

Mobile Application Development

Lesson 2

Dr. Syed Asim Jalal

Department of Computer Science

University of Peshawar

Prerequisites

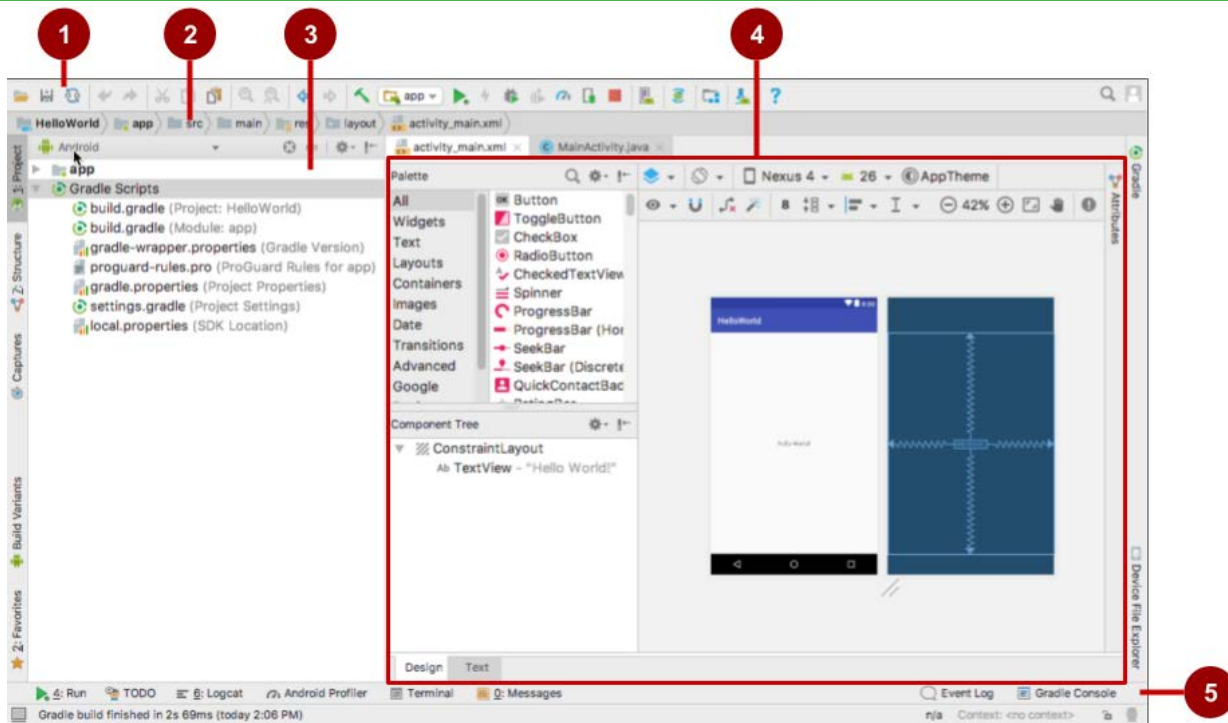
- Java Programming Language
- Object-oriented programming
- XML - properties / attributes
- Using an IDE for development and debugging

Android Studio

What is Android Studio?

- Android integrated development environment (IDE)
- Project and Activity templates
- Layout editor
- Testing tools
- Gradle-based build
- Log console and debugger
- Emulators

Android Studio interface



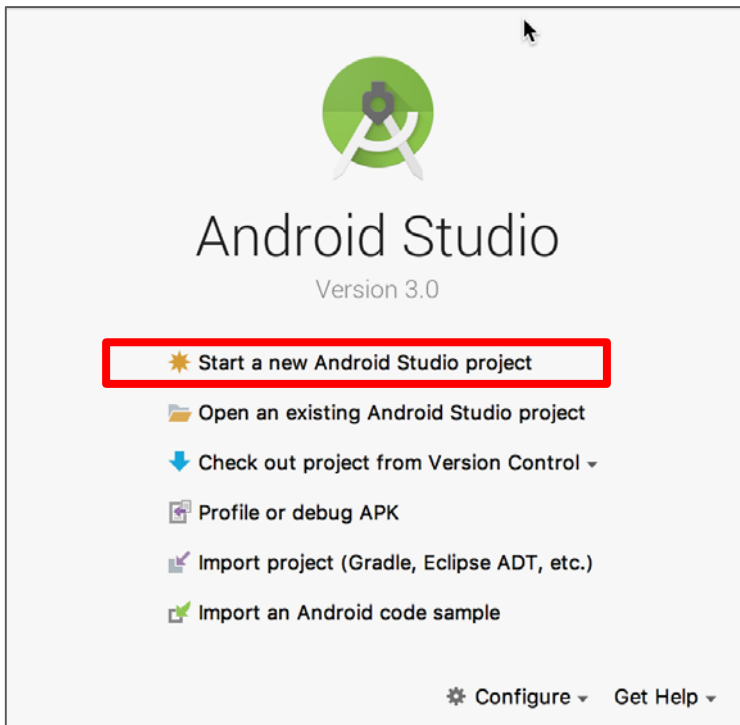
1. Toolbar
2. Navigation bar
3. Project pane
4. Editor
5. Tabs for other panes

