Troubleshooting Guide - Developing Android Apps in Kotlin

Android Studio Versions and What to Install

Installing the stable version of Android Studio

We recommend that you use the latest stable versions of the Android SDK, Kotlin plugin, Gradle, libraries, and Android Studio, whenever possible.

You can download the latest stable version of Android Studio here.

In our courses, we strive to teach the most up-to-date features when a course is released. While a course is being developed, these features may still be under development, and instructors may be using beta or canary versions of Android Studio. Note that as a result, the UI shown in the videos may be slightly different from the UI you are working with.

If you need to download and install a beta or canary build, you can find links to them here.

Android Studio preview channels

Android Studio builds are available in several "preview channels". In general, which channel you choose for receiving updates depends upon whether you prefer to work in the most stable and bug-free environment, or enjoy to preview new features at the cost of occasional bugs.

For this course, we recommend that you use the latest stable version, unless a feature used in the course is not yet released in stable.

There are four different preview channels, described below. See the Preview Channels documentation for details.

- **Stable channel:** Contains the most recent stable version of Android Studio.
- **Beta channel:** When Android Studio reaches a beta milestone for its next version, the beta builds are posted here.
- **Dev channel:** Dev builds are hand-picked older canary builds that survived the test of time. The channel should be updated roughly bi-weekly or monthly.
- **Canary channel:** Canary builds are the bleeding edge, released about weekly. While these builds do get tested, they are still subject to bugs, as the Android team wants people to see what's new as soon as possible. This build is not recommended for production development.

To select your preferred channel for updates:

1. Open **Android Studio -> Preferences** (on Mac) or **File -> Settings** (on Windows).
2. Select **Appearance & Behavior > System Settings > Updates**.
3. Click the dropdown box next to **Automatically check updates for** and pick one.

Working with Lesson Code on GitHub

All of the code for our courses is on GitHub in the Udacity repository.

- We recommend that you use a local Git installation to work with your code, and you can find details below.
- If you choose not to use Git for your projects, you find instructions below for downloading the code as a zip file from GitHub.

I'm new to Git and GitHub, how do I get started?

Git and GitHub are code repositories with version control. You use GitHub to store your code online, and Git to work with it locally. Your local Git repository can be connected to your remote GitHub repository, allowing you to upload your local changes to GitHub and, optionally, collaborate with others on a project.

We recommend that you create an account on GitHub and use it for this course. In addition to keeping your exercises organized, your GitHub repositories can eventually serve as your coding and project portfolio.

Learn more about Git and GitHub:

- How to use Git and GitHub (new users)
- Version control with Git (new users)
- Git website and documentation
- Optimize your GitHub
- GitHub collaboration

How do I install Git?

Install Git on your computer following the instructions in getting started and installing Git.

Where can I find the source code for this course?

The source code for this course is stored in our main Udacity GitHub repository.

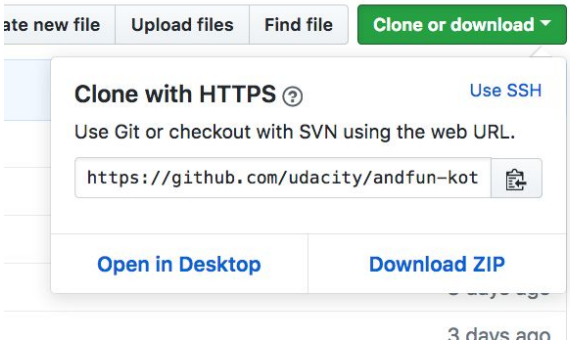
How to use this repo while taking the course describes the structure of our repository and how you can find solutions to lesson exercises.

Import Lesson Code into Android Studio

If I don't want to use Git, how do I download the sample apps?

Download the code for each project from GitHub as a zip file and import it into Android Studio.

