# Android Programing 101

Dr. Charles Yu

# Outlines

- Introductions and Android Studio
- General Android App UI setup
- Activities and its jumping
- Fragment

# Why Android Studio?

- Officially Supported / Suggested by Google for Apps development
- If you are a Python lover, you should know PyCharm?
- Android Studio <-> PyCharm (looks very similar?)
- Developed by JetBrains

# Why Android Studio + Java?

- You need to deal with tons of archive code
- Most of archive code for Android App are written in Java
  - Your 1st job in the company:
  - Maintain the old archived code
  - We still need to survive in our 1st year. Isn’t it?

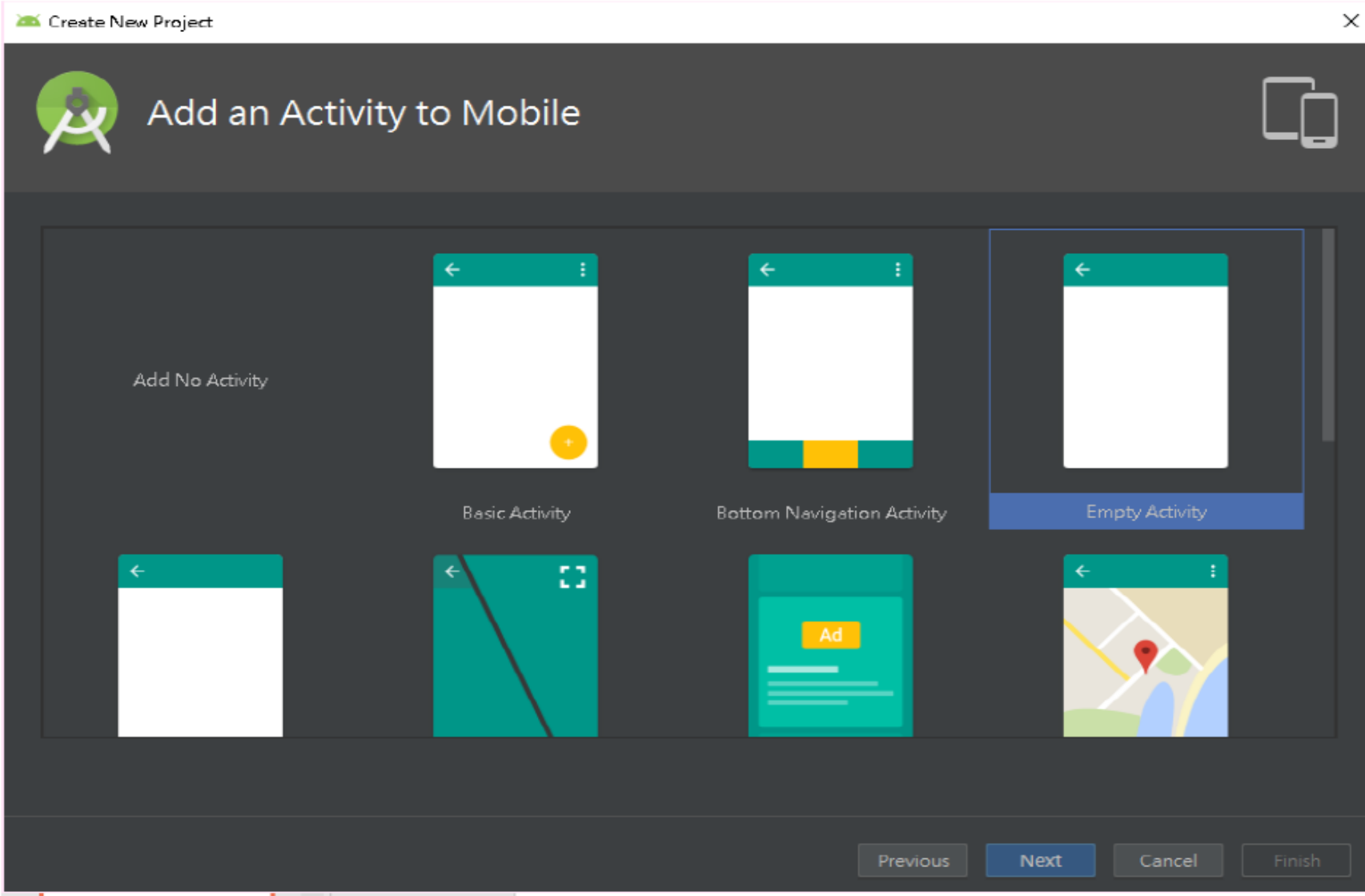
# Why Android Studio + Java?

- Kotlin? (like Scala or Ruby). A good choice
- This is a trend: more and more new projects are written in Kotlin
- We can take advantage on new features of Android by using Kotlin
- Their new demo code in android developer's website are written in Kotlin
- One of my friend, she just returned from FB as a summer intern lady.
  - We use Kotlin in new project development!

