
# Basic Android Development

*Prepared by Leslie Ho*

*14/5/2018*

# Pre-Installation(s)

- Android Studio 3.1.2 : https://developer.android.com/studio/#downloads
- Min SDK: 4.4 KitKat
- Compile SDK: 8.0 Oreo
- Target SDK: 8.0 Oreo
- AVD: Nexus 4 (4.7”); System Image: Oreo

**Instructions**

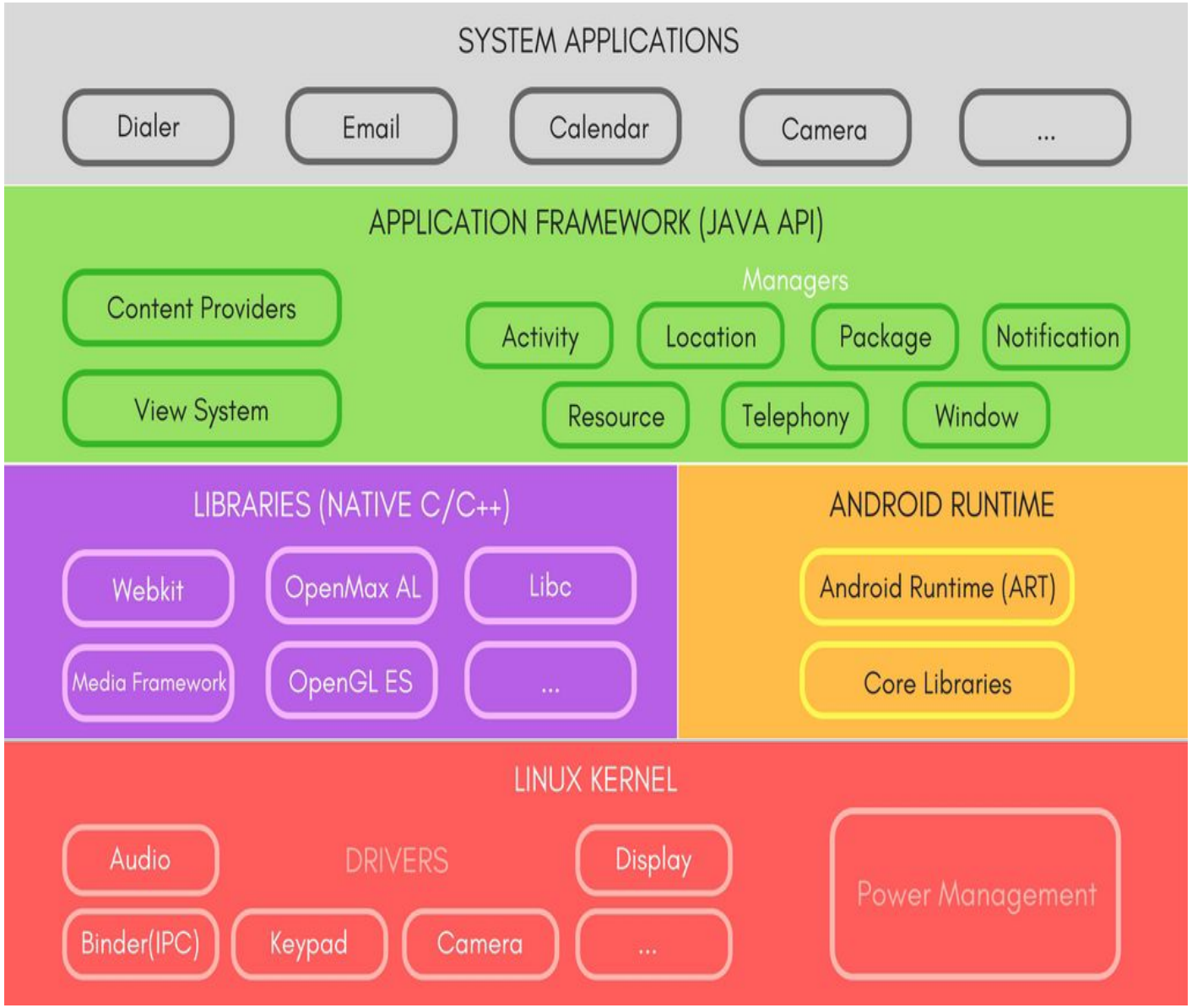
https://drive.google.com/open?id=1PzVQG-mFSuUrsV6G9RTEnuWE8mkJ73luw8aVhtYyL74

# Content.

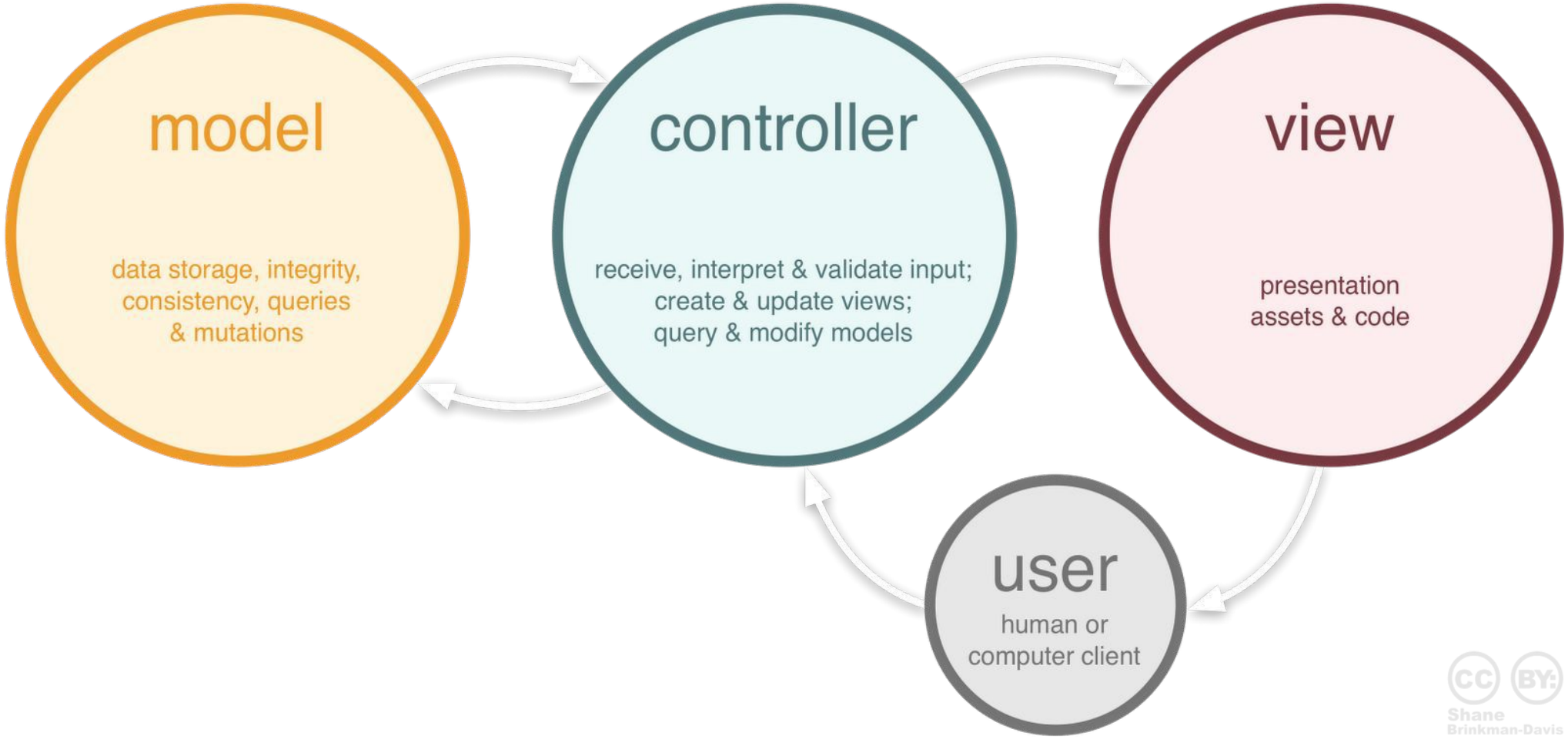
- Android Architecture
- Studio Interface
- UI Controls (element + layouts)
- Importing Images
- Scroll View
- Class Activity
- Scripting (wiring your UI)
- Optimization

