This document will show our process for phase 2 of the language project. Insert your screenshots and change any text as needed. We are pretty much explaining how to get an IDE / compiler working on a system.

# TYPES OF INSTALLS FOR KOTLIN

There are two main ways to start developing in Kotlin: with an IDE or using a text editor and standalone compiler. The compiler can be used with any editor once installed. Most IDE’s that support Java will support some type of Kotlin plugin. Most notably Eclipse, Android Studio, and IntelliJ IDEA.

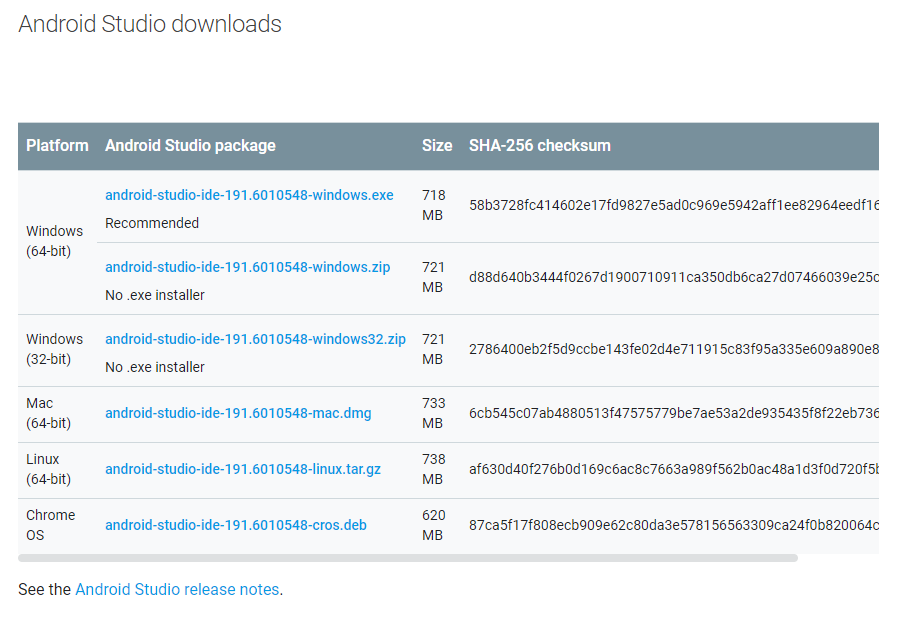
* Eclipse
  + https://marketplace.eclipse.org/content/kotlin-plugin-eclipse
* Android Studio
  + As of Android Studio v 3, Kotlin comes bundled with the IDE.
* IntelliJ
  + <https://plugins.jetbrains.com/plugin/6954-kotlin>
* Popular Editor Kotlin support
  + Vs code
    - <https://marketplace.visualstudio.com/items?itemName=fwcd.kotlin>
  + Atom
    - <https://atom.io/packages/language-kotlin>
  + Notepad ++
    - <https://www.kotlinresources.com/library/nppextension/>
  + Sublime
    - <https://packagecontrol.io/packages/Kotlin>

# PROCESS OF ANDROID Studio Installation

# Android Studio is a versatile IDE based on the IntelliJ IDE and fully supports Kotlin. Android Studio allows you to create Kotlin projects or convert existing Java code into Kotlin. Android Studio provides code completion, debugging tools, and code inspection through Lint.

