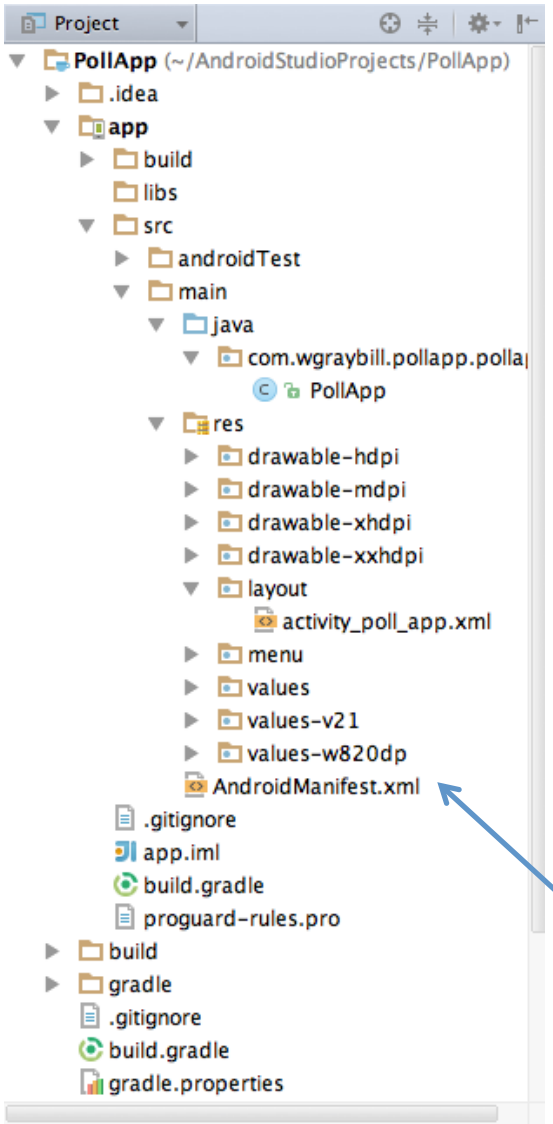
# Meet-up 4: Intro to Android

# Today's Meet-up

- Anatomy of an Android Project
- Android Layouts
- Practice with Layouts

# Android Project Structure

# Android Project Structure




Automatically generated files, act as "glue" between .java and .xml files

Your Java Code goes in here

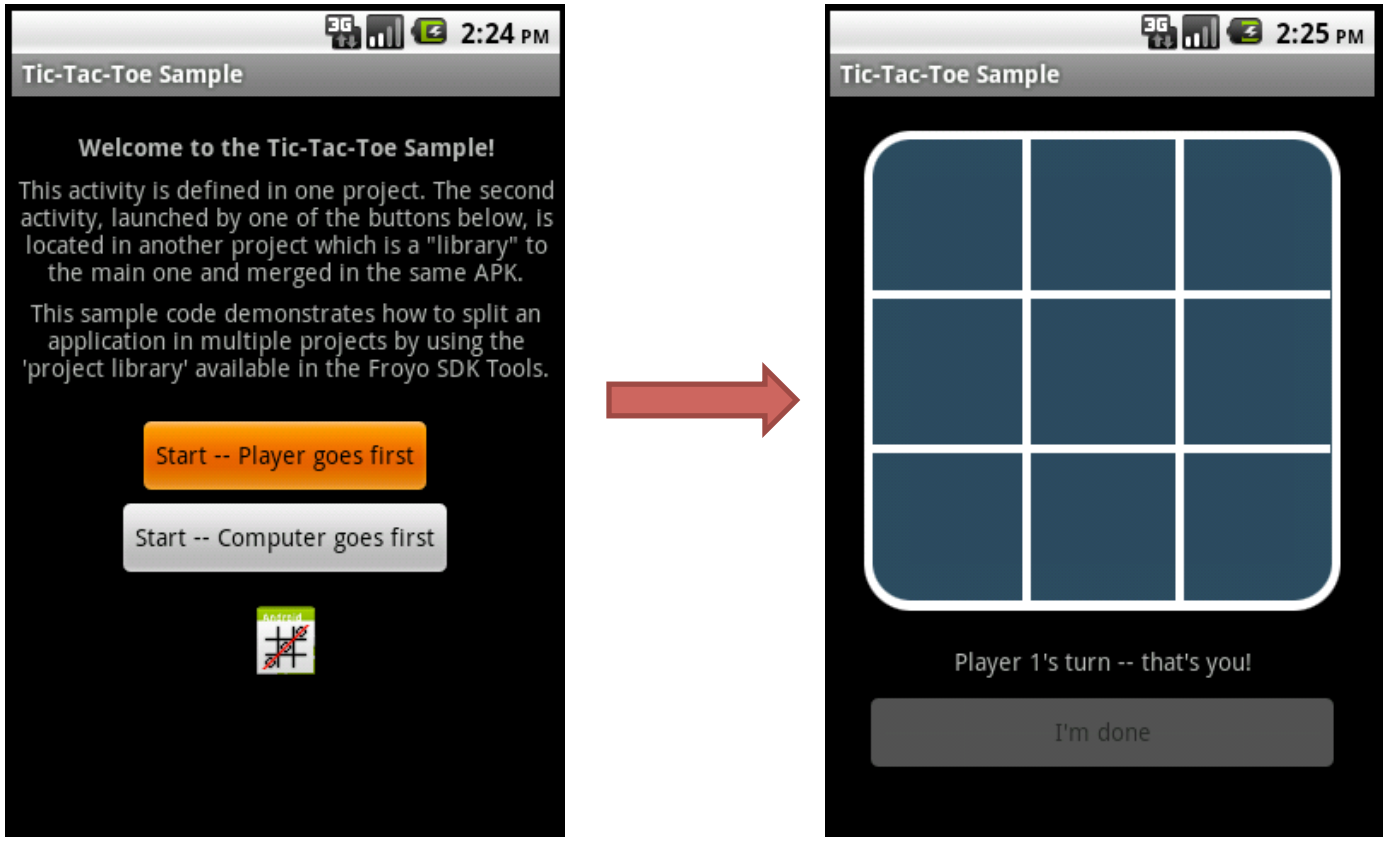
Folders for graphic resources

Defines UI layout

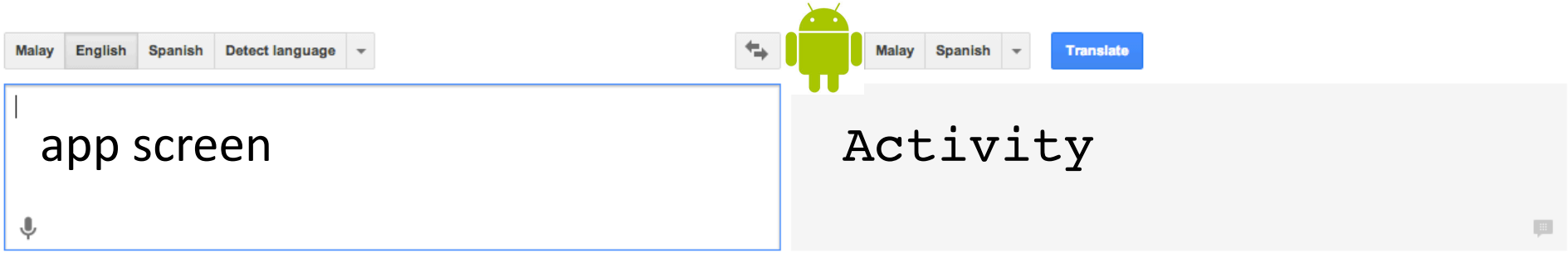
Values (strings, styles, dimensions) are defined here

`AndroidManifest.xml`
First file to be executed, points to Java code and contains system definitions/permission

# Example: Tic-Tac-Toe




Google
Translate


Malay
English
Spanish
Detect language
Malay
Spanish
Translate
app screen
Activity

# Activity File .java

MainActivity.java

```
package com.example.exampleapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

}
```

Package declaration

Import statements

Extend Activity Class

Create UI (as defined in Layout .xml)