Installation Overview

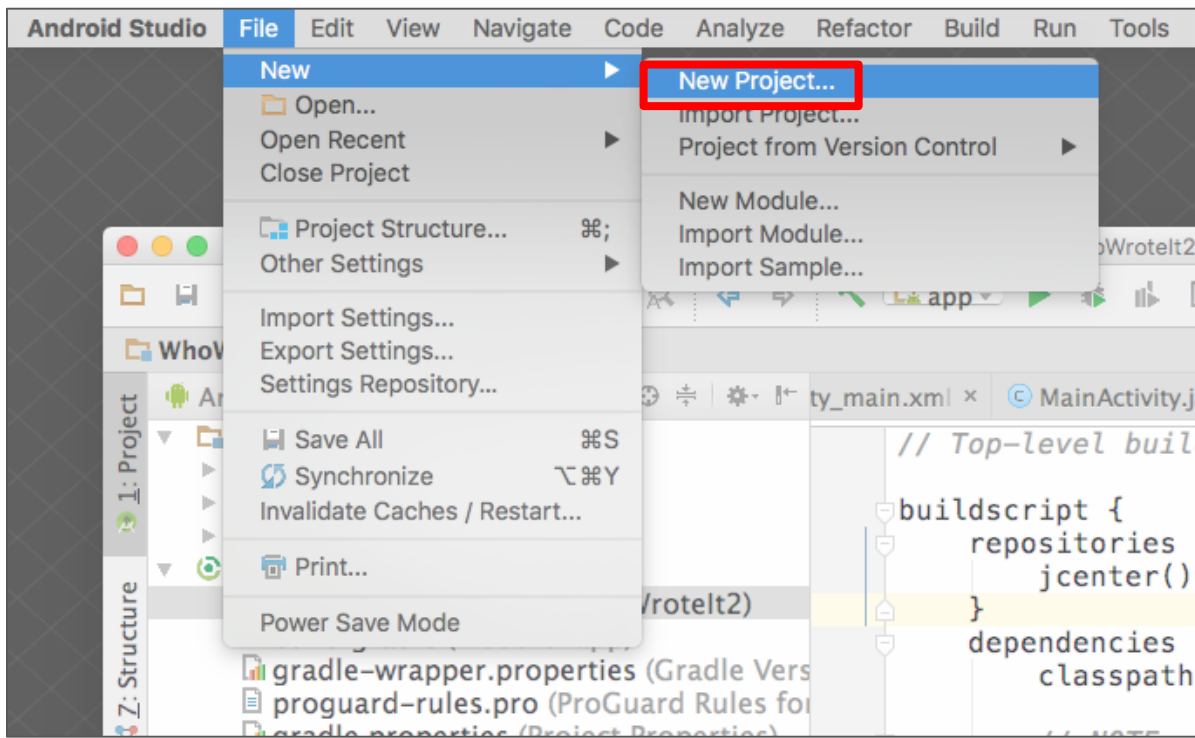
- Mac, Windows, or Linux
- Download and install Android Studio from <https://developer.android.com/studio/>

Creating your first Android app

Start Android Studio



Create a project inside Android Studio



Name your app

Create New Project

Create Android Project

Application name
Hello World

Company domain
android.example.com

Project location
/Users/tbove/AndroidStudioProjects/HelloWorld

Package name
com.example.android.helloworld Edit

Include C++ support
 Include Kotlin support

Cancel Previous Next Finish



Target Android Devices

Select the form factors and minimum SDK

Some devices require additional SDKs. Low API levels target more devices, but offer fewer API features.