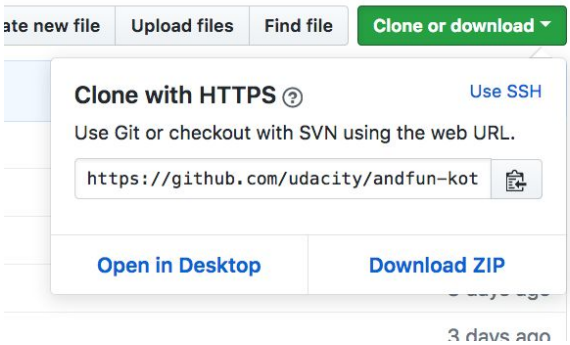
1. On GitHub, find the repository for the app.
2. Make sure that you are on the master branch.
3. Click on the **Clone or download** button.
4. Click **Download Zip**.
5. Save and unzip anywhere on your computer.



6. Start Android Studio.
7. On the Startup screen, open the project, or open it with **File > Open** if you have other projects open.
8. To get a different branch, find it on GitHub and download its zip file.

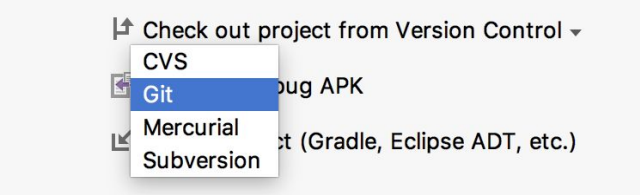
If I use Git, how do I import sample apps into Android Studio?

1. Go to the project on the Udacity GitHub.
2. Click on the **Clone or download** button.
3. Click **Use HTTPS**. (If the link says Use **SSH** then you are already in the correct mode.)



4. Copy the URL.
5. Start Android Studio.

6. On the Android Studio Welcome screen, click **Check out project from Version Control** and choose **Git**.



   If you do not see **Git** as an option, you have to enable it:
   - From the Android Studio menu, select **Android Studio > Preferences**.
   - Click **Plugins**.
   - Find **Git Integration** and check it.
   - Click **OK** and accept restarting Android Studio.
   - Then on the Welcome screen, **Check out project from Version Control** and choose **Git**.

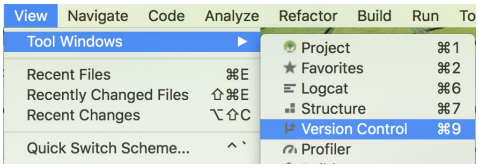
7. Paste the HTTPS web URL that you copied a few steps ago into the **URL** field.
8. Choose a location for the local repository and click the **Clone** button.
9. Click **Yes** when prompted to open the project.
10. Click **OK** when prompted to Import from gradle.
11. Click **Don't remind me again for this project** if prompted to update gradle files.
12. Wait until the build finishes.
    - If you get error messages about missing components, click the links in the messages to install the required components for your project. This may take a while.

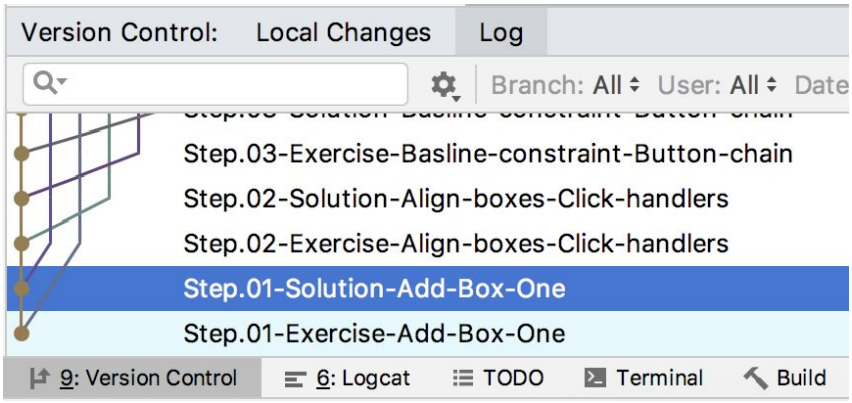
See the next section on how to work with this code.

Using Lesson Code with Git and Android Studio

1. Click **Version Control** at the bottom of Android Studio to open a pane that shows all the branches and their relationships.