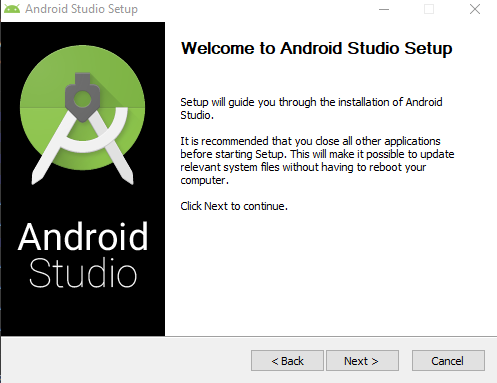
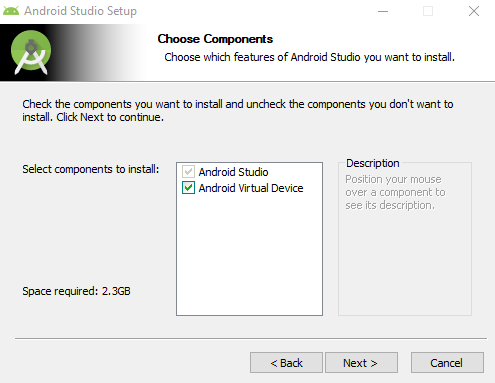
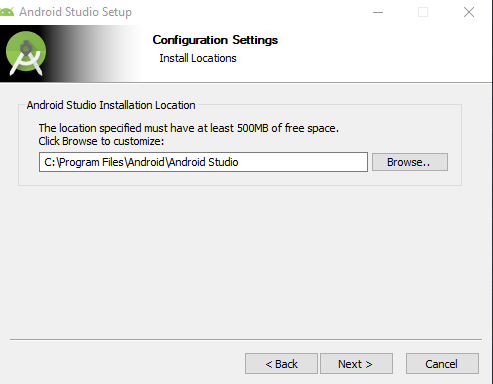
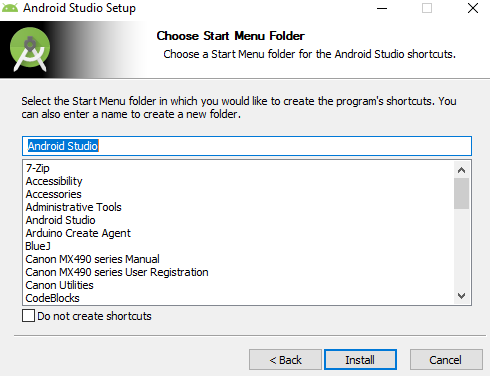
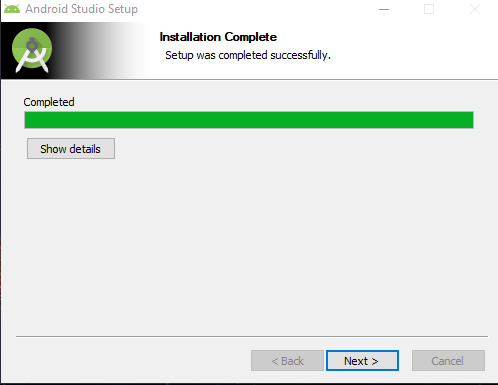
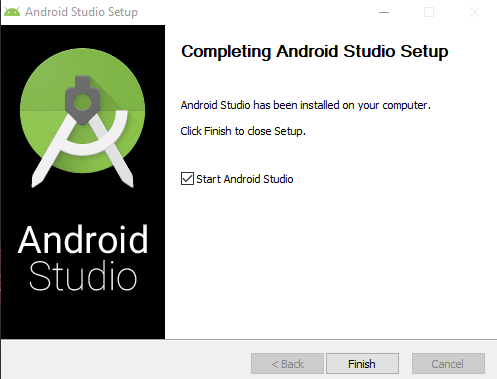
Step 1-Download:

Navigate to <https://developer.android.com/studio> and scroll down to the section labeled Android Studio Downloads. Download the appropriate files according to your operating system. If you are utilizing the windows operating system you have 2 possible installation methods. You can utilize the .exe installer for Windows 64-bit. If you are utilizing windows 32-bit or prefer to, you can utilize the .zip download. Mac 64-bit is provided with a .dmg download. Linux 64-bit is provided with a .tar.gz download.

