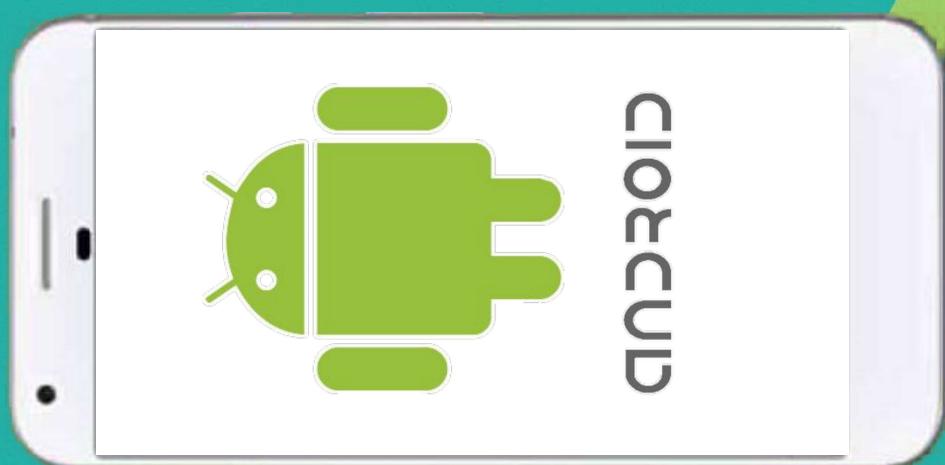


Basic Android Development

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

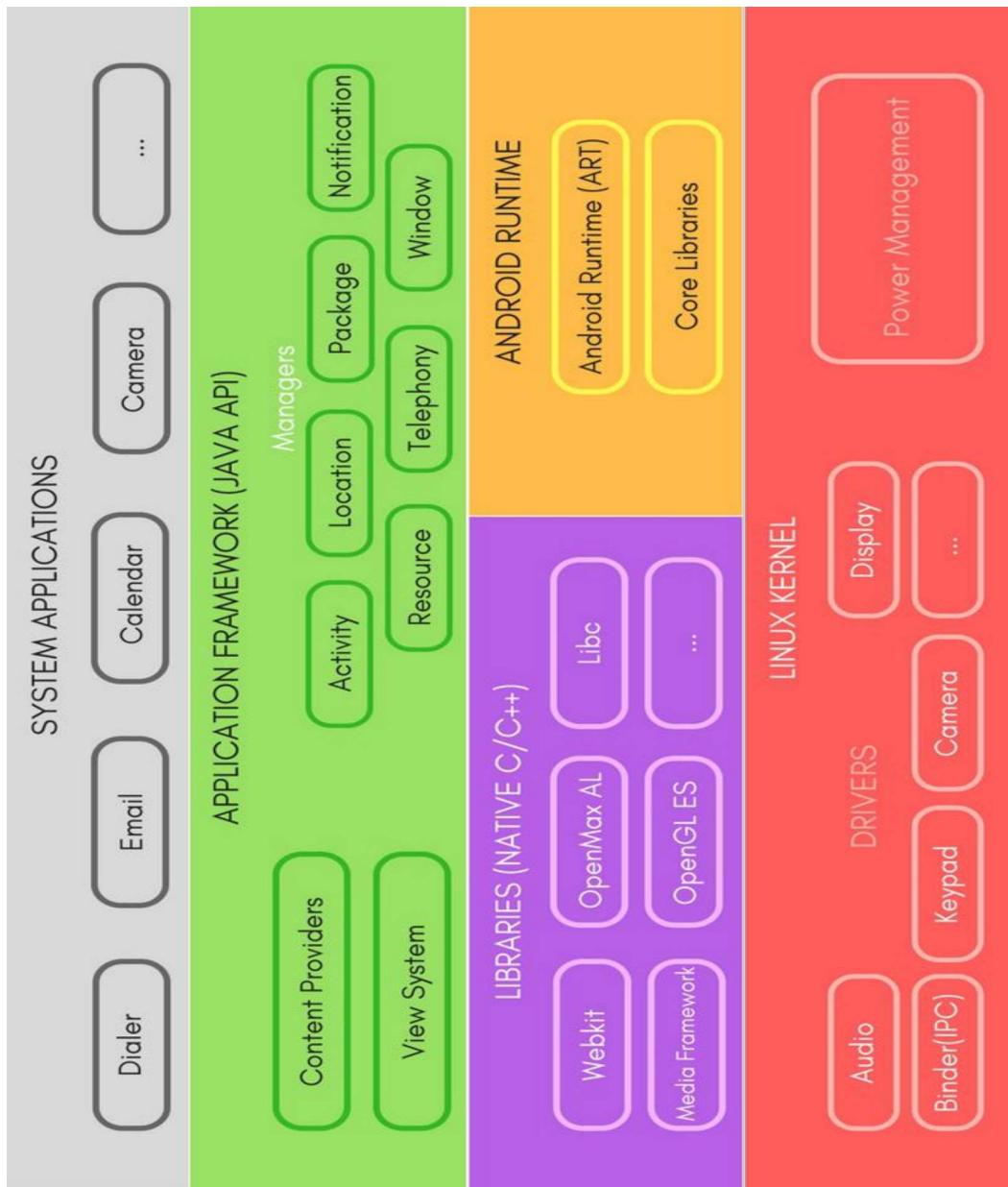
Instructions

<https://drive.google.com/open?id=1PzVQG-mFSuUrsV6G9RTEnuWE8mkJ73Iuw8aVhtYyL74>

Content.

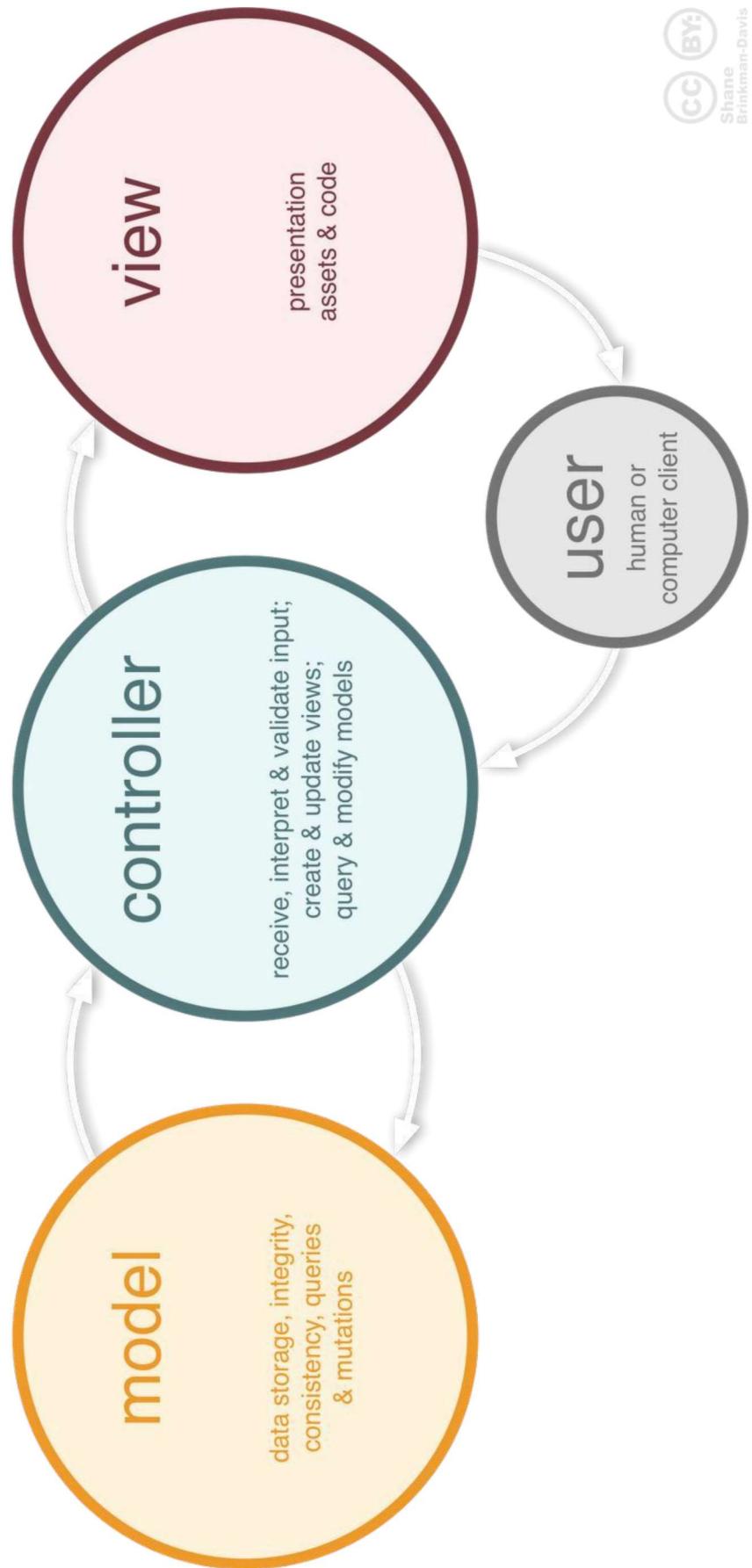
- Android Architecture
- Studio Interface
- UI Controls (element + layouts)
- Importing Images
- ScrollView
- Class Activity
- Scripting (wiring your UI)
- Optimization