*Appendix: AVD set-up*

# System Architecture

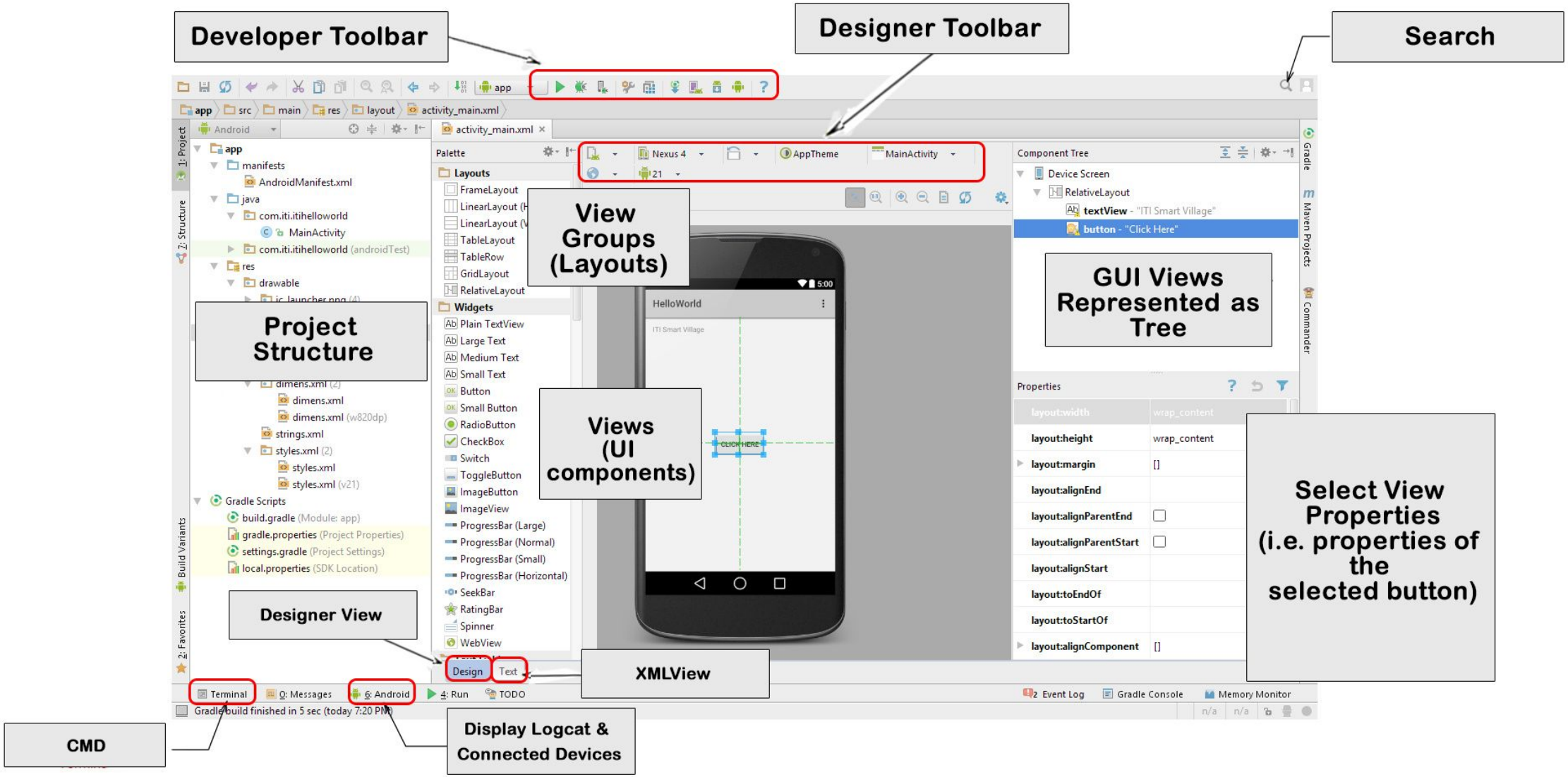


# Android MVC

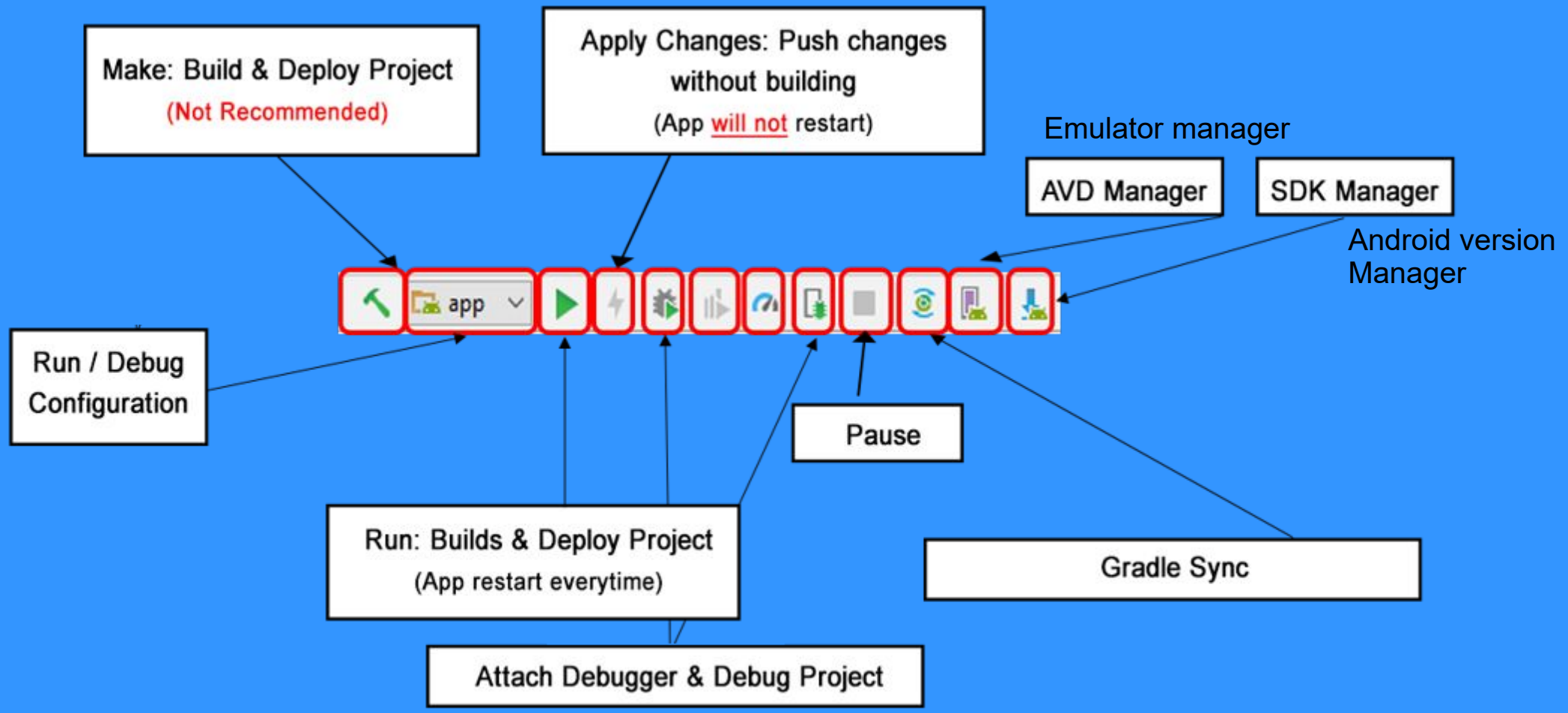


# Android Studio

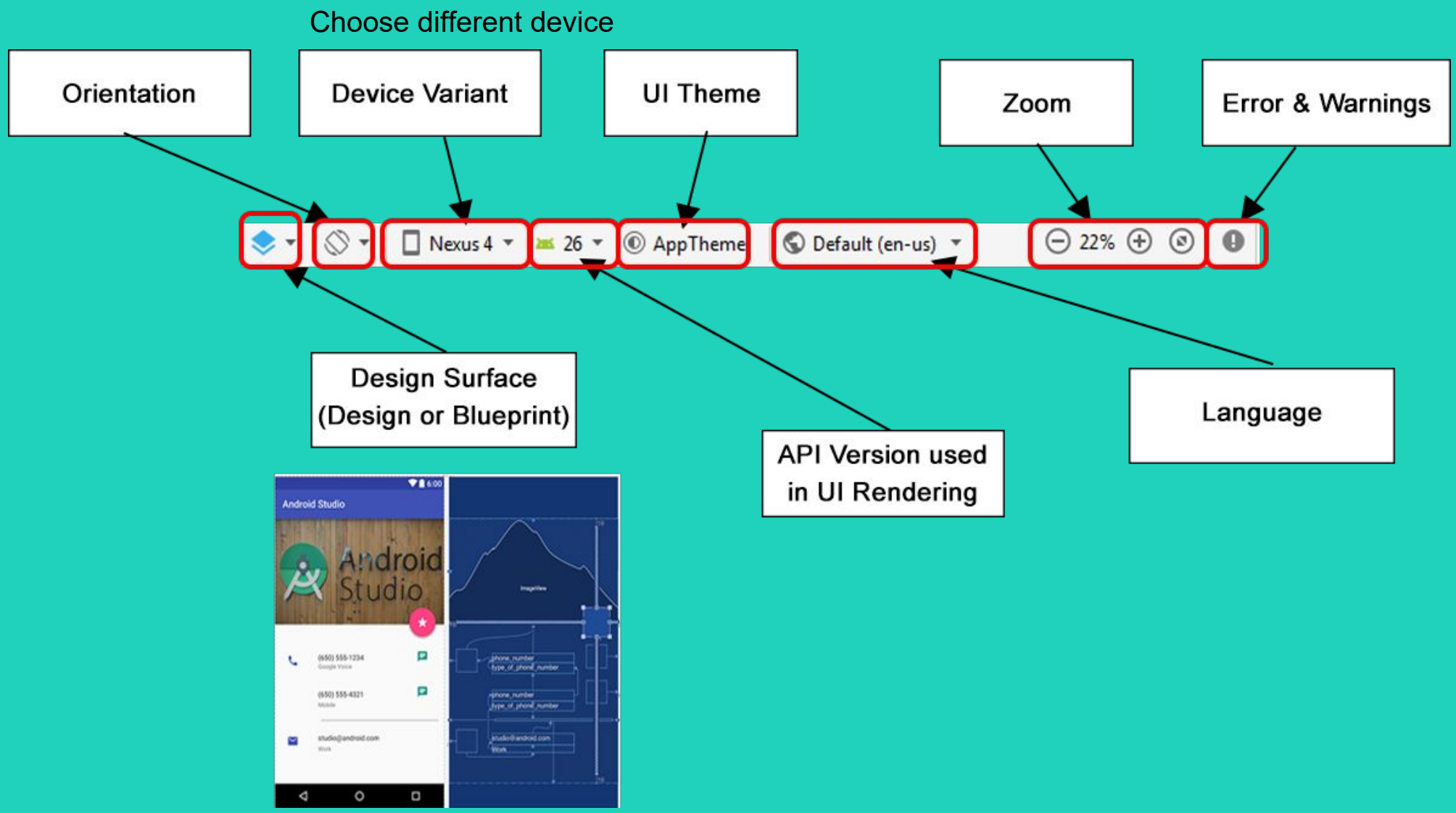
# Studio Interface



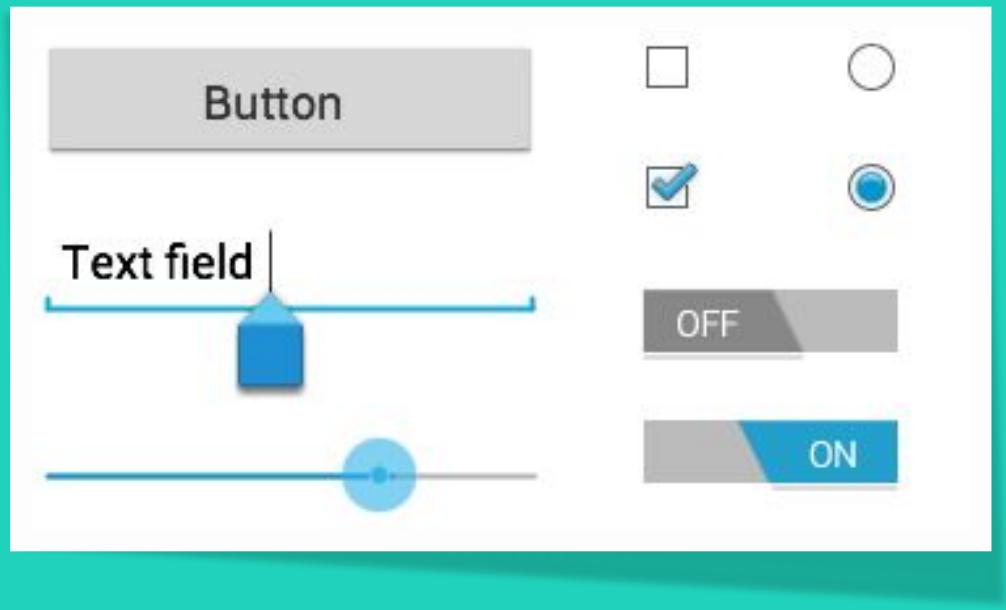
# Developer Toolbar



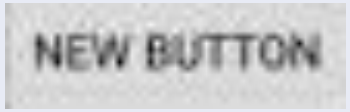
# Designer Toolbar



# UI Controls/ Element



# UI Controls

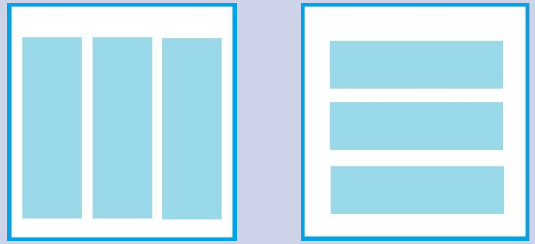
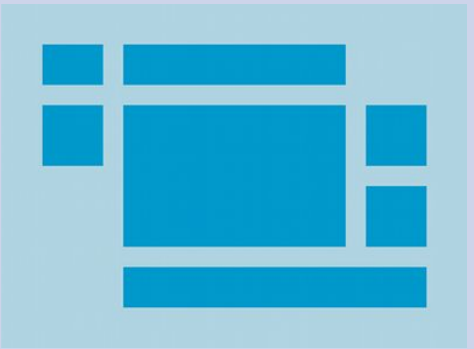
| S.N. | Layout & Description | Preview |
|---|---|---|
| 1. | TextView<br>Display Text to user |  |
| 2. | Button<br>Element for user to tap or click to perform an action. |  |
| 3. | Switch<br>Two-state toggle switch widget that can select between two option |  |
| 4. | RadioButton<br>Allow the user to select one option from a set. |  |
| 5. | ImageView<br>Display Image Resources |  |
| 6. | ProgressBar<br>Indicates the progress of an operation. |  |
| 7. | SeekBar<br>Extension of ProgressBar that adds draggable thumb |  |

# Advanced Controls

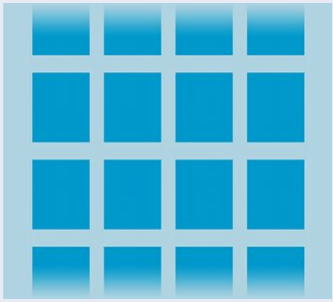
| | | |
|---|---|---|
| 8. | MapView<br>Displays a map (with data obtained from the Google Maps service) |  |
| 9. | WebView<br>Display web pages as a part of your activity layout. |  |
| 10 | VideoView<br>Display a video file |  |
| 11 | CalendarView<br>Calendar widget for displaying and selecting dates |  |
| 12 | RatingBar<br>Extension of SeekBar and ProgressBar that shows a rating in stars |  |
| 13 | SearchView<br>Allow user to enter a search query and submit a request to a search provider. |  |

# Layouts

# Layout Types

| S.N. | Layout & Description | View |
|---|---|---|
| 1. | Linear Layout<br>Aligns all children in a single direction, vertically or horizontally. Creates scrollbar if length of the window exceeds the length of the screen. |  |
| 2. | Relative Layout<br>Enable you to specify the location of child objects relative to each other or to the parent |  |
| 3. | Table Layout<br>Position children into rows and columns |  |
| 4. | Table Row<br>Arrange children horizontally. **Used as a child of TableLayout**, otherwise behave as a horizontal linear layout | |

# Cont...

| S.N. | Layout & Description | View |
|---|---|---|
| 5. | List View<br>Display a column list of scrollable items |  |
| 6. | GridView<br>Display a two-dimensional grid of scrollable items |  |

# Relative Layout

## Steps

1. If default layout is not relative, right click the layout and click *convert view > relative layout*
2. Introduce an element
3. Clear margin, padding and any LayoutParam (e.g. layout_alignParentStart)
4. Choose your LayoutParam (you can check more than 1)
5. Apply any margin or padding

# LayoutParam

THIS IS YOUR PARENT!

