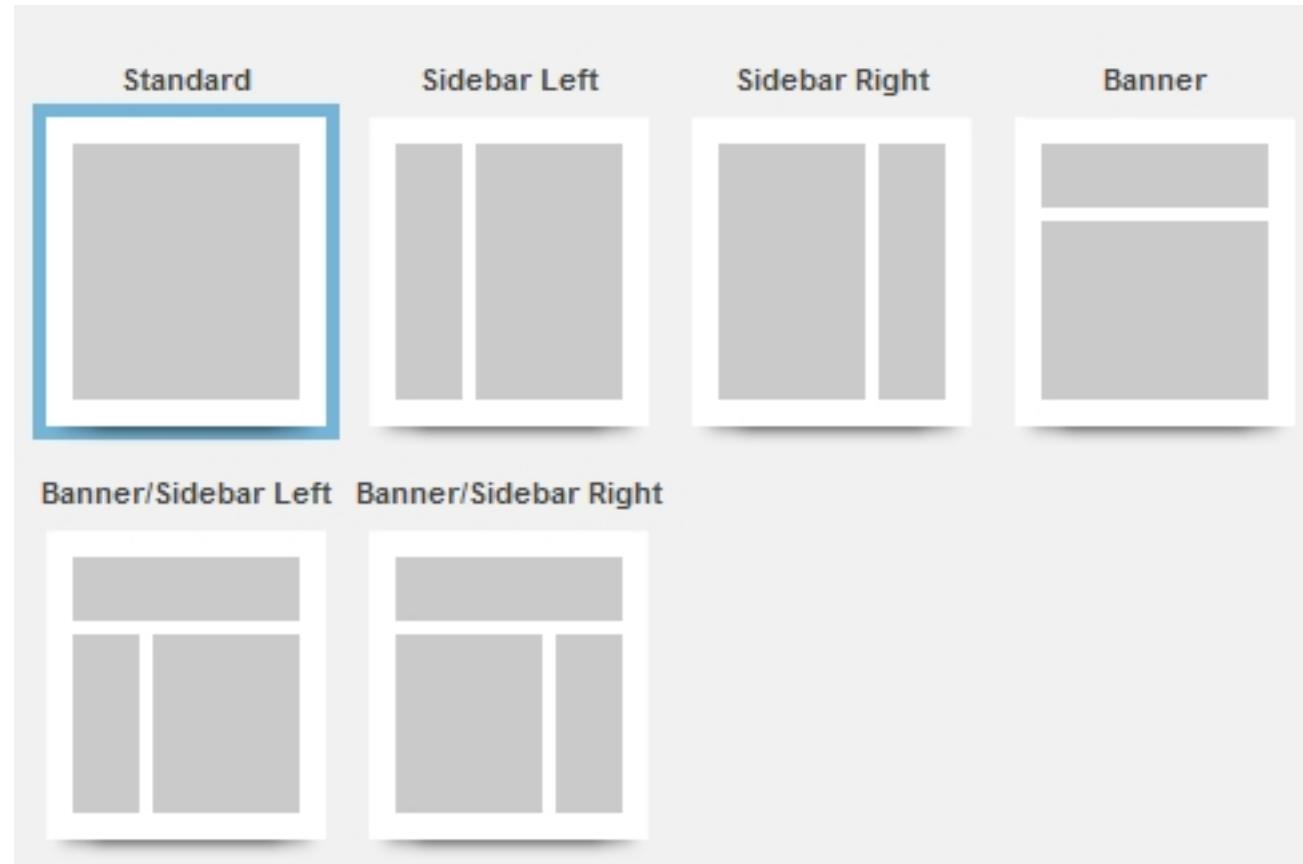
## *Calling Activity*

```
Intent intent = new Intent(this, SecondActivity.class);  
startActivity(intent);
```

