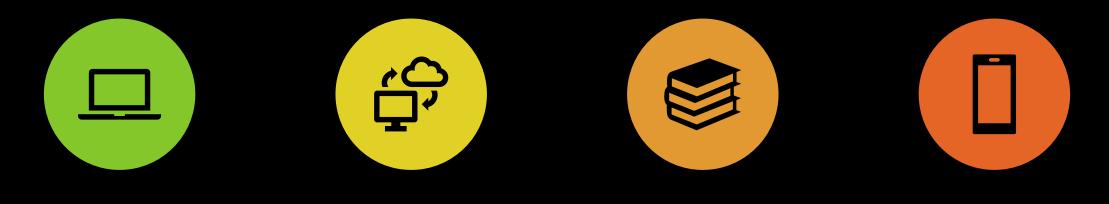
# ANDROID STUDIO PURE ANDROID

#### REQUIREMENTS



LAPTOP & CHARGER

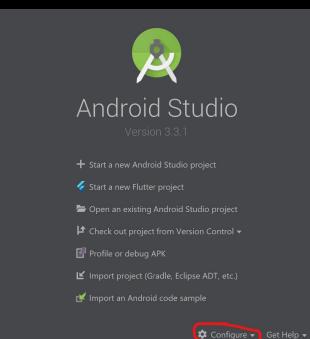
ANDROID STUDIO INSTALLED KNOWLEDGE OF JAVA (PREFERRED, NOT REQUIRED) ANDROID PHONE OR EMULATOR SETUP

### DEVICES

#### Emulator or Android Phone Setup

#### EMULATOR

- 1. Start Android Studio, click Configure (Tools if Project is open)
- 2. Open AVD Manager and click Create Virtual Device
- 3. Create based on any device (Default is fine) and click Next



Select Hardware									
Choose a devic	Choose a device definition								
	Q								
Category	Name 🔻	Play Store		Resolution	Density	C Nexus 5X			
TV									
Phone						1080px	Ì I	Size: large Ratio: long Density: 420dpi	
				480x800		5.2"	1920px		
				480x800					
		۵		1080x1920					
	New Hardware Profile Import Hardware Profiles Clone Device								
							lext		

#### EMULATOR CONTINUED

#### 4. Select any version of Android and click Next

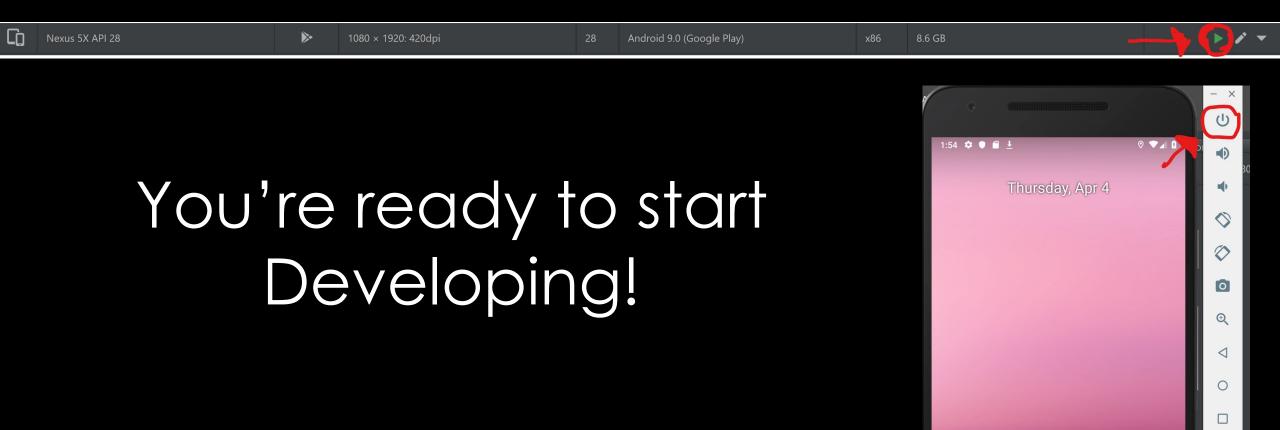
- If you have an older version, it should be fine
- 5. Verify Settings and Finish

System Image Android Studio						
Select a system image						
Recommended x86 Images	Other Images	ABI	Target	Pie		
<b>Q</b> Download						
Pie	28		Android 9.0 (Google Play)	× *	28	
Oreo Download					20	
Oreo Download						
Nougat Download					9.0	
Nougat Download					Google Inc.	
					x86	
				We recommend these G device is compatible with	oogle Play images because this n Google Play.	
See the API level distribution chart						
?					ext Cancel Finish	