Default positioning

Layout_centerInParent
Layout_alignParentTop

Layout_centerInParent
Layout_toLeftOf: A

Layout_centerInParent
Layout_above: A

Layout_centerInParent

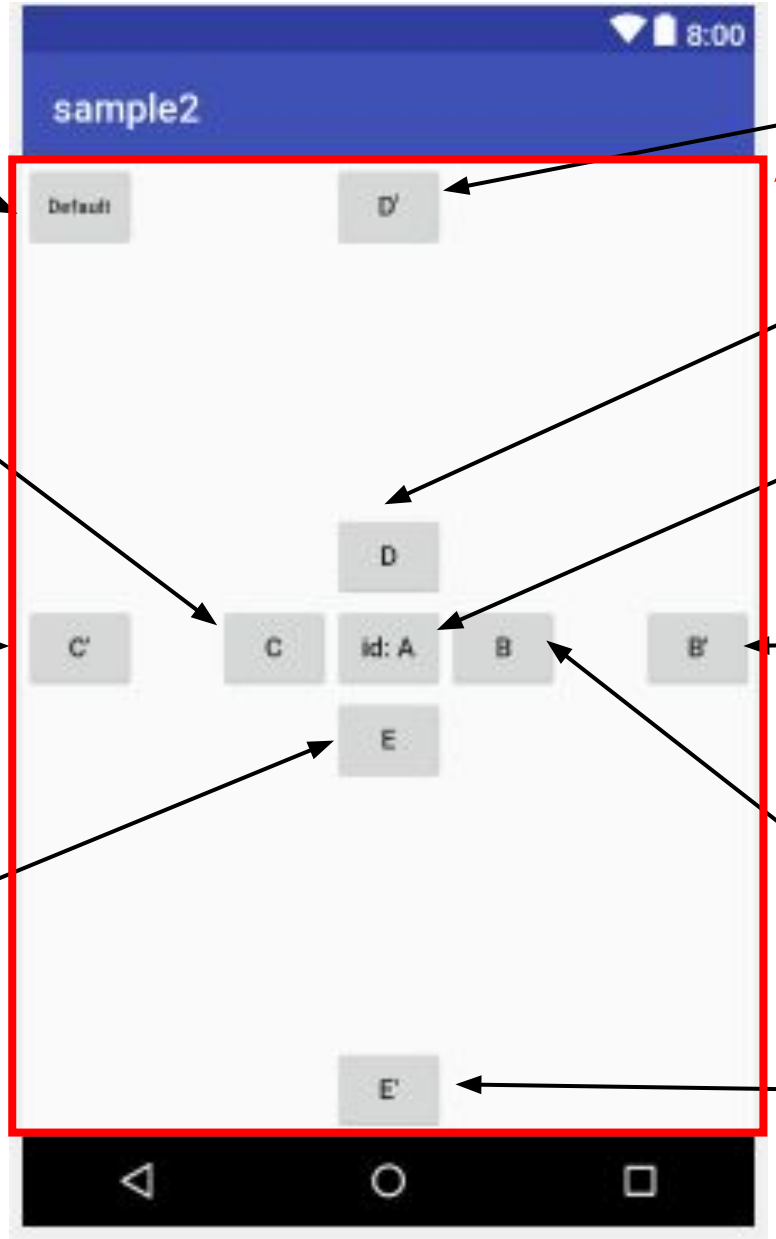
Layout_centerInParent
Layout_alignParentRLeft

Layout_centerInParent
Layout_alignParentRight

Layout_centerInParent
Layout_Below: A

Layout_centerInParent
Layout_toRightOf: A

Layout_centerInParent
Layout_alignParentBottom



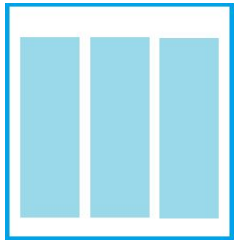
**What if I want to position the element independent of each other?**

# Use Linear Layout!

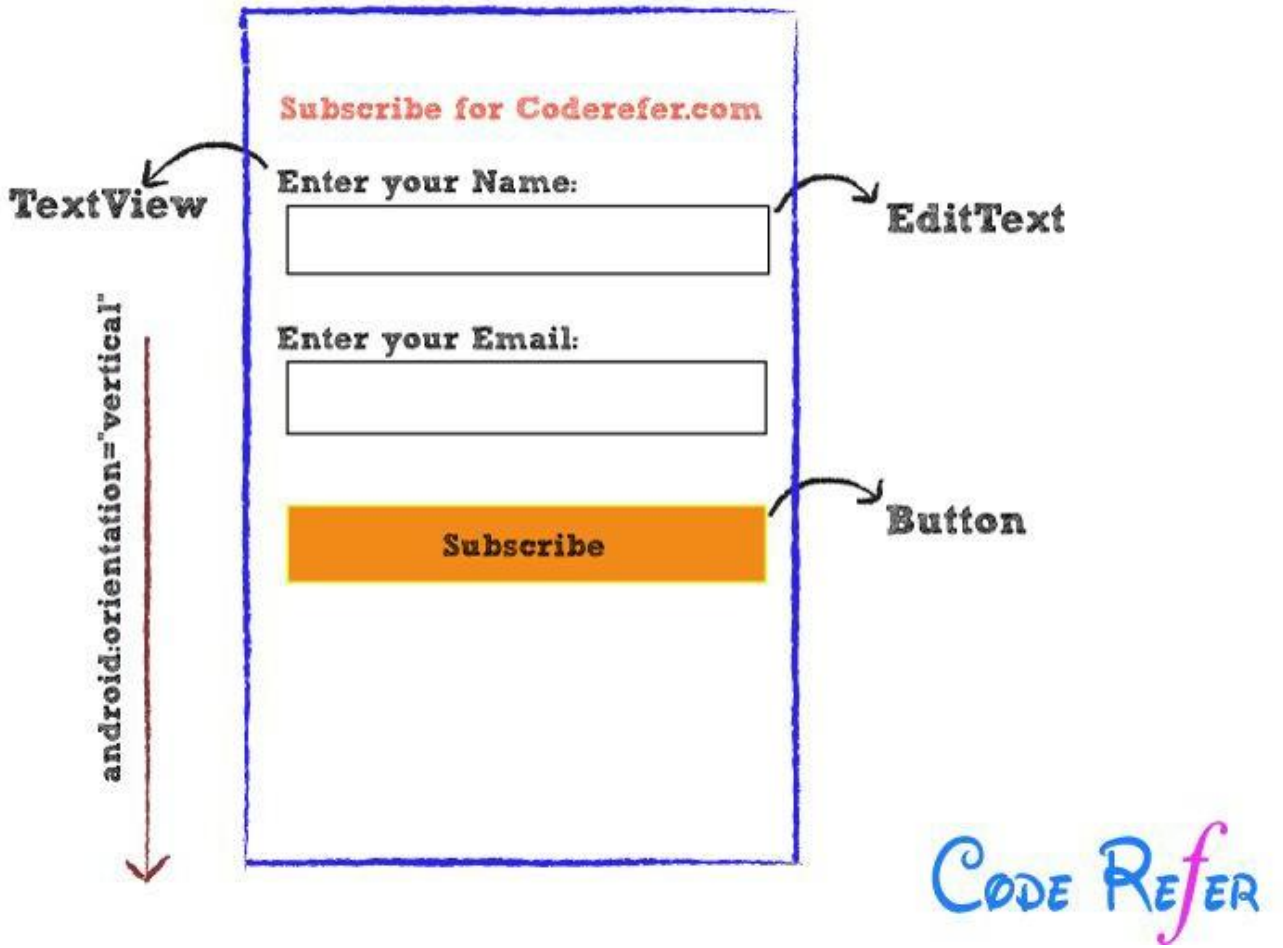
# Linear Layout

## Steps

1. If default layout is not linear, right click the layout and click *convert view > linear layout.* Default is horizontal. To switch to vertical, right click again, click *linear layout > convert orientation to vertical*
2. Introduce an element
3. Clear margin and padding
4. Repeat (2) and (3) until u have all element row-wise or col-wise
5. Apply any margin, padding, weight or spacing



# Example




Code Refer

# Other Layouts

- Table Layout

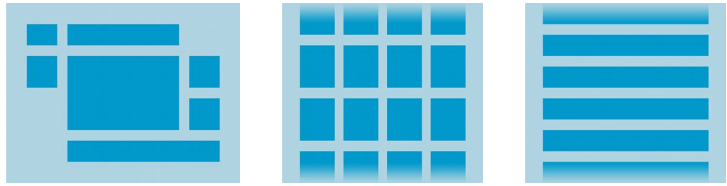
https://www.youtube.com/watch?v=9jzI8JDHFtA&t=345s

- GridView

https://www.youtube.com/watch?v=VUPM387qyrw

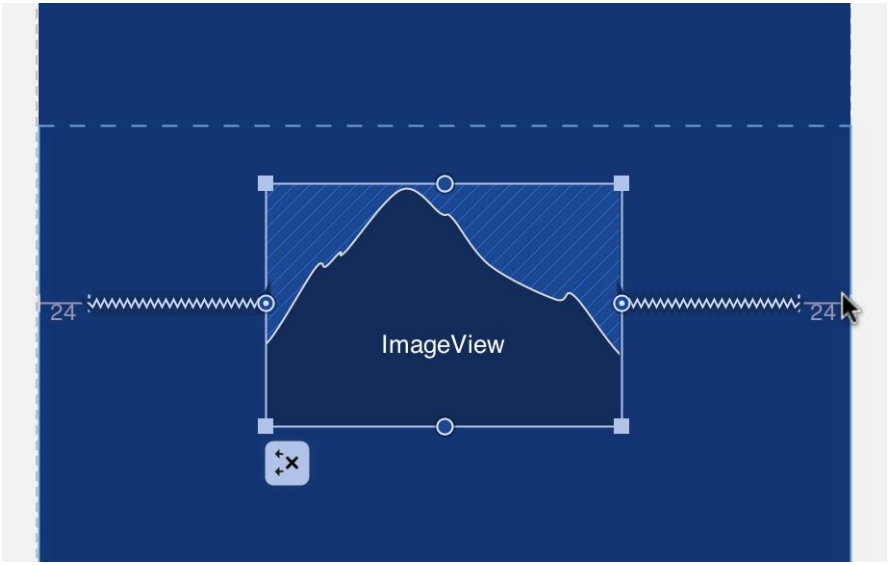
- ListView

https://www.youtube.com/watch?v=FKUlw7mFXRM



# In Android Studio 2.2, Google introduce a new layout….

*Constraint Layout!*

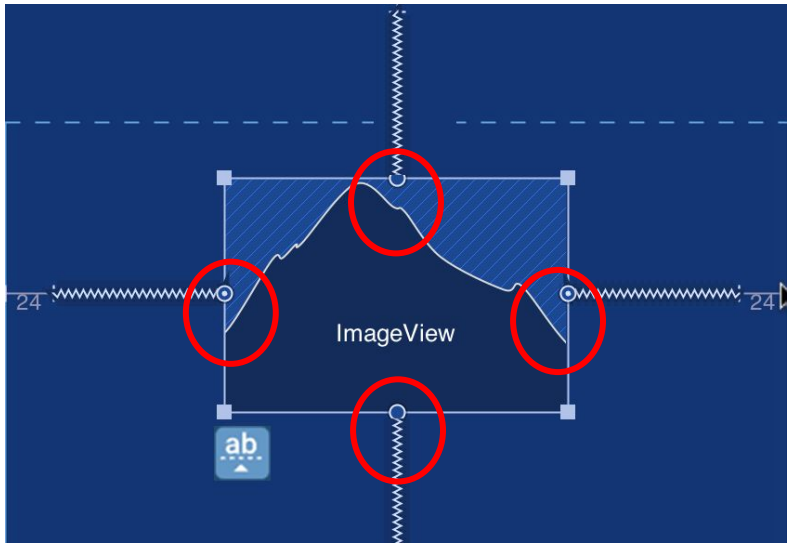



- Flat View hierarchy
- Form groups via “chain method”
- Does not break layout if an element is set to gone
- Can do both relative and linear