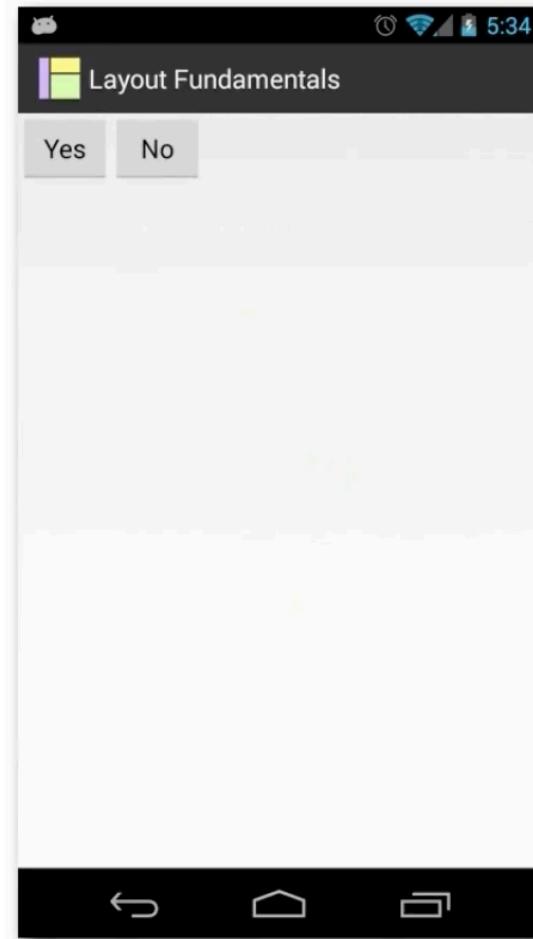
## *Calling Activity*

```
private static final String TAG = "MyActivity";  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    Log.v(TAG, "activity create");
```