# First time, “new” a project, Entry Activity Selection

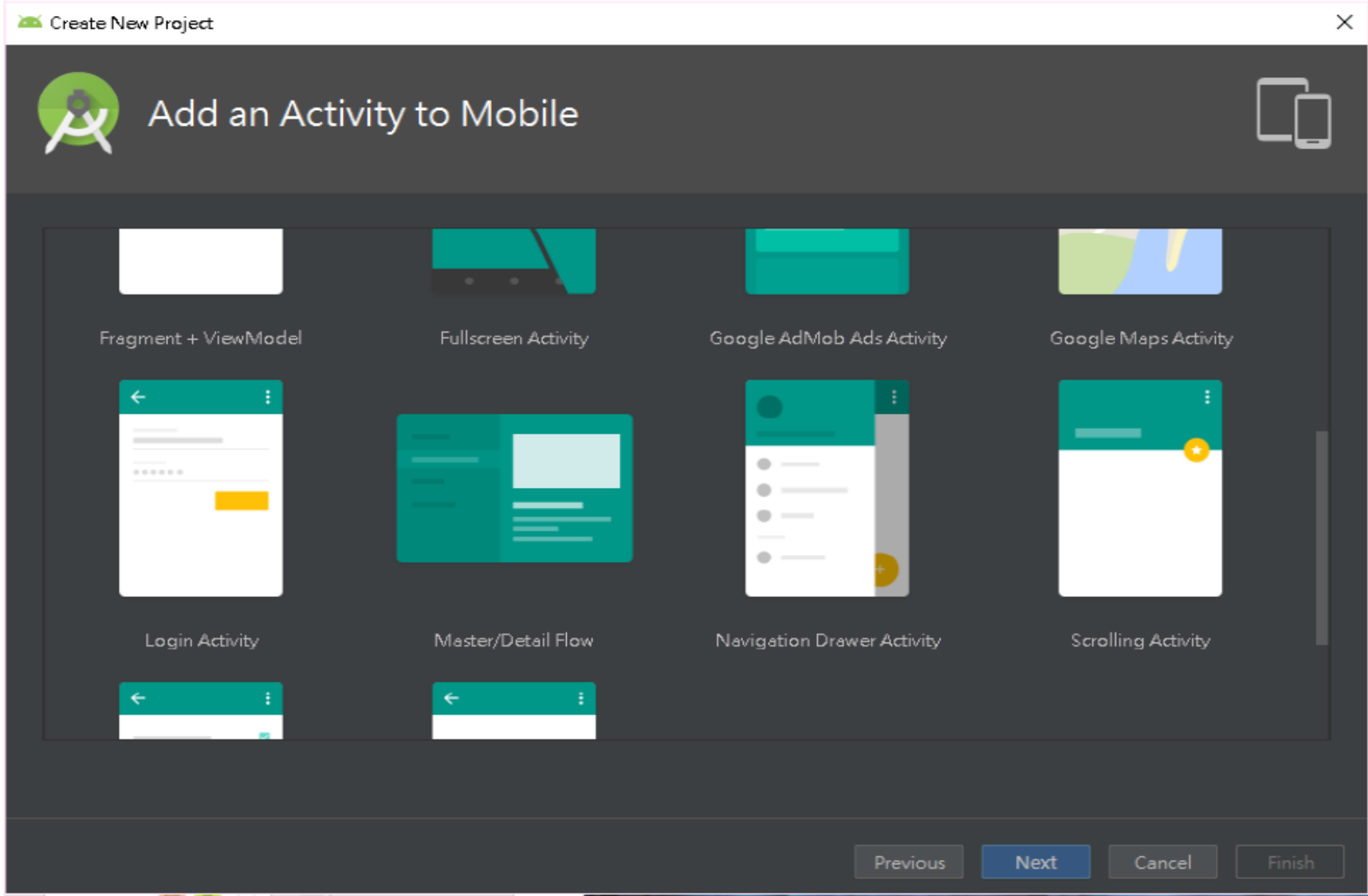
- Empty activity is my “best favorite”
- If you choose the Basic Activity, you can see the “+” button on the corner after it launches

(go the next page…)



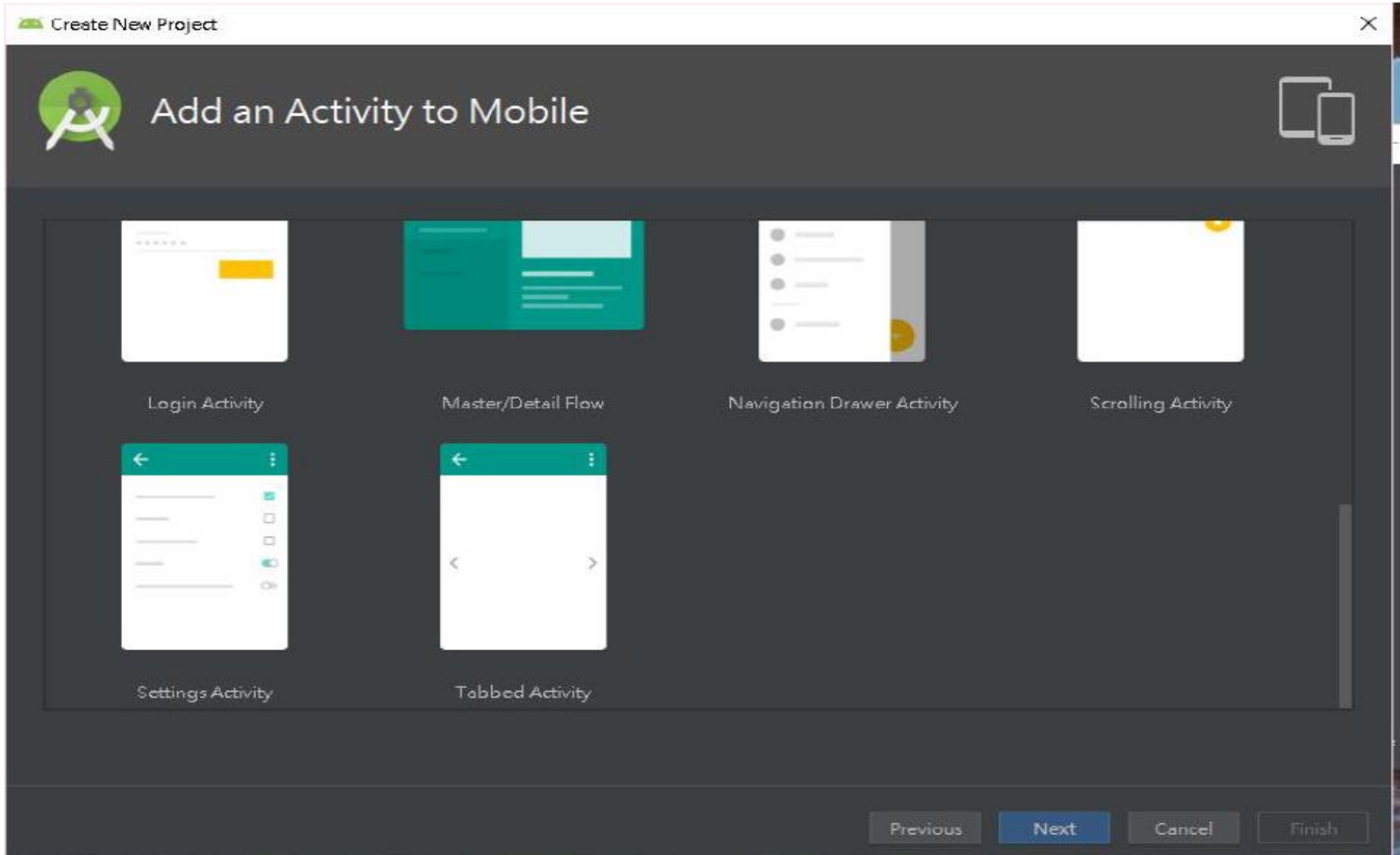
# First time, “new” a project, Entry Activity Selection

