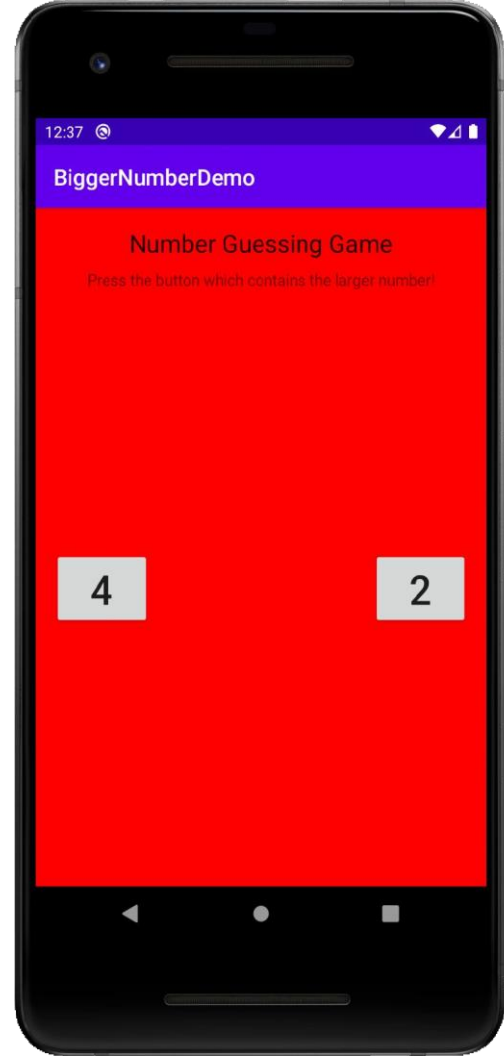
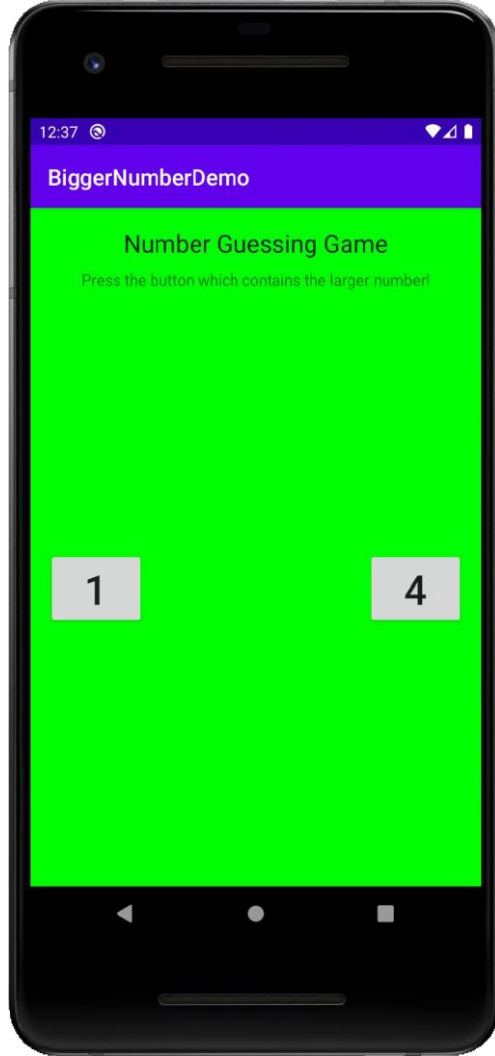
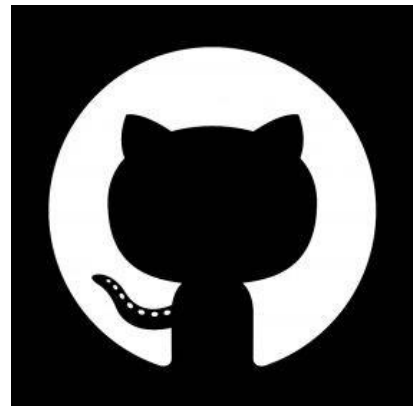
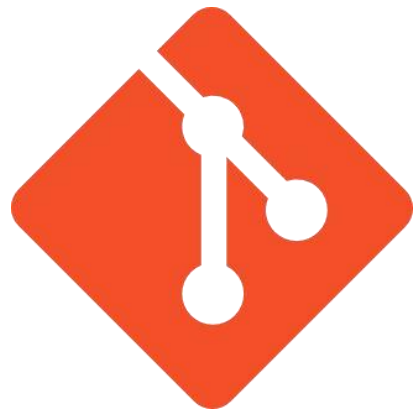