Create Options Menu

# AndroidManifest.xml

ExampleApp Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exampleapp"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.exampleapp.MainActivity"
            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>
```

Compatibility Information (in `build.gradle` file in Android Studio)

Icon/ App name/
Style defined

Points to main Activity file

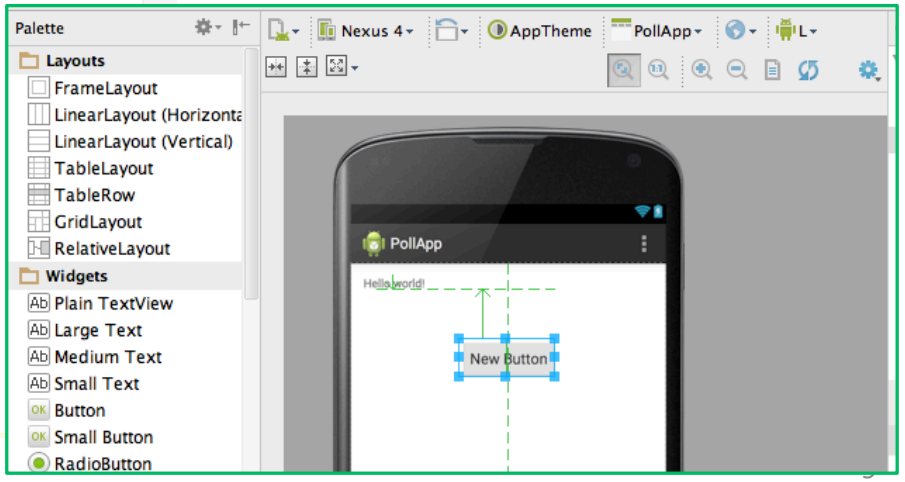
# The Layout File .xml

Activities – one page of app, only one main activity can exist

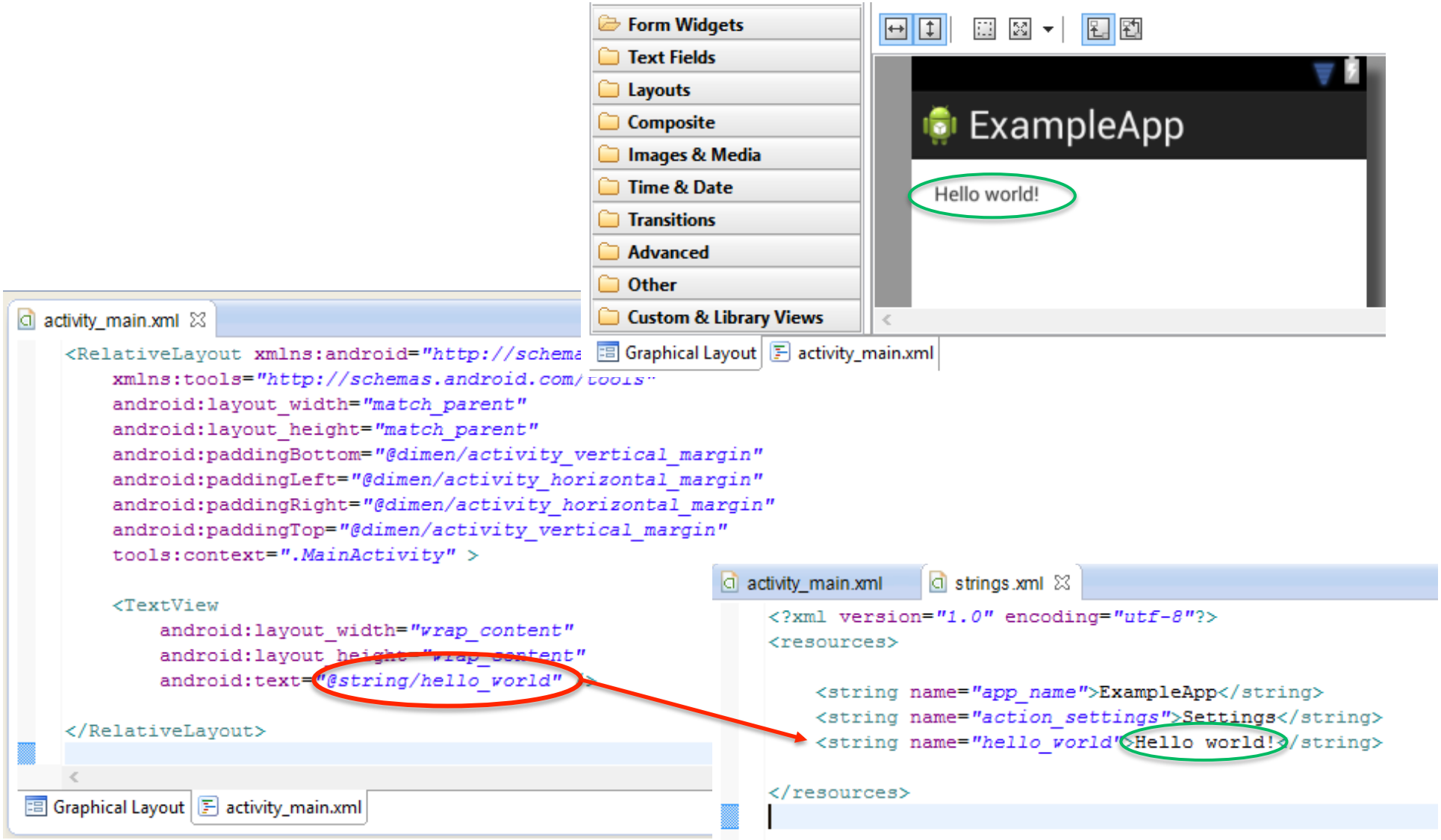
PollApp.java | activity_poll_app.xml | strings.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/an
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="64dp"