- **Login** Activity: They “dig” everything ready for you (buttons, text views…)
- **Navigation** Drawer: In your 1st (entry) activity, you will have the navigation bar. You can swipe it out from the LHS
- **Scrolling** is similar to news reader app. Two parts are separated but the bottom part is scrollable



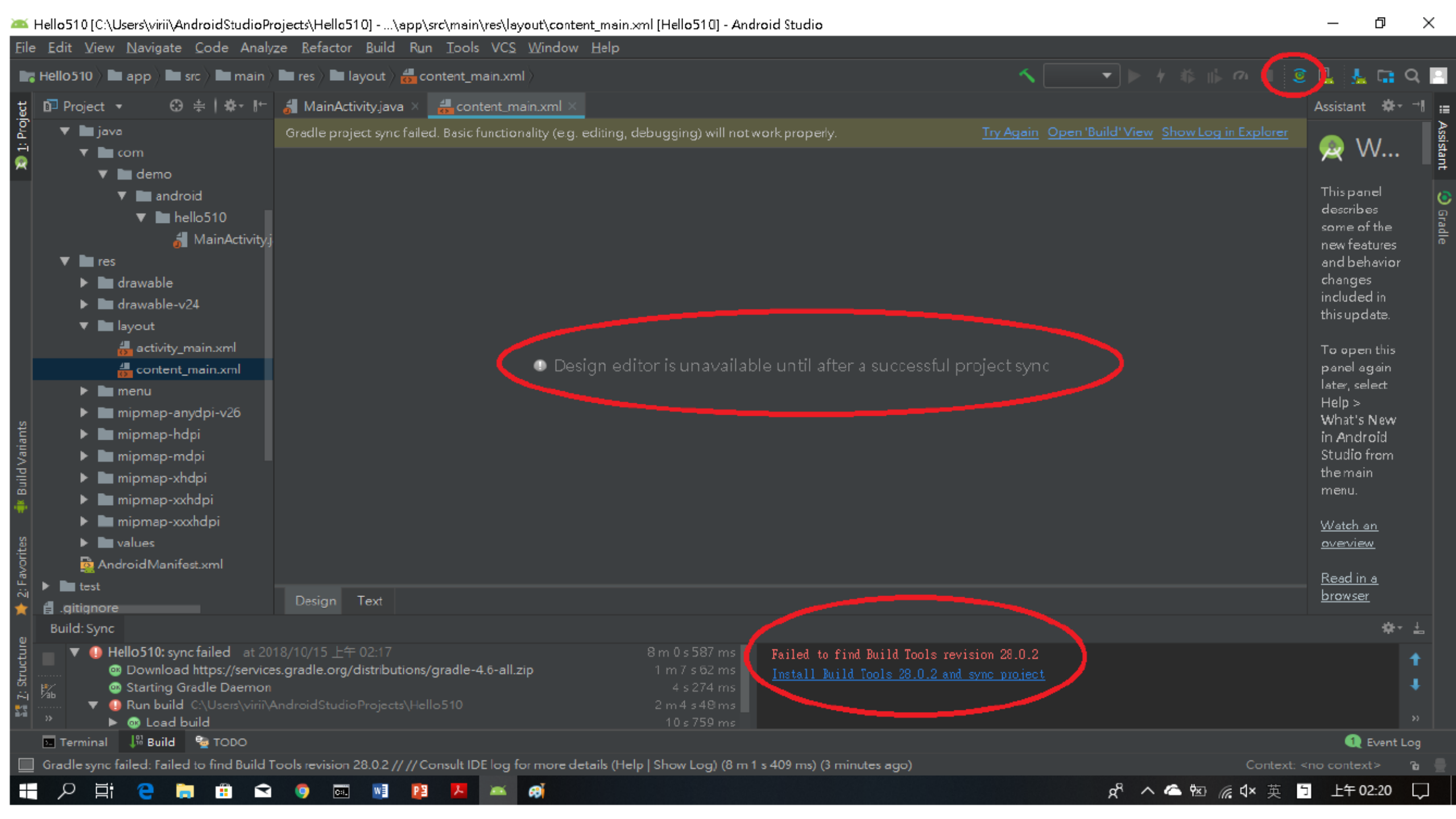
# First time, “new” a project, Entry Activity Selection