# Android Layout fundamentals



# Elements positioning and size



# Size (height & width)

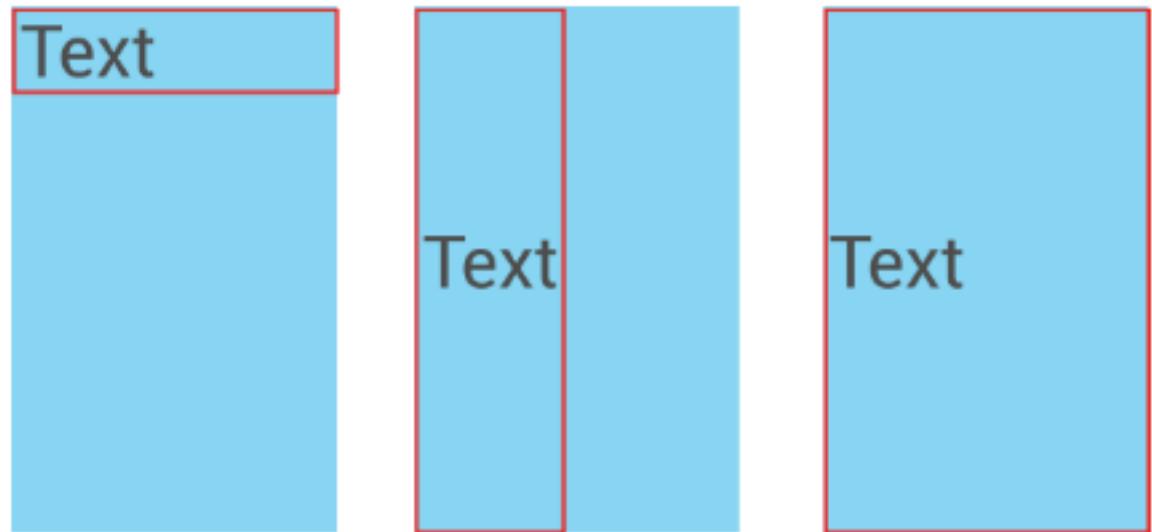
- wrap\_content
- match\_parent

**match\_parent**

`android:layout_width="wrap_content"  
 android:layout_height="match_parent"`

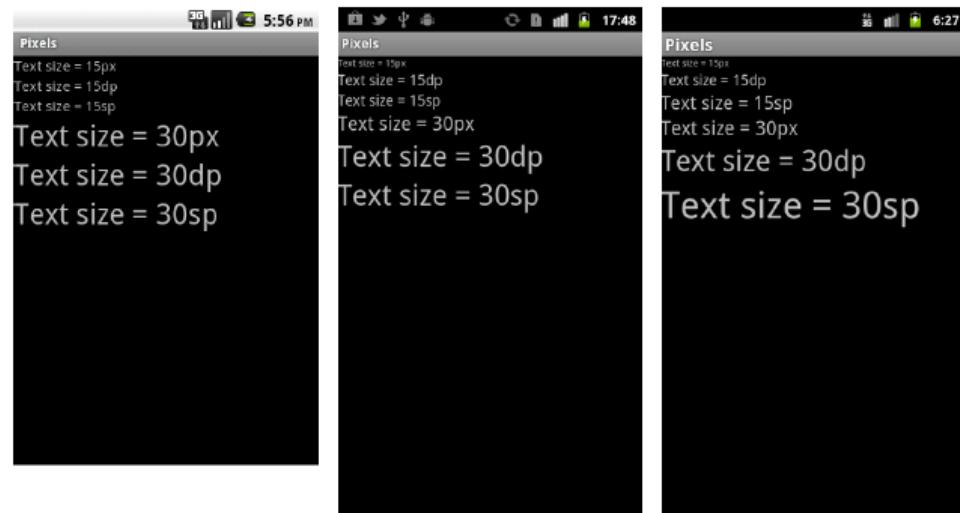
`android:layout_width="match_parent"  
 android:layout_height="wrap_content"`

`android:layout_width="match_parent"  
 android:layout_height="match_parent"`



# Text Size

- **Dp** Density-independent Pixels
- **Px** Pixels
- **Mm** Millimeters
- **In** Inches



# Margin, Padding

bis diam neo delenit vindico ne. Iriure letalis singularis, secundum ad  
luptatum importunus vel quidne. Vel cogo antehabeo antehabeo, velit  
defui abigo ~~percedo~~ <sup>oppeto</sup> paulatim proprius, valde exputo. Suscipere

**Padding**



## This is a Header

Nimis exerci illi in oppeto mara epulae diam imputo eros duis, vindico.