    android:paddingRight="64dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context=".PollApp">

    <TextView
        android:text="@string/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button"
        android:layout_marginTop="59dp"
        android:layout_below="@+id/textView"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

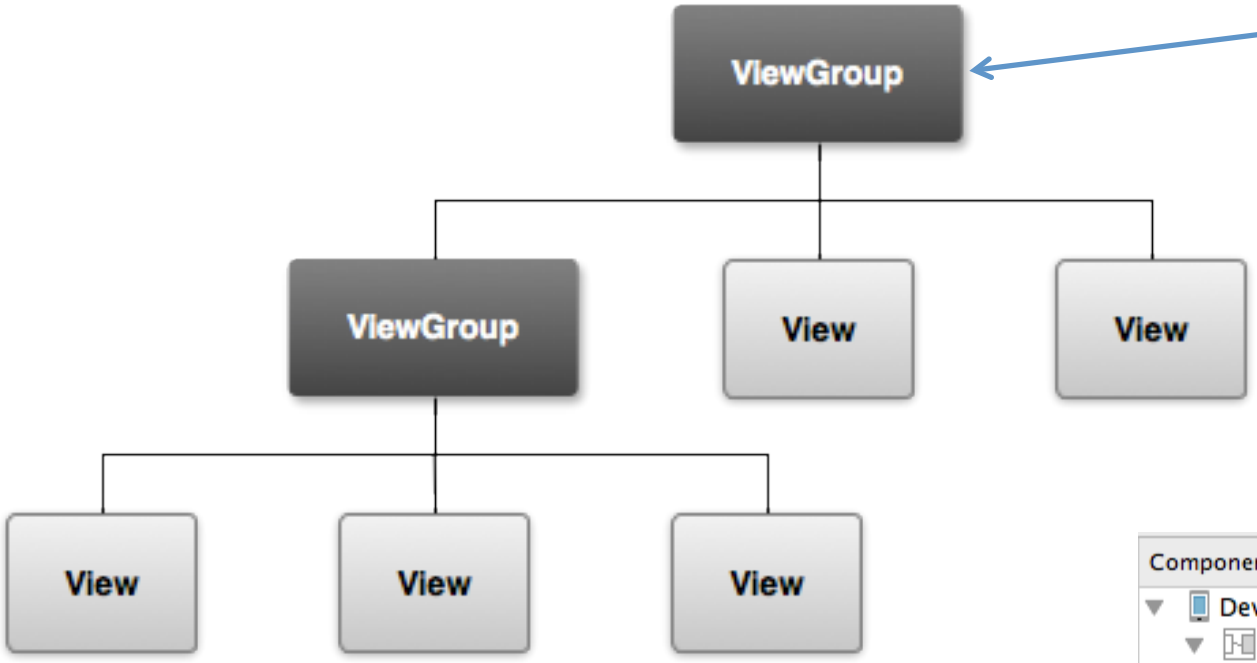
Graphical Layout



# Value Files .xml

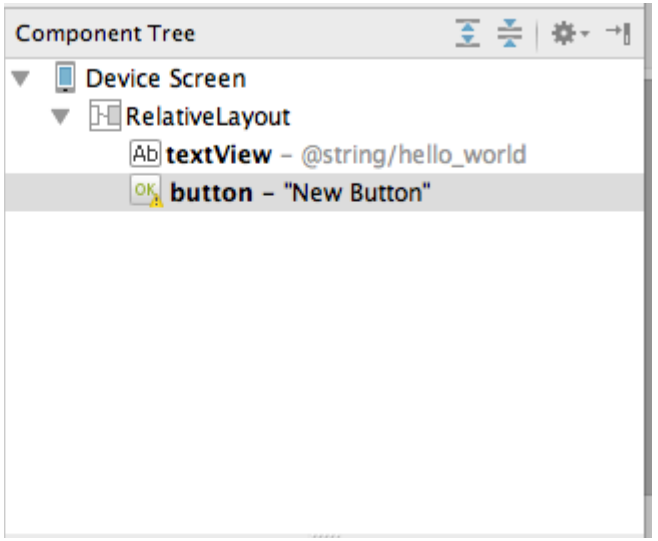


# Building a UI: Layouts




RelativeLayout,
LinearLayout,
etc.

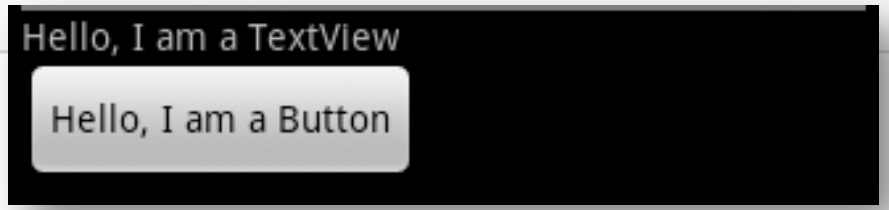
Button,
MapView, etc.



# LinearLayout

Arrange components one after another, left-to-right, top-to-bottom:

```
<?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:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```



# Relative Layout

Position widgets relative to each other: good for more complicated UIs

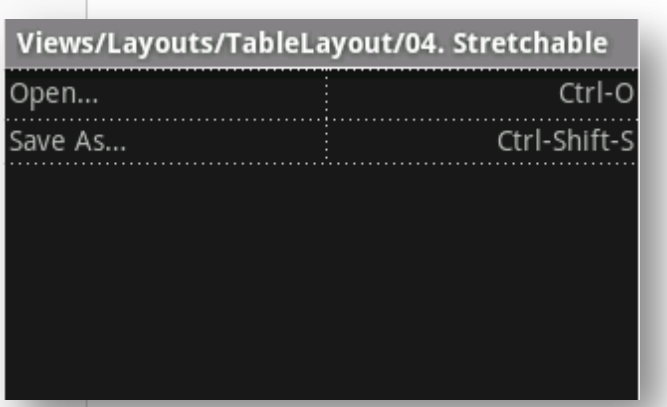
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:background="@drawable/blue"
                android:padding="10px" >

    <TextView android:id="@+id/label"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:text="Type here:" />

    <EditText android:id="@+id/entry"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:background="@android:drawable/editbox_background"
              android:layout_below="@id/label" />

</RelativeLayout>
```


`android:layout_below` is an attribute that can be used only with RelativeLayout. Other such attributes include `layout_alignParentRight`, and `layout_toLeftOf`.

# TableLayout

## Position components in rows and columns:

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_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>
```



# Many more layouts and widgets…

- Linear Layout
- Relative Layout
- Grid Layout
- Frame Layout
- Table Layout

- Context Menu
- Option Menu
- Sub Menu
- Sliders

- Graphics
- Animations
- Videos

- Etc, etc, etc…

- Button
- EditText (a text box)
- TextView (a text label)
- ListView
- GridView
- TabView
- Spinner (a drop-down menu)
- CheckBox
- RadioButton
- ToggleButton
- RatingBar
- MapView (for embedding Google Maps objects in applications)
- WebView (for embedding web browsers in applications)

## …which are furthermore all customizable

# Today's Assignment

# Play with Layout design

- Work in pairs
- Use both xml and graphical interface
- Be creative! You don't need to match the example