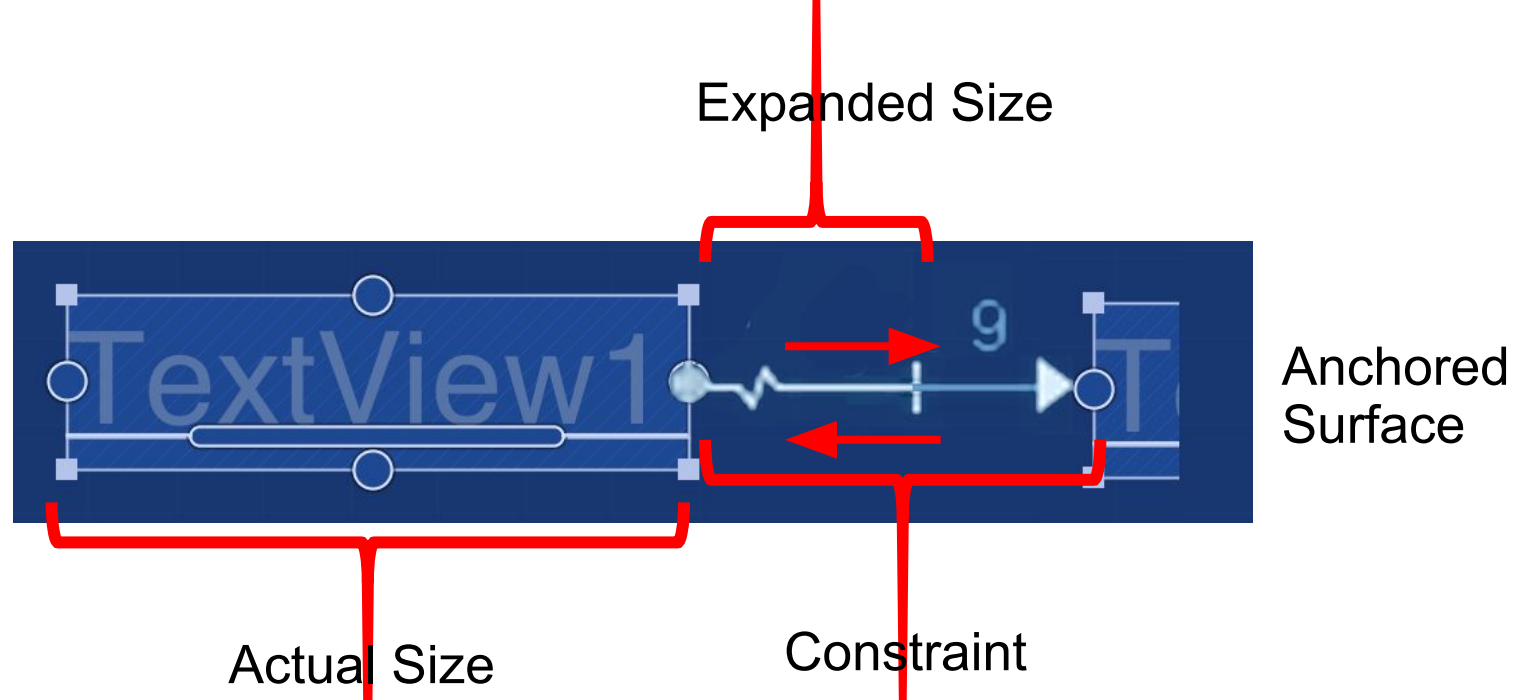
# What is Constraint Layout?

Similar to relative layout, but uses constraints to determine how UI element are position relative to others.  A constraint can be thought of as a connection or alignment to another element / parent via

**Anchor Points**

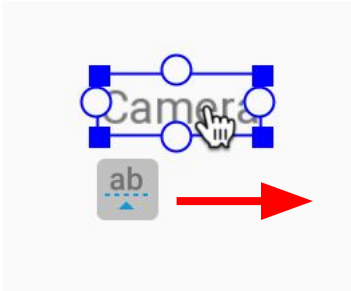


# Features



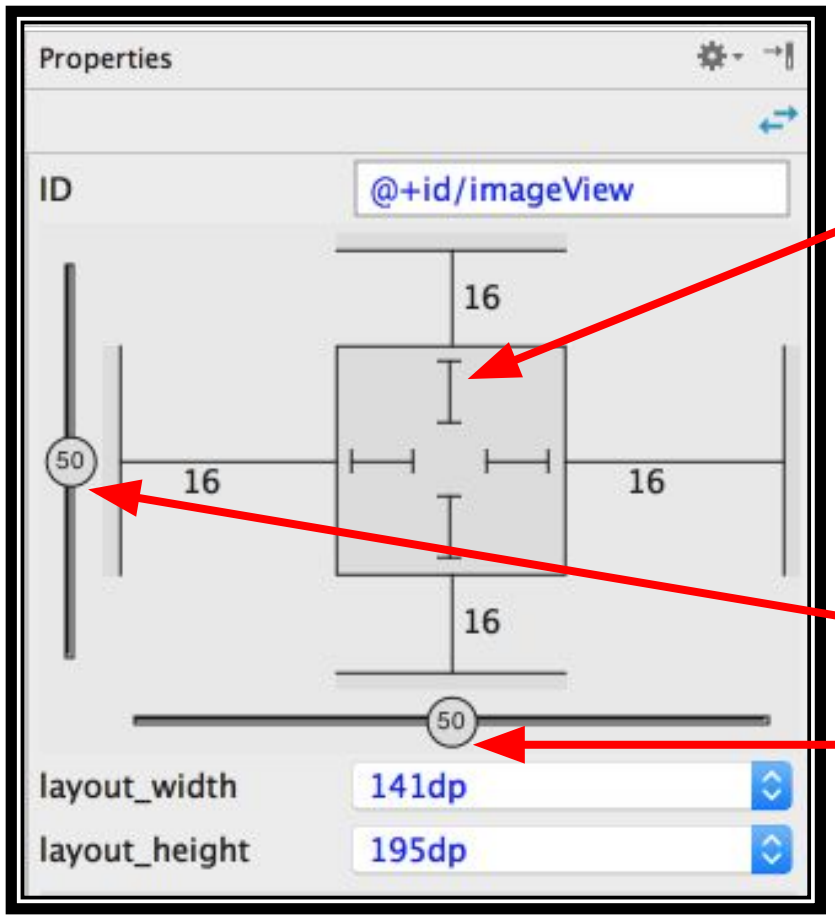

Resize handle: used for resizing view

Side handle / Anchor: used to specify the position of the element by connecting to other element. By default, anchored to constraint layout itself




Baseline: used for aligning  the text of a element with the baseline of another element

# Using the Inspector plane



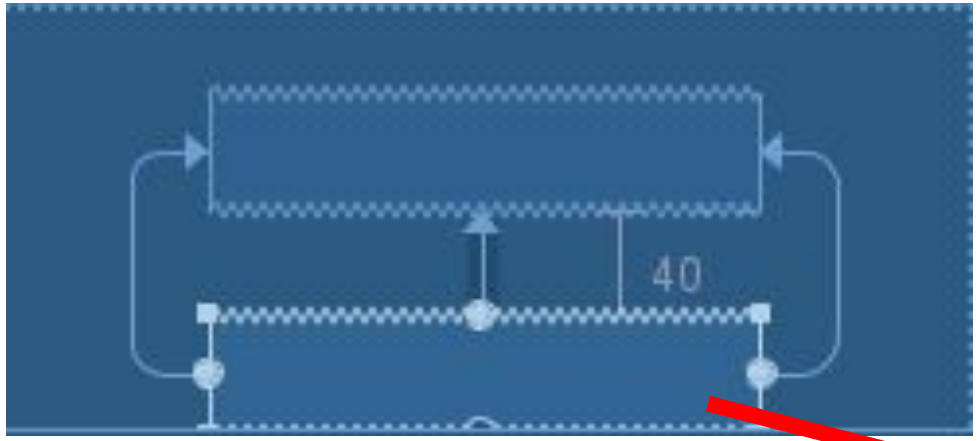

**Fixed:** Actual Element. Specify Constraint

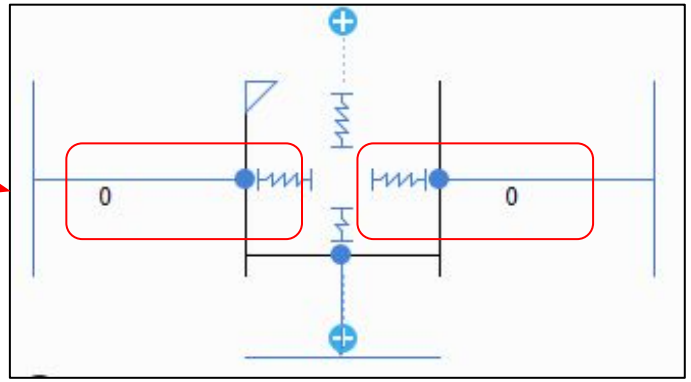
**Wrap**: Actual Element. Specify distance between the actual element and the anchored surface.

**Any size**: Expands Element. Specify distance between the expanded element and the anchored surface.

**Vertical / Horizontal Biasness:**
+/- constraint

# Vertical Alignment




# Horizontal Alignment (Baselining)

# Importing Images

# Steps

1) Find a non-copyrighted / open-source image
   - Wallpaper: https://unsplash.com/
   - Icons (.svg, .psd, etc.): https://www.flaticon.com/