Phone and Tablet

API 15: Android 4.0.3 (IceCreamSandwich)

By targeting **API 15 and later**, your app will run on approximately **100%** of devices. [Help me choose](#)

Include Android Instant App support

Wear

API 21: Android 5.0 (Lollipop)

TV

API 21: Android 5.0 (Lollipop)

Android Auto

Android Things

API 24: Android 7.0 (Nougat)

Cancel

Previous

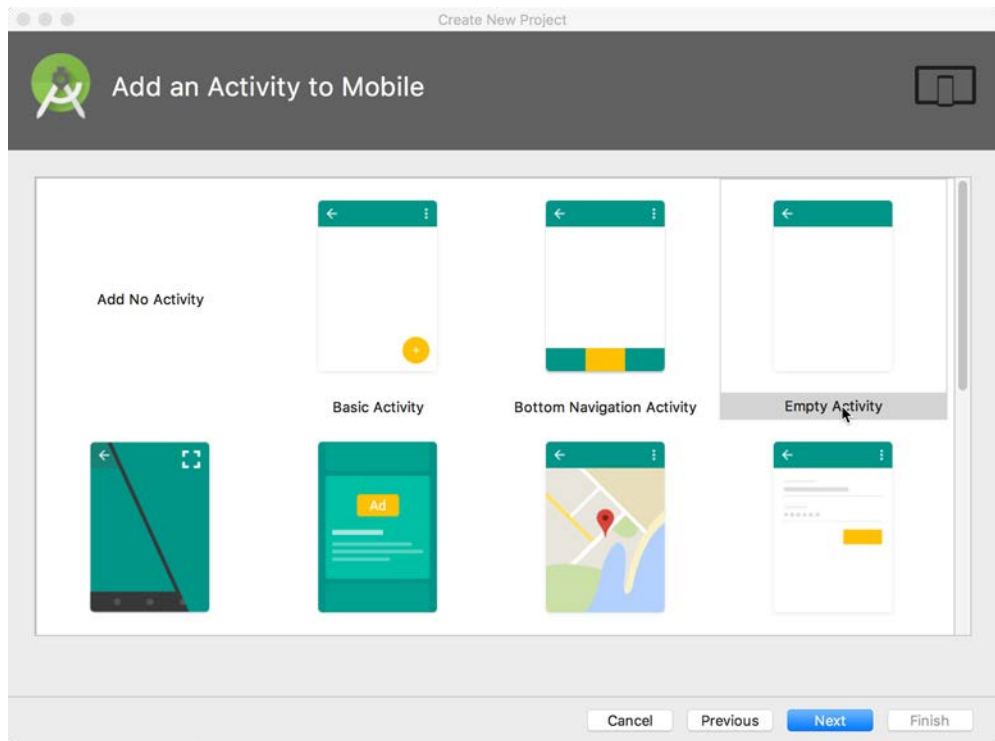
Next

Finish

Pick activity template

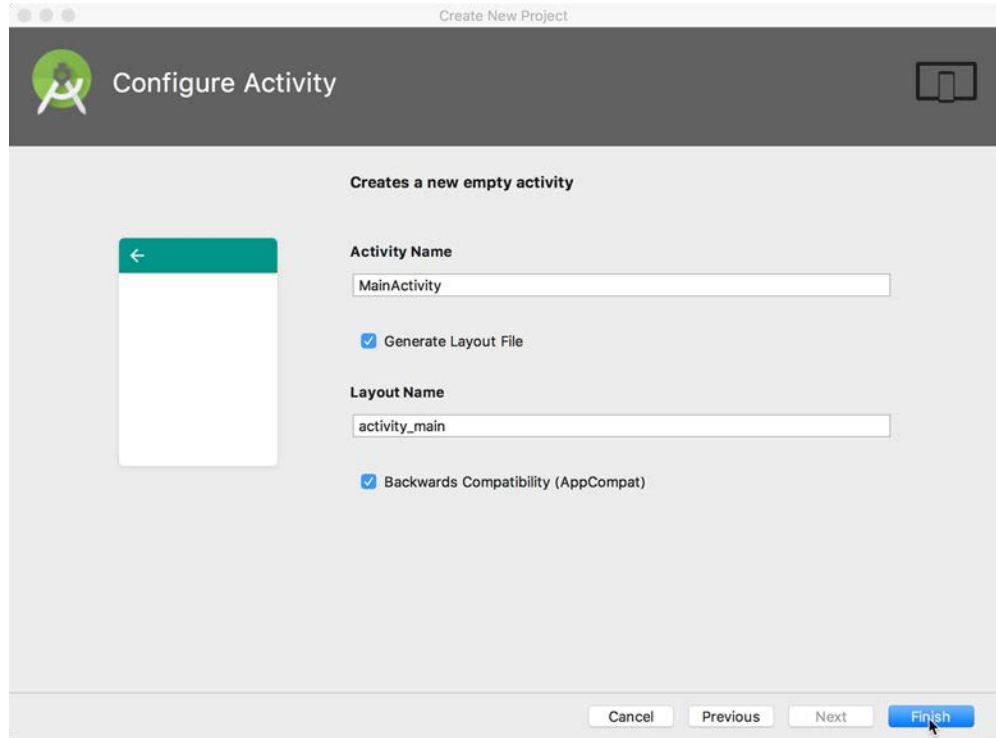
Choose templates for common activities, such as maps or navigation drawers.