Appendix: AVD set-up



System Architecture

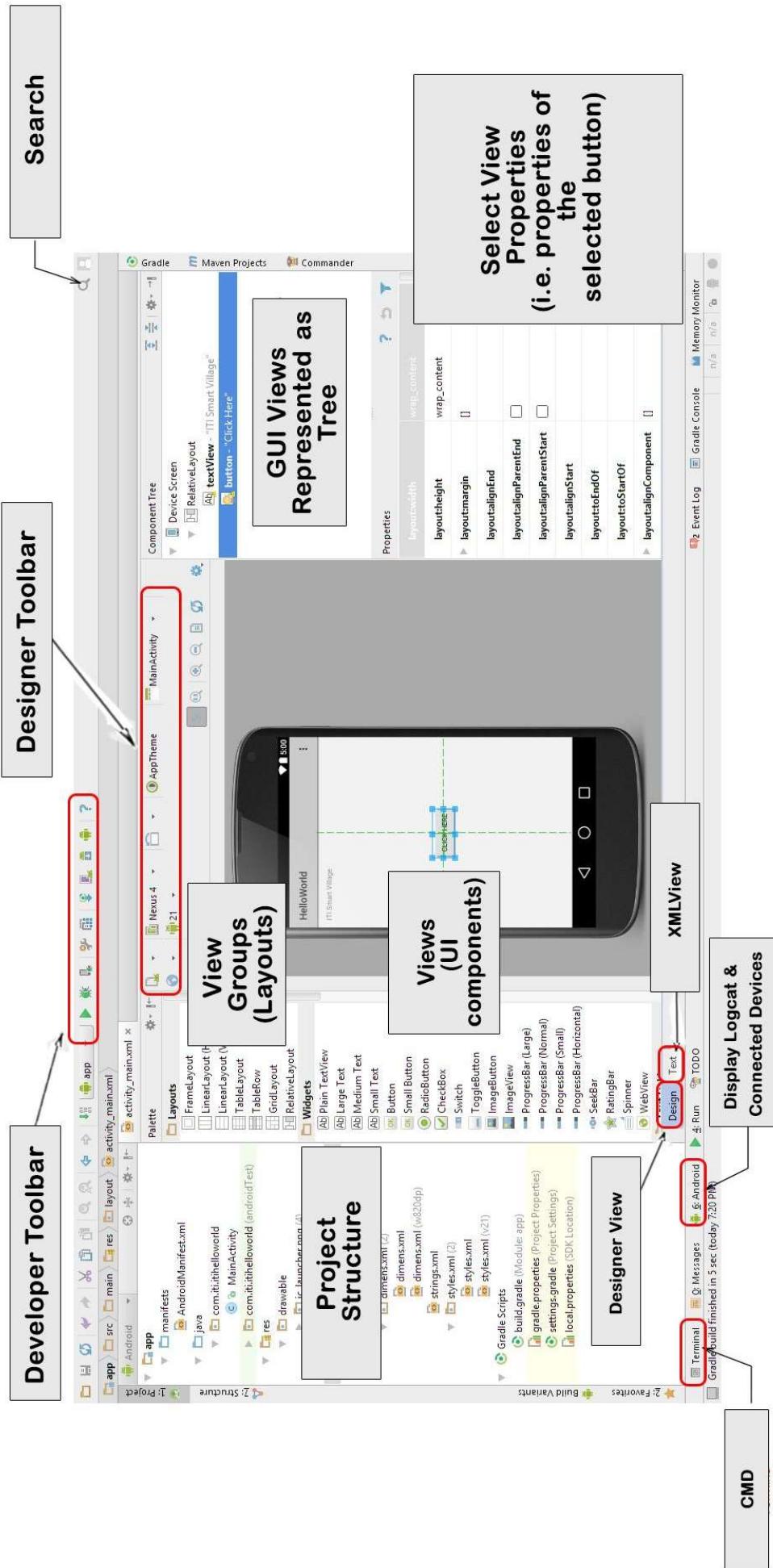
Android MVC



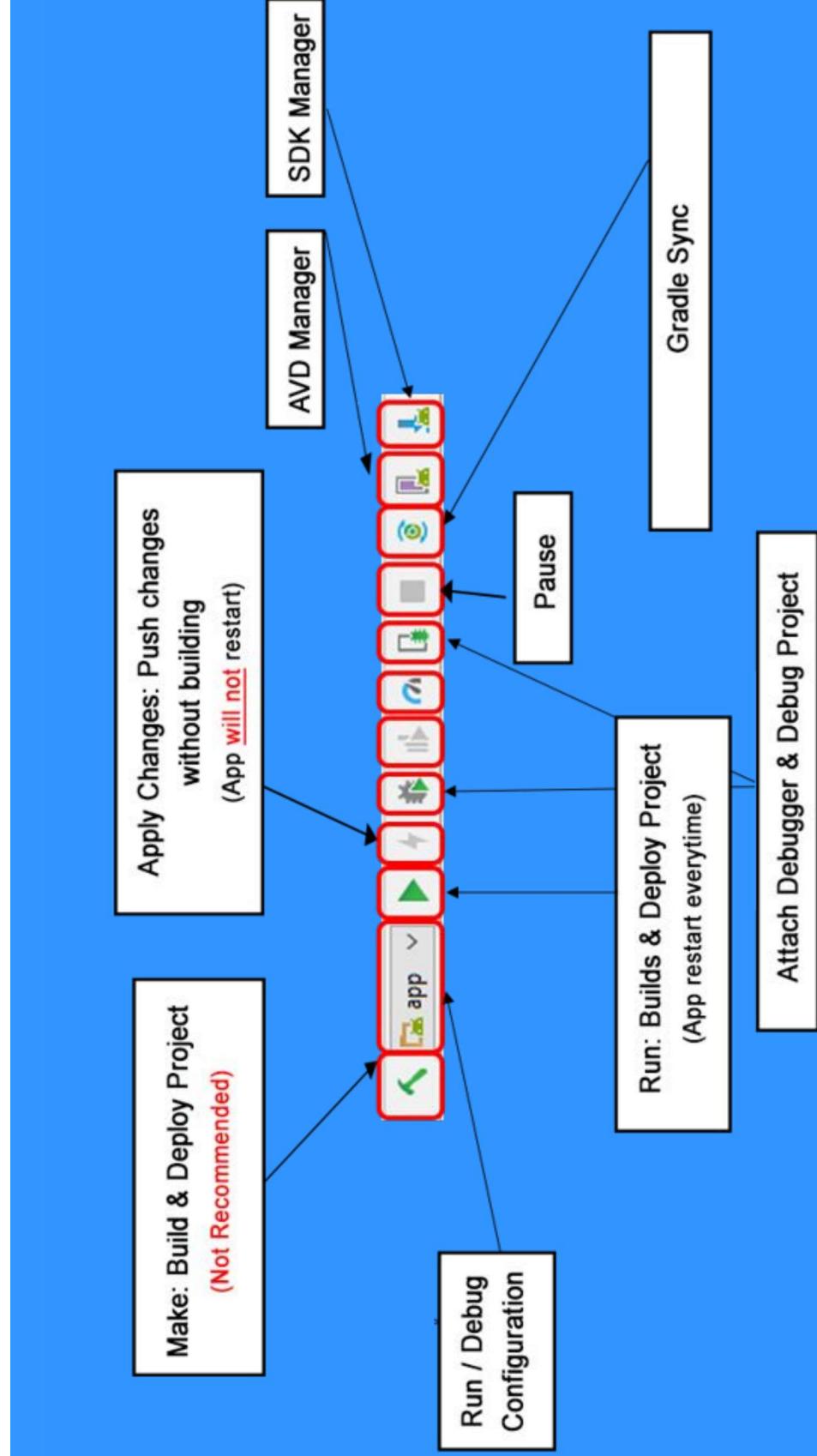
Android Studio



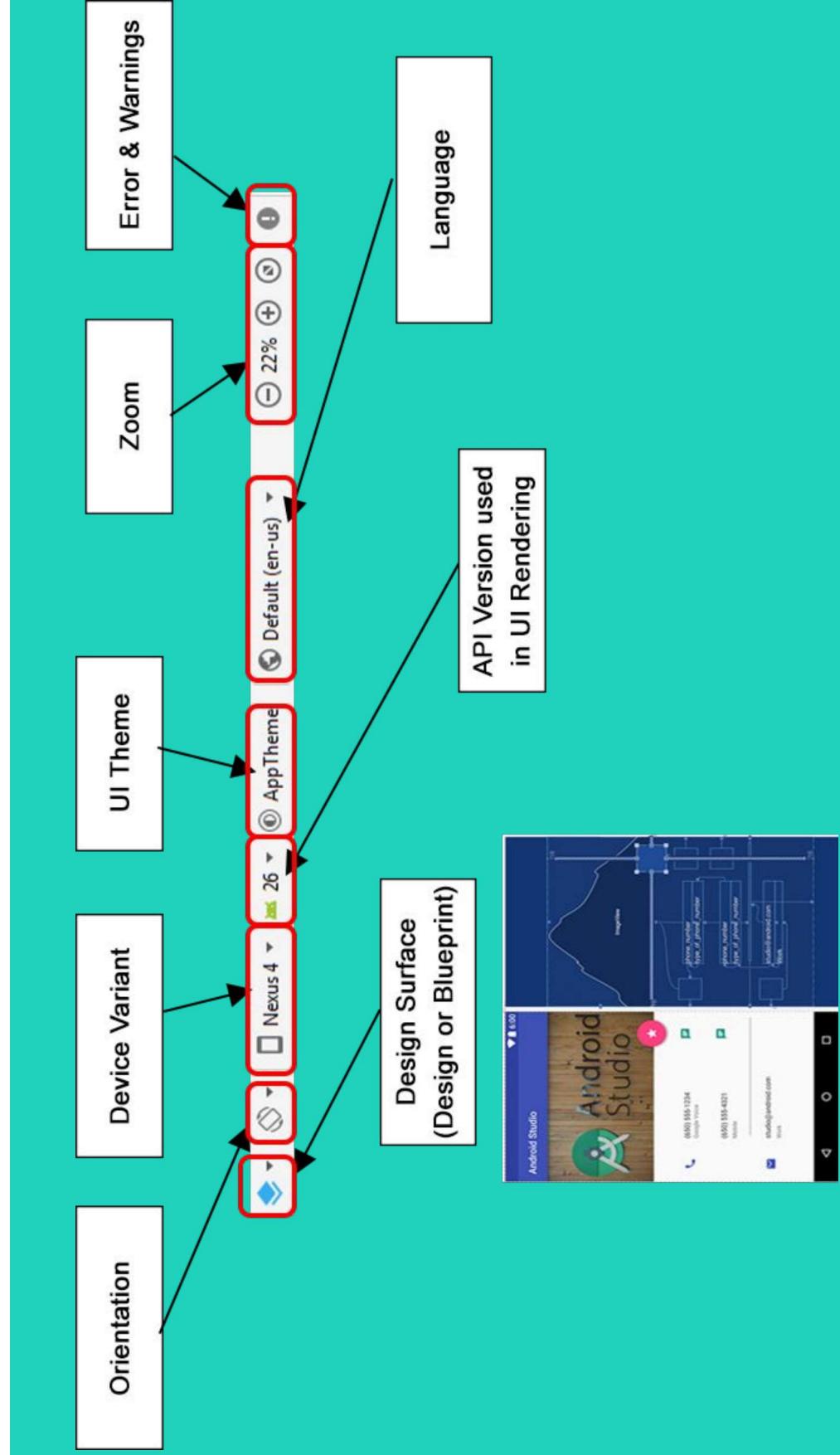
Studio Interface

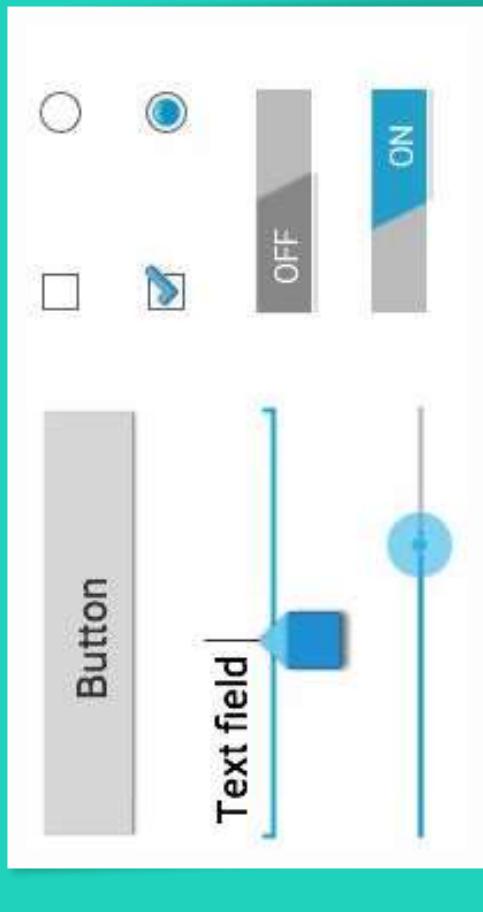


Developer Toolbar



Designer Toolbar





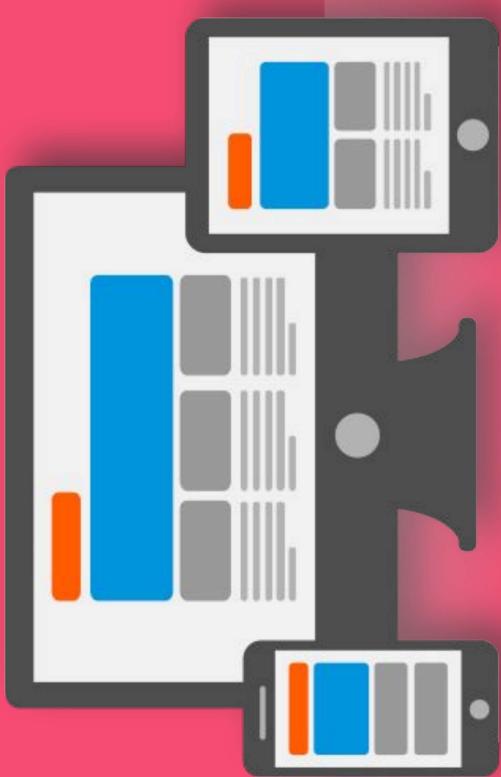
UI Controls/ Element

UI Controls