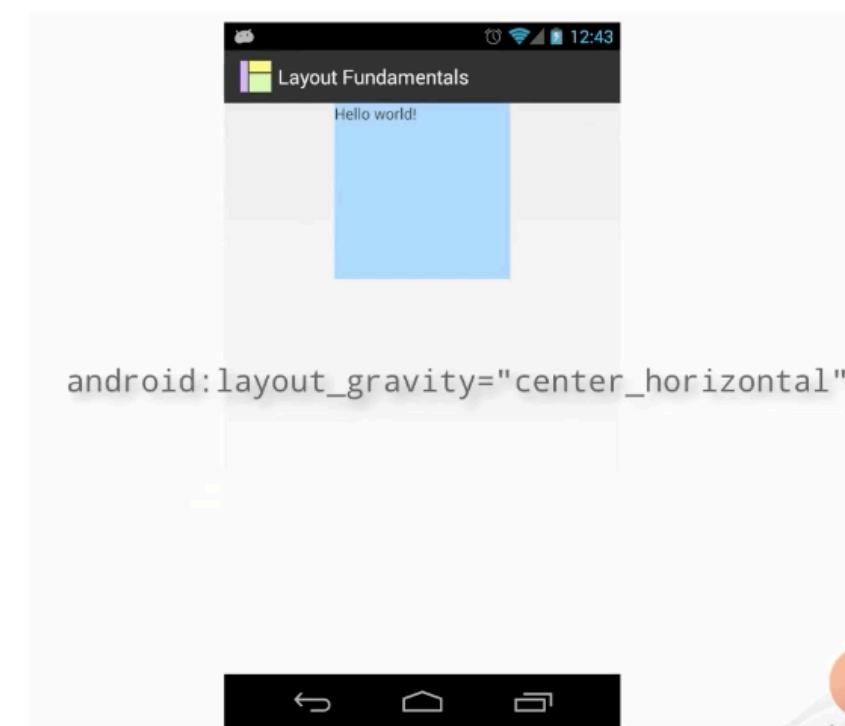
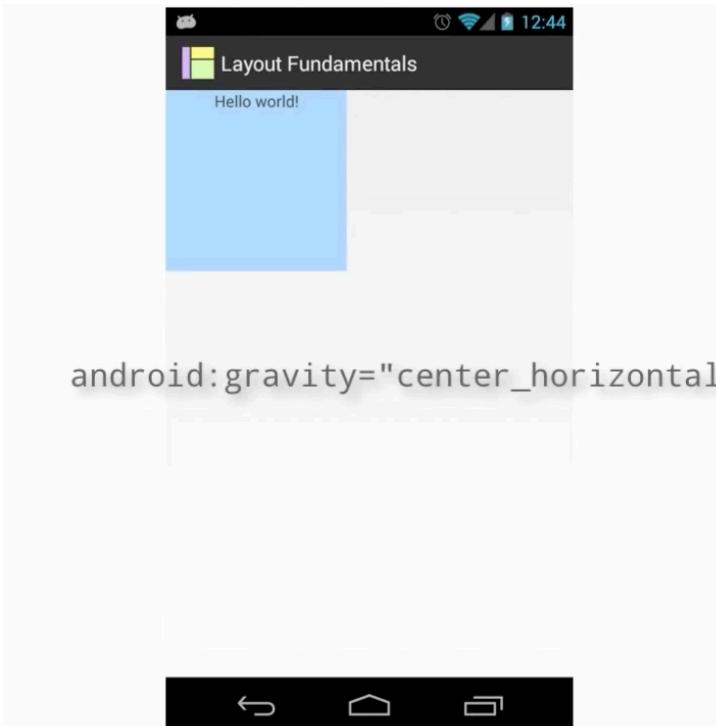
Vel bis diam neo delenit vindico ne. Iriure letalis singularis, secundum ad  
luptatum importunus vel quidne. Vel cogo antehabeo antehabeo, velit