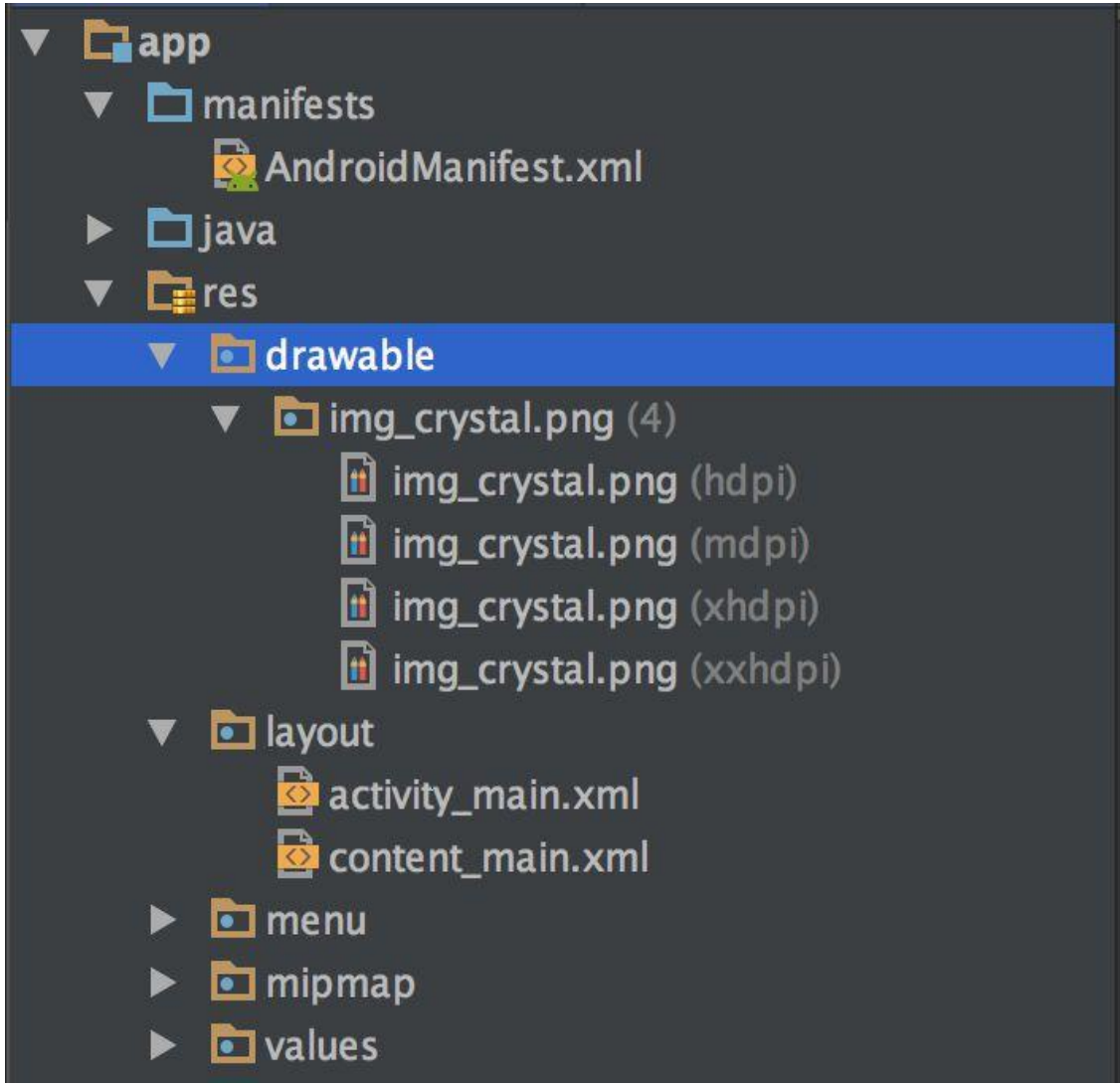
2) Download and Ctrl + C

3) Navigate to app/res/drawable in Android Studio and Ctrl + P

** Dragging and dropping the image into the respective folder will not work*

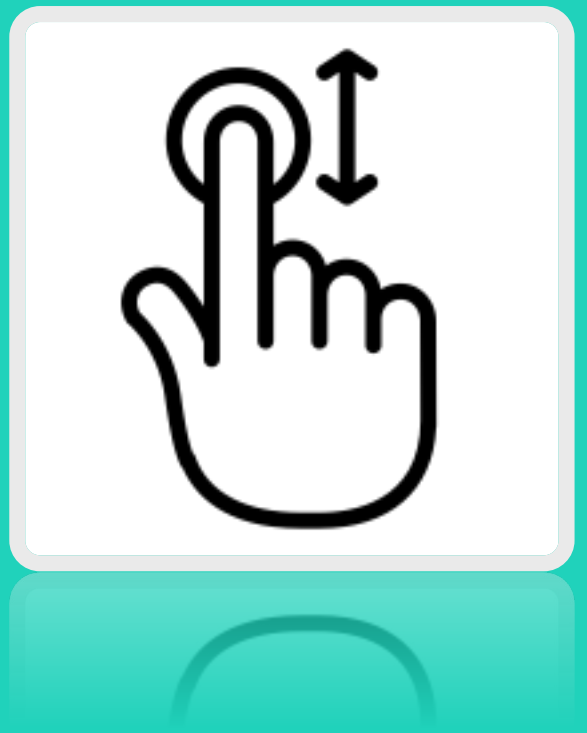
4) Rename the image and ensure that there is no special character or symbols

5) To display image in your app, use ImageView. To crop image, set its scaleType to *centreCrop*



*Screen too long? No problem!*

# Scroll View



# Steps

1. Add scroll view only after you have all your element in place
2. For text-based UI element (i.e. multiline text) switch layout_height to wrap_content
3. Switch to XMLView and insert the following snippets