- Layouts



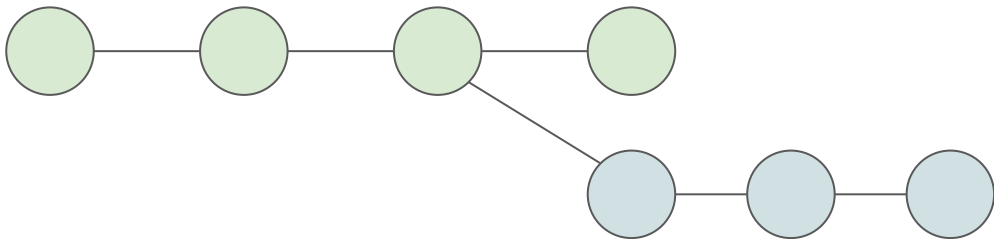
Source Control

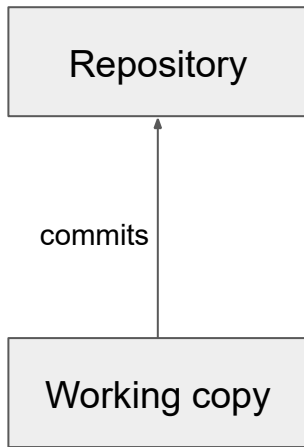
- Records all your code changes
- Single source of truth for all developers ● Key functionality
 - Revert files back to previous state
 - Compare changes over time
 - Easily collaborate on features
 - “Blame” a code change :) for bug triage or learning



Git: a distributed version control system

- One of many version control systems
 - Git (git)
 - Mercurial (hg)
 - Subversion (svn)
- Understand the fundamentals of git: <https://git-scm.com/book/en/v2>