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

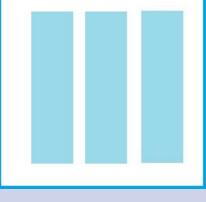
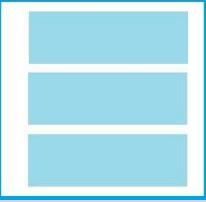
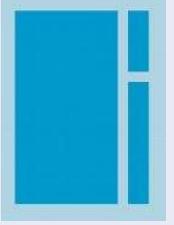
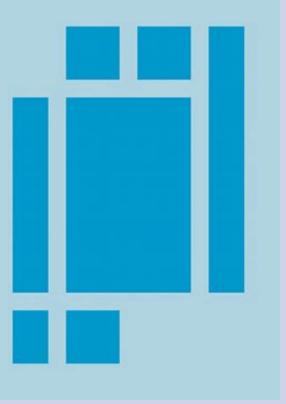
Advanced Controls

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

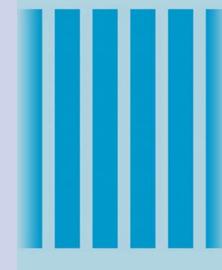
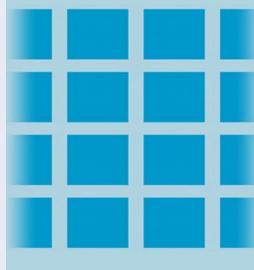