- **Settings** Activity: Making something setup related UI
- **Tabbed** Activity: Like the 1st time launch, welcoming messages (tips), this tab → next tab →…



# Why my Android Studio has so many error messages?

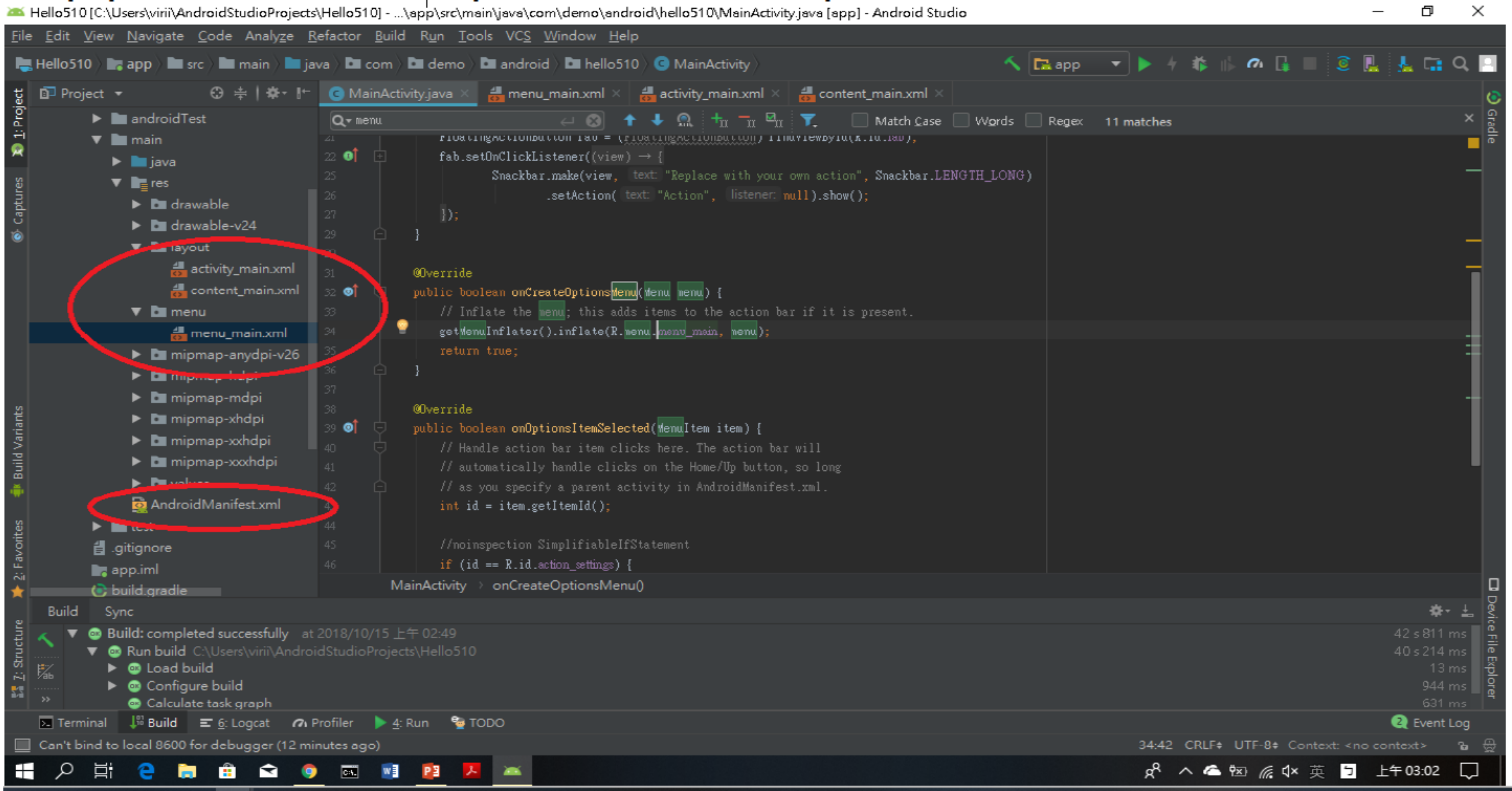
- You need to refresh, every time, after you installed something or synced new libraries from the cloud


Hello510 [C:\Users\virii\AndroidStudioProjects\Hello510] - ...\app\src\main\res\layout\content_main.xml [Hello510] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Hello510 app src main res layout content_main.xml
Project
MainActivity.java
content_main.xml
Gradle project sync failed. Basic functionality (e.g. editing, debugging) will not work properly.
Try Again
Open 'Build' View
Show Log in Explorer
Assistant
java
com
demo
android
hello510
MainActivity
res
drawable
drawable-v24
layout
activity_main.xml
content_main.xml
menu
mipmap-anydpi-v26
mipmap-hdpi
mipmap-mdpi
mipmap-xhdpi
mipmap-xxhdpi
mipmap-xxxhdpi
values
AndroidManifest.xml
test
.gitignore
Design editor is unavailable until after a successful project sync
This panel describes some of the new features and behavior changes included in this update.
To open this panel again later, select Help > What's New in Android Studio from the main menu.
Watch an overview
Read in a browser
Design
Text
Build: Sync
Hello510: sync failed at 2018/10/15 上午 02:17 8 m 0 s 587 ms
Download https://services.gradle.org/distributions/gradle-4.6-all.zip 1 m 7 s 62 ms
Starting Gradle Daemon 4 s 274 ms
Run build C:\Users\virii\AndroidStudioProjects\Hello510 2 m 4 s 48 ms
Load build 10 s 759 ms
Failed to find Build Tools revision 28.0.2
Install Build Tools 28.0.2 and sync project
Terminal
Build
TODO
Event Log
Gradle sync failed: Failed to find Build Tools revision 28.0.2 /// Consult IDE log for more details (Help | Show Log) (8 m 1 s 409 ms) (3 minutes ago)
Context: <no context>
上午 02:20