**Vertical Scrolling**

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent" >
                    <!-- you content here -->
</ScrollView>
```

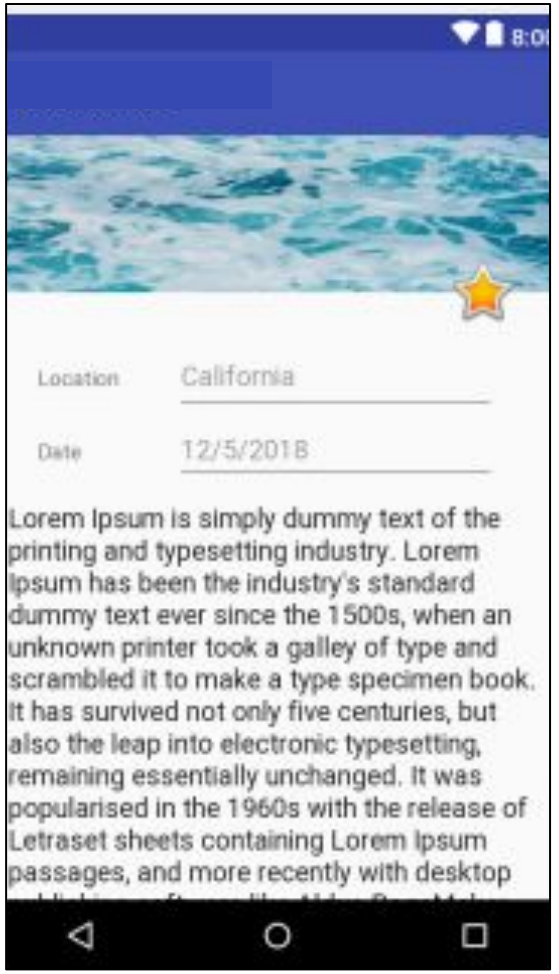
**Horizontal Scrolling**

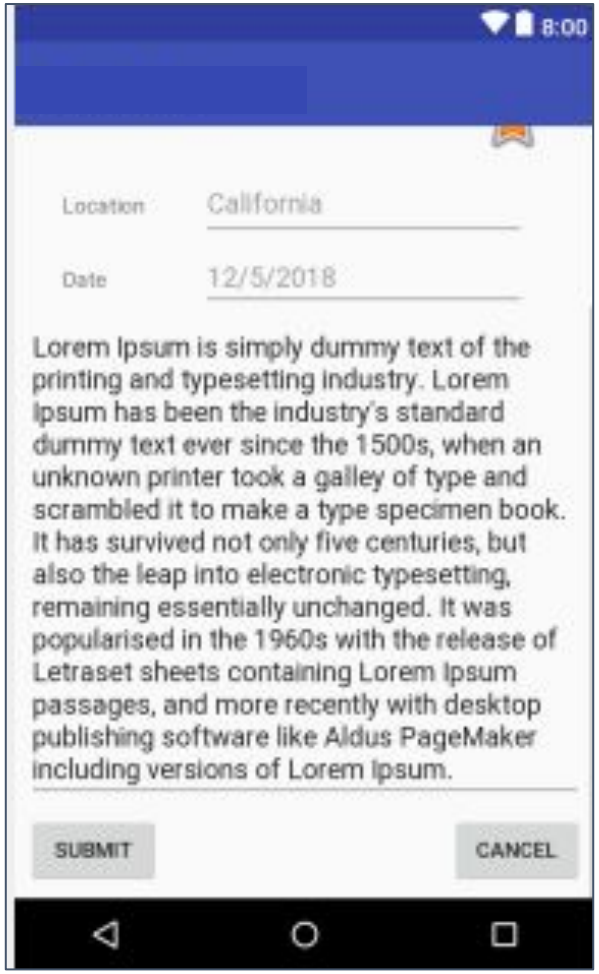
```
<HorizontalScrollView xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent" >
                    <!-- you content here -->
</ScrollView>
```

# Try it Yourself!

**You will need…**

- Constraint Layout
- Image view
- Scroll View
- Text View & Plain Text (field)
- Multiline Text (description)
- Buttons



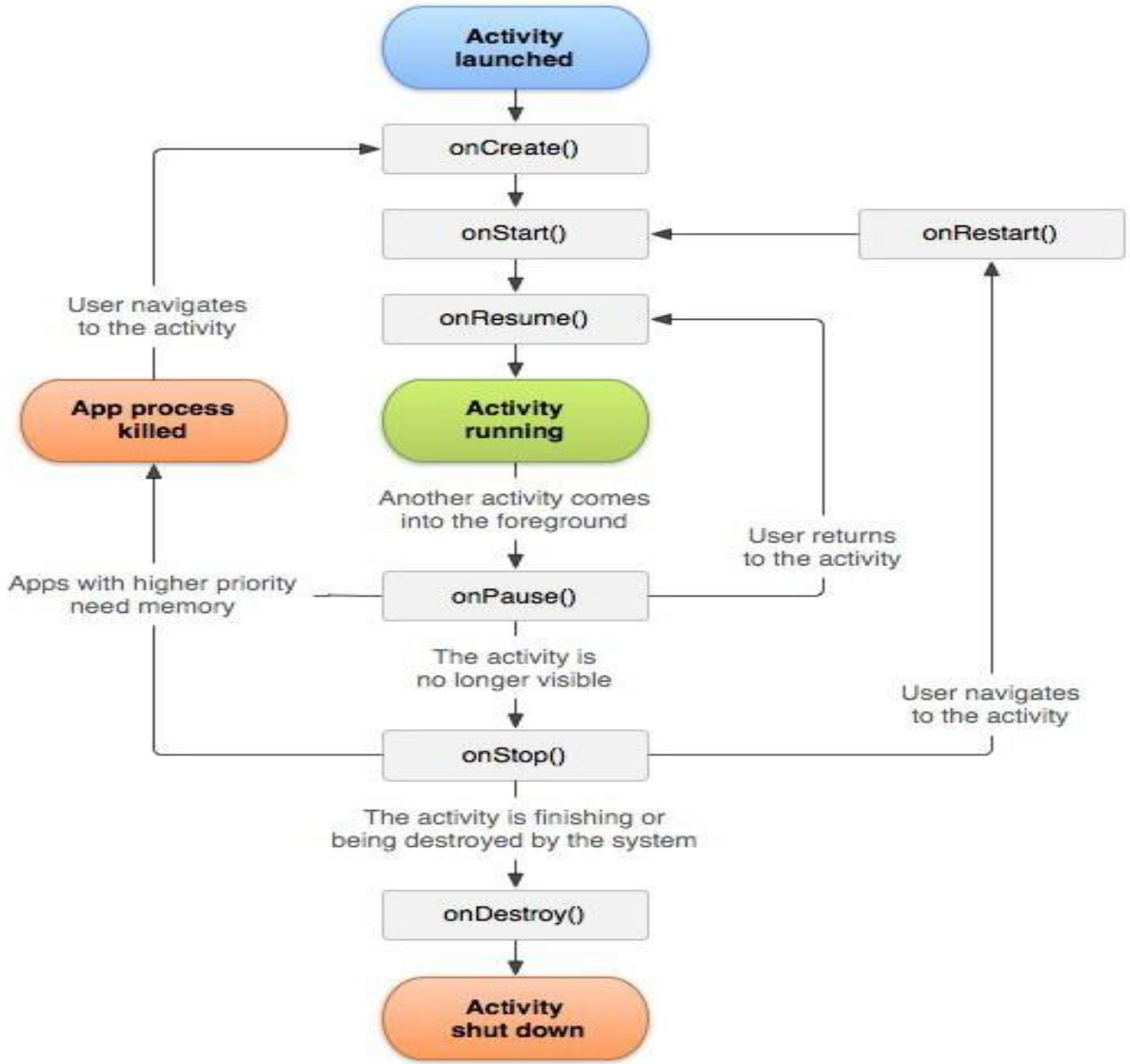




**Make sure it works in landscape mode too!**

# Scripting


a.fn.scrollspy=d,this},a(window).on(
function b(b){return this.each(function(){
TRANSITION_DURATION=150,c.prot
"use strict";function b(b){return this.each(function(){
var c=function(b){this.element=a(b)};c.VERSION="3.3.7",c.TRANSITION_DURATION=150,c.prot
e[b]()})}var c=function(b){this.element=a(b)};c.VERSION="3.3.7",c.TRANSITION_DURATION=150,c.pro
opdown-menu)"),d=b.data("target");if(d||(d=b.attr("href"),d=d&&d.replace(/.*(?=#[^\s]*$)/,"")),
st a"),f=a.Event("hide.bs.tab",{relatedTarget:b[0]}),g=a.Event("show.bs.tab",{relatedTarget:e[0]
faultPrevented()){var h=a(d);this.activate(b.closest("li"),c),this.activate(h,h.parent(),functio
rigger({type:"shown.bs.tab",relatedTarget:e[0]})})})}},c.prototype.activate=function(b,d,e){func
u > .active").removeClass("active").end().find('[data-toggle="tab"]').attr("aria-expanded",!1),
ia-expanded",!0),h?(b[0].offsetWidth,b.addClass("in")):b.removeClass("fade"),b.parent(".dropdow
().find('[data-toggle="tab"]').attr("aria-expanded",!0),e&&e()}var g=d.find("> .active"),h=e&&
e")||!!d.find("> .fade").length);g.length&&h?g.one("bsTransitionEnd",f).emulateTransitionEnd
;var d=a.fn.tab;a.fn.tab=b,a.fn.tab.Constructor=c,a.fn.tab.noConflict=function(){return a.fn.t
"show")};a(document).on("click.bs.tab.data-api",'[data-toggle="tab"]',e).on("click.bs.tab.data
se strict";function b(b){return this.each(function(){var d=a(this),e=d.data("bs.affix"),f="ob
typeof b&&e[b]()})}var c=function(b,d){this.options=a.extend({},c.DEFAULTS,d),this.$target=a
",a.proxy(this.checkPosition,this)).on("click.bs.affix.data-api",a.proxy(this.checkPositionWi
null,this.pinnedOffset=null,this.checkPosition()};c.VERSION="3.3.7",c.RESET="affix affix-top
State=function(a,b,c,d){var e=this.$target.scrollTop(),f=this.$element.offset(),g=this.$targ
"bottom"==this.affixed)return null!=c?!(e+this.unpin<=f.top)&&"bottom":!(e+g<=a-d)&&"bottom"
!=c&&e<=c?"top":null!=d&&i+j>=a-d&&"bottom"},c.prototype.getPinnedOffset=function(){if(this
.RESET).addClass("affix");var a=this.$target.scrollTop(),b=this.$element.offset();return
WithEventLoop=function(){setTimeout(a.proxy(this.checkPosition,this),1)}
nt.height(),d=this.options.offset,e=d.top,f=d.bottom
peof e&&(e=d.top(this.$element))
nt.css("top","")

# Android Activity



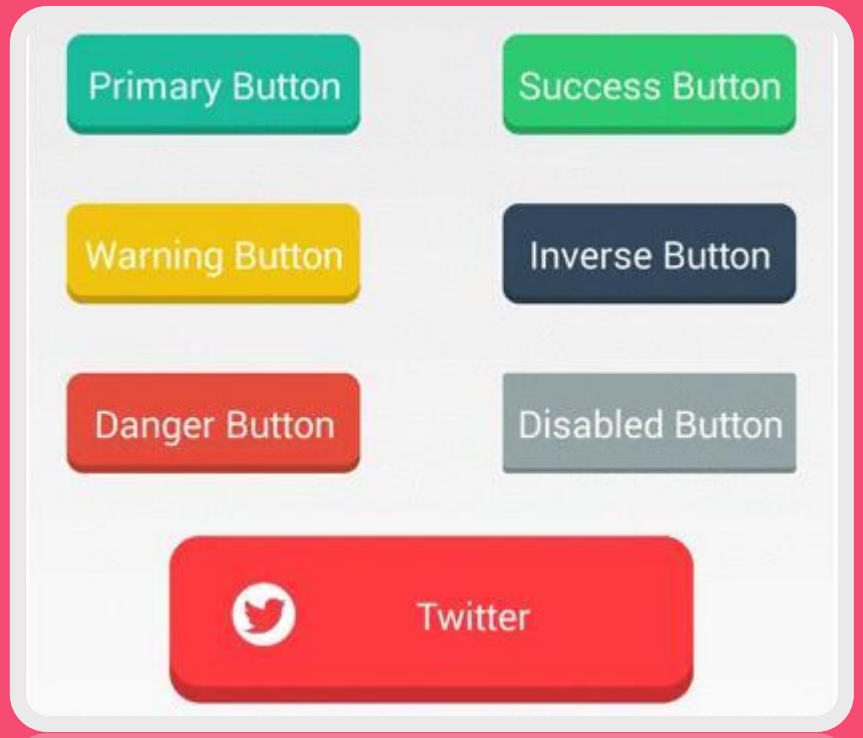
# Event Handling
(in controller)

| Event Handler | Event Listener & Description |
|---|---|
| onClick() | **OnClickListener()**<br>This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. You will use onClick() event handler to handle such event. |
| onLongClick() | **OnLongClickListener()**<br>This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. for one or more seconds. You will use onLongClick() event |
| onKey() | **OnFocusChangeListener()**<br>This is called when the user is focused on the item and presses or releases a hardware key on the device. You will use onKey() event handler to handle such event. |
| onTouch() | **OnTouchListener()**<br>This is called when the user presses the key, releases the key, or any movement gesture on the screen. You will use onTouch() event handler to handle such event. |
| onMenuItemClick() | **OnMenuItemClickListener()**<br>This is called when the user selects a menu item. You will use onMenuItemClick() event handler to handle such event. |

# Navigate to app/java/com/MainActivity

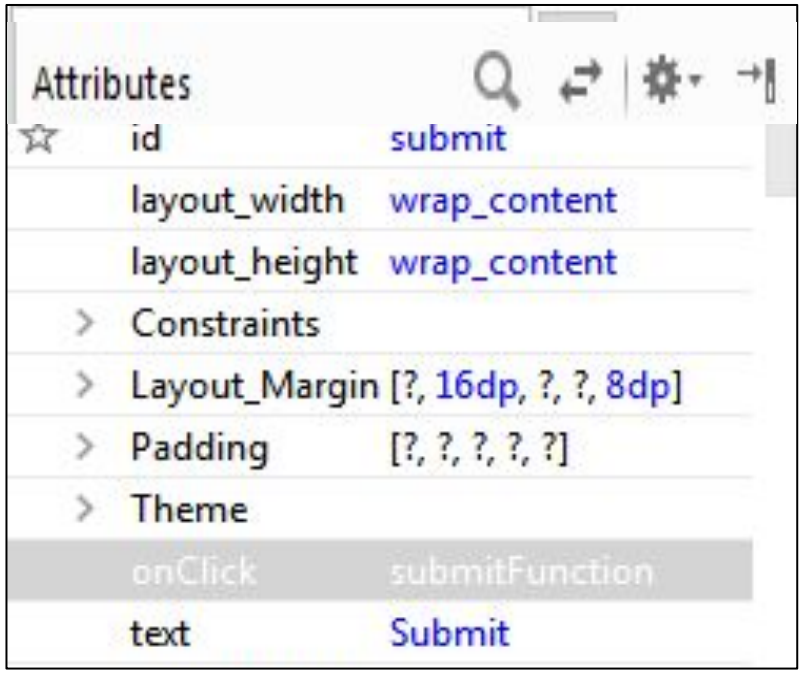
# Wiring your Buttons

Import android.view.View



# Steps

1. *import android.view.View;*
2. Create your function, argument is type View
   *e.g. public void clickFunction(View view) {}*
3. In your design view, navigate to your button, under attributes, look for “onClick” and input the function name

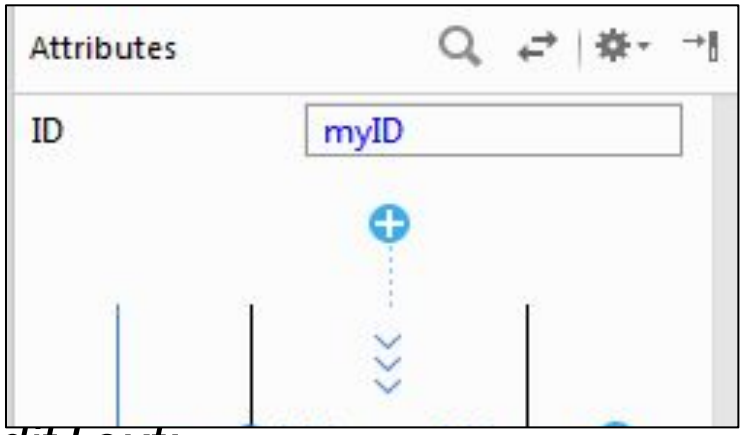


# Wiring your TextView

Import android.widget.EditText

# Steps

1. In your designer view, navigate to your EditText UI, under attributes, set an ID




*2. import android.widget.EditText;*

3. Locate view using findViewByID, parameter type ID

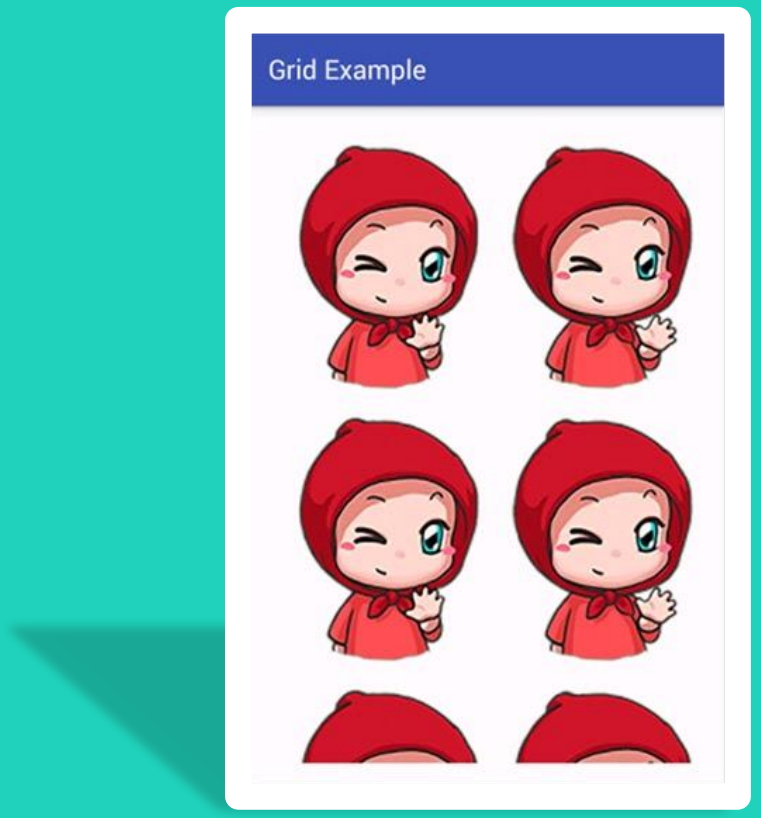
*e.g. EditText myTextfield = findViewById(R.id.myID);*

* To get current text: *myTextfield.getText().toString();*

* To set text: *myTextfield.setText("This is a sample text");*

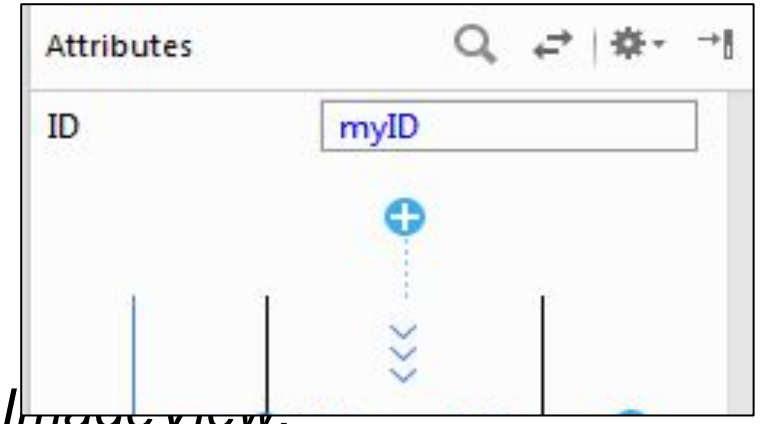
# Wiring your Image

Import android.widget.ImageView;



# Steps

1. In your designer view, navigate to your imageView UI, under attributes, set an ID




2. *import android.widget.ImageView;*
3. Locate view using findViewByID, parameter type ID

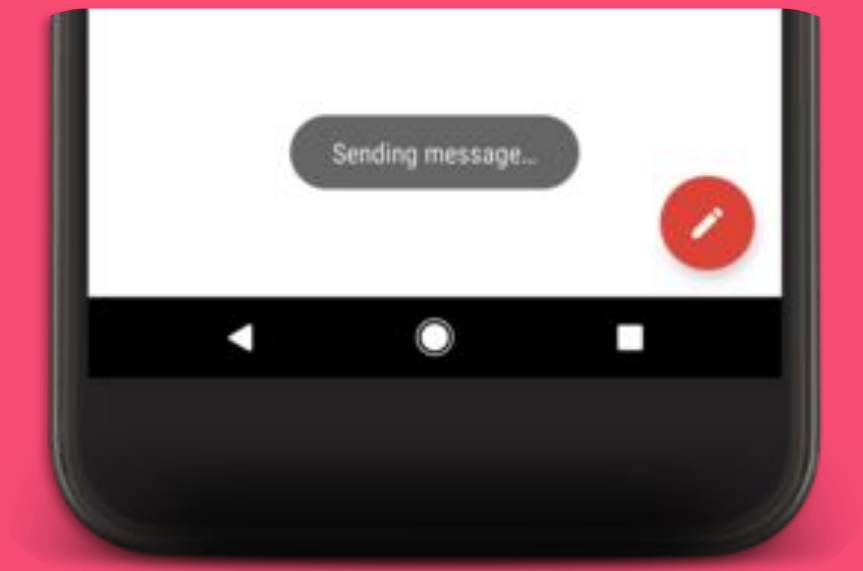
   *e.g. ImageView image= findViewById(R.id.myID);*

* To get current image: *image.getDrawable(); // returns Drawable*
* To set image: *myTextfield.setImageResource(R.drawable.myImage)*

  *// argument is location of your image*

# Toast Message?

Import android.widget.Toast;

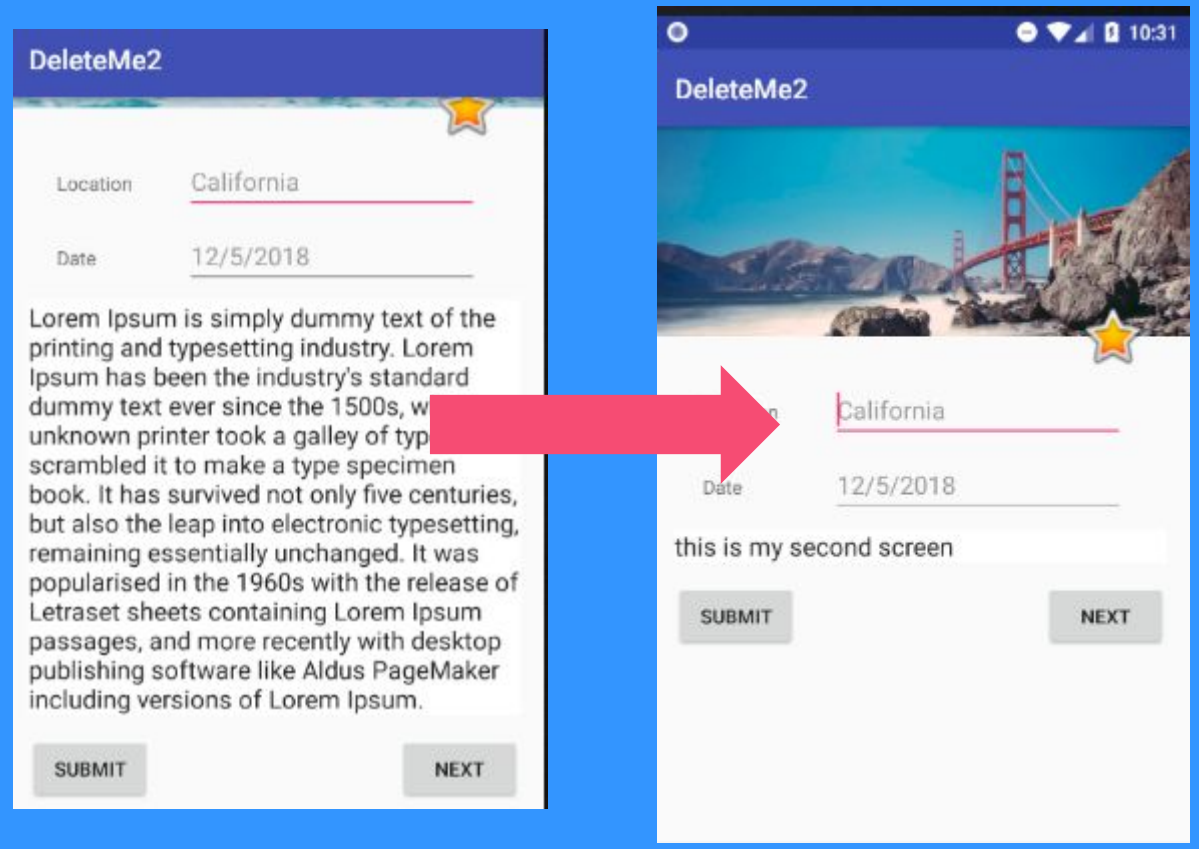


# Steps

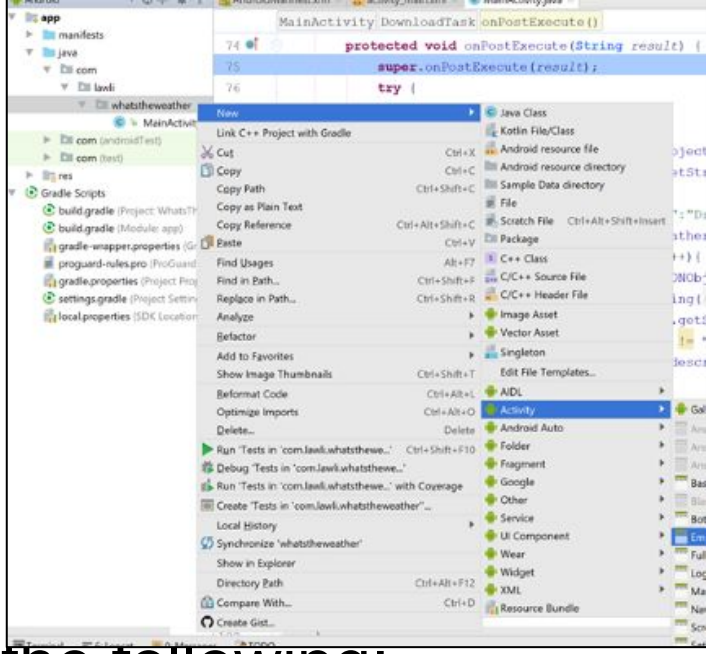
1. *Import android.widget.Toast;*
2. *Toast.makeText(this, “justAToastMsg”, Toast.LENGTH_SHORT), show(); // this refers to main activity*

# Switching Screen

## Import android.content.intent



# Steps



1. Create a second Activity (e.g. Activity2) in app / java / com
2. Go back to your main Activity and import:
   *import android.content.Intent;*
3. In your function responsible for switching screen, insert the following:

```
public void switchScreenFunction(view View) {
    Intent intent = new Intent(this, Activity2.class)
    startActivity(intent);
}
```

* Remember to wire the function to the button responsible for switching screen

# More useful functions…

**Logger**

Prints messages to your run terminal (Logcat)
*e.g. Log.i (“myTag”, “pressed!”); //* **import android.util.Log**

**Show/Hide SoftKeyboard onCreate()**

Hide:
*getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_HIDDEN);*

Show:
*getWindow().setSoftInputMode(WinowdManager.LayoutParams.SOFT_INPUT_VISIBLE):*
*//* **import android.view.WindowManager**

# HELP!

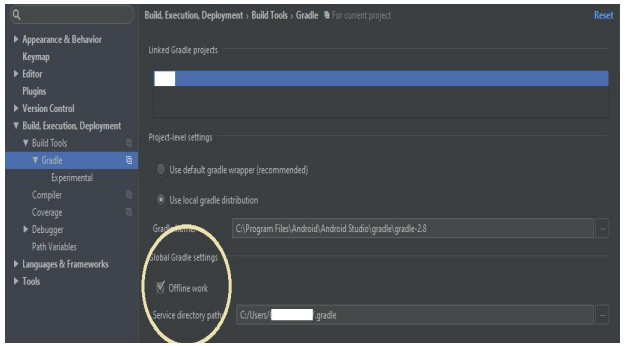
# My Gradle Build is too slow!

# Ways to optimize your build speed

**Method 1:** Offline Work

1) In Android Studio go to File -> Settings -> Build, Execution, Deployment -> Build Tools -> Gradle

2) Check the 'Offline work' under 'Global Gradle settings’




**Method 2**: Disable Saving State

1) Open your AVD Manager.


2) Select your AVD of choice, and gs.

3) Tap on the Show Advanced Settings button.

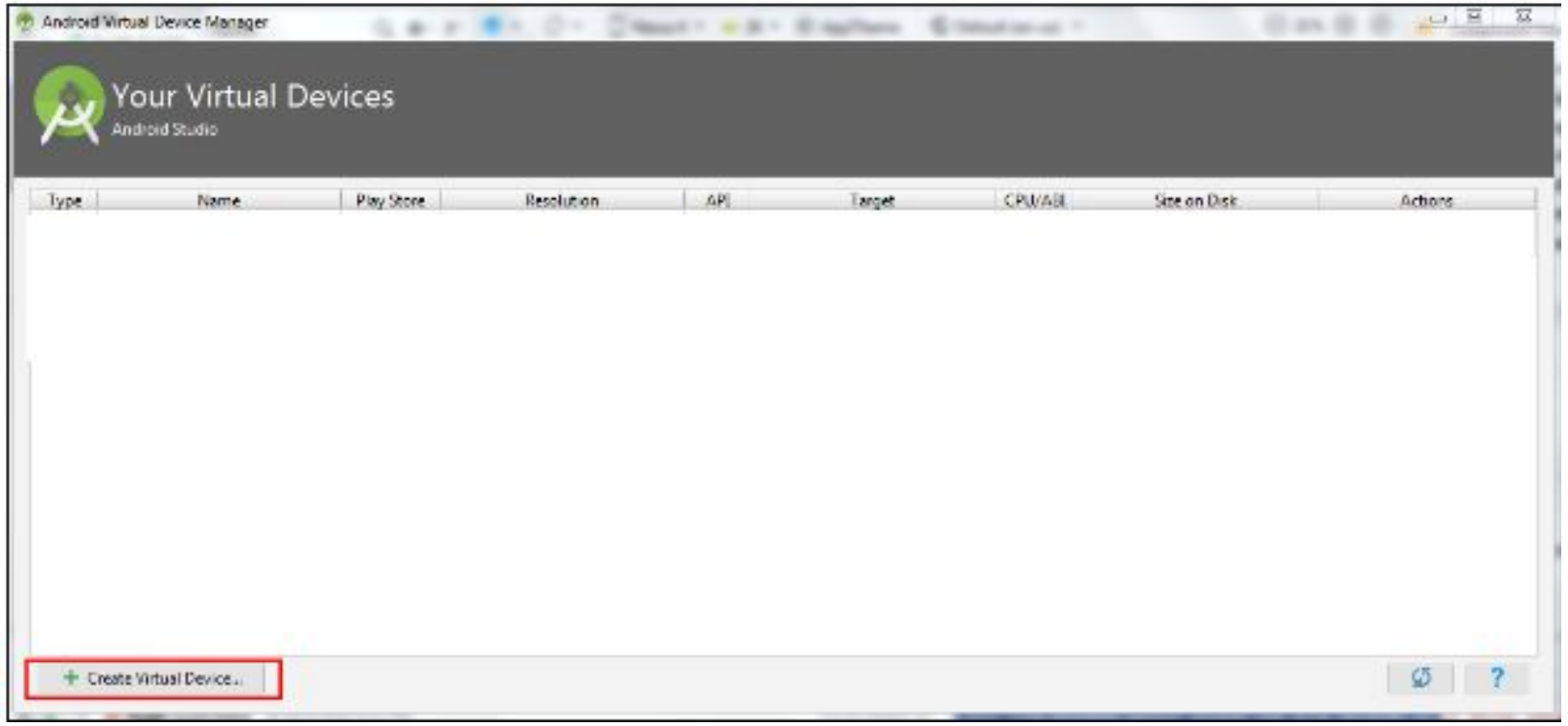
4) Under the "Emulated Performance" sub-category, "Boot option" menu, check "Cold boot".

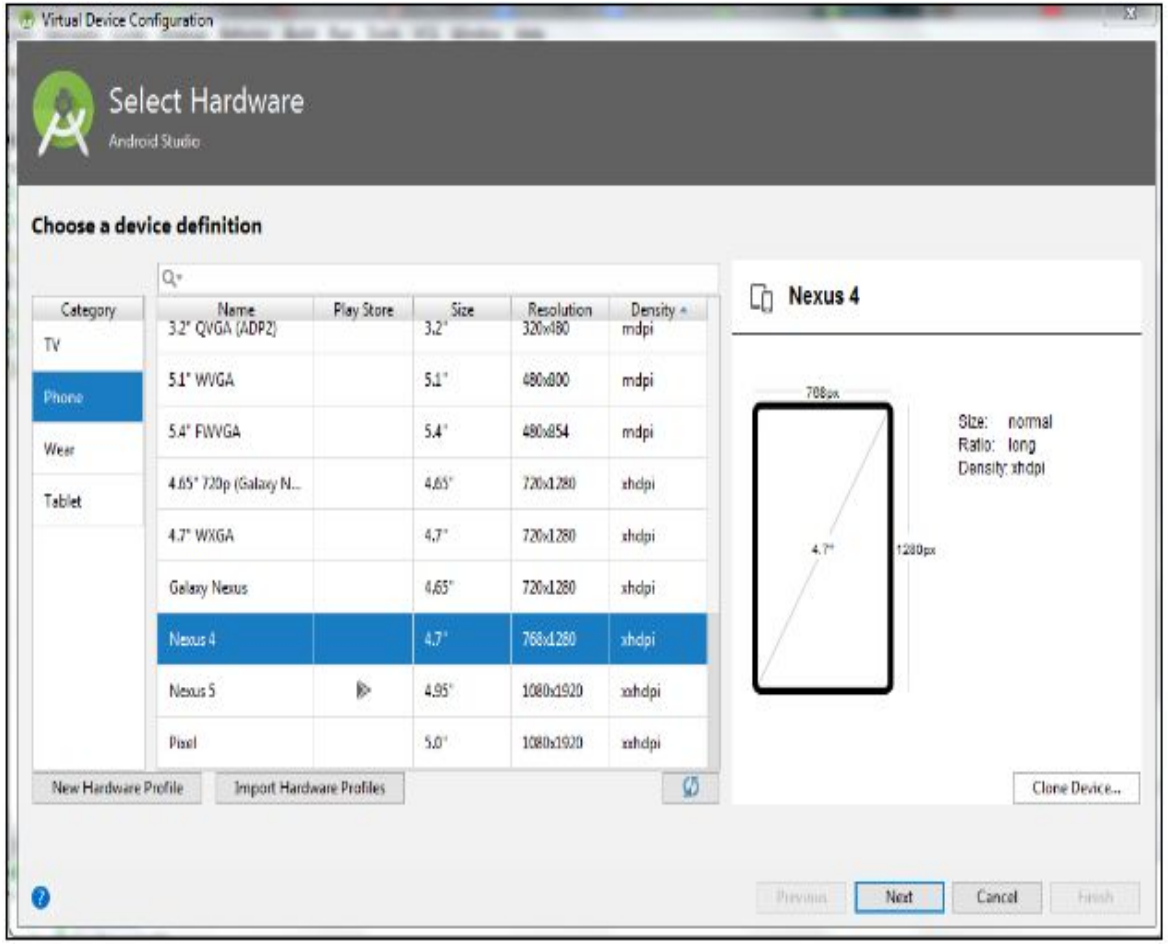
# Thank You

Email: leslie.nma@gmail.com
LinkedIn: https://www.linkedin.com/in/lesliehozonghong

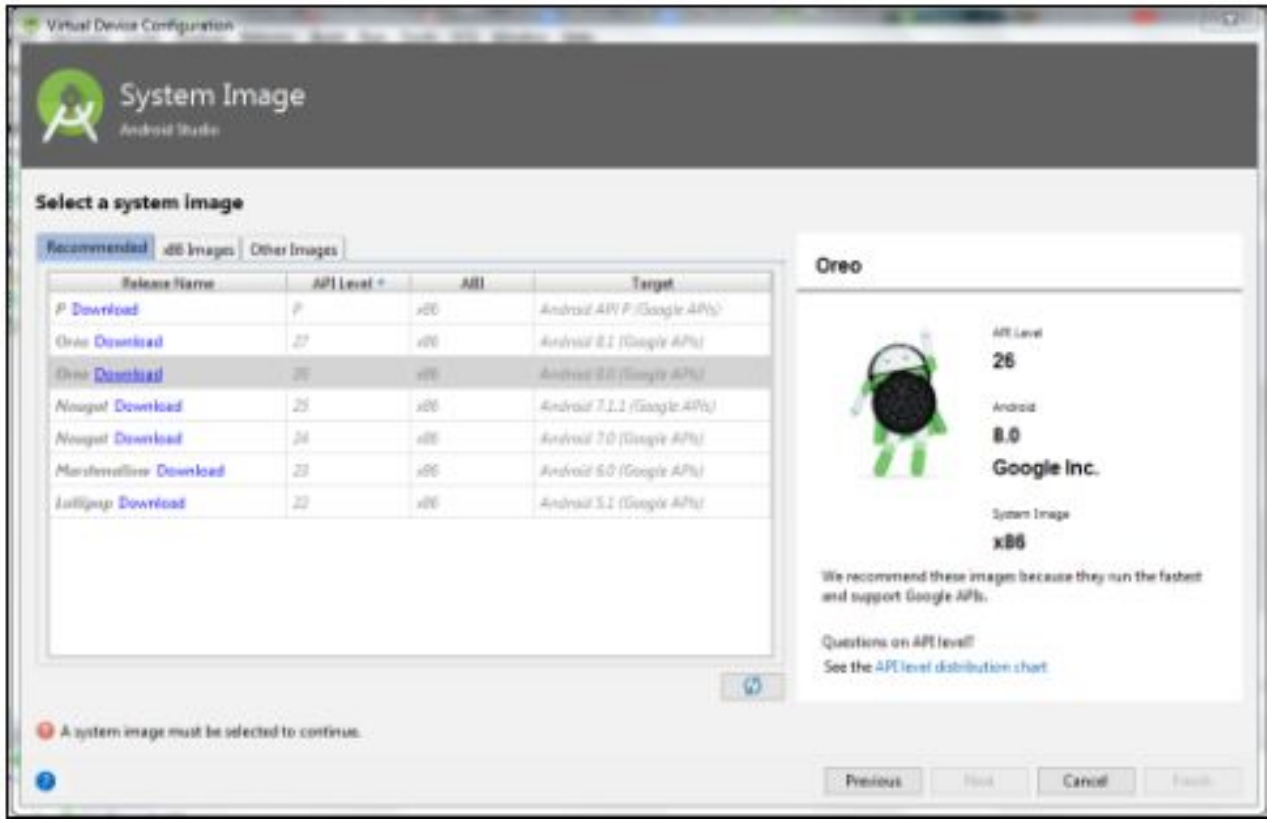
# Appendix: Running AVD
## (from installation guide)

1) In Android Studio menu, go to *Tools > AVD Manager*
2) Click “Create Virtual Device”






3) Select Nexus 4 and click “Next”

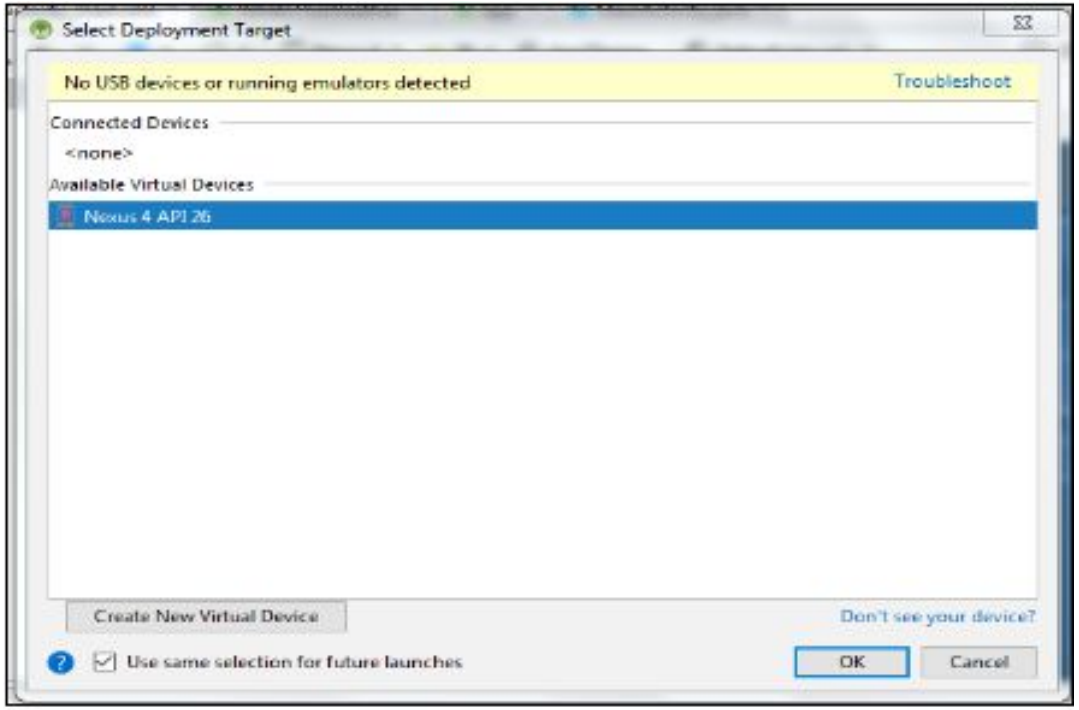



4) Select “Oreo” (API Level 27) for system image and click the Download button next to it

5) Click “Next” once done. Stick to the default configuration and click “Finish”

6) In the menu bar, click the run button indicated by a green arrow


7) Select Nexus 4 API 26 as your deployment target. Check “Use same selection for future launches.” Click “Ok” and launch the emulator.

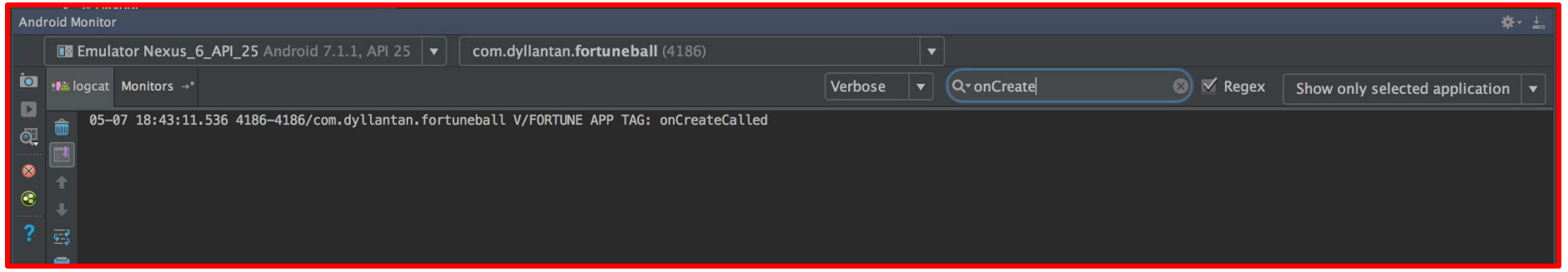


# Debugging

# Android Monitor - Logcat