Layouts

Layout Types

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

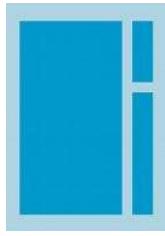
Cont...•

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

Relative Layout

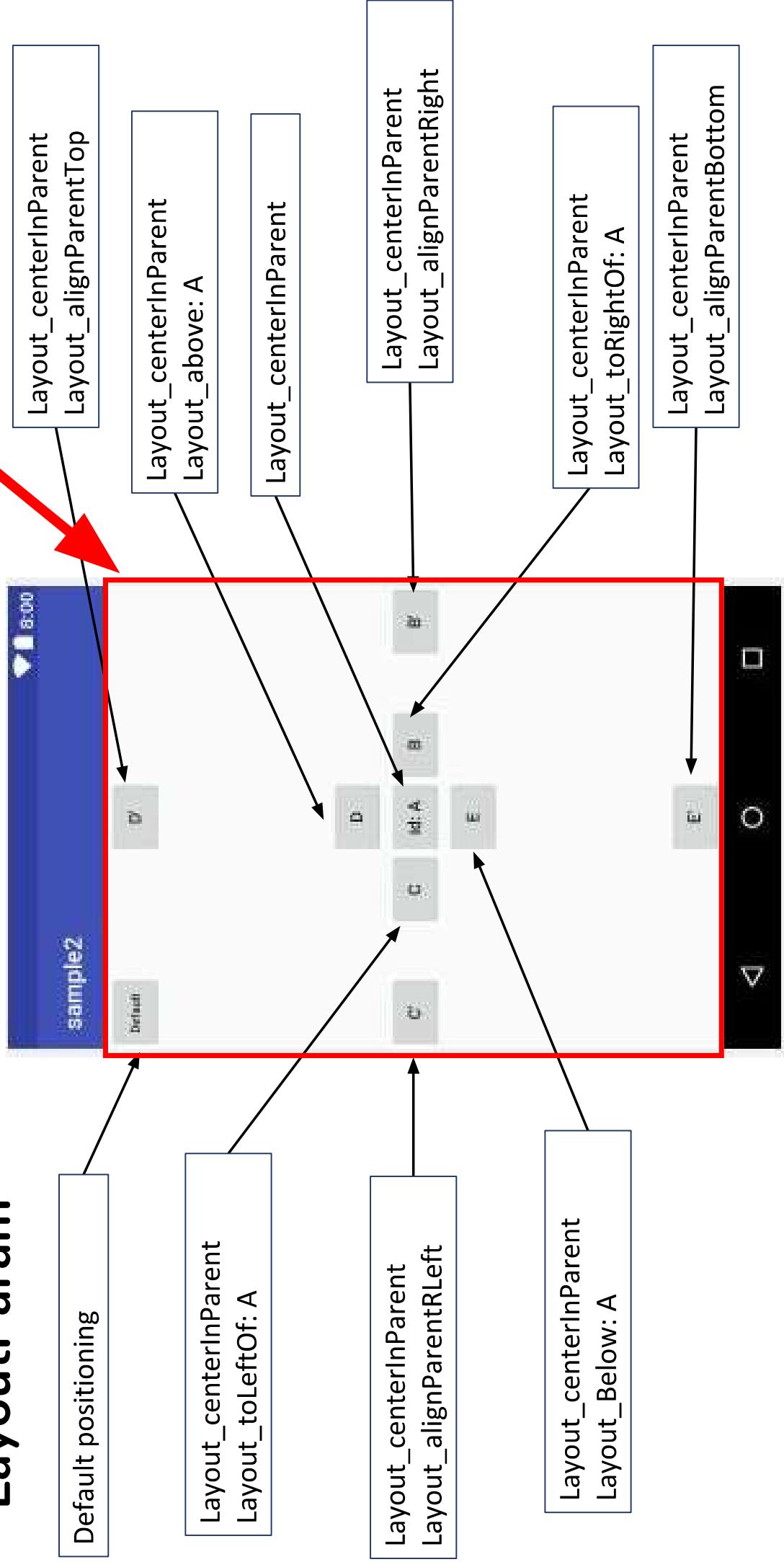
Steps

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



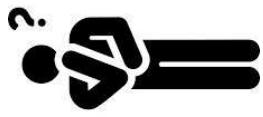
LayoutParams

THIS IS YOUR PARENT!



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

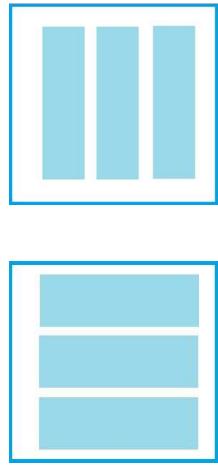
Use Linear Layout!



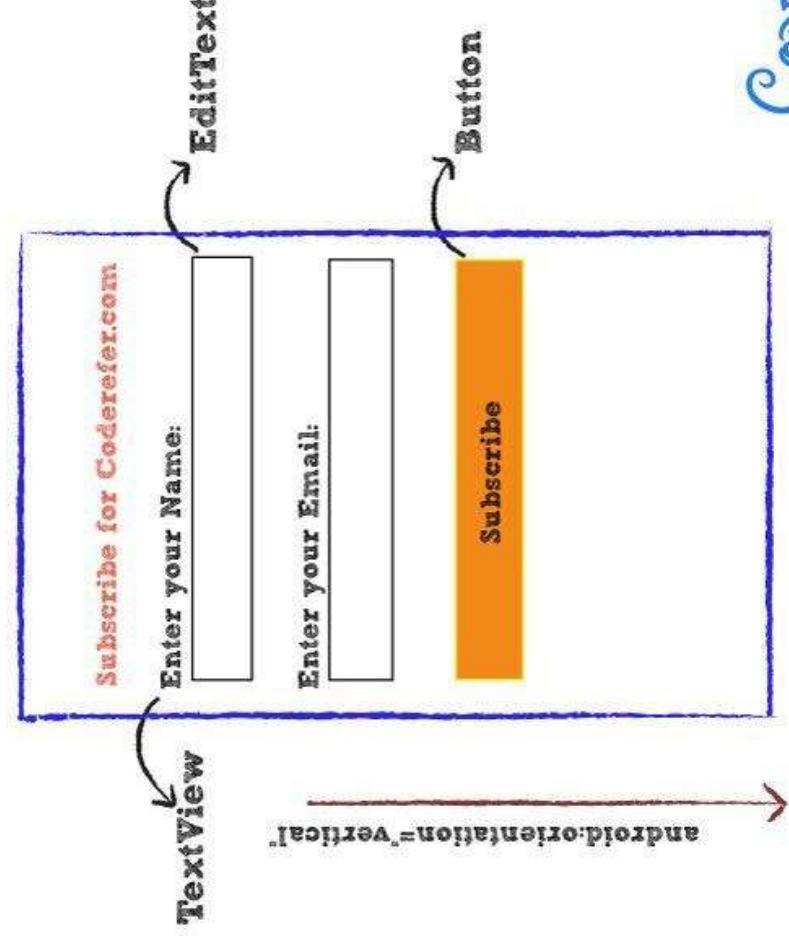
Linear Layout

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



Coderefer

Other Layouts

- Table Layout

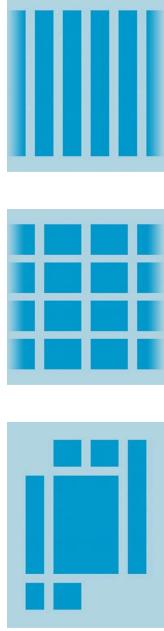
<https://www.youtube.com/watch?v=9jzI8JDHFtA&t=345s>

- GridView

<https://www.youtube.com/watch?v=VUPM387qryw>

- ListView

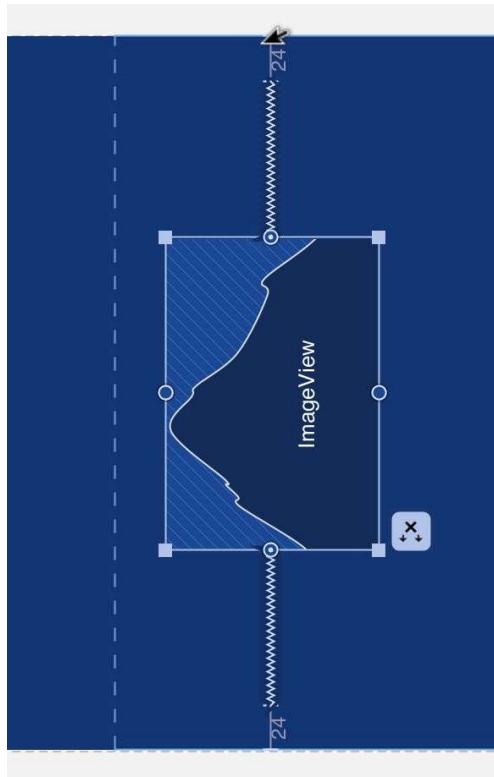
<https://www.youtube.com/watch?v=FKUlw7mFXRM>



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

Constraint Layout!

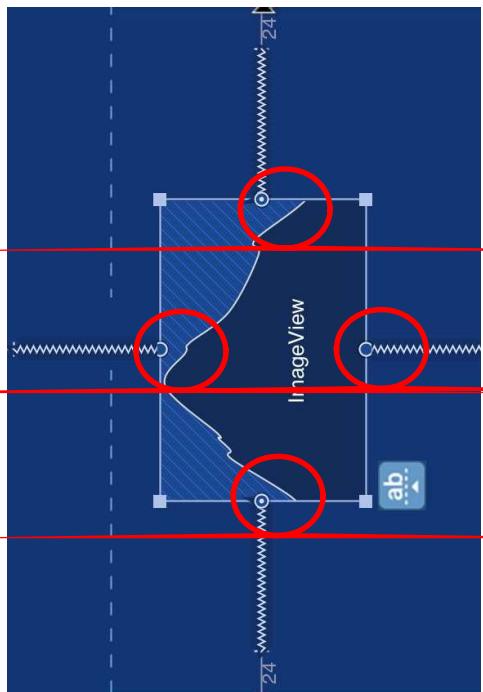
- Flat View hierarchy
- Form groups via “chain method”
- Does not break layout if an element is set to gone
- Can do both relative and linear



What is Constraint Layout?

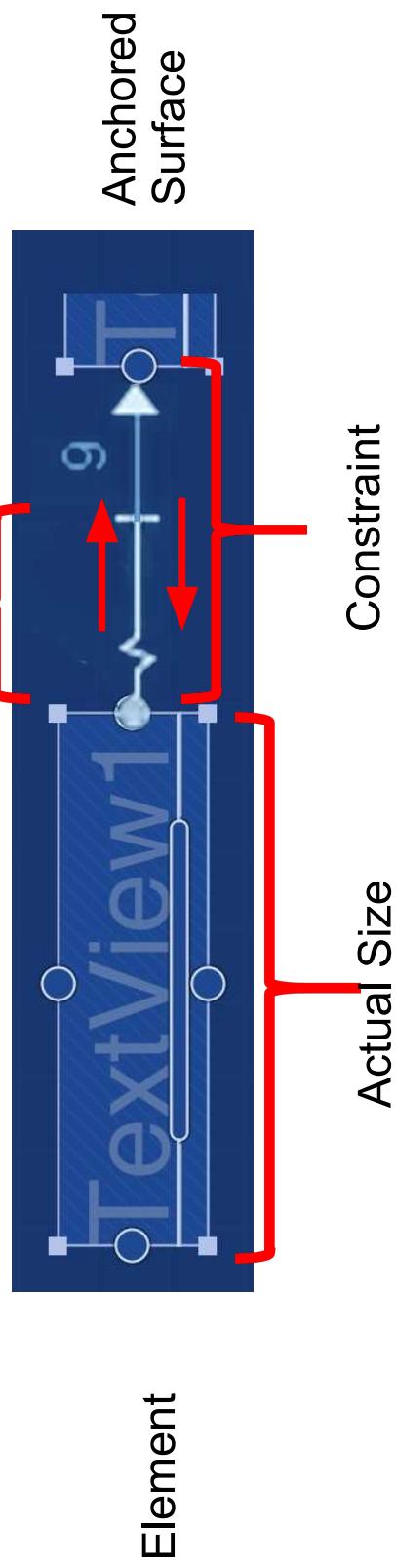
Similar to relative layout, but uses constraints to determine how UI elements are positioned relative to others. A constraint can be thought of as a connection or alignment to another element / parent via