# What if my Android Studio is very abnormal?

- Note that my version might be old. You need to take care of your version
- How to fix it? (Windows, for example)
  - Kill everything underneath:
  - C:\Users\[user_name]\.AndroidStudio3.2
- Restart Android Studio

# UI setup (previous 2 xml) and App wide setup



# UI setup and App wide setup

- Couple of xml files
  - AndroidManifest.xml
    - In src/main
- activity_main.xml
  - In res/layout
- content_main.xml
  - In res/layout
- menu_main.xml
  - In res/menu

- There is only **AndroidManifest.xml** about **application wide software components setup**. Other xml(s) are about UI!

# UI setup and App wide setup

- UI and the calling sequence
- R.layout.activity_main get called in onCreate()
- In activity_main.xml, it **includes** content_main.xml (**You won't see that if you choose the Empty Activity**)
- menu_main was get called in onCreateOptionsMenu()
- That's take a look in MainActivity.java, as an entry point to your App and try to find out all of these xml files

# UI setup and App wide setup

- AndroidManifest.xml
  - You can clearly see how many software components in this app
  - Activity(ies), Service(s),…
- Another purpose is to setup access rights
  - After Android 6.0, you **not only** have to **setup** access rights in **AndroidManifest.xml but also** need to **ask** App **user** to **grant** the hardware or system resources in the running time.
  - Hardware or system resources include camera, location (GPS), sensors, file system,…, etc
  - (See the next page, our AndroidManifest in MySQLDemo2022)

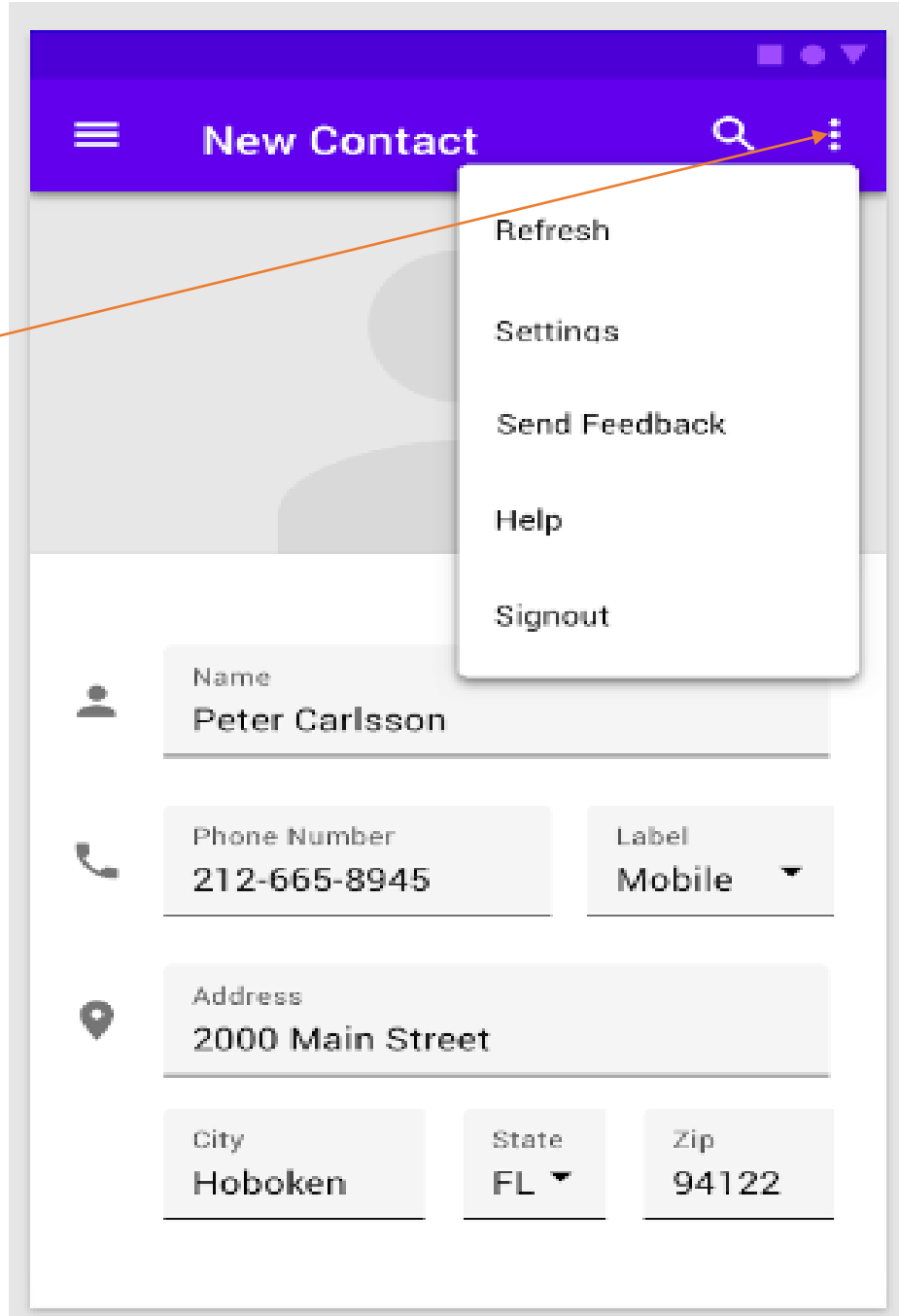
# UI setup and App wide setup

- See? It is asking access rights!