**Margin**



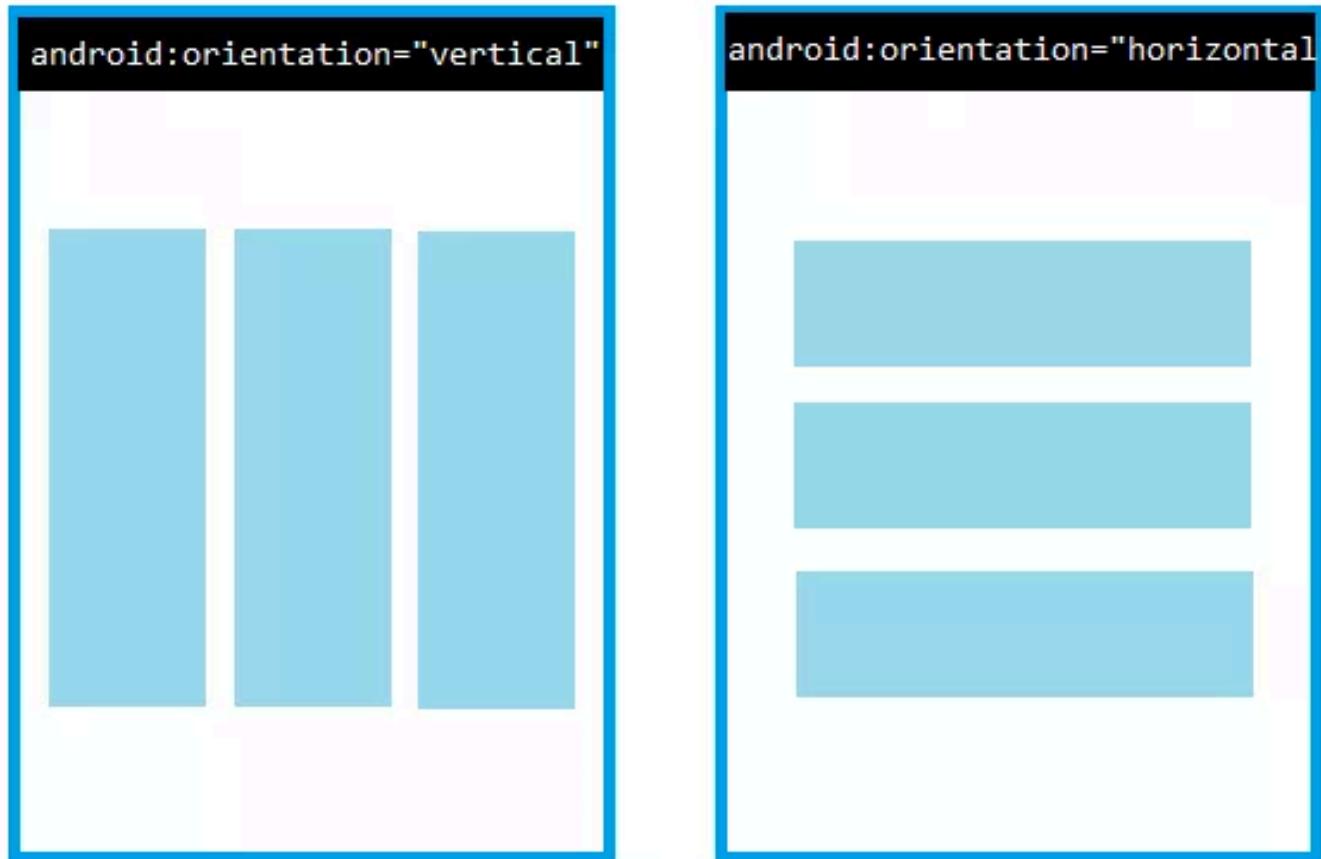
# Gravity

- layout\_gravity
- gravity

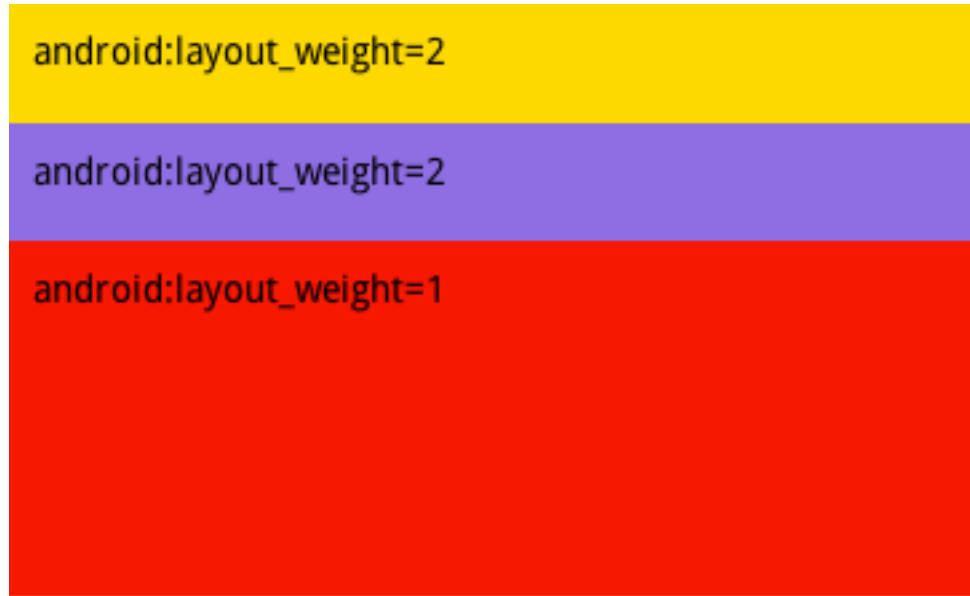