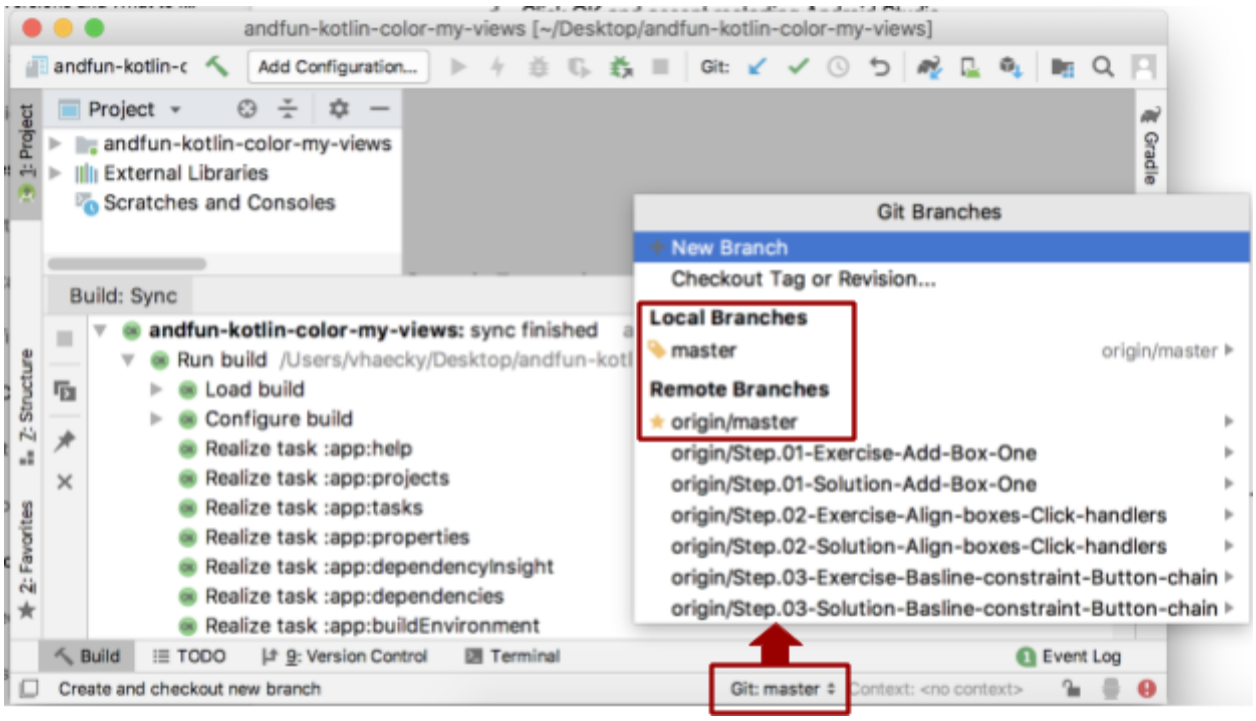
   - If you don't see it, select View > Tool Windows > Version Control to show it.





2. Click on the Git button at the bottom right of Android Studio. This will show you all the branches.

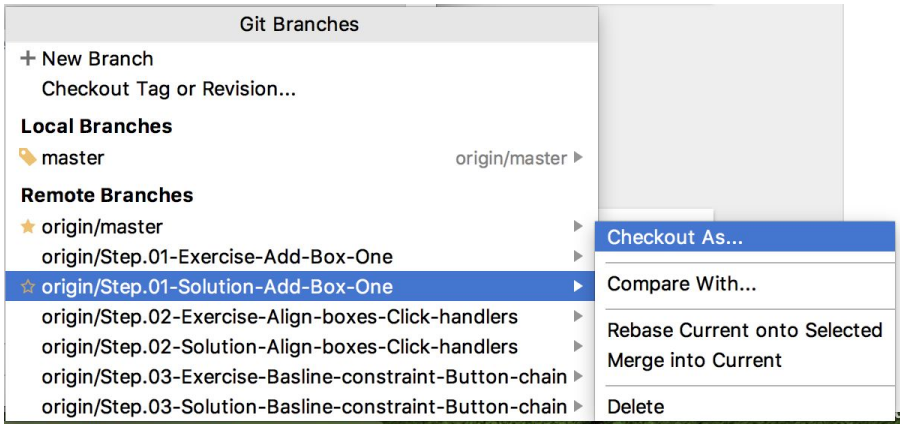
    - Local branches are branches you have cloned to your local Git repository.
    - A yellow tag indicates which branch you're currently on.
    - The name of the current branch is also always visible in the lower right corner.
    - Remote Branches is the list of all available branches for this project.
    - Commits for all course projects are organized as Exercise-Solution pairs, providing you with a clean start and the instructor's solution for each exercise.



