

# Android Development

## layouts in Android Studio

① Linear layout - linear layout is a <sup>view</sup> ~~workgroup~~ that aligns all the children v/h

two types of linear layout

① horizontal

② vertical

layout gravity → alignment of text

layout weigh → how much size of screen

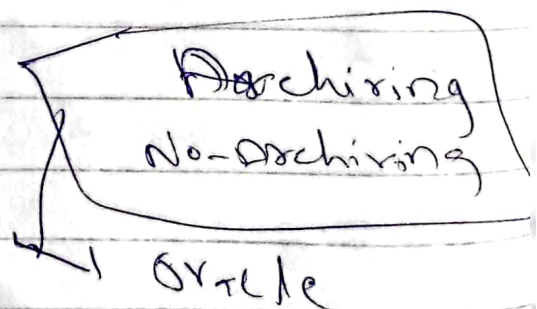
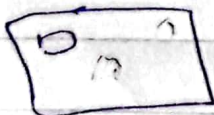
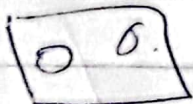
layout Divider → to see the layout division

Android: Orientation: "horizontal"

### Divider

① Divider width

② Divider color



## Android Functions

- ① findViewById (R.id.myButton) → resourceid
- ② `setOnClickListener()` → this is an interface event listener
- ③ `Resource.getText()` → this value get the value of Particular Resource
- ④ `parseInt()` → change the data type
- ⑤ `textView.setText()` → Put a text into Resource
- ⑥ Intent is for change the activity inside the app
- ⑦ `uses-Permission` → use to give Permission to like Internet.  
in Manifests folder →  
androidmanifest.xml file.  
<uses-permission> → for allowing Internet

## Fragments



## ① Udacity, Basic Android Development course

① Views in AS  
every thing we see in our app is used by views.

① Button

① TextView

`android:text = "Hello world"`  
→ Attributes Name → Attributes Value

DP → Density-Independent Pixels.  
↳ we atleast 48DP to for perfect resolution

Creating errors Part

SP → Scalable Independent Pixels

\* ImageView / \*

scaleType → center and centerCrop

textAllCaps → Capitalize the text

ViewGroup = layout



## Linear layout Weight

Using Linear layout Weight

- ① Vertical  $\rightarrow$  height equal to ODP
- ② Horizontal  $\rightarrow$  width equal to ODP

Note - if ODP given to other childrens and 1 weight given to other view it will take the rest of screen.

## ③ Relative layout

Attributes

- $\rightarrow$  layout\_alignParentTop = "True/False"
- $\rightarrow$  layout\_alignParentLeft = "T/F"
- $\rightarrow$  layout\_alignParentRight = "T/F"
- $\rightarrow$  layout\_alignParentBottom = "T/F"

## ③ Positioning Relative layout

- $\rightarrow$  android:layout\_toLeftOf = "id name of that"
- $\rightarrow$  layout\_above = "id"

## Dynamic Intent

ca. PutExtra()  $\rightarrow$  for giving Extra  
 $\rightarrow$  name and value Data

Bundle  $\rightarrow$  class  $\rightarrow$  startActivity().getExtra()

# Android Architecture

- ① Application Layer
- ② Application Framework
- ③ Libraries → Graphics → Media → SQLite, SSL  
network
- ④ Kernel Linux → OS layer android base on this
- ⑤ Android Runtime
  - ↳ Android Libraries
  - ↳ Virtual Machine

---

## Animation in Android

- ① translate → to move the object from x,y
- ② Rotate
- ③ Reverse
- ④ Alpha

android:duration = "4000"

repeatCount = "infinite"

repeatMode = "reverse"

Animation view → AnimationUtils.loadAnimation(  
this, R.anim.move)

setAnimation() → load  
startAnimation() → click



## 5 Fragments in Android

Code

```
FragmentManager fragName = new getSupportFragmentManager();
```

```
fragName.beginTransaction().replace(R.id.lytMain, fragName, "class",  
    .setReorderingAllowed(true)  
    .addToBackStack("Main").commit();
```

~~Android Life in Android~~

Fragment represents a reusable portion of your app's UI.