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.mysqldemo2022">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MySQLDemo2022"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

# UI setup and
# App wide setup

- What is the purpose of menu_main?
- When the dot-dot-dot is clicked,
  - its job is to setup how does that look like?
  - I can guess, in this app (from someone's website), it only put 5 items in the xml file



# UI setup and
# App wide setup

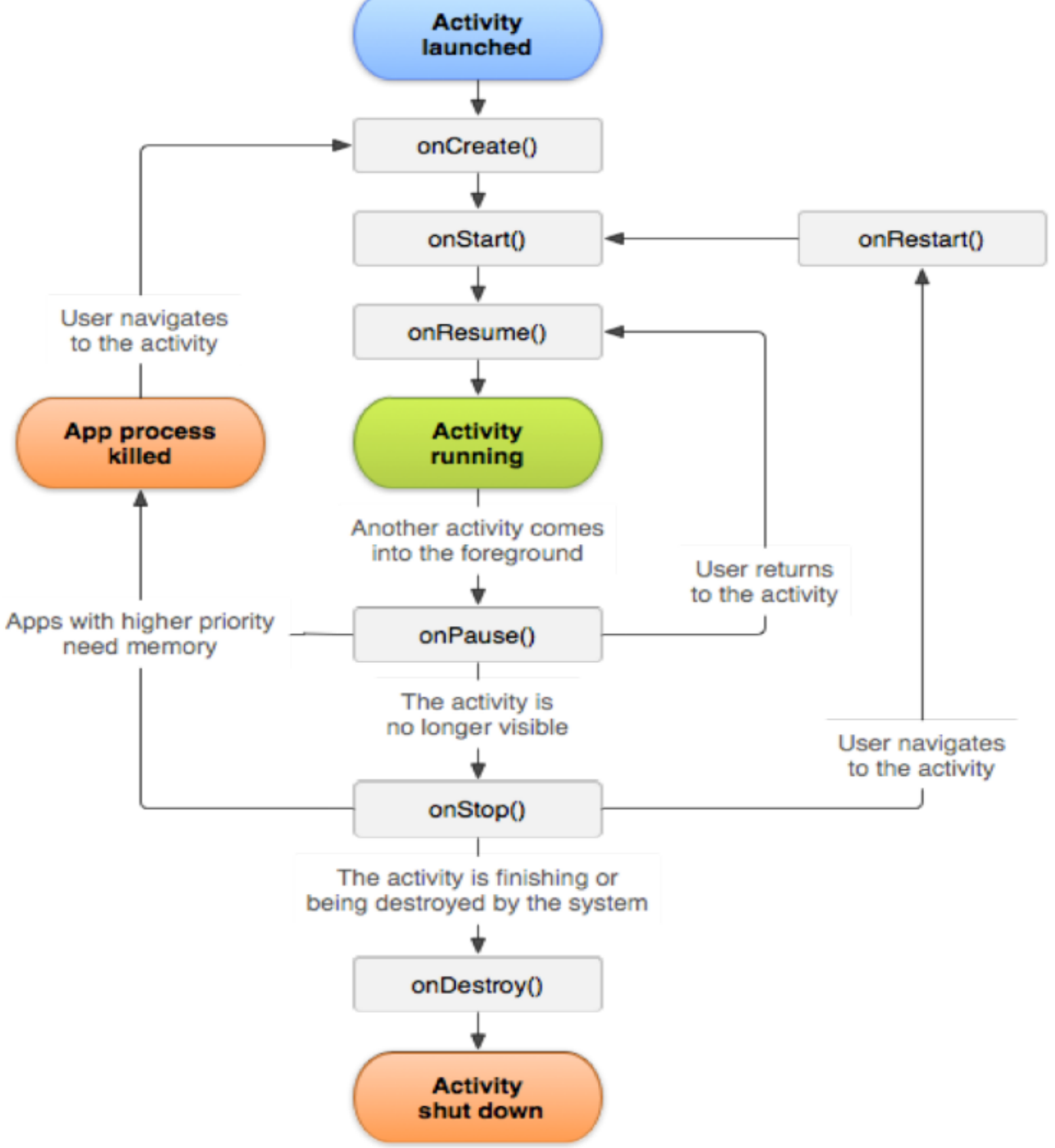
- In conclusion of Android UI:
  - Event driven code implementation when the UI got clicked
  - i.e. It is possible that, I can swipe a tab and upload a file to the server.