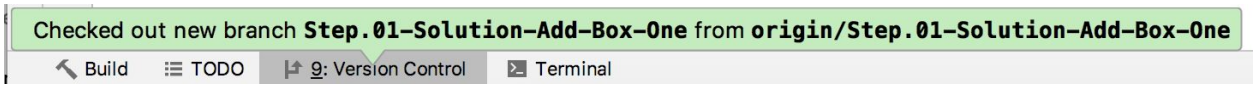
To check out a branch that corresponds to an exercise:

1. Open the **Git Branches** list.

2. Click on the branch you want to work on locally.
3. Select **Checkout as**.



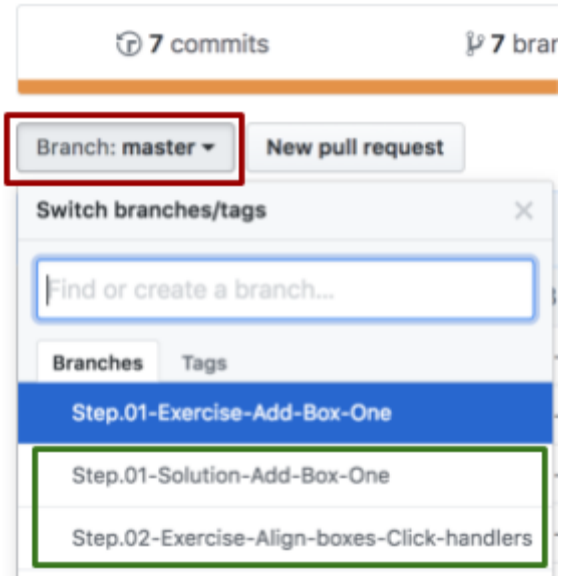
4. Click **OK**. The branch is checked out and the code is opened in Android Studio.



Completing Lesson Exercises

You can see starter and solution code in two ways:

- Locally, through Git, or by downloading the corresponding solution branch as a zip file.
- On GitHub, you can view a branch by clicking the *Branch* button and selecting the branch from the dropdown box.

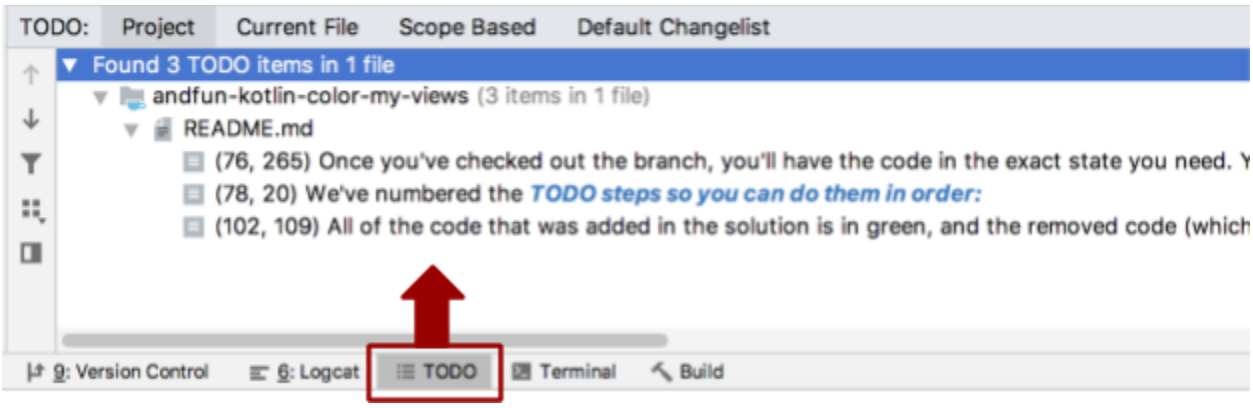


- The source code for each exercise has the starting code and the solution code, as shown in the previous screenshot.
- The starter code for each step includes a set of TODOs to guide you through the changes that you need to make for that exercise.