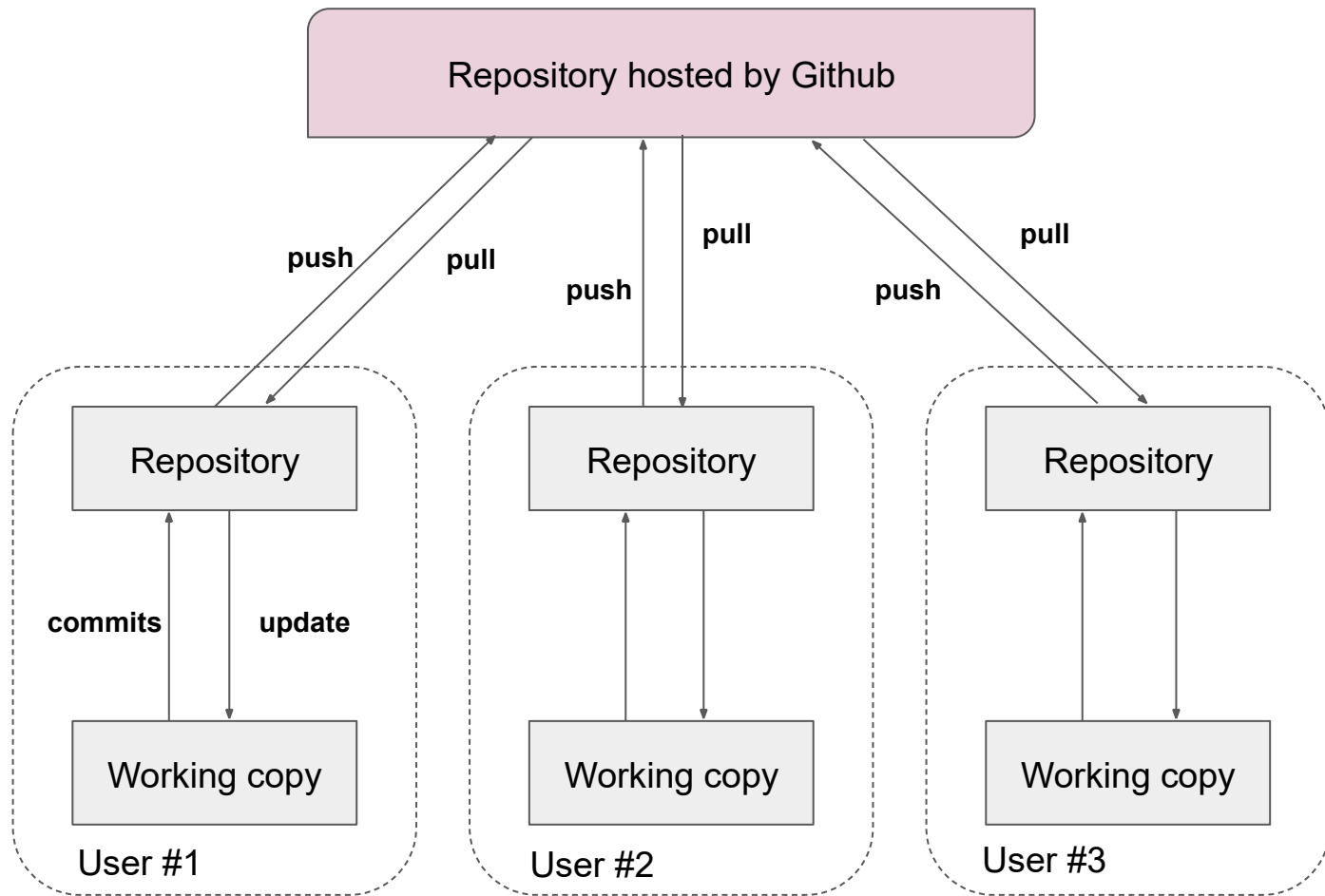


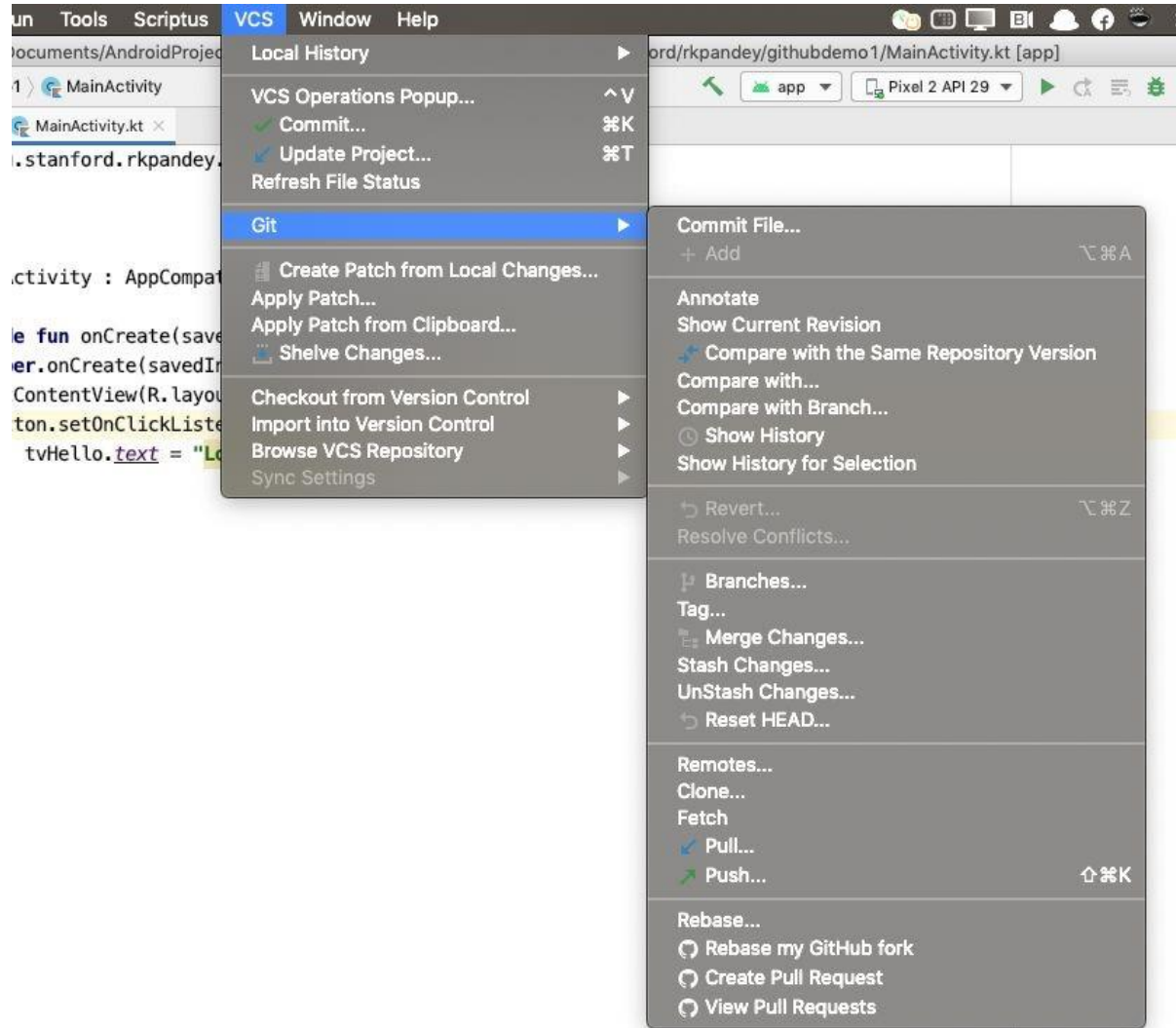


Github

- A website/community that provides hosting for Git repositories
- Allows you to easily view files in the browser and comment on changes
 - Register for an account at github.com