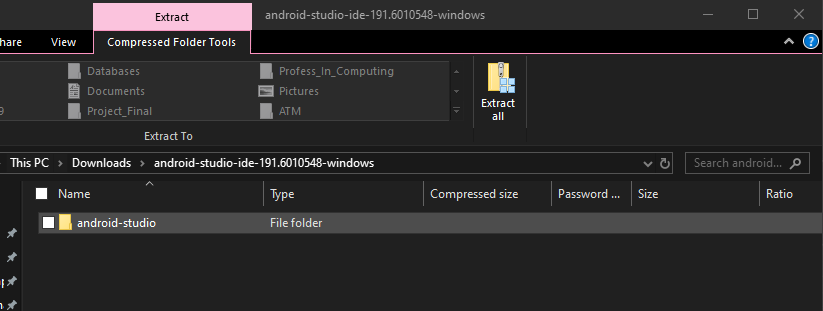
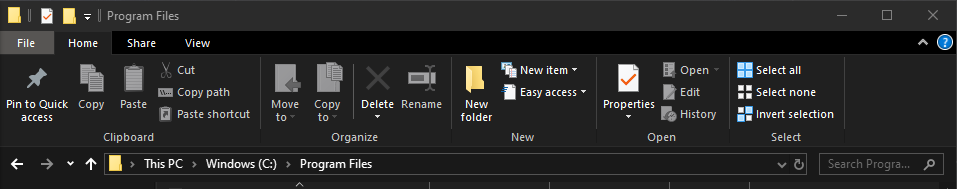
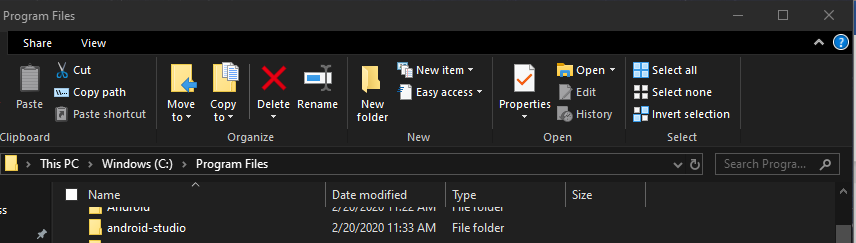
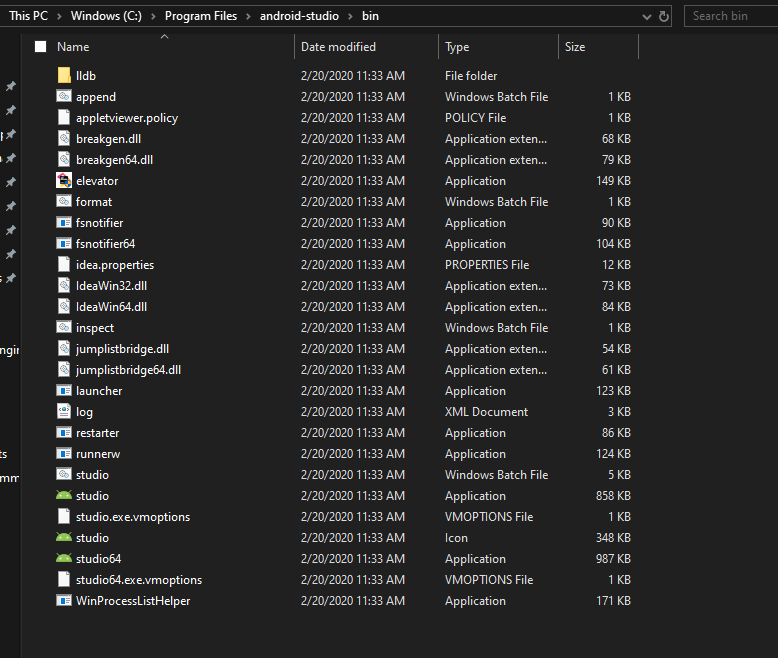
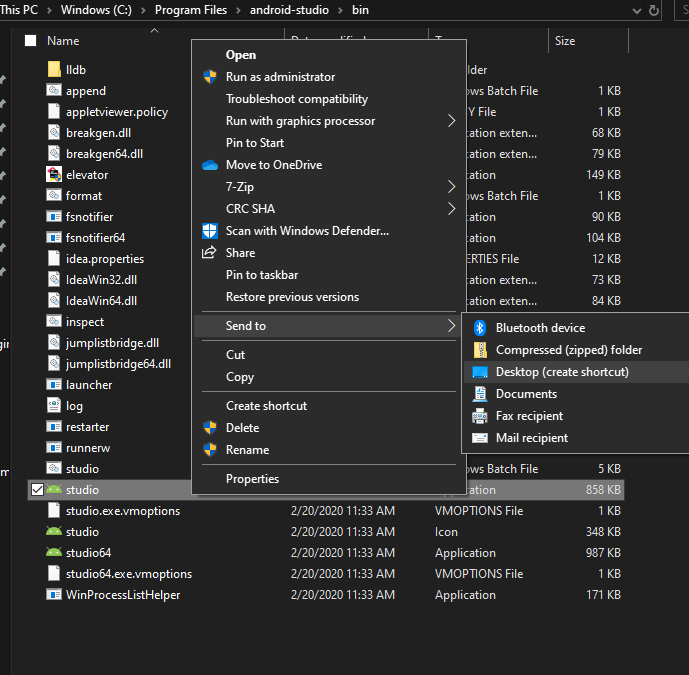