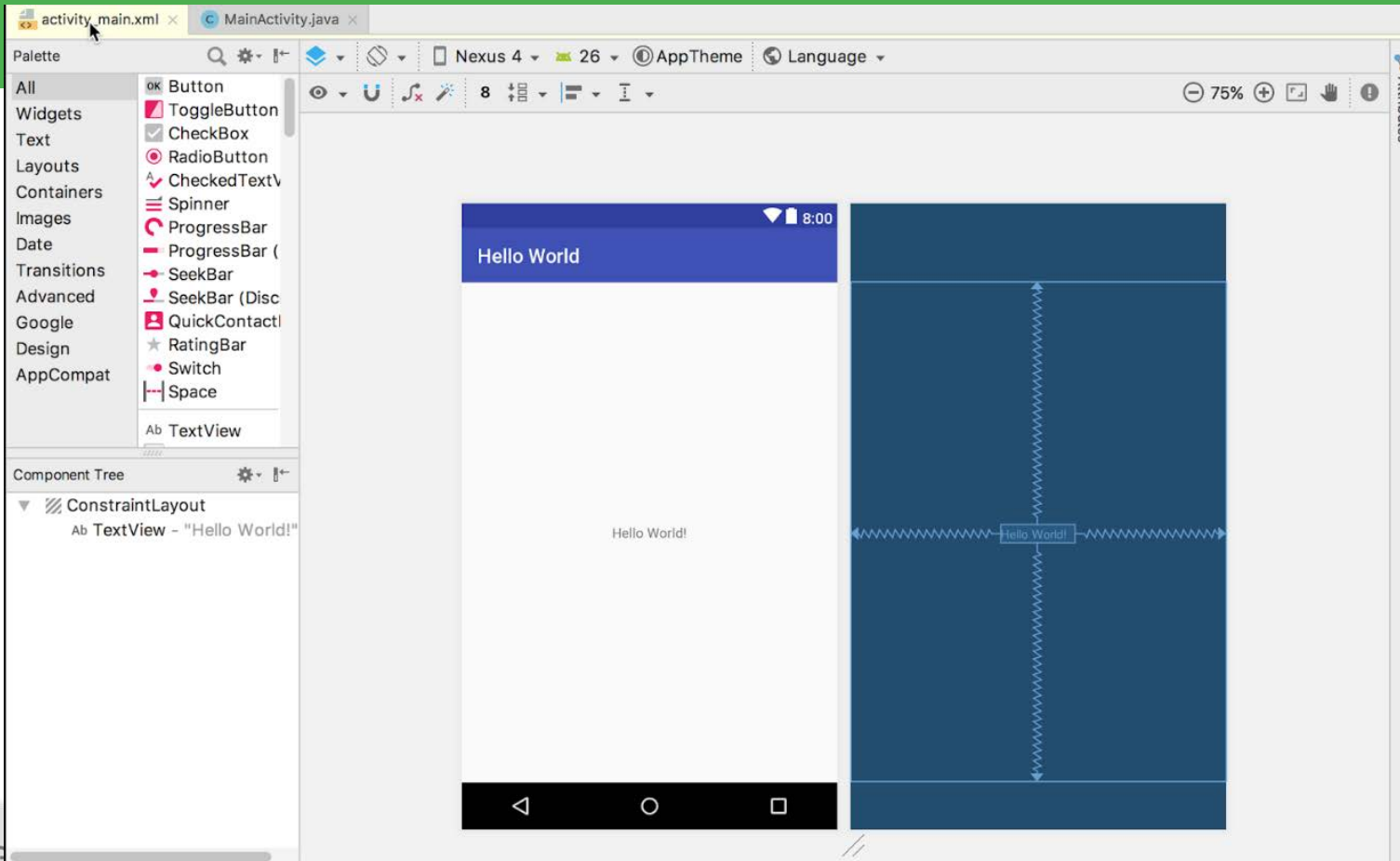
Pick Empty Activity or Basic Activity for simple and custom activities.