Server



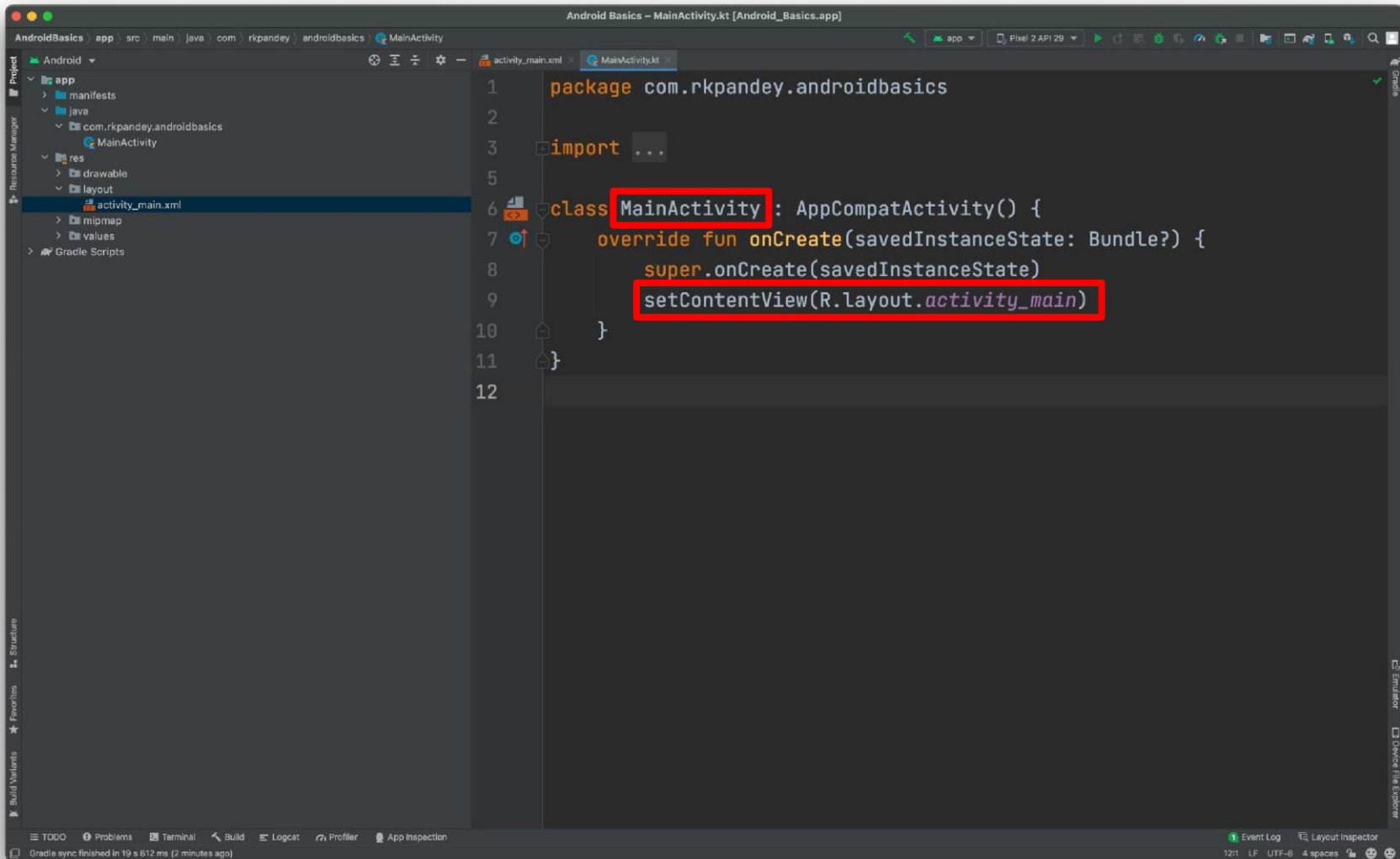


Public vs private repos

- Prefer public repositories:
 - Better for building a portfolio
 - Easier for sharing
- Be careful with what you put in Github
 - Secret API keys or access tokens
 - Use .gitignore file

Android Activity

- A “screen” in Android is called an Activity
- Activities are where we write our application logic:
 - handle user interaction
 - change what’s on the screen
 - tell views to update