#### EMULATOR CONTINUED

#### 6. Start your Emulator and Turn it on using Power Button



#### PHYSICAL DEVICE DEVELOPER OPTIONS

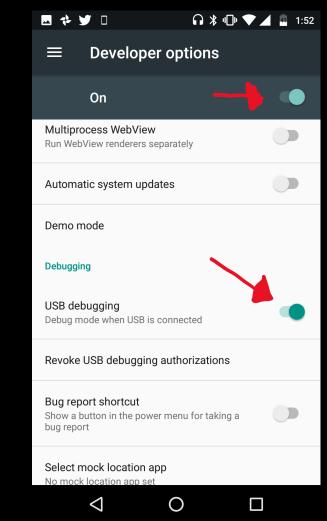
	<b>Y</b> 0	🞧 🎗 🕕 💎 🚄 📱 1:50
	Setti	ngs Q
one	S	Date & time GMT-05:00 Central Daylight Time
	Ť	Accessibility
veloper	ē	<b>Printing</b> 0 print jobs
	{}	Developer options
	٩	System UI Tuner
	5	Legal information
	i	About phone Android 7.0

- 1. Go to Settings
- 2. Scroll to down to bottom and select About Phone
- 3. Under about Phone, find Build Number
  - Placement can very, you can try searching for it
  - Tap that ~7 times.
  - You should get a message about becoming a develope

#### DEVELOPER OPTIONS CONTINUED

- 4. Go back to the Main Settings Page
- 5. Select Developer Options, Toggle top switch to On
- 6. Scroll slowly to the section on Debugging
  - Toggle USB Debugging to On
- 7. Connect Device to Computer
- 8. If necessary, allow USB Debugging

You're ready to start Developing!



# ANDROID STUDIO

#### GETTING STARTED

- 1. Create a new Empty Activity Project, click Next
- 2. Call the app Countdown Timer and click Finish

Choose your project			
Phone and Tablet Wear OS T			
	¢ :	¢	¢
		Empty Activity	Bottom Navigation Activity
			Google Maps Activity
Empty Activity Creates a new empty activity			
			Next Cancel Finish

Configure your project		
÷	com.lkbrough.countdowntimer	
	Save location	
	C:\Users\laury\AndroidStudioProjects\CountdownTimer	
	Language	
	Java	
	Minimum API level	
	<ul> <li>Your app will run on approximately 100% of devices.</li> <li>Help me choose</li> </ul>	

# ANDROID BASICS -BACKGROUND

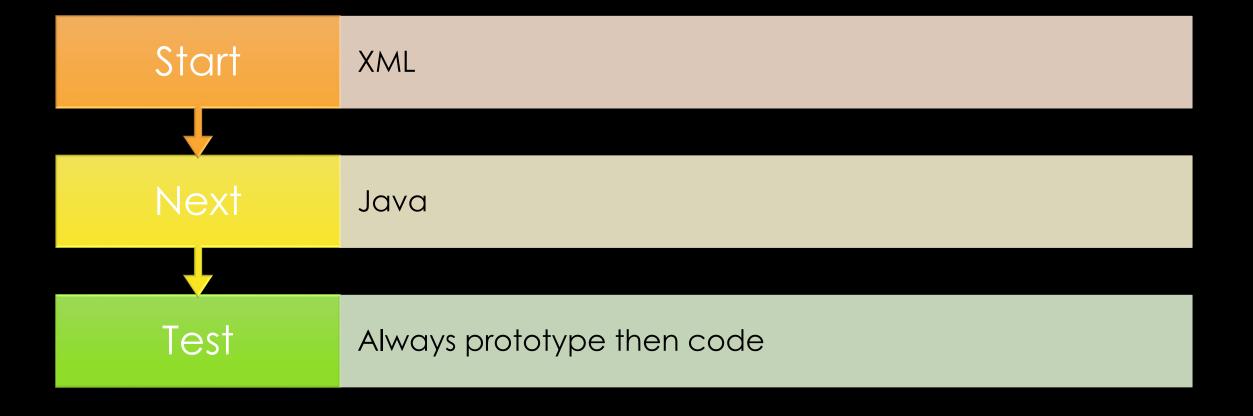
- Android Code is comprised of two main files
  - XML File (Layout/Display)
  - Java File (Code)
- XML can be done with either the GUI or by Text/Code
  - I prefer Text, but for this we will be using GUI since it's quicker
  - If you want to use the Text option, you can



## BACKGROUND CONTINUED

- Java is the main part of your program
  - Connects to XML
  - Can create parts of the XML
- In Java files, refer to the app outside of a java file as R
  - i.e. R.layout.textViewName
    - Refers to an object in the XML file with the id of textViewName
  - Useful for updating layouts with new information

#### PROJECT: CREATE A COUNTDOWN TIMER



### XML

- Things you'll need
  - EditText for Time
  - Button to Start Timer
  - TextView for Timer Display
  - Optional: Button to Stop Timer
  - Optional: Progress Bar
- ALWAYS ID your objects in XML
  - Allows access in Java files