Working with TODOs

Android Studio makes it straightforward to see and work with TODOs.

1. Open the TODO pane to display a complete list of the TODOs you complete for every step.

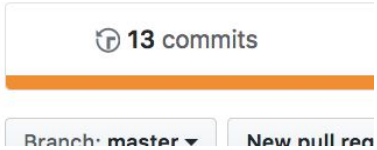


2. Each TODO indicates a place in the code where you will need to make changes.
3. Double clicking a TODO opens the file and takes you directly to its place in the code. Sometimes you'll be given extra information or tips in the TODOs, so make sure to read each one carefully.

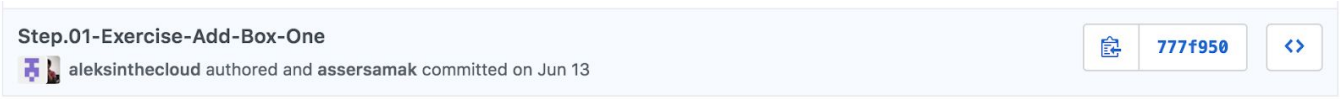
Looking at Udacity solution code on GitHub

If you get stuck on an exercise or want to check if you did the right thing, you can look at the difference between the starter and solution code.

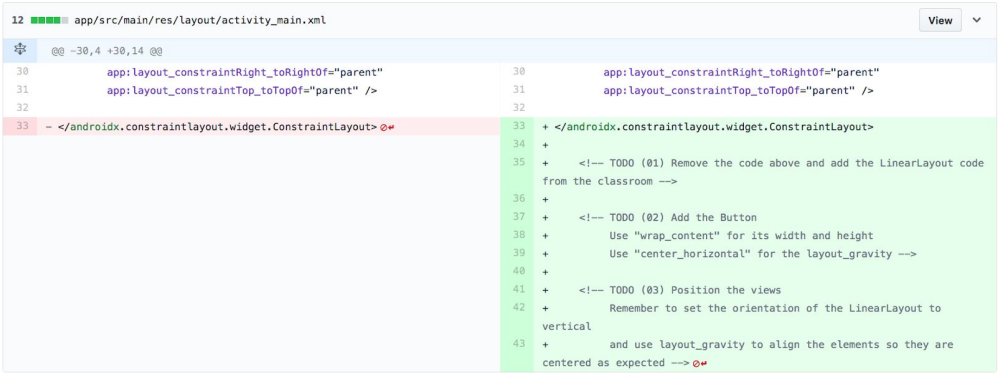
1. On GitHub click the button that shows the number of commits (toward the upper left of the screen).



2. The next screen lists all the commits for the current branch. Click on a commit to view all the changes made since the prior commit.



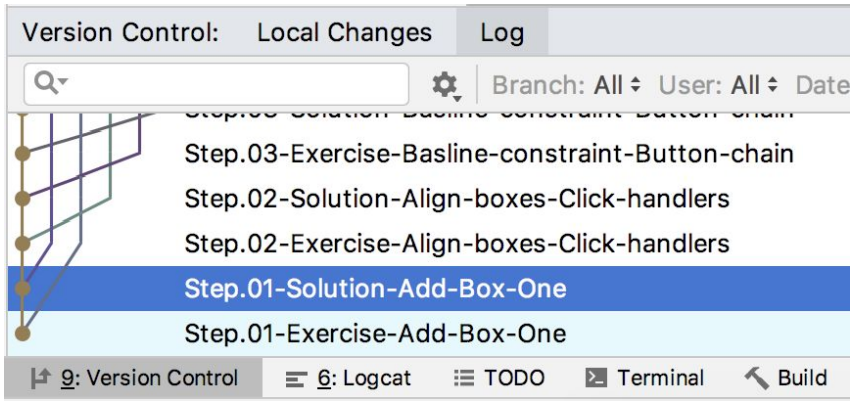
3. Clicking on the commit will open a page showing the differences. This can be in split, side-by-side, or as a unified view. Click the corresponding button in the upper right corner of the page to change this view.



5. See the GitHub documentation for detailed instructions for reading diff screens.

Looking at