# Activity

- Your 1st App will start with dealing with “Activity” (newer versions of API will bring you a Fragment)
- For each of the “page”, the unit is called “Activity”. This is true before the “Fragment” comes into the world
- How to see Activity’s documentation?
  - Well formatted in Java’s style
  - https://developer.android.com/reference/android/app/Activity

# Activity

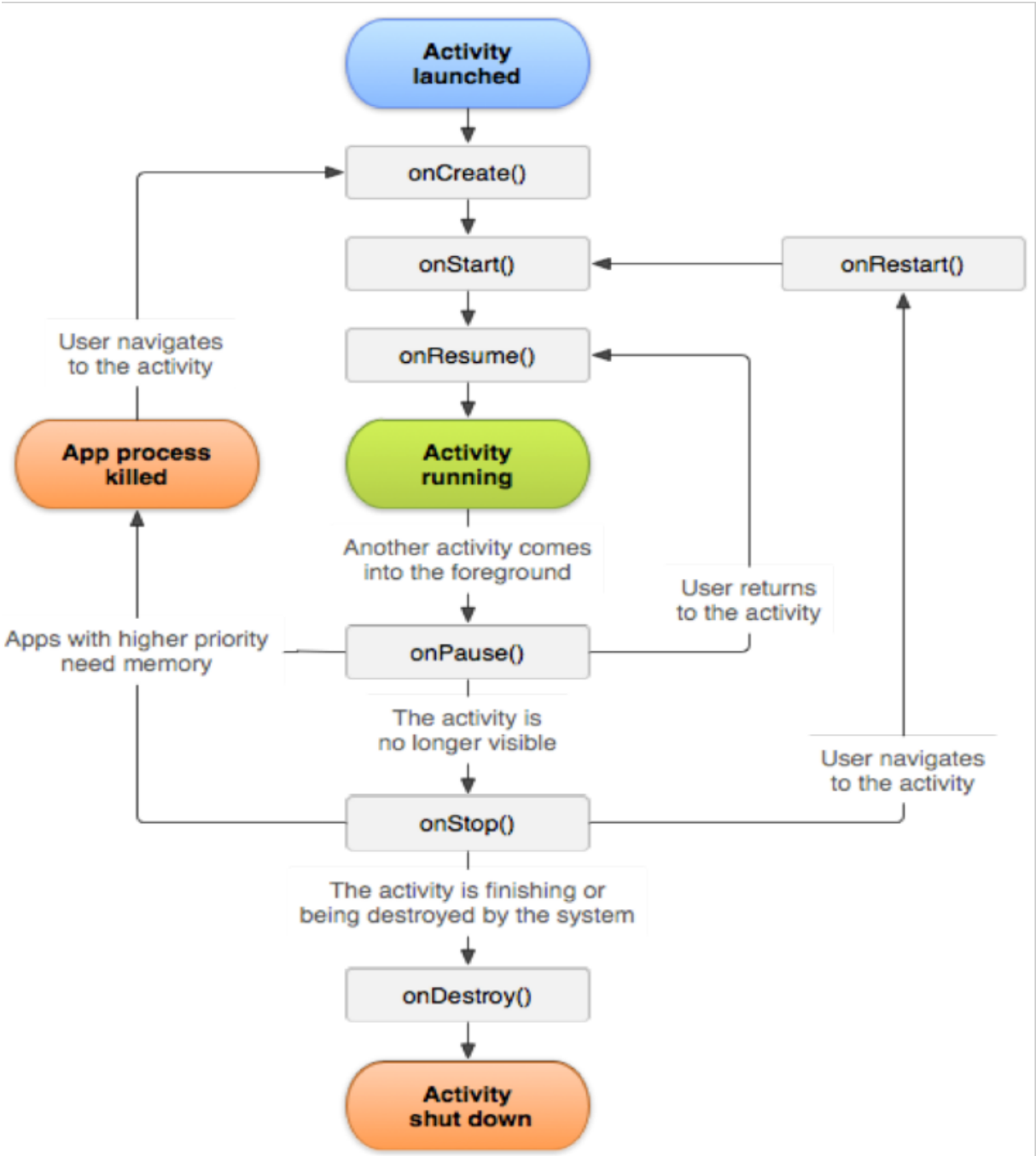
- The activity does have its lifecycle!
- Get UI ready in onCreate(). This is a one-time setup (most cases)
- If an Activity enters onResume(), It can re-run the core jobs again. For example, the logic when you "return" to the App, you want to check something.
    - i.e. internet is still available?



# Activity

- **onPause** and **onStop** will get executed by pressing [**Back**] key and [**Home**] key respectively!

- Do the right thing, in the right time
- If you are expecting the user is about to exit this app, for example, going back to the home screen and won't see this app for a while, the pre-occupied system resources has to be released by unregistering  them
  - i.e. Cameras, Sensors



# Activity

- [Demo1] Activity Jumps

# Fragment

- What is Fragment?
- A Parasite?
- A fragment must always be hosted in an Activity and the Fragment's lifecycle is

directly affected by the host Activity's lifecycle

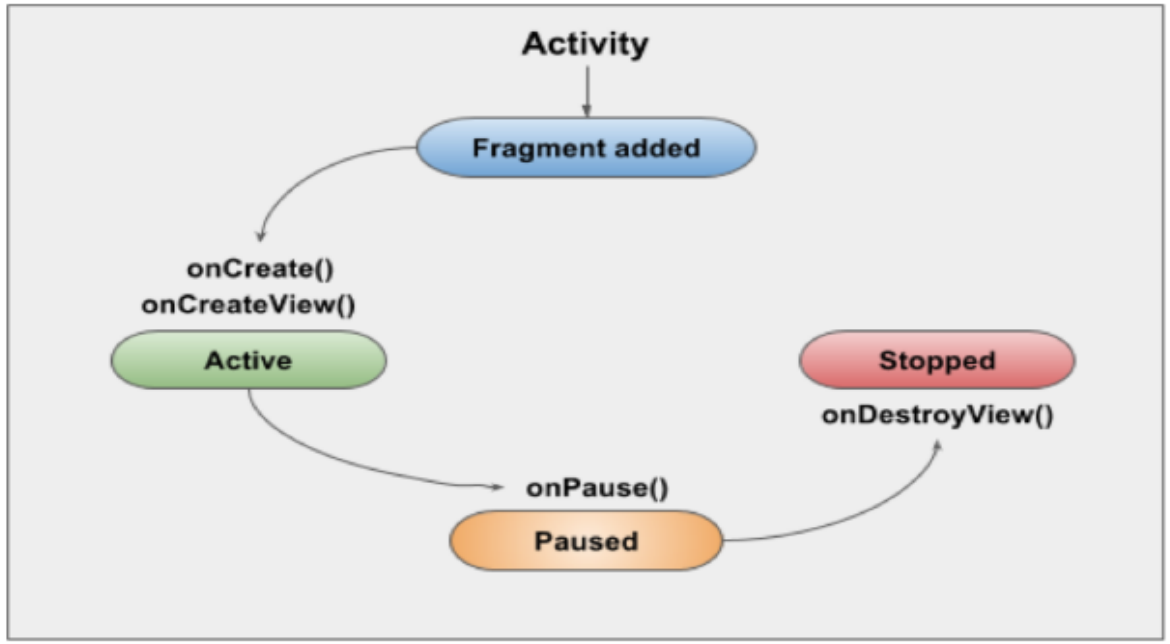
- For example, when the Activity is paused, so are all the Fragments in it, and when the Activity is destroyed, so are all the Fragments.