Name your activity

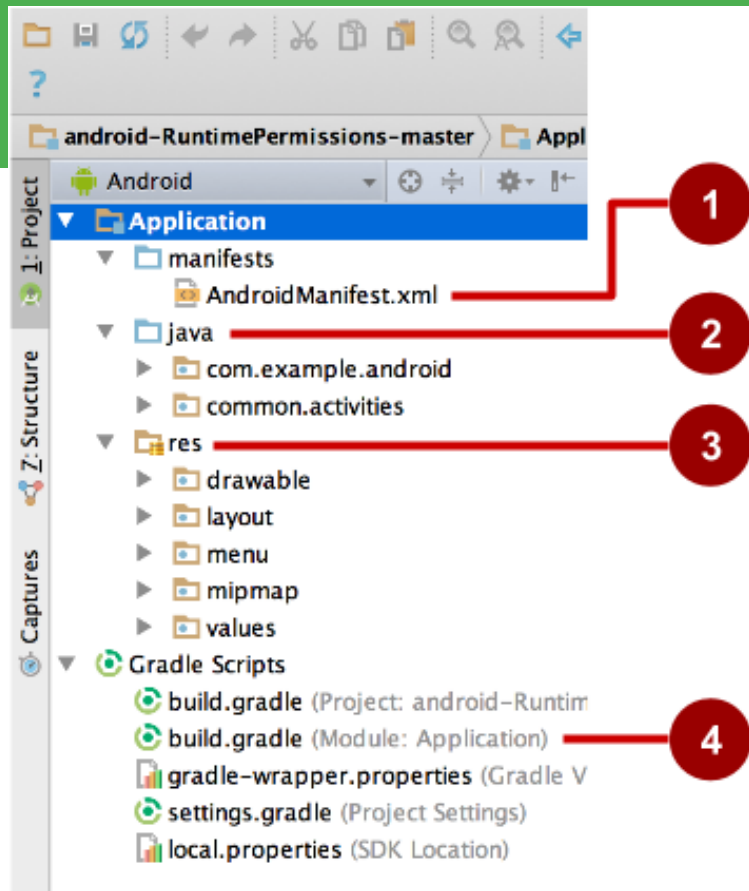
- Good practice:
 - Name main activity
MainActivity
 - Name layout
activity_main
- Use AppCompatActivity
- Generating layout file is convenient





Project folders

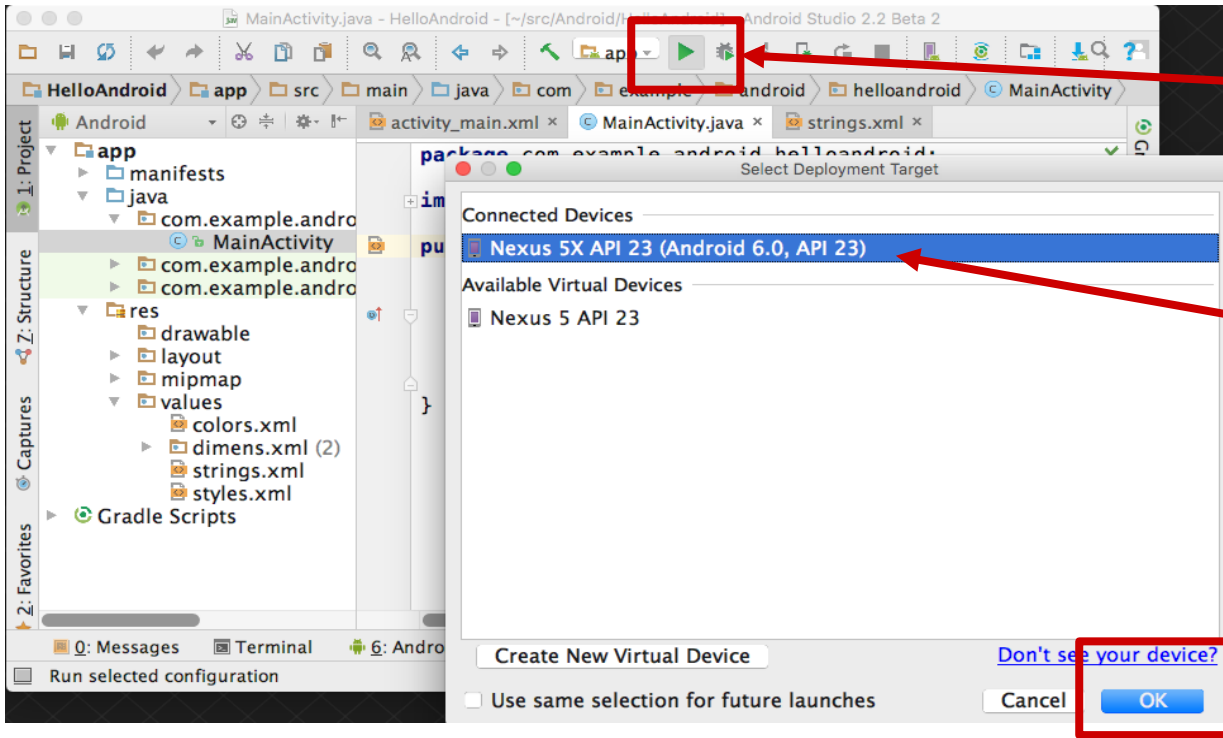
- 1. manifests**—Android Manifest file - description of app read by the Android runtime
 - examples
- 2. java**—Java source code packages
- 3. res**—Resources (XML) - layout, strings, images, dimensions, colors...
- 4. build.gradle**—Gradle build files