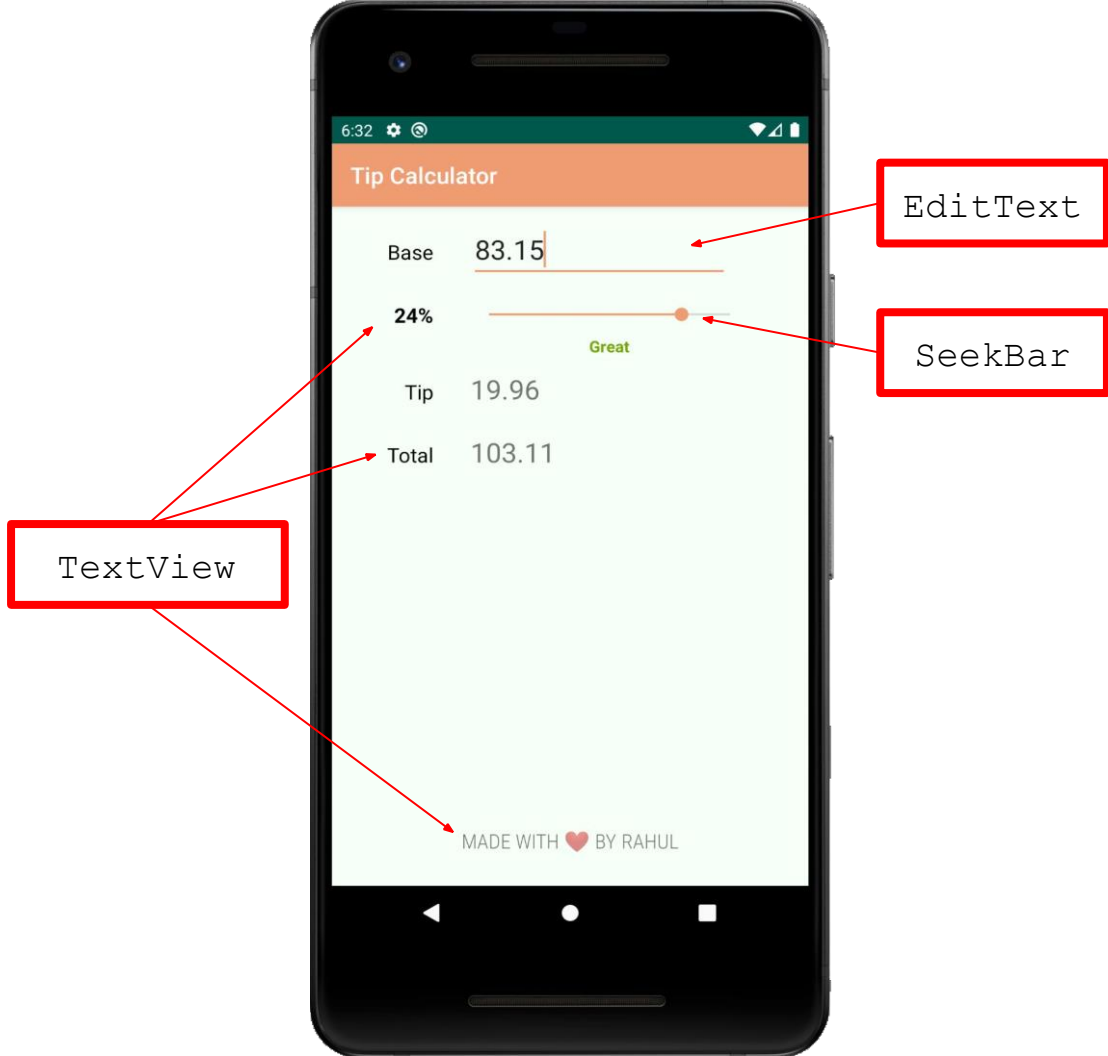
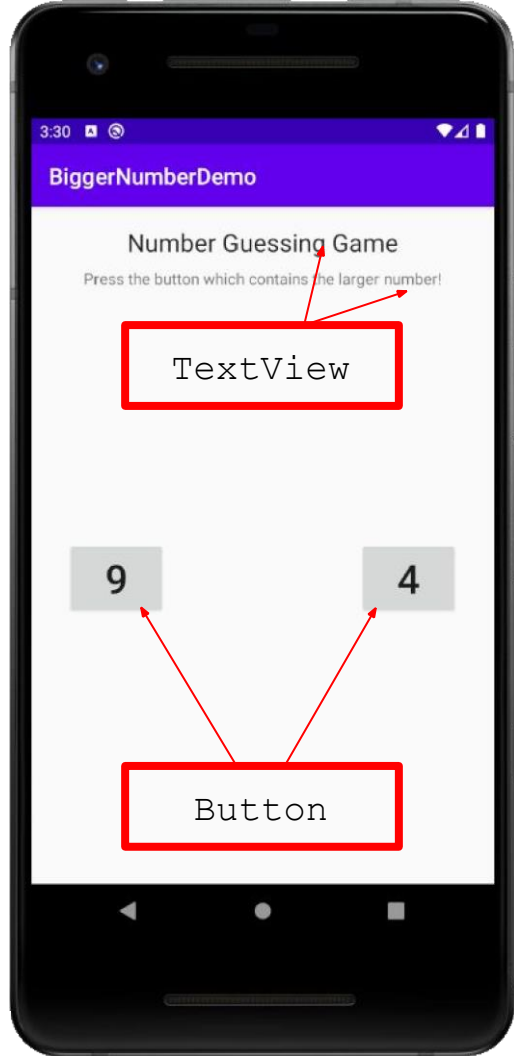
Layouts