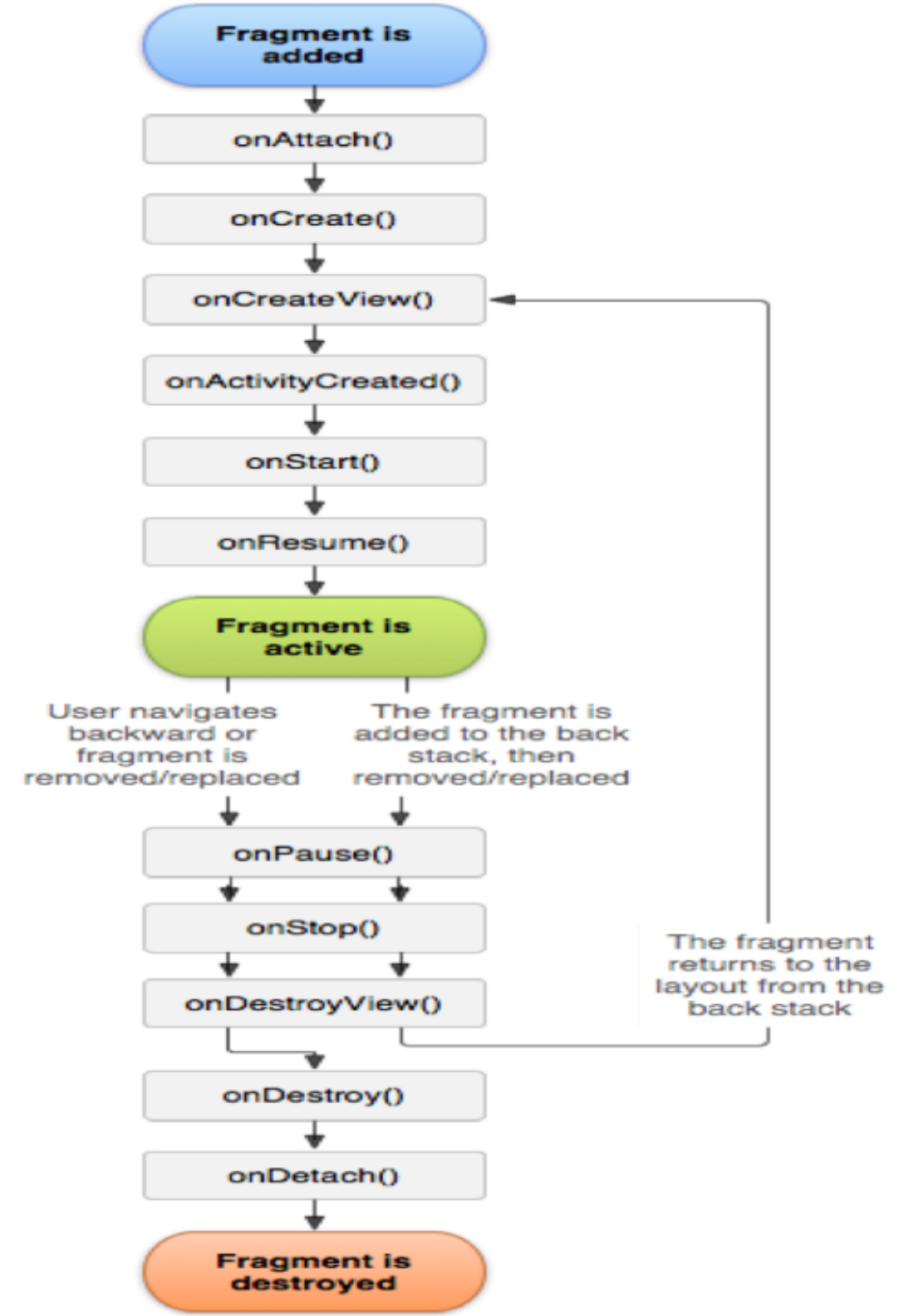
# Fragment

- Why we need to have Fragments?
  - Light weight
  - Faster
  - Better user experience
  - Fragments VS. Activity, any differences?
    - I will show you later.
    - They can be different, visually!

# Fragment Lifecycle

Macro (high-level) view and Micro View



# Fragment

- [Demo2] Activity and Fragment

Note that, in the AboutMe.java, there is no dot-dot-dot. It is because I do not do the setup like MainActivity. You know what I mean and please check the slide #17