	Attributes	Q ,
	ID	timeInput
4	layout_width	p_content 💌 …
	layout_height	p_content 💌 …
		•
		<b>•</b>
	<b>.</b> >	» × < ⊕
		<u>^</u>
		<b></b>
Time in Seconds		
START TIMER	▼ EditText	
START TIMER	inputType	number
TimerDisplay	hint	
······	style	at.EditText 💌 …
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	singleLine	
STOP TIMER	selectAllOnFocus	
	▼ TextView	
	text	Time in Seconds
	🗲 text	
	contentDescripti	
	► textAppearanc	Material 🔹

Τi

#### XML - CONSTRAINTS

- Time in Seconds Time in Seconds Time in Seconds START TIMER STARTTIME imeDispla TimerDisplay Timer merDisplay rogressBa J<sub>x</sub> ab ∫× STOP TIMER STOP STOP TIME
- Describes position on screen
- Drag arrows from each side of a box to define constraints to different edges
  - Easiest to have some sort of base in the very center of the screen
- Can be defined by spacing against sides or by spacing from another object

#### JAVA & XML

- Right now, we have a button that does nothing. Let's change that!
- Write a method to retrieve data inputted into our EditText and send it to the timer
- Note: Anytime that a method is being called using a button, the button passes the view that it is (the button itself). Account for that in the parameters
  - startTimer(View view)

#### ONCREATE

- All android activities start with an onCreate method
  - Sets up display based on the attached XML file
- You can add other startup pieces onto it like we will now.
- In the Class, create a variable for a TextView for your display
- In the onCreate, save our display to it by finding it by id.

```
public class MainActivity extends AppCompatActivity {
  TextView display;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    display = (TextView) findViewById(R.id.timerDisplay);
}
```

### STARTTIMER (VIEW VIEW)

- <u>https://we.tl/t-tC92pmB9fW</u>
- Copy file using File Explorer and Paste into Android Studio (Project/Java/Main)
- We need a timer, text to display/update, and the data from our textview
- Retrieve Data from EditText
  - Save the Edit Text Field
    - EditText userInputField = (EditText) findViewById(R.id.timeInput);
  - Save the text from it
    - String input = userInputField.getText().toString();
  - Parse your string to an int

#### CREATE A TIMER

- Create a new timer
  - Class has been provided! Feel free to look through this file at one point as there are a lot of cool things going on (updating the view)
  - UpdatingCountDownTimer(int time, int CountDownInterval, TextView display, ProgressBar progress)
- CountDownInterval should simply be 1 for 1 second as we want the timer to update every second.
- If you're not using the progress bar (which none should be yet) pass it as a null
- After the timer is created, start the timer by calling it's start method.

#### STARTTIMER

```
public void startTimer(View view){
  EditText yes = (EditText) findViewById(R.id.UserInput);
  String input = yes.getText().toString();
  int time = Integer.parseInt(input);
  UpdatingCountDownTimer timer = new UpdatingCountDownTimer(time, 1, display, null);
  timer.start();
```

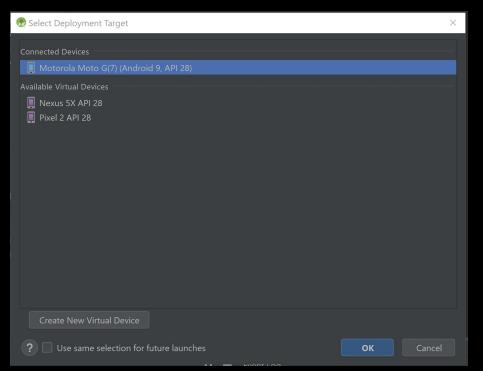
#### ADD THE METHOD TO THE BUTTON

- Switch back to the XML Document and Click on the Button
- In the Attributes panel, you'll see an onClick after a little bit of scrolling
- As the onClick attribute, type in our startTimer method (without parameters!)

#### RUN

- 1. Run the app with the play button in the top right hand corner
- 2. Select your device
  - Could be emulator or physical device, which are split into two different sections





#### ERRORS?

- Raise your hand if you have errors
- Talk to the person next to you
- Read through the error

#### PROJECT FINISHED

- Optional: Make it Pretty
- Optional: Add progress bar
- Optional: Add stop timer button
- Optional: Add another screen

#### BEAUTIFY

- Use the style, colors, and other xml files in the res/values directory of your android app
  - Defining Style, not applying
- Apply the style in the Android Manifest

#### PROGRESS BAR

- UpdatingCountDownTimer was written to support progress bars
- Add a progress bar similar to how you added other elements and save it similar to how we saved the TextView

#### STOP BUTTON

- Another button, another method (similar to start)
- Save current time, destroy previous timer, modify start timer
  - Next time user presses start timer, start based on your saved time.

#### PROBLEMS? COMMENTS? IMPROVEMENTS?

Email: <u>lauryn.brough@hotmail.com</u>

Project: <a href="https://github.com/lkbrough/CountDownTimer">https://github.com/lkbrough/CountDownTimer</a>

Hope everyone enjoyed this workshop, learned some very basic android, and enjoys the rest of the Hackathon!