# Linear Layout

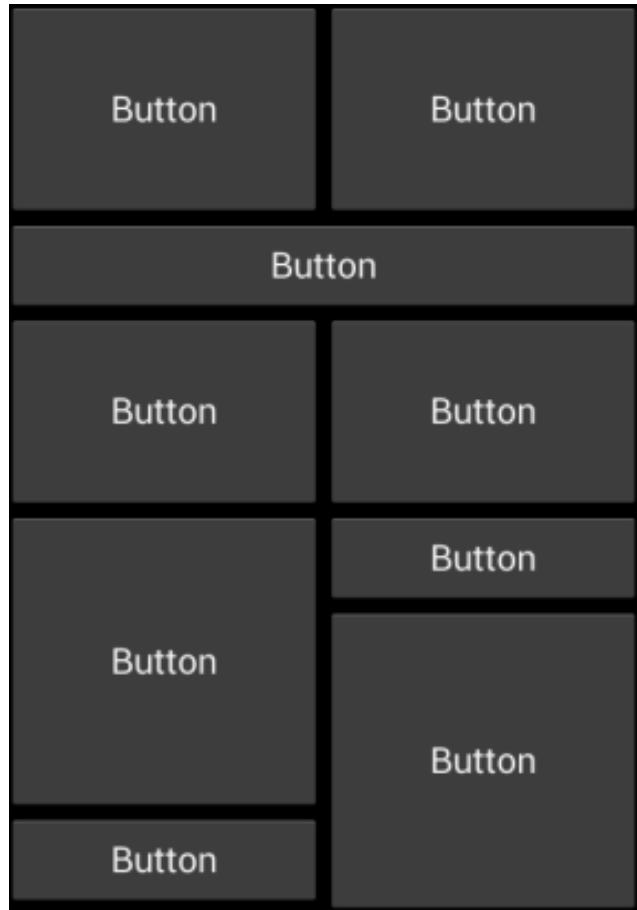
“Put elements one after each other”



# Layout - weight



# Lab



# Relative Layout

Nested Layout (**Poor performance**)