Step 2-Installation:

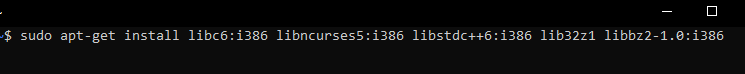
Windows 64-bit .exe installation:

1. Launch andriod-studio-ide.exe
2. Follow the Android Studio Setup wizard instructions
   1. Click Next
   2. Make sure both Android Studio and Android Virtual Device are selected and click next.  
      
   3. Check that your Android Studio Installation Location is the drive and folder you wish Android Studio to be installed and click next.  
      
   4. Click install  
      
   5. Once installation is complete you will see the below screen. Click next.  
      
   6. Click finished as Android Studio is now installed.   
      

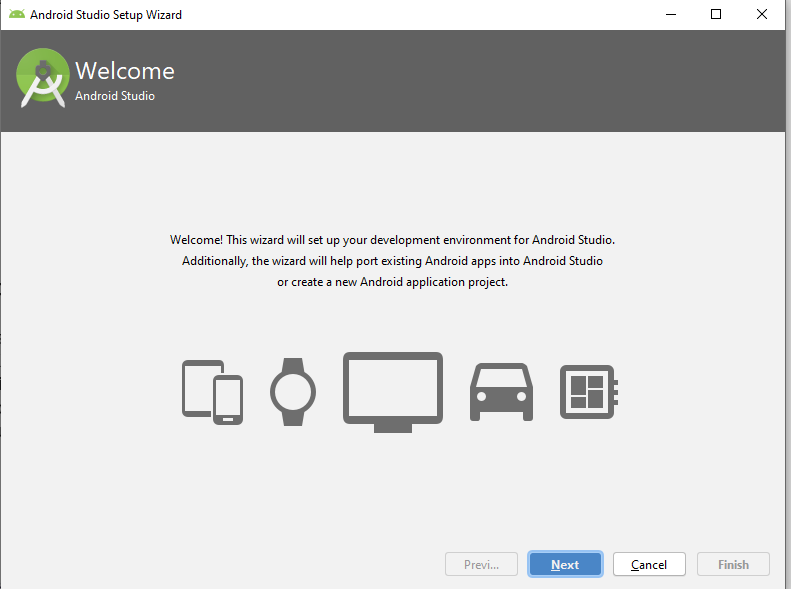
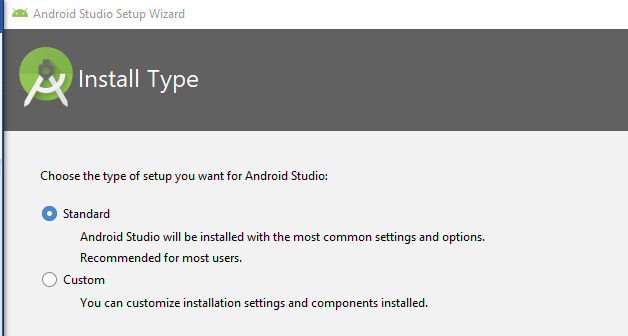
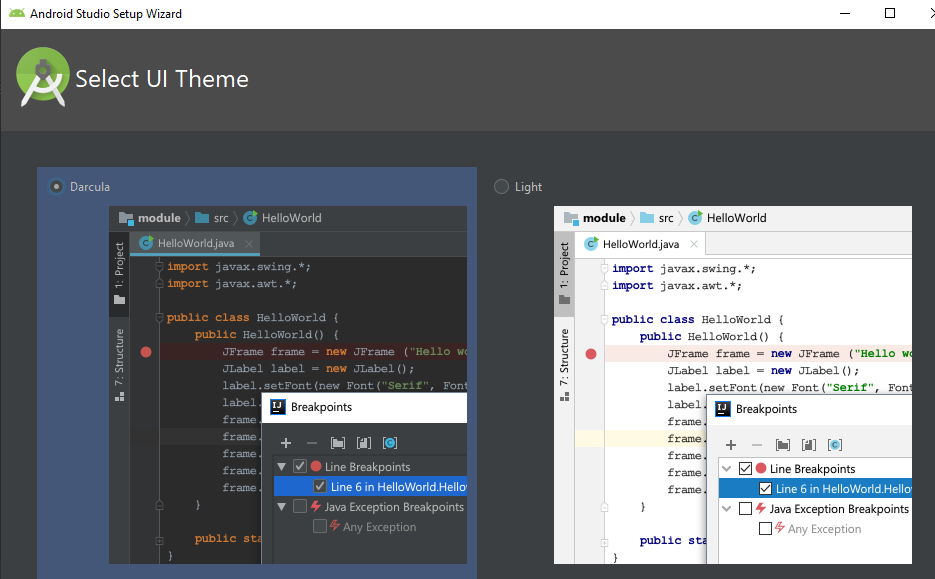
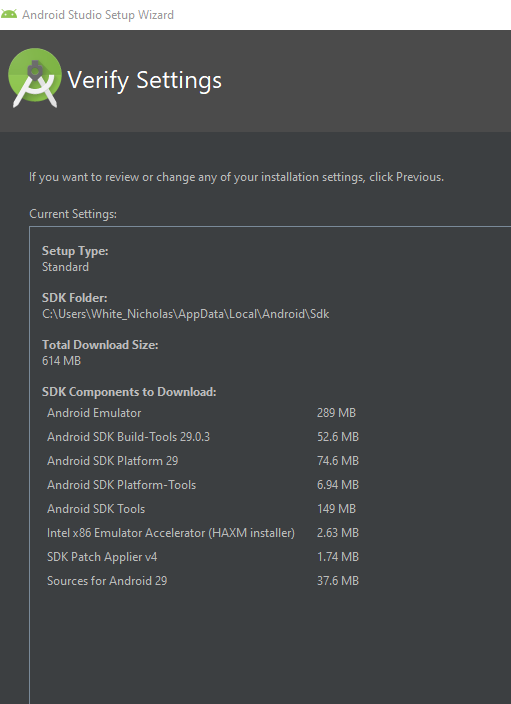
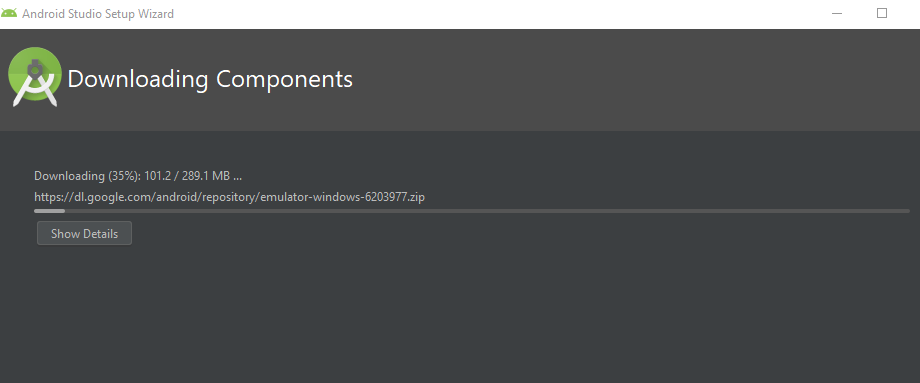
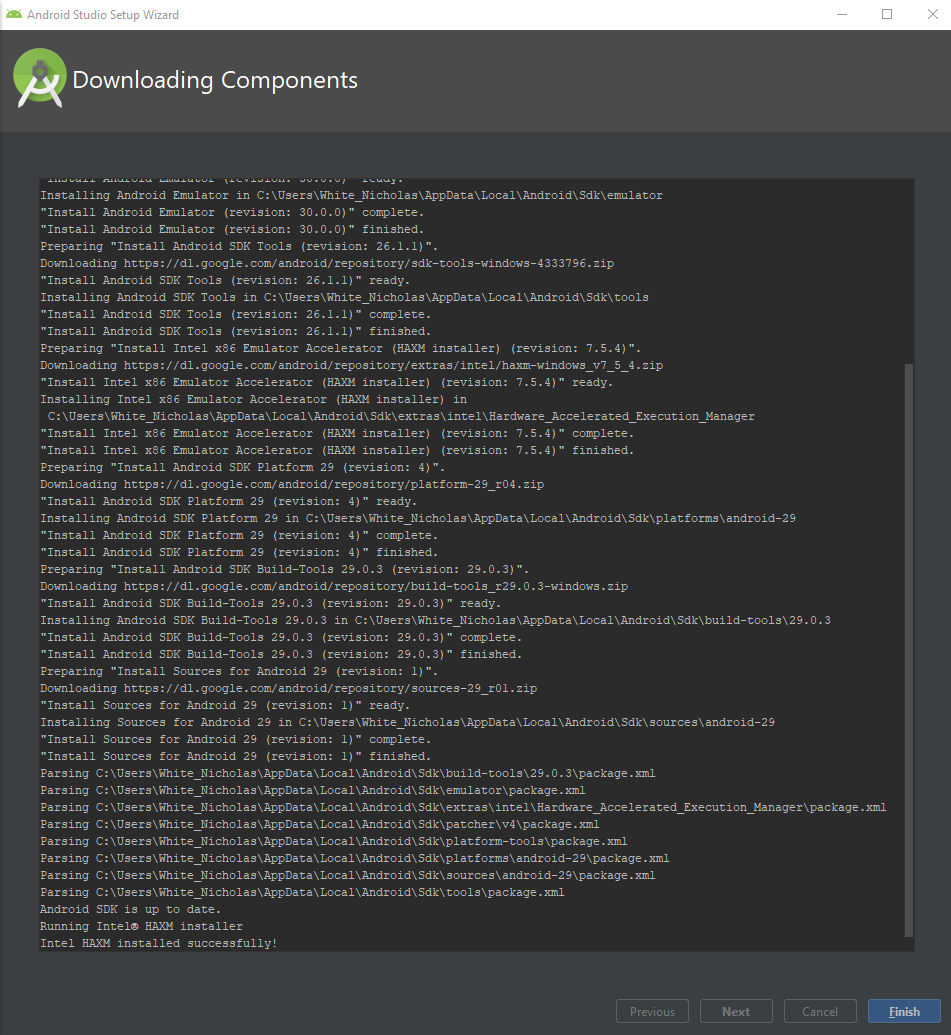
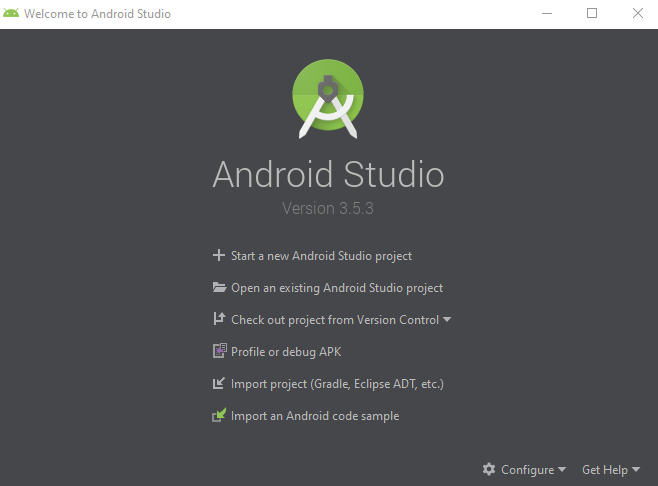
Windows 32-bit or Widows 64-bit .zip installation:

1. Locate android-studio-ide.zip file  
   
2. Open file with an application that can extract files from a zip file such as 7-zip or WinRaR   
   
3. In a second Windows Explorer window navigate to Program Files under the desired installation drive Ex. C:\Program Files  
   
4. Next drag and copy android-studio file into Program File  
   
5. Next navigate into the android-studio file. Locate and navigate into the bin file.   
   
6. Lastly create a shortcut to desktop. For Windows 32-bit utilize studio.exe. For Window 64-bit utilize the studio64.exe. You can then launch the application from your desktop short cut.   
     
   

Linux Installation guide:

1. Unpack the zip into your /usr/local/ or /opt/ file location.
   1. For 64-bit Linux you will need to install the required libraries.   
      Using the command:  
      
2. To launch you will navigate to the installation location and follow the path android-studio/bin/. You will then execute the studio.sh
3. You will then be prompted to select whether you will import Android Studio setting or not. Click ok. Then continue to setup instructions

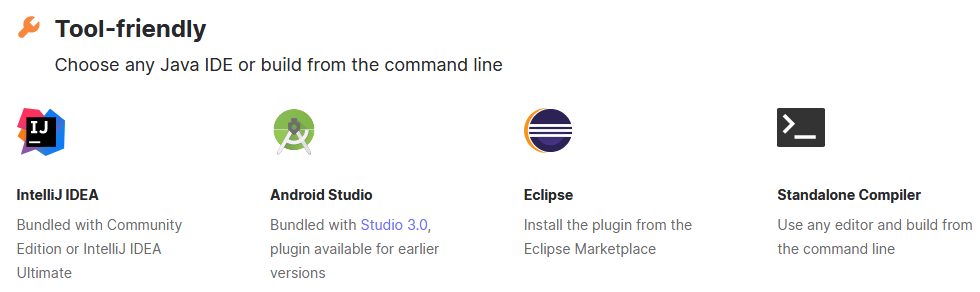
Step 3-Setup:

1. You will be prompted by the Android Studio Setup Wizard on your first execution. Click next.  
   
2. Select Standard for the Install Type. Click Next.   
   
3. Select your preferred UI Theme. Click next.   
   
4. Preview settings if correct click finish.  
   
5. Android Studio will then download the appropriate components.  
   
6. Once complete you will receive the below log of the installation process. Click finish.   
   
7. You are now ready to start a new Android Studio project.  
   

# PROCESS FOR TEXT EDITOR AND COMPILER

Preface: The environment that I am using for this installation process is WSL using the Ubuntu

Kernel. This process should look the same on any UNIX/OSX Platform. I will also being using VS code as my editor, but you can use any editor you wish. The process of install a plugin for Kotlin on the editor might change slightly. Note, this act of installing a Kotlin plugin is completely optional. You can still compile and run code without it, but it does make everything look way nicer in the editor.

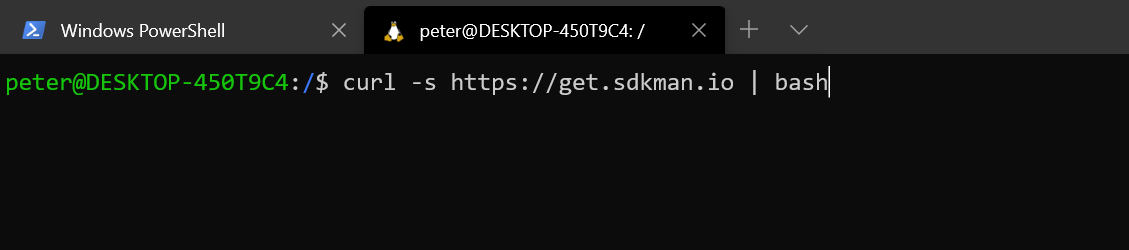


STEP 1: install a package manager if needed

There are a few packages that will install the comelier for you. ‘Homebew’ for mac os, snap if you are Ubuntu, and ‘SDKMAN!’. Because I am using WSL we will be using ‘SDKMAN!’, because ‘Snap’ is not currently working for WSL at the moment (Feb 2020).

Once you are at the command line, enter the following commands

‘ curl -s https://get.sdkman.io | bash ‘

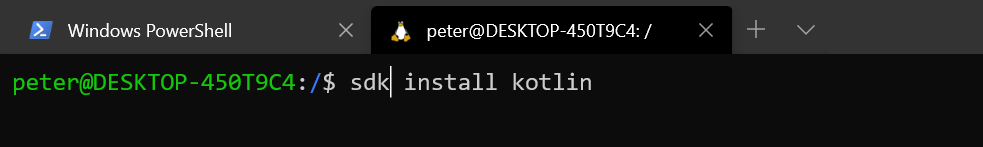


You might be instructed at this time to install ‘Unzip’ if you do not currently have this package on your system. To do this simply run ‘Sudo apt-get install unzip’. After that runs, make sure it was installed using ‘unzip -v’. Then run the curl command again and sdk man should be installed.