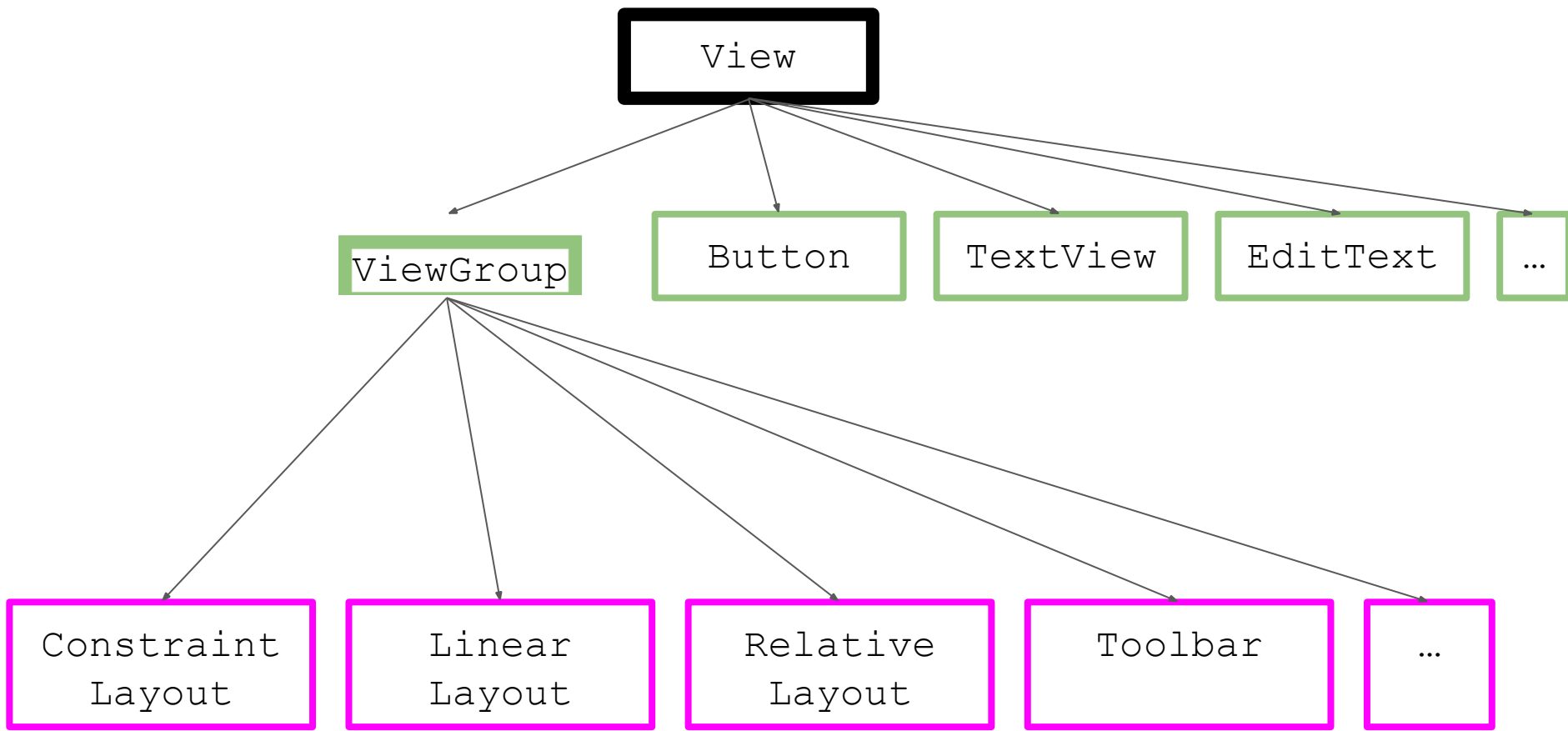
- Layouts are defined in XML
- Consist of various “views”
 - `TextView`
 - `ImageView`
 - `Button`
 - `EditText`
 - ...

Activity ↔ Layout Communication

```
activity_main.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical">
6
7     <TextView
8         android:id="@+id/tvHelloWorld"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="Hello World!" />
12
13 </LinearLayout>
```

```
MainActivity.kt
1 package com.rkpandey.androidbasics
2
3 import ...
4
5
6
7 class MainActivity : AppCompatActivity() {
8     private lateinit var tvHelloWorld: TextView
9
10
11
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_main)
15         tvHelloWorld = findViewById(R.id.tvHelloWorld)
16         tvHelloWorld.text = "Updated!"
17     }
18 }
```





Layouts (ViewGroups)

- ViewGroup: a special view that can contain other views (called **children**)
- Subclasses:
 - ConstraintLayout
 - LinearLayout
 - FrameLayout
 - RelativeLayout

XML

- Language used to describe hierarchical data (e.g. views inside a viewgroup)