Anchor Points



Features

Expanded Size



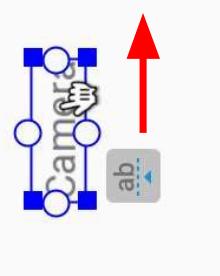
Resize handle: used for resizing view



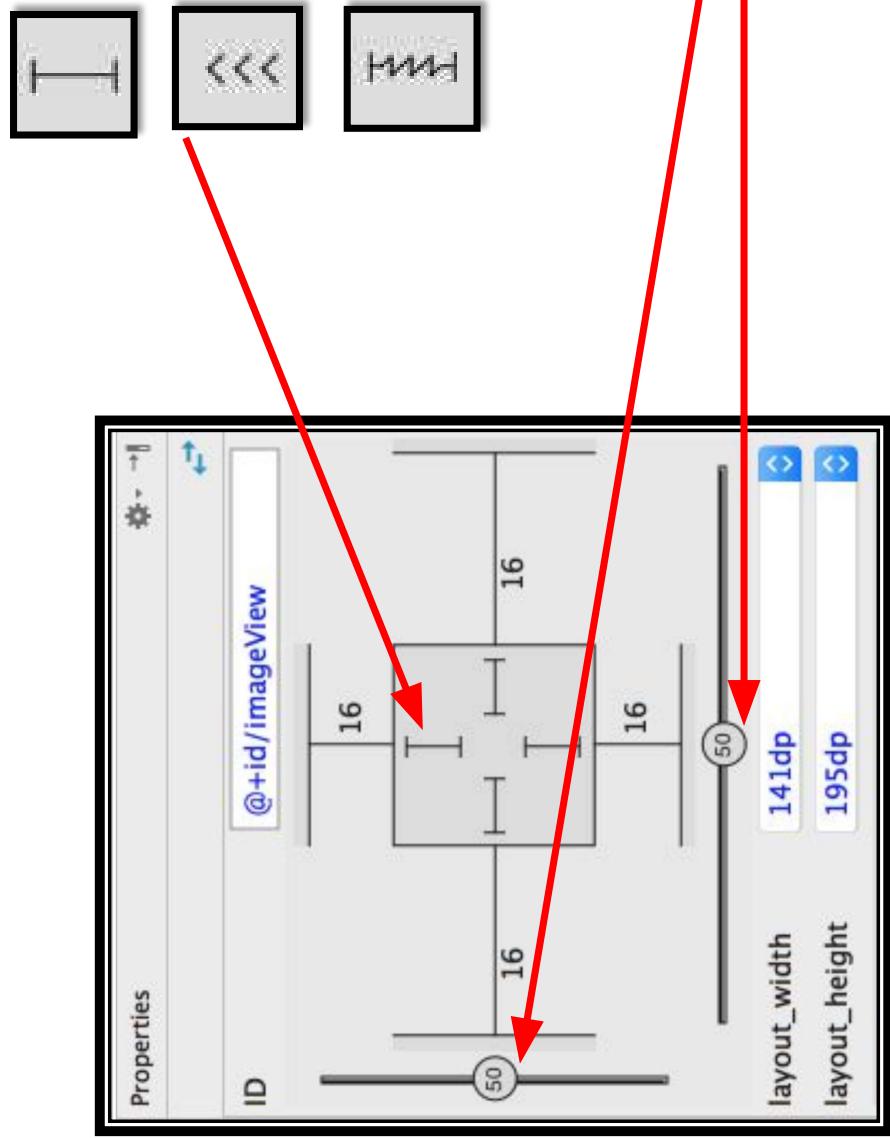
Side handle / Anchor: used to specify the position of the element by connecting to other element by default, anchored to constraint layout itself



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



Using the Inspector plane



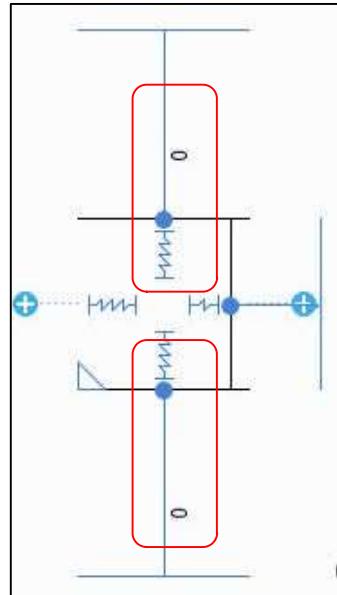
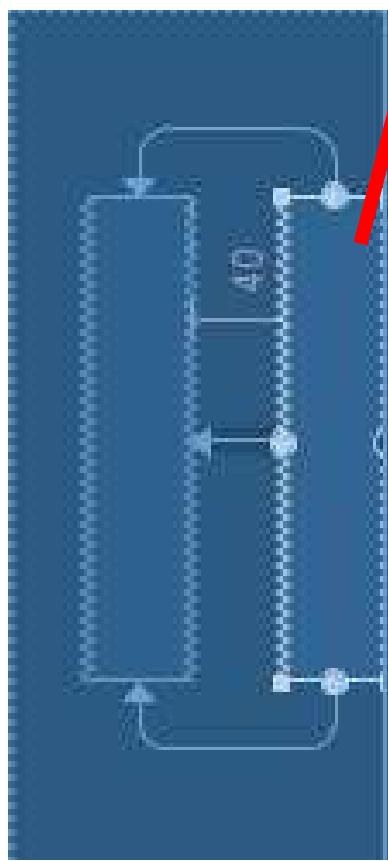
Fixed: Actual Element. Specify Constraint

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

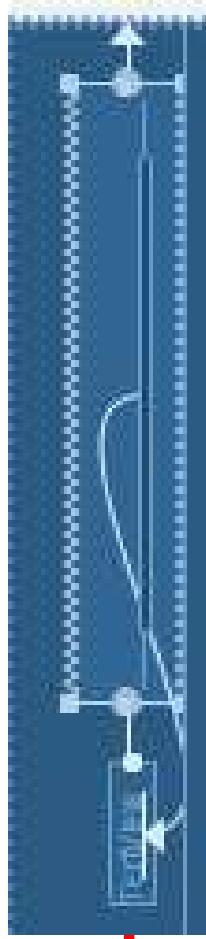
Any size: Expands Element. Specify distance between the expanded element and the anchored surface.

Vertical / Horizontal Biasness:
+/- constraint

Vertical Alignment



Horizontal Alignment (Baseline)





Importing Images

Steps

1) Find a non-copyrighted / open-source image

- Wallpaper: <https://unsplash.com/>
- Icons (.svg, .psd, etc.): <https://www.flaticon.com/>