Relative

- Position
- Alignment

To

- Parent
- Sibling

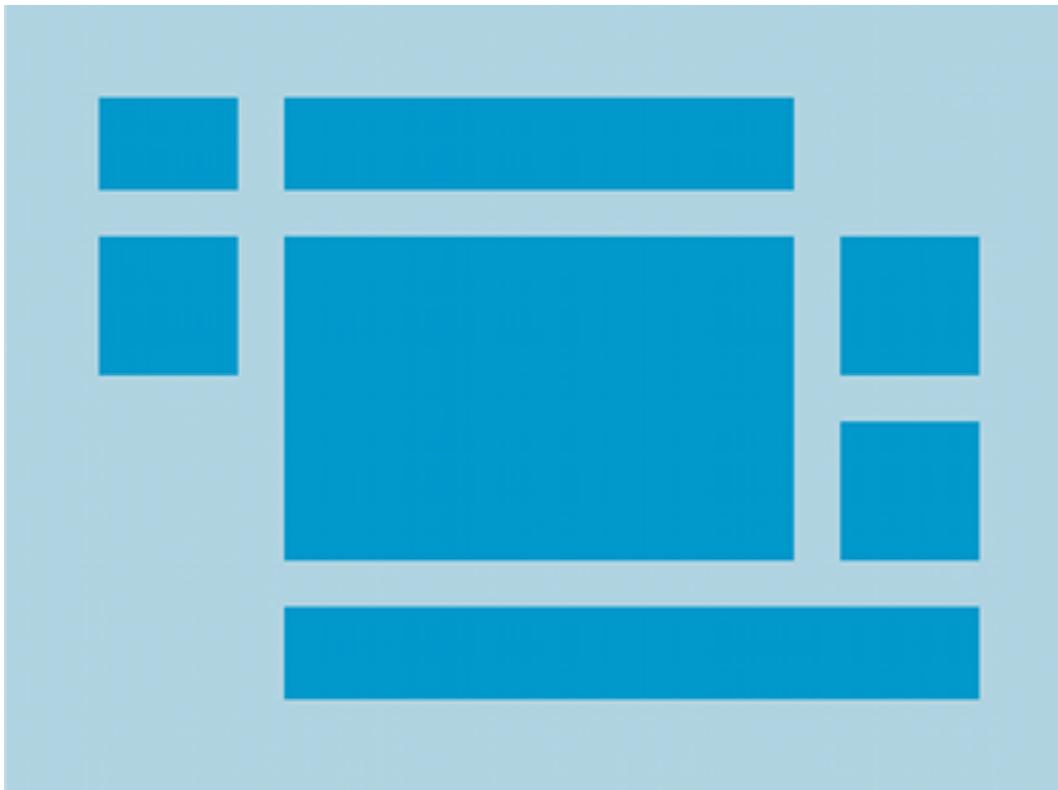


# Relative Layout

Attribute Name	Description
<code>android:layout_above</code>	Positions the bottom edge of this view above the given anchor view ID.
<code>android:layout_alignBaseline</code>	Positions the baseline of this view on the baseline of the given anchor view ID.
<code>android:layout_alignBottom</code>	Makes the bottom edge of this view match the bottom edge of the given anchor view ID.
<code>android:layout_alignEnd</code>	Makes the end edge of this view match the end edge of the given anchor view ID.
<code>android:layout_alignLeft</code>	Makes the left edge of this view match the left edge of the given anchor view ID.
<code>android:layout_alignParentBottom</code>	If true, makes the bottom edge of this view match the bottom edge of the parent.
<code>android:layout_alignParentEnd</code>	If true, makes the end edge of this view match the end edge of the parent.
<code>android:layout_alignParentLeft</code>	If true, makes the left edge of this view match the left edge of the parent.
<code>android:layout_alignParentRight</code>	If true, makes the right edge of this view match the right edge of the parent.
<code>android:layout_alignParentStart</code>	If true, makes the start edge of this view match the start edge of the parent.
<code>android:layout_alignParentTop</code>	If true, makes the top edge of this view match the top edge of the parent.
<code>android:layout_alignRight</code>	Makes the right edge of this view match the right edge of the given anchor view ID.
<code>android:layout_alignStart</code>	Makes the start edge of this view match the start edge of the given anchor view ID.
<code>android:layout_alignTop</code>	Makes the top edge of this view match the top edge of the given anchor view ID.
<code>android:layout_alignWithParentIfMissing</code>	If set to true, the parent will be used as the anchor when the anchor cannot be found for <code>layout_toLeftOf</code> , <code>layout_toRightOf</code> , etc.
<code>android:layout_below</code>	Positions the top edge of this view below the given anchor view ID.
<code>android:layout_centerHorizontal</code>	If true, centers this child horizontally within its parent.
<code>android:layout_centerInParent</code>	If true, centers this child horizontally and vertically within its parent.
<code>android:layout_centerVertical</code>	If true, centers this child vertically within its parent.
<code>android:layout_toEndOf</code>	Positions the start edge of this view to the end of the given anchor view ID.
<code>android:layout_toLeftOf</code>	Positions the right edge of this view to the left of the given anchor view ID.
<code>android:layout_toRightOf</code>	Positions the left edge of this view to the right of the given anchor view ID.
<code>android:layout_toStartOf</code>	Positions the end edge of this view to the start of the given anchor view ID.