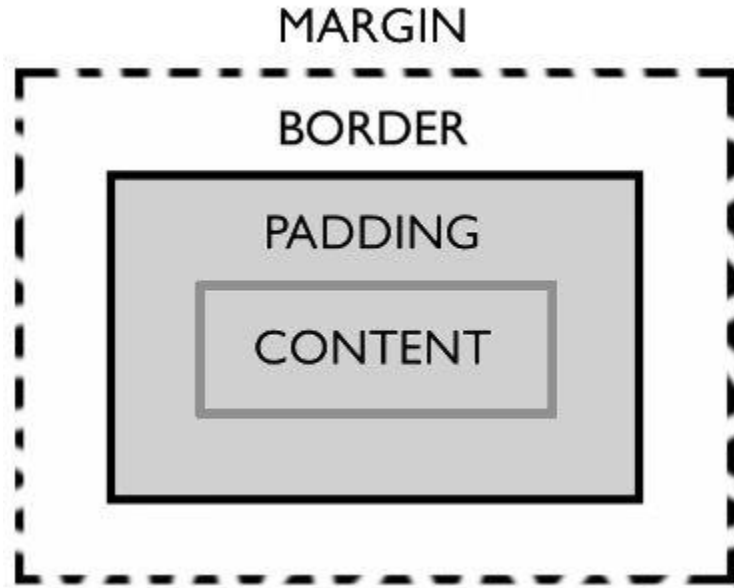
- `<element attr1="value" attr2="value"> ... </element>`

- `<element attr1="value" attr3="value" />` (self-closing) ●

Case sensitive!

- Let autocomplete help you

Padding inside, margin outside



Prefer design tab, but understand XML

Most professional devs will edit the XML directly

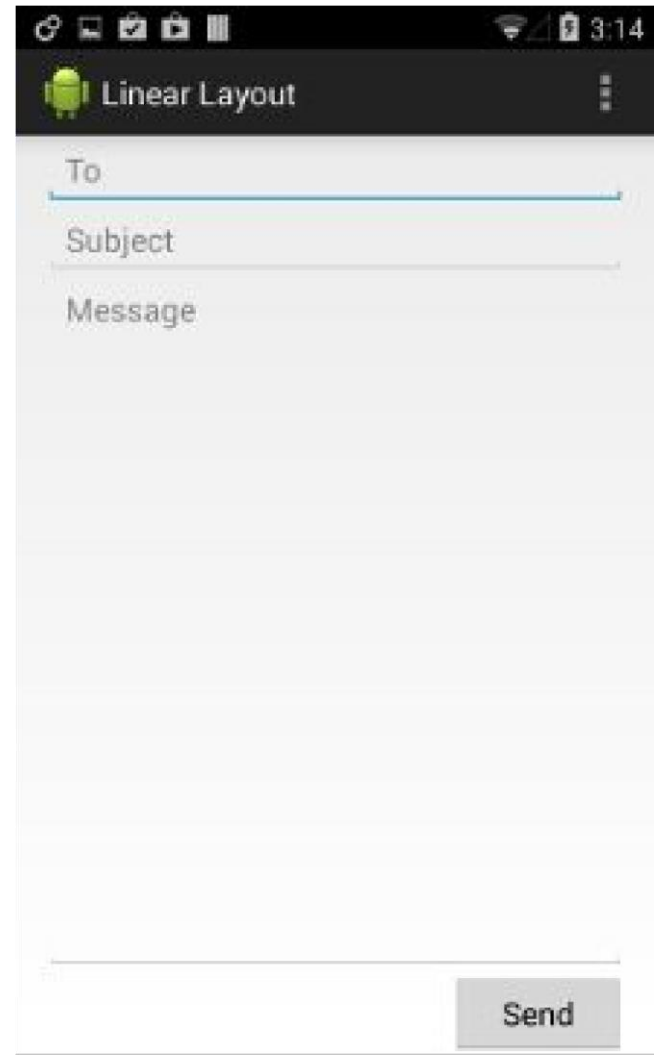
- Faster to specify attribute names instead of searching
- Company-specific abstractions = less value from design preview

ConstraintLayout

- Modern default layout system, intended to create powerful and flat view hierarchies
- Views are laid out according to relationships (or **constraints**) between sibling views and the parent layout.