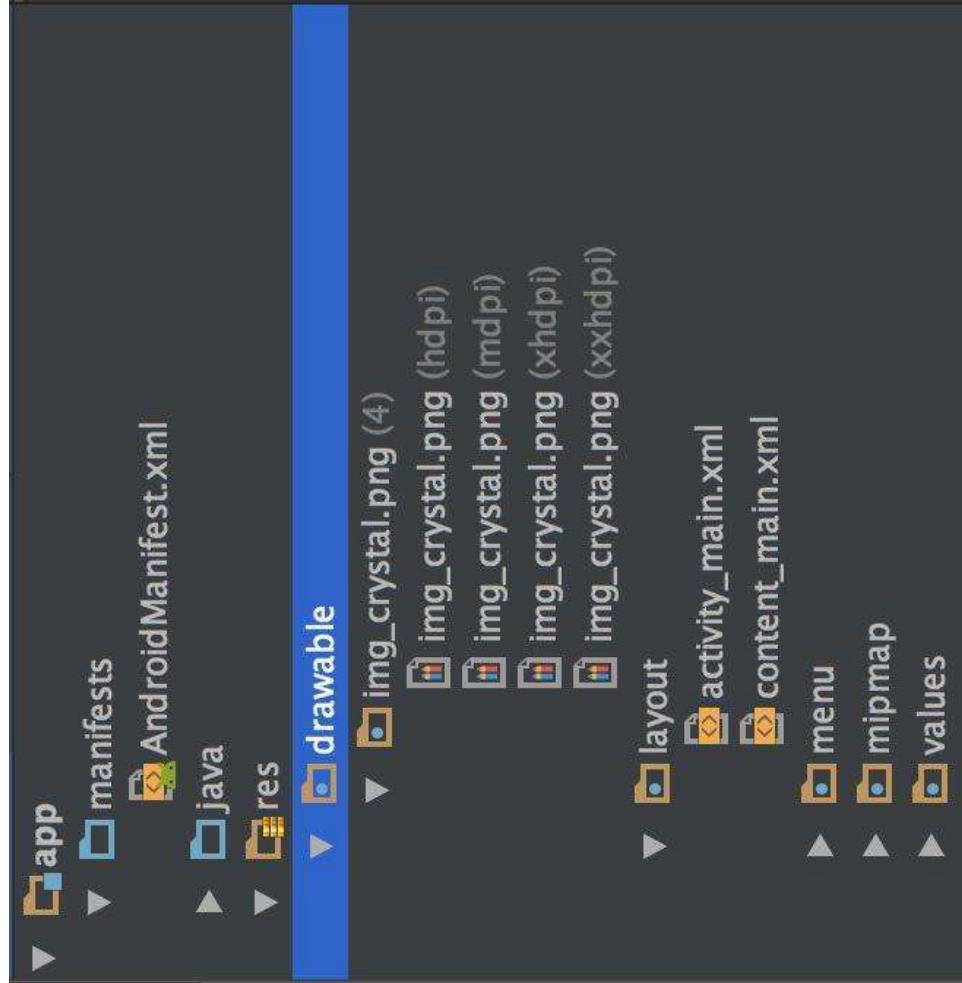
2) Download and Ctrl + C

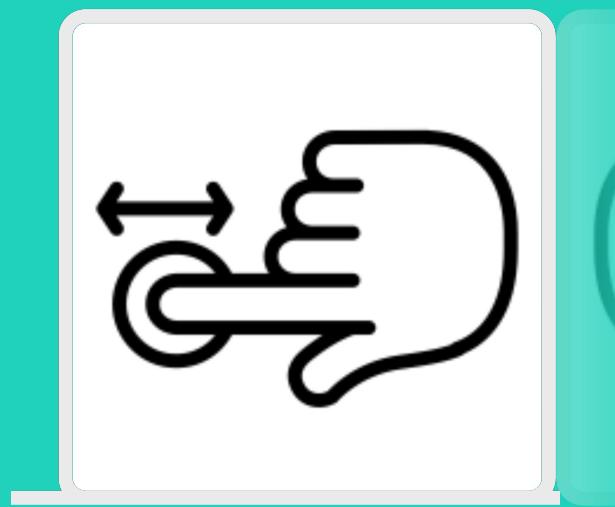
3) Navigate to app/res/drawable in Android Studio and Ctrl + P

* 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 *centerCrop*





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>
```

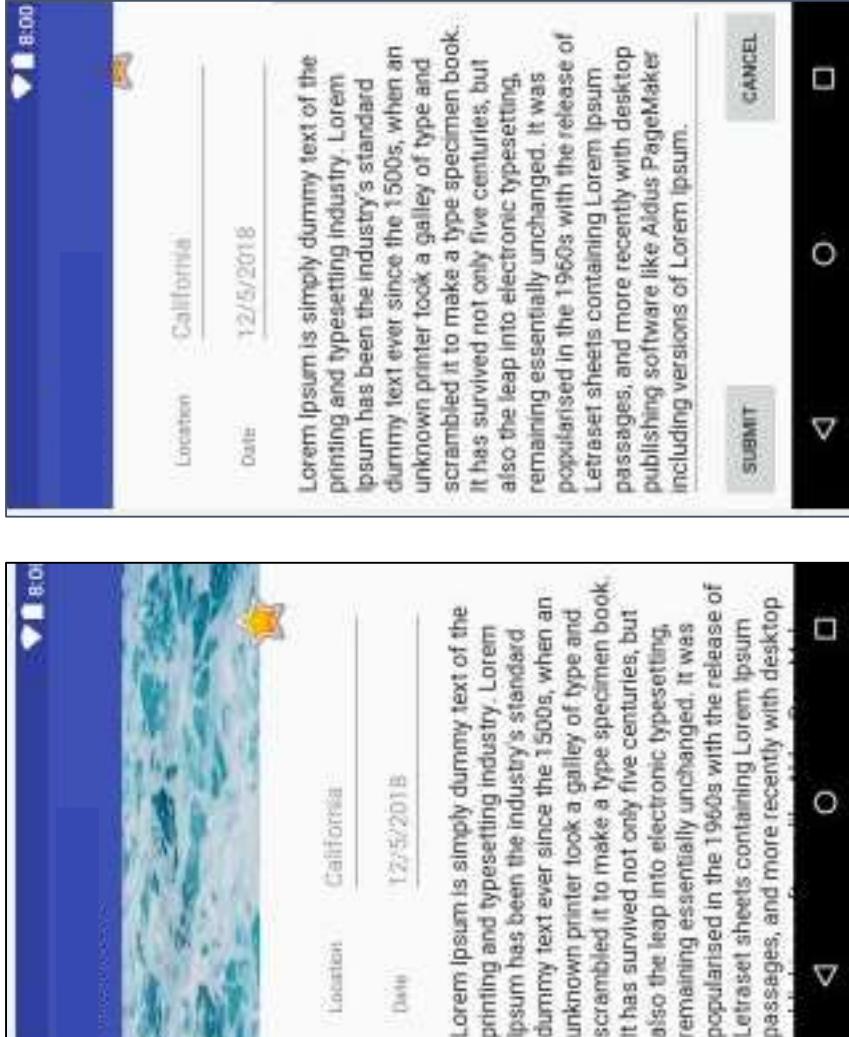
Horizontal Scrolling

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

Try it Yourself!

You will need...

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

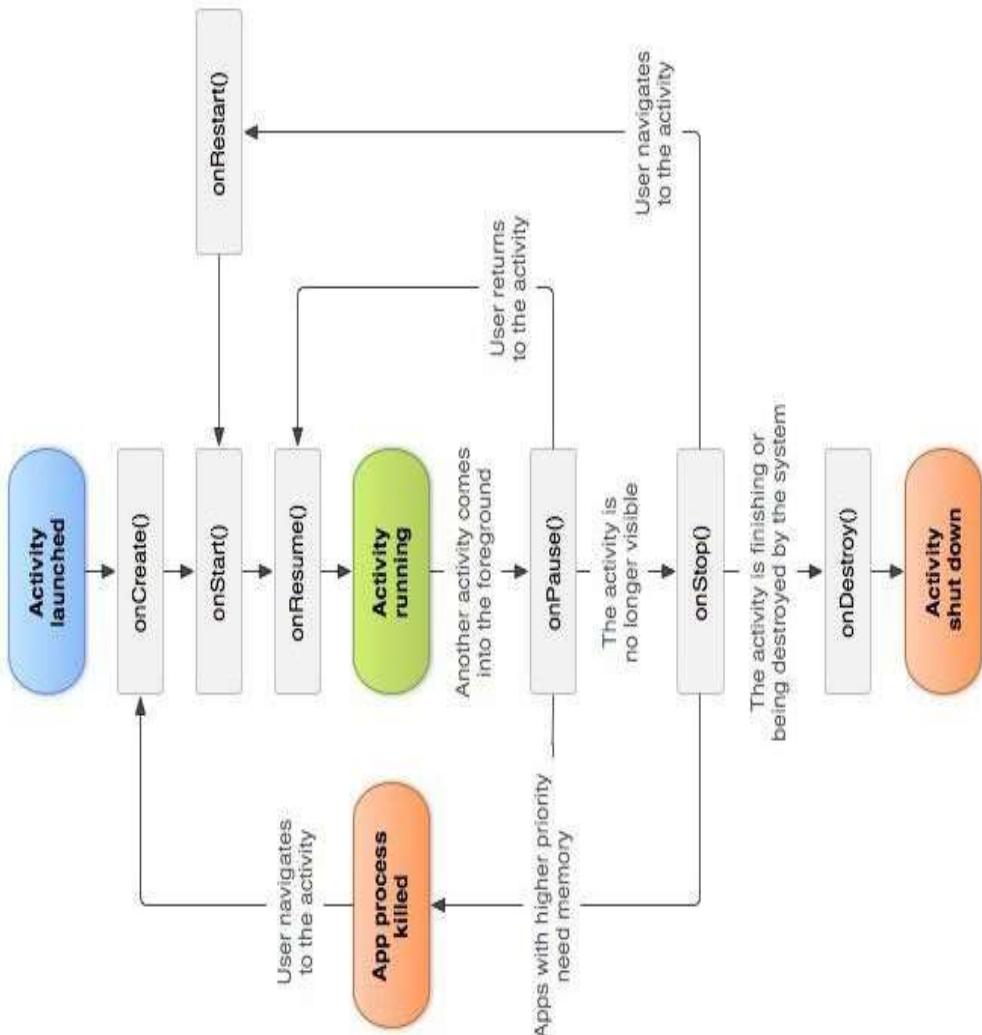


Make sure it works in landscape mode too!

Scripting