Gradle build system

- Modern build subsystem in Android Studio
- Three build.gradle:
 - Project: information about the gradle itself.
 - Module: information about each module. Phone, Wear, TV, Car
 - Settings: information about which module to compile, Phone+TV
- Typically not necessary to know low-level Gradle details

Run your app



1. Run

2. Select virtual or physical device

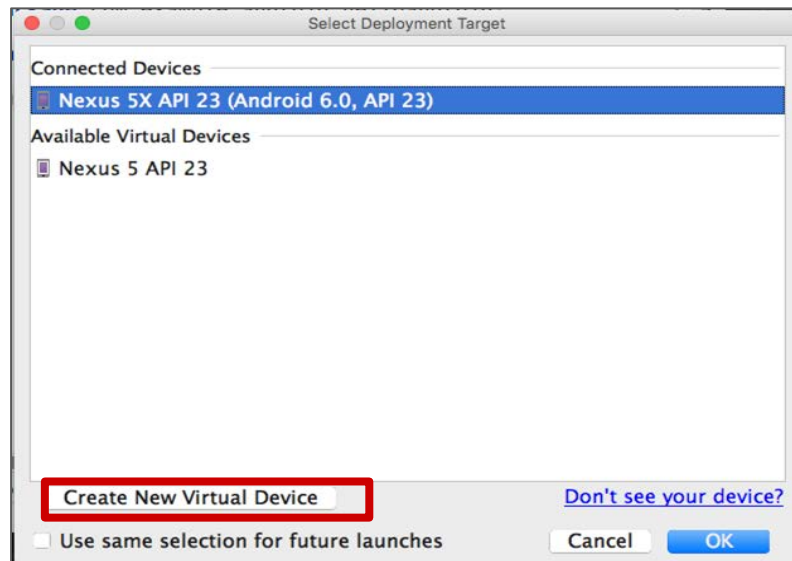
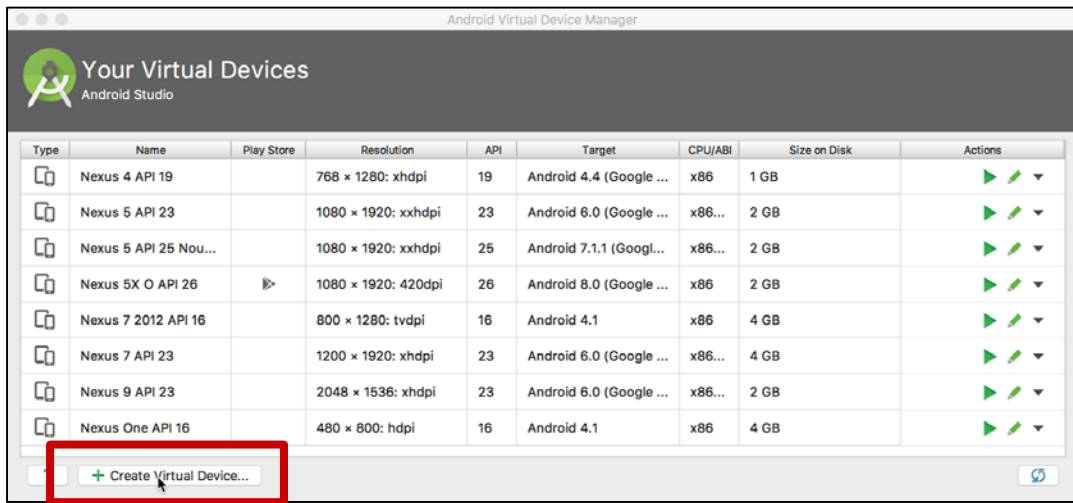
3. OK

Create a virtual device

Use emulators to test app on different versions of Android and form factors.