STEP 2: install the compiler

run the following command to use SDK to install Kotlin

‘ sdk install kotlin’



After the install is done, check that Kotlin was installed correctly using

‘ kotlin -version ‘

STEP3: Create your first Kotlin program

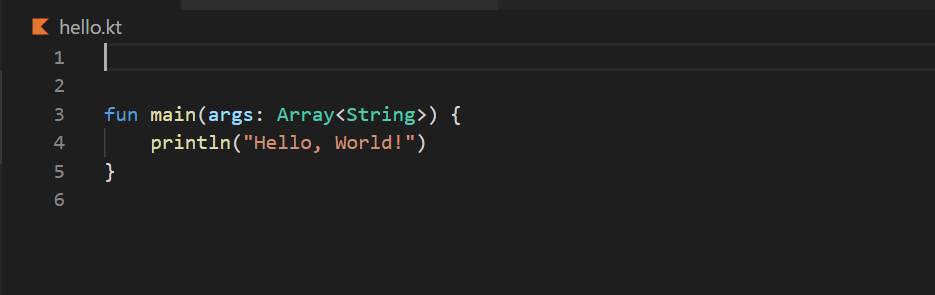
We are going to create a simple ‘hello world’ program just to make sure everything was set up correctly.

Create a new file called ‘hello.kt’ and put the following code into that file

‘ fun main(args: Array<String>) {

println("Hello, World!")

}’



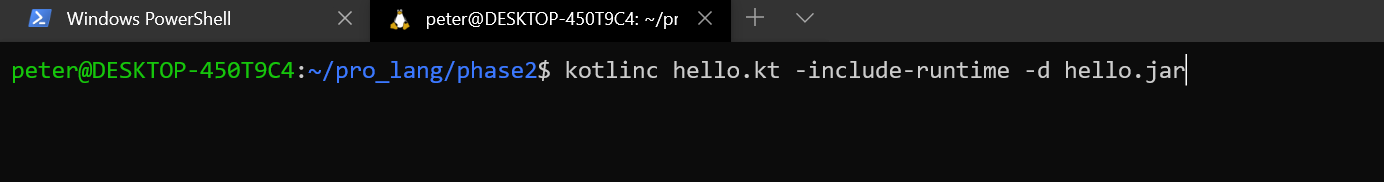
This will simply just print hello world out to the screen. Make sure to save the file before leaving the editor!!!

STEP4: compile your program

Now we are going to use some commands to compile this file into a working Kotlin program.

Use the following command to compile the file

‘ kotlinc hello.kt -include-runtime -d hello.jar ‘



What this does is it compiles our ‘hello.kt’ file and turns it into a ‘hello.jar’ executable file. The ‘include runtime’ option just the executable ‘self-contained’ by adding the java runtime library too it. The -d option just allows us to put a name for the executable file.

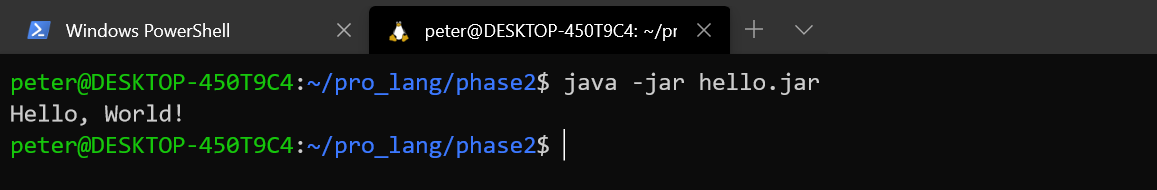
STEP5: run the application

Now we need to run the executable we created to make sure it works properly.

Enter the following command

‘ java -jar hello.jar ’

This will run our ‘hello.jar’ file that was created when we compiled our file in step 4



STEP6: celebrate, you are now a Kotlin programmer!

And they say programming is hard! Granted, this was a very simple program but it shows us how to create and run Kotlin programs using the command line instead of an IDE. To recap what we did in case you forgot, we used a packaged to install the compiler, we then created a simple hello world program file, we then compiled that file to turn it into an executable, and finally we ran the executable.