```
    ).removeEvent();
    a.fn.scrollspy=d, this}, a(window).on("scroll", function(b){return this.each(function(c){c.VERSION="3.3.7", c.TRANSITION_DURATION=150, c.prototype={}), !y}, +function(a){("use strict"; function b(){this.element=a(b); }c.VERSION="3.3.7", c.TRANSITION_DURATION=150, c.prototype={}).replace(/.*(?=\#[^\s]*$)/, "")}), !sel[b[1]()]})var c=function(b){this.element=a(b); }c.VERSION="3.3.7", c.prototype={}.replace(/.*(?=\#[^\s]*$)/, "")}, !opdown-menu"), d=b.data("target");if(d||(d=b.attr("href")), d=d&&d.replace(/\#/), e[d].relatedTarget:b[0]), g=a.Event("show.bs.tab", {relatedTarget:e[0]};st a"), f=a.Event("hide.bs.tab", {relatedTarget:b[0]}), g=a.Event("show.bs.tab", {relatedTarget:a}), h=a.parent(), function(faultPrevented()){var h=a(d); this.activate(h.closet("li"), c), this.activate(h.h.parent(), function(trigger){(type:"shown.bs.tab", relatedTarget:e[0]))}), c.prototype.activate=function(b,d,e){function(y){active").removeClass("active").end().find('[data-toggle="tab"]').attr("aria-expanded", !1), y).find('[data-toggle="tab"]').attr("aria-expanded", !0), h?b[0].offsetWidth,b.addClass("in")):b.removeClass("fade"), b.parent().dropdov(0).find("[data-toggle='tab']").attr("aria-expanded", !0), e&&e().var g=d.find("> .active"), h=e&&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-typeof b&e[b]()), var c=function(b,d){this.options=a.extend({}, c.DEFAULTS, d), this.$target=a.proxy(this.checkPosition, this)).on("click.bs.affix.data-api", a.proxy(this.checkPosition, this).RESET="affix", null, this.pinnedOffset=null, this.checkPosition());c.VERSION="3.3.7", c.RESET="affix", f="ob", state=function(a,b,c,d){var e=this.$target.scrollTop(), f=this.$element.offset(), g=this.$target="bottom"==this.affixed)?return null:c!=e-this.$target.scrollTop(), f!=this.$element.offset(), g=this.$target, l=c&&e<=c;"top":null!=d&&i+j=a-d&&i+j=a-d&&"bottom"}, c.prototype.getPinnedOffset(): !(e+g<=a-d)&&"bottom"!RESET).addClass("affix");var a=this.$target.scrollTop(), b=this.$element.offset(), g=this.$targetWithEventLoop=function("affix");var a=this.$target.setTimeout(a.proxy(this.$element.offset(), {if(this height(), d=this.$element.offset(), e=d.top, f=d.bottom)\\".css("top":this.$element.offsetTop, "bottom":this.$element.offset(), this).offset();return
```

Android Activity



Event Handler	Event Listener & Description
onClick()	OnTouchListener() 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() 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 handler to handle such event.
onKey()	OnFocusChangeListener() 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() 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() This is called when the user selects a menu item. You will use onMenuItemClick() event handler to handle such event.

Event Handling (in controller)

Navigate to app/java/com/MainActivity

```
activity_main.xml activity_2.xml MainActivity.java Activity2.java

1 package com.example.leslie.Sample;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 import com.example.leslie.deleteme2.R;
7
8 public class MainActivity extends AppCompatActivity {
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14    }
15
16 }
```

```
import  
    android.view.View
```



Wiring your
Buttons

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



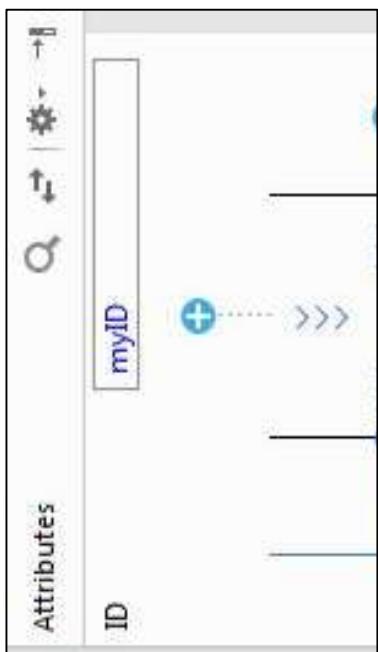
Import
android.widget.EditText



Wiring your
Textview

Steps

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

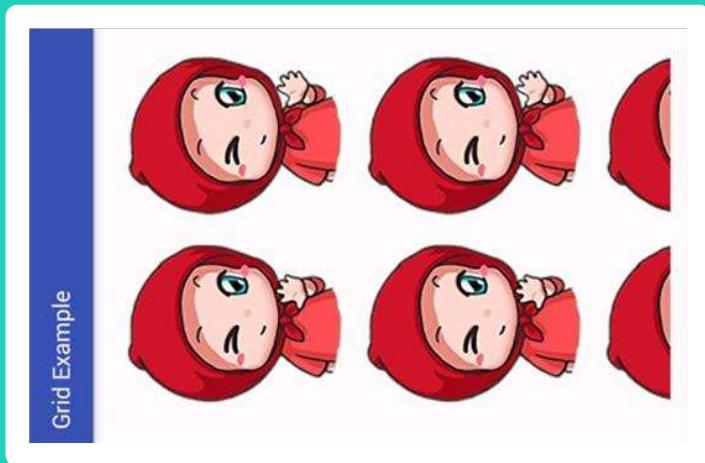


2. import android.widget.EditText;
3. Locate view using findViewById, parameter type ID
e.g. EditText myTextfield = findViewById(R.id.myID);

* To get current text: `myTextfield.getText().toString();`
* To set text: `myTextfield.setText("This is a sample text");`

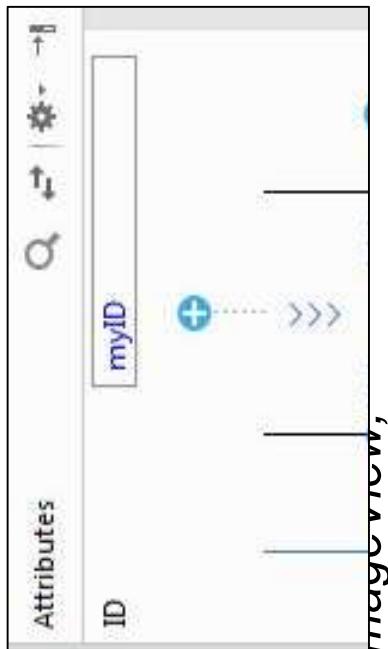
Wiring your
image

Import
`android.widget.ImageView;`



Steps

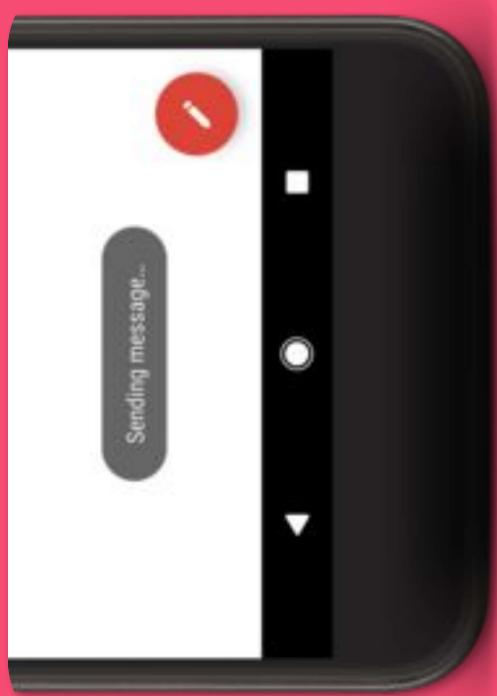
1. In your designer view, navigate to your imageView UI, under attributes, set an ID



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

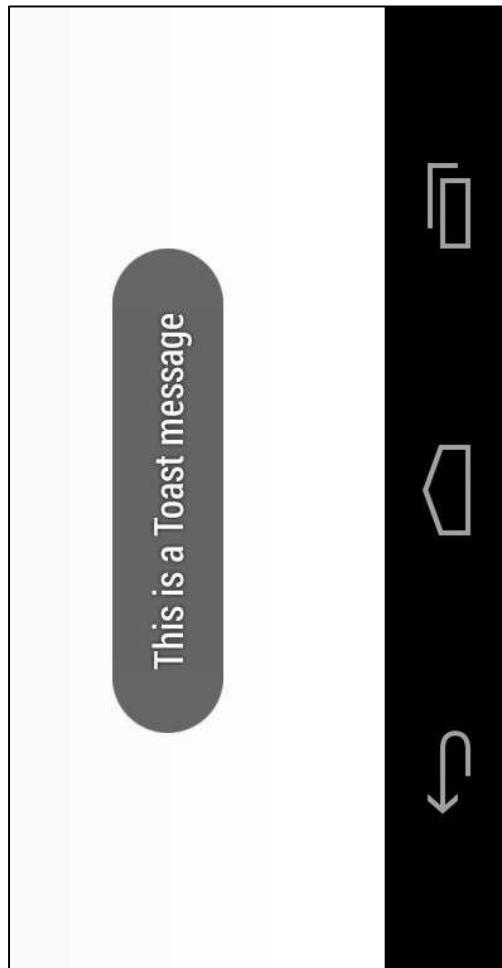
Toast Message?