```
import android.util.Log;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    //mFortuneText = (TextView) findViewById(R.id.fortuneText);
    mFortuneBallImage = (ImageView) findViewById(R.id.fortunateImage);
    mGenerateFortuneButton = (Button) findViewById(R.id.fortuneButton);

    mGenerateFortuneButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            int index = new Random().nextInt(fortuneList.length);
            mFortuneText.setText(fortuneList[index]);

            YoYo.with(Techniques.Swing)
                    .duration(500)
                    .playOn(mFortuneBallImage);
        }
    });

    Log.v("FORTUNE APP TAG","onCreateCalled")
}
```




Source: https://www.raywenderlich.com/154676/android-studio-tutorial-introduction

# Android Monitor - Logcat

```
22  public class MainActivity extends AppCompatActivity {
23
24      String fortuneList [] = {"Without a doubt","Outlook not so good","It's decidedly so","Signs point to yes","Yes definitely","Yes","My
25
26      TextView mFortuneText;
27      Button mGenerateFortuneButton;
28      ImageView mFortuneBallImage;
29
30      @Override
31      protected void onCreate(Bundle savedInstanceState) {
32          super.onCreate(savedInstanceState);
33          setContentView(R.layout.activity_main);
34          Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
35          setSupportActionBar(toolbar);
36
37          //mFortuneText = (TextView) findViewById(R.id.fortuneText);
38          mFortuneBallImage = (ImageView) findViewById(R.id.fortuneImage);
39          mGenerateFortuneButton = (Button) findViewById(R.id.fortuneButton);
40
41          mGenerateFortuneButton.setOnClickListener(new View.OnClickListener() {
42              @Override
43              public void onClick(View view) {
44
45                  int index = new Random().nextInt(fortuneList.length);
46                  mFortuneText.setText(fortuneList[index]);
47
48                  YoYo.with(Techniques.Swing)
49                          .duration(500)
50                          .playOn(mFortuneBallImage);
51              }
52          });
53
54          Log.v("FORTUNE APP TAG","onCreateCalled");
55      }
```

```
--------- beginning of crash
FATAL EXCEPTION: main
Process: com.dyllantan.fortuneball, PID: 4186
java.lang.NullPointerException: Attempt to invoke virtual method 'void android.widget.TextView.setText(java.lang.CharSequence)' on a null object reference
    at com.dyllantan.fortuneball.MainActivity$1.onClick(MainActivity.java:46)
    at android.view.View.performClick(View.java:5637)
    at android.view.View$PerformClick.run(View.java:22429)
    at android.os.Handler.handleCallback(Handler.java:751)
    at android.os.Handler.dispatchMessage(Handler.java:95)
    at android.os.Looper.loop(Looper.java:154)
    at android.app.ActivityThread.main(ActivityThread.java:6119) <1 internal calls>
    at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:886)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:776)
```

Source: https://www.raywenderlich.com/154676/android-studio-tutorial-introduction