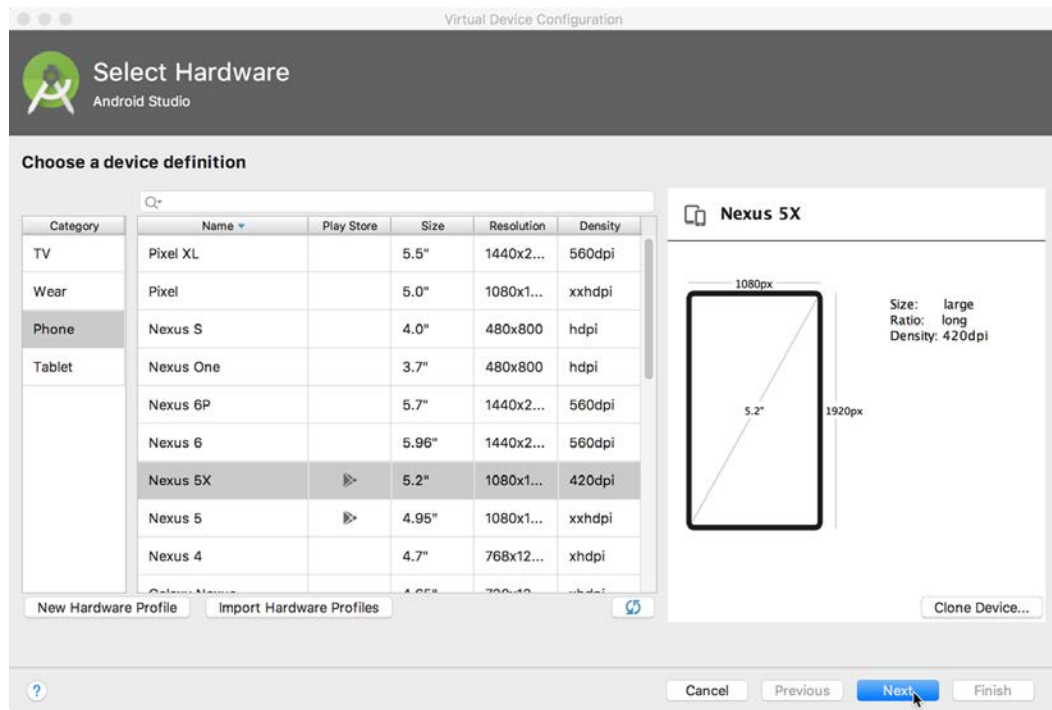
Tools > Android > AVD Manager

or:



Configure virtual device

1. Choose hardware
2. Select Android version
3. Finalize





System Image

Android Studio

Select a system image

Recommended x86 Images Other Images

Release Name	API Level	ABI	Target
API 27	27	x86	Android API 27 (Google Play)
Oreo	26	x86	Android 8.0 (Google Play)
Nougat	25	x86	Android 7.1.1 (Google Play)
Nougat Download	24	x86	Android 7.0 (Google Play)

Oreo



API Level

26

Android

8.0

Google Inc.

System Image

x86

We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?

See the [API level distribution chart](#)



Cancel

Previous




Next

Finish

Connected Devices

<none>

Available Virtual Devices

-  Nexus 5X API 26
-  Nexus 5X API 25
-  Nexus 9 API 25 Tab

Create New Virtual Device

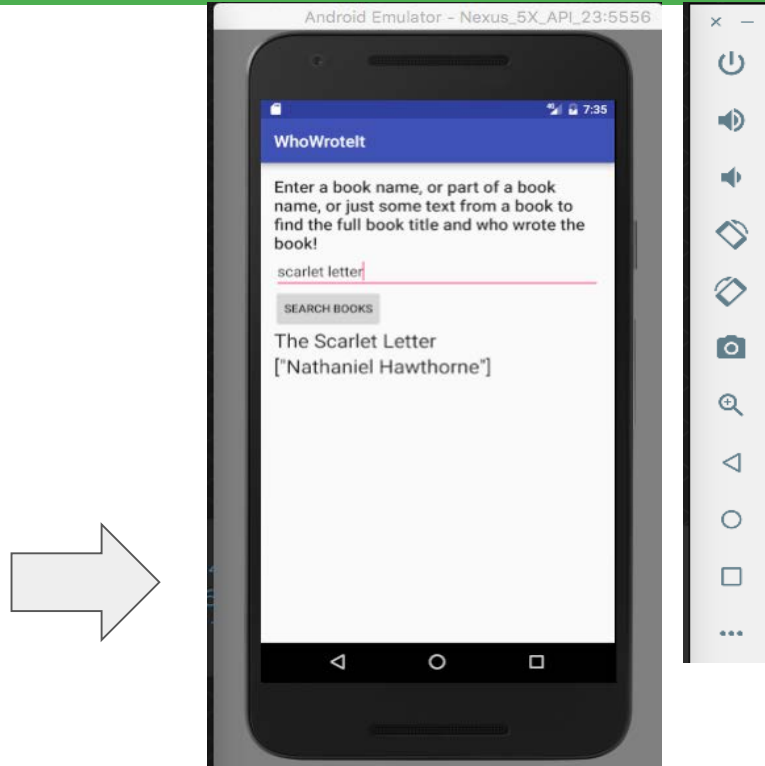
[Don't see your device?](#)