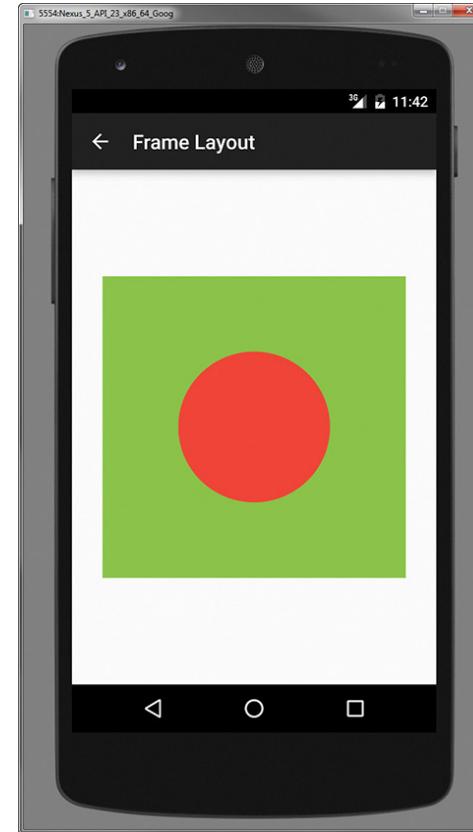
# Table Layout

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="1">
    <TableRow>
        <TextView
            android:text="@string/table_layout_4_open"
            android:padding="3dip" />
        <TextView
            android:text="@string/table_layout_4_open_shortcut"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>
    <TableRow>
        <TextView
            android:text="@string/table_layout_4_save"
            android:padding="3dip" />
        <TextView
            android:text="@string/table_layout_4_save_shortcut"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>
</TableLayout>
```



# Frame Layout

- Each View is drawn from the top-left corner of the layout



# Grid Layout

android:columnCount="4"

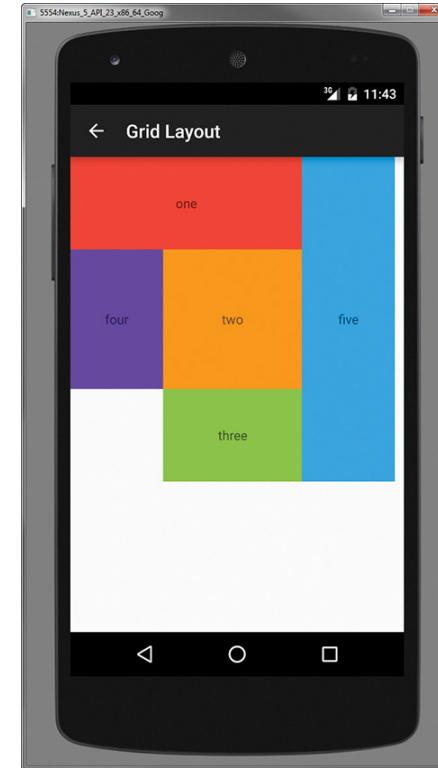
android:rowCount="4"

android:layout\_column="0"

android:layout\_columnSpan="3"

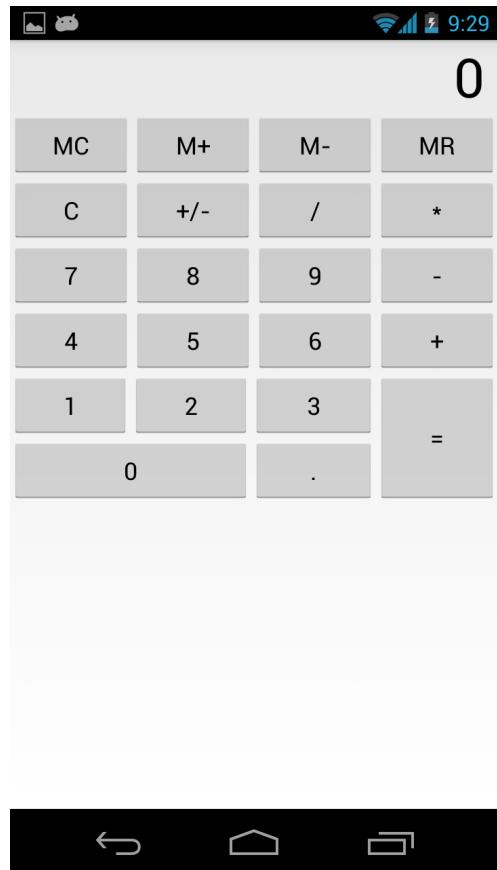
android:layout\_row="0"

android:layout\_rowSpan="2"



# Lab

- Create calculator  
Layout using  
Relative or table  
Layout



D H M S → LD<sup>®</sup>  
\* LD = LernTech Date