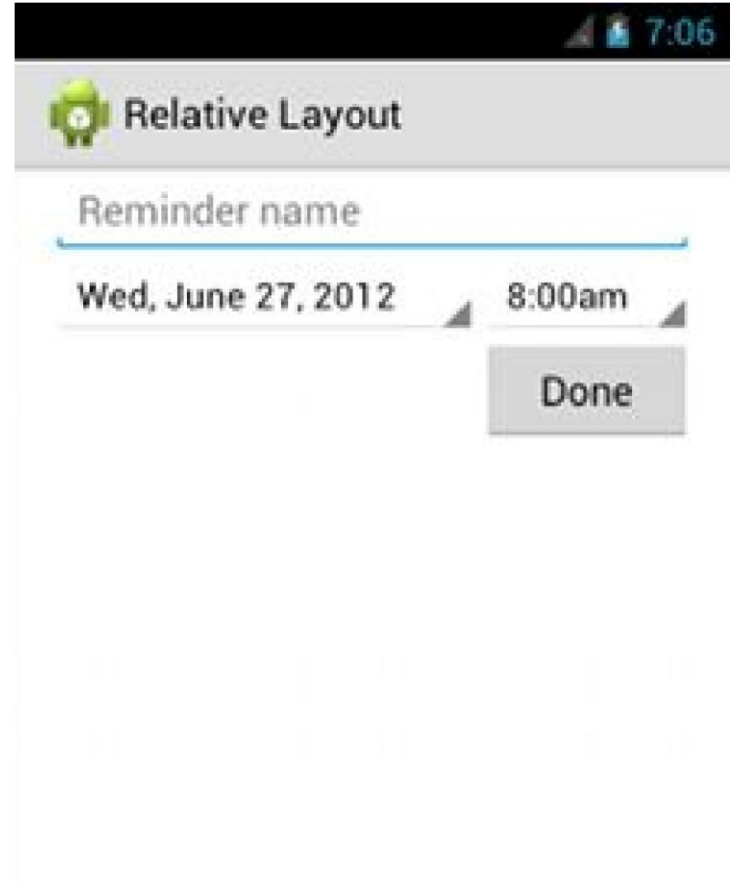
LinearLayout

Aligns all children in a single direction,
vertically or horizontally

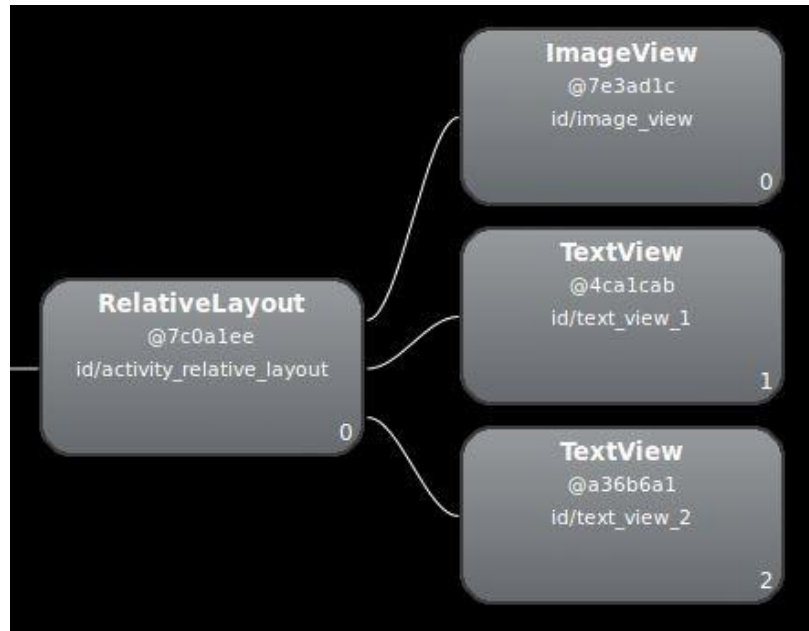
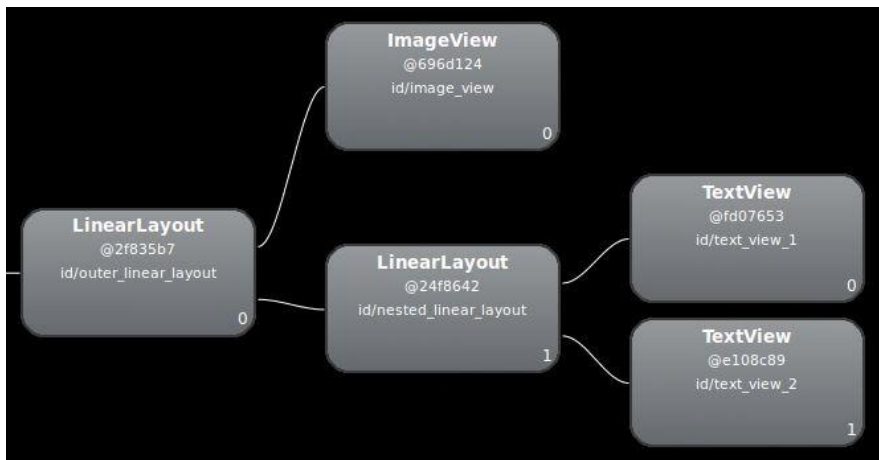
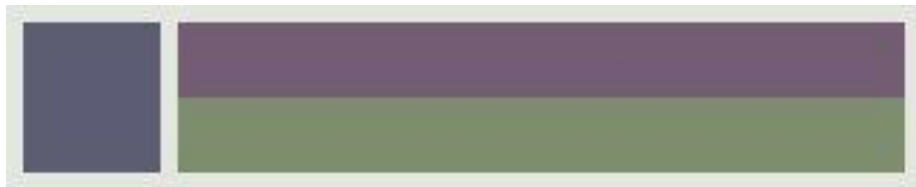


RelativeLayout

Displays child views in relative positions



Flat hierarchies



Prep for next week

- Build the Tip Calculator
- If you're stuck for more than 30 minutes, post on Ed or come to office hours