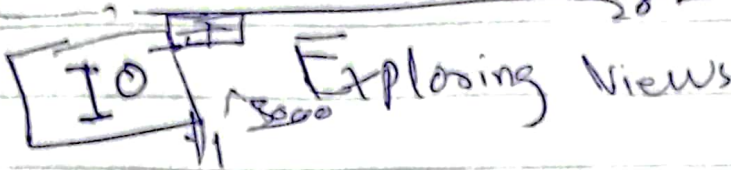
getSupportFragmentManager -> when we get a fragment from an activity then we call get


Add Buttons one by one

```
substring(0, data.calculate.length() - 1);
```

# Android Development

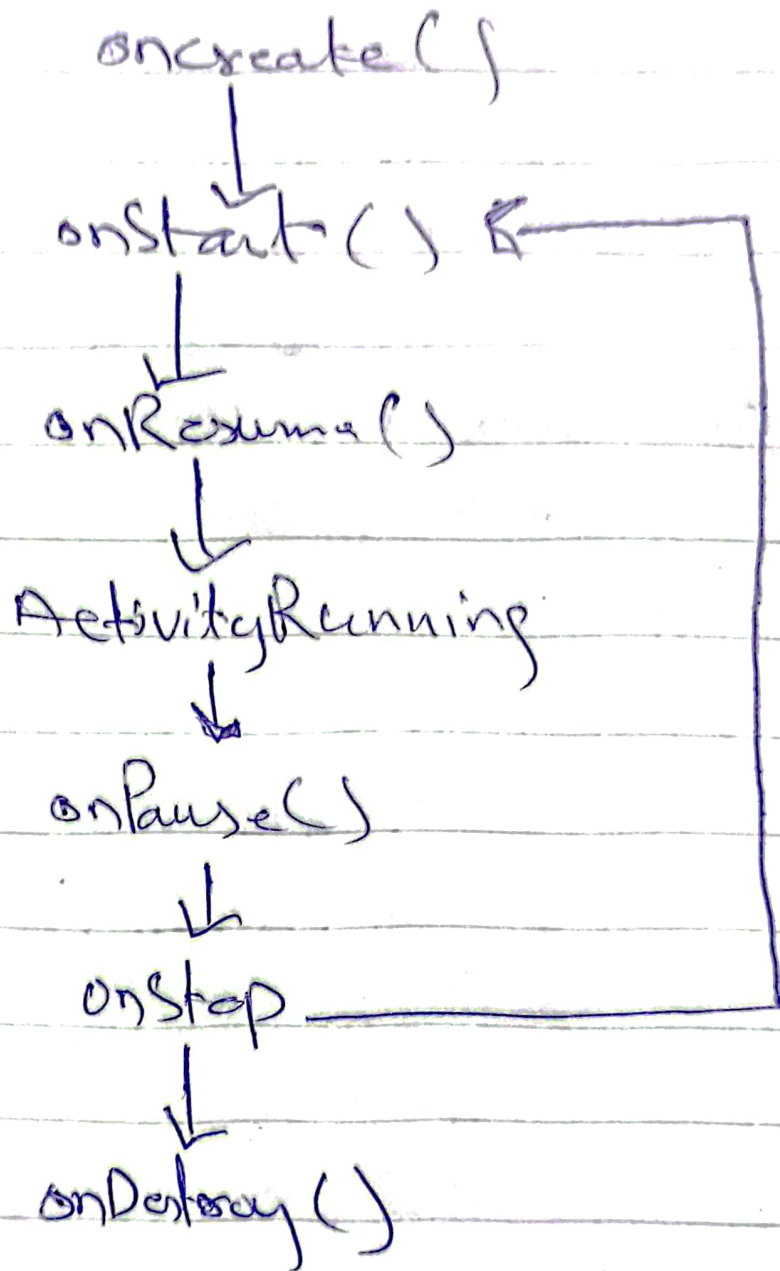
## RecyclerView in Android



- ① SearchBar → gives a searching textBox with search icon
- ② CalendarView → Gives us calendar view starting from current date.
- ③ ProgressBar → Circular ProgressBar which runs continuously.
- ④ ProgressBar (horizontal) →  horizontal Progress Bar static
- ⑤ SeekBar → it is a range defining Bar with movable.
- ⑥ Discrete seekbar → ~~same~~ range defines with Particular discrete dots
- ⑦ RatingBar → gives us horizontal stars ~~with~~ <sup>for</sup> rate something
- ⑧ EditTextView → gives us textBox with editable feature
- ⑨ PasswordText → Gives us text for Password feature with ~~xxx~~ like dots in alphabets
- ⑩ PasswordTextNumbers → Gives us Password feature in Numbers



# Android Activity Life Cycle



requirement Analysis and Design