Cancel

OK

Run on a virtual device



Run on a physical device

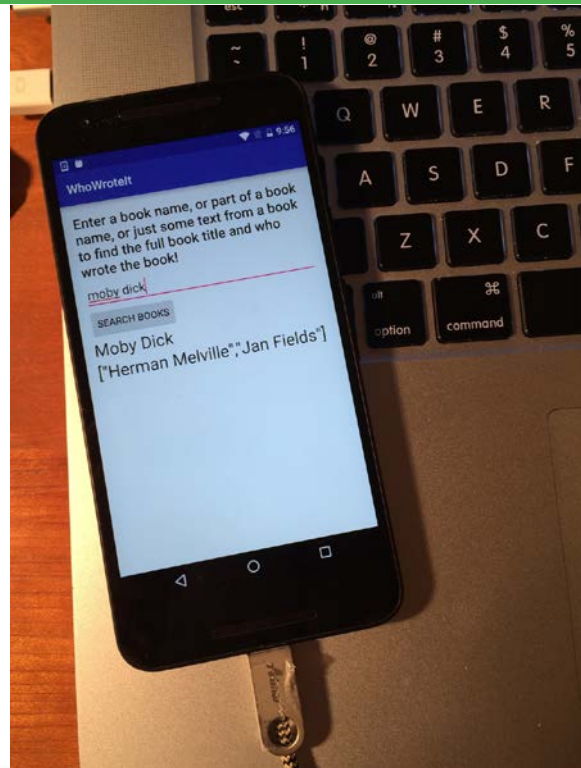
1. Turn on Developer Options:
 - a. **Settings > About phone**
 - b. Tap **Build number** seven times
2. Turn on USB Debugging
 - a. **Settings > Developer Options > USB Debugging**
3. Connect phone to computer with cable

Windows/Linux additional setup:

- [Using Hardware Devices](#)

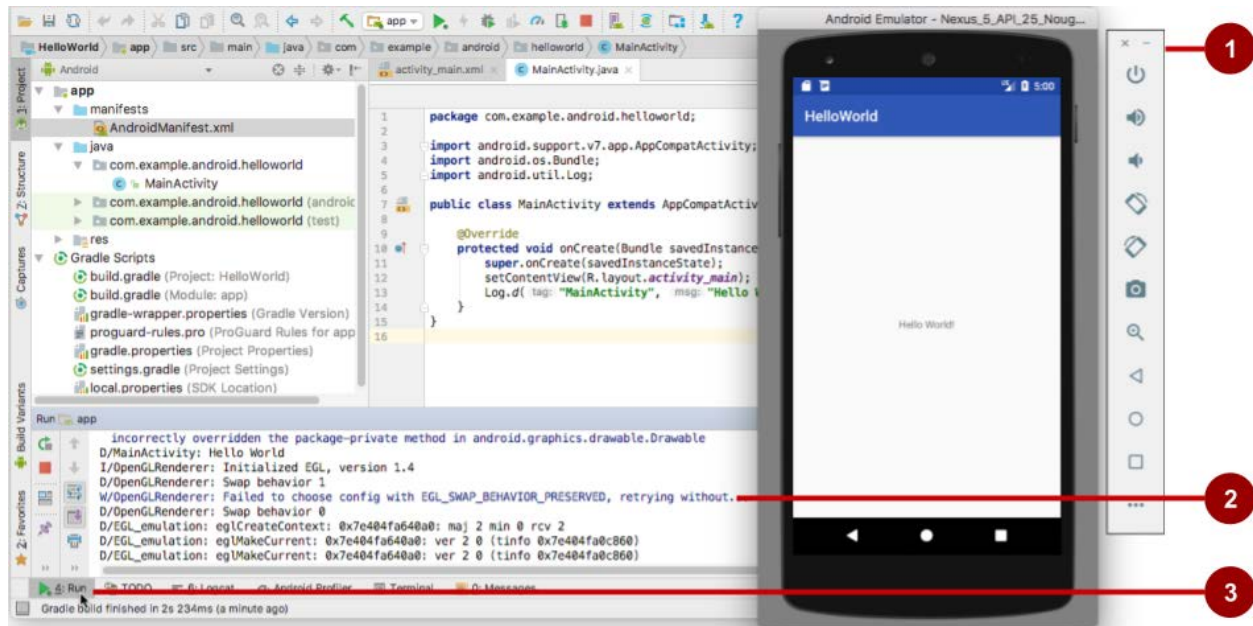
Windows drivers:

- [OEM USB Drivers](#)



Get feedback as your app runs

1. Emulator running the app
2. Run pane
3. Run tab to open or close the Run pane



Adding logging to your app

- As the app runs, the **Logcat** pane shows information
- Add logging statements to your app that will show up in the Logcat pane
- Set filters in **Logcat** pane to see what's important to you
- This will be covered in later lectures.