```
Import  
    android.widget.Toast;
```



Steps

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



Import android.content.Intent

Switching Screen



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`

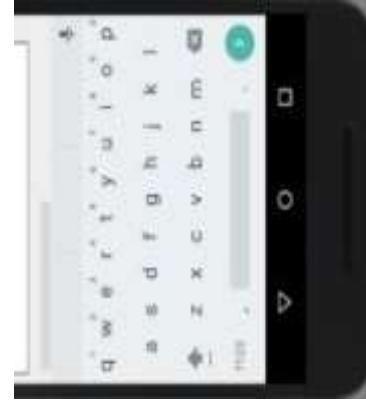
Show/Hide SoftKeyboard onCreate()

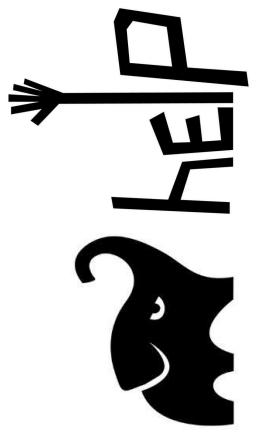
Hide:

```
getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_HIDDEN);
```

Show:

```
getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_VISIBLE);  
// import android.view.WindowManager
```





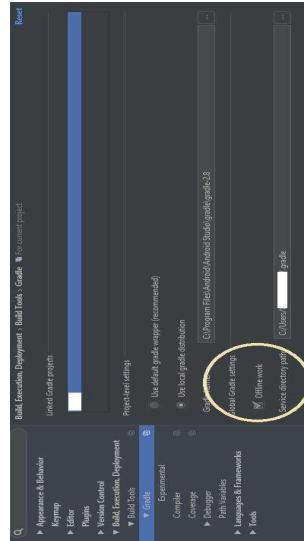
My Gradle Build is too slow!

HELP!

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 tap on the three-dot menu icon.
- 3) Tap on the 'Show Advanced Settings' button.
- 4) Under the "Emulated Performance" sub-category, "Boot option" menu, check 'Cold boot'.



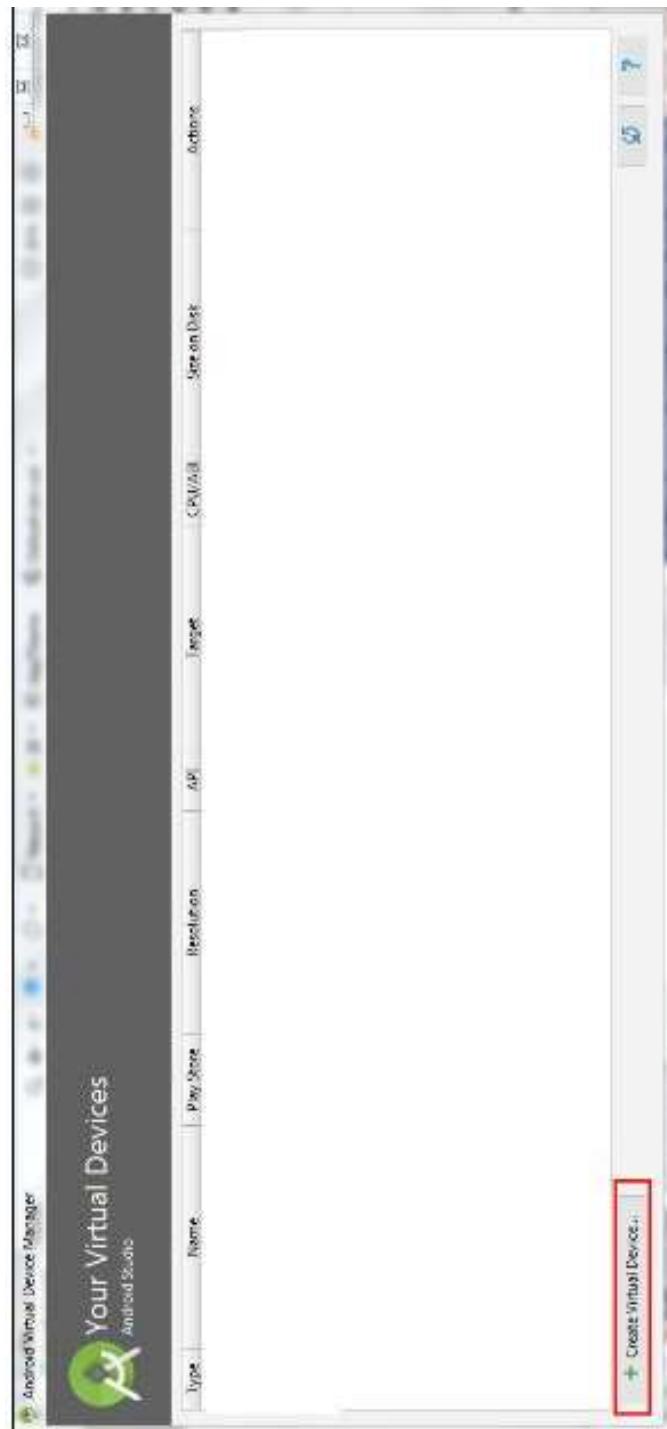
Thank You

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”





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

Category	Name	Play Store	Size	Resolution	Density
TV	3.2" QVGA (2D)		3.2"	320x180	mdpi
Phone	5.1" WVGA	5.1"	480x800	mdpi	
Phone	5.4" FWVGA	5.4"	480x854	mdpi	
Tablet	4.85" 720p (Scaling Nexus)	4.85"	720x1280	hdpi	
Tablet	4.7" WVGA	4.7"	720x1280	hdpi	
Tablet	Galaxy Nexus	4.65"	720x1280	hdpi	
Tablet	Nexus 4	4.7"	720x1280	hdpi	
Tablet	Nexus 5	4.95"	1080x1920	xhdpi	
Tablet	Pixel	5.0"	1080x1920	xhdpi	

Choose a device definition

Virtual Device Configuration | **Android Studio**

Next | **Cancel** | **Previous**

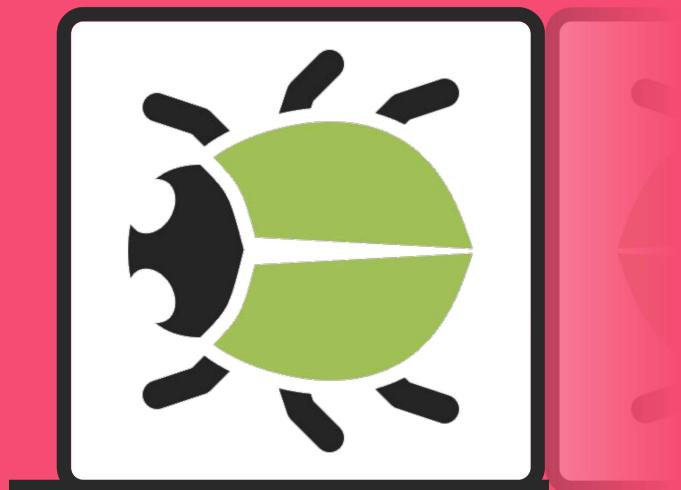
3) Select Nexus 4 and click “Next”

- 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

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

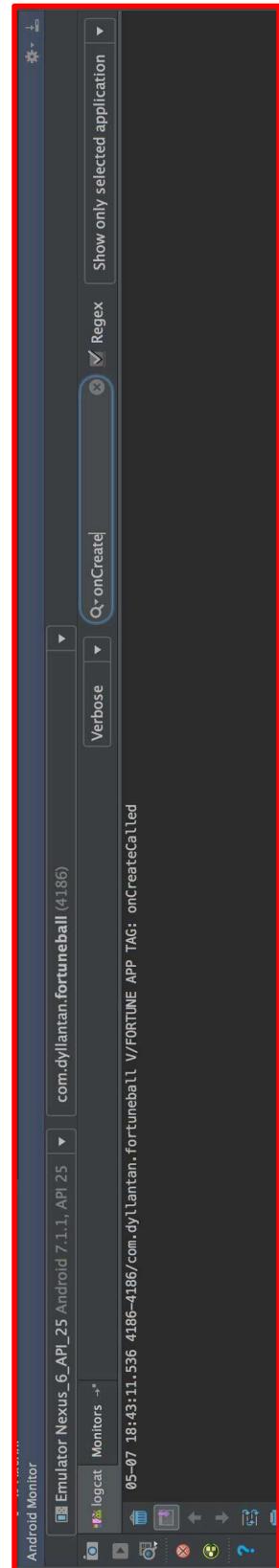
    mFortuneText = (TextView) findViewById(R.id.fortuneText);
    mFortuneBallImage = (ImageView) findViewById(R.id.fortuneImage);
    mGenerateFortuneButton = (Button) findViewById(R.id.fortuneButton);

    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);
}

import android.util.Log;

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     String fortuneList[] = {"Without a doubt", "Outlook not so good", "It's decidedly so", "Signs point to yes", "Yes definitely", "Yes", "My
24
25     TextView mFortunetext;
26     Button mGenerateFortuneButton;
27     ImageView mFortuneBallImage;
28
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         setContentView(R.layout.activity_main);
33         Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
34         setSupportActionBar(toolbar);
35
36         //mFortunetext = (TextView) findViewById(R.id.fortunetext);
37         //mFortunetext.setText("Generating Fortune Ball Image");
38         //mGenerateFortuneButton = (Button) findViewById(R.id.fortuneButton);
39
40         mGenerateFortuneButton.setOnClickListener(new View.OnClickListener() {
41             @Override
42             public void onClick(View view) {
43
44                 Intent intent = new Intent(MainActivity.this, FortuneBall.class);
45                 intent.putExtra("list", fortuneList);
46                 startActivity(intent);
47             }
48         });
49
50         YoYo.with(YoYo.DURATION_500)
51             .playOn(mFortuneBallImage);
52
53         Log.v("FORTUNE APP TAG", "onCreateCalled");
54     }
55 }
```

```
beginning of crash
FATAL EXCEPTION: main
Process: com.dylantan.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.dylantan.fortuneball.MainActivity$1.onClick(MainActivity.java:46)
at android.view.View.performClick(View.java:5637)
at android.view.View$PerformClick.run(View.java:2429)
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) <-- internal call
at java.lang.reflect.Method.invoke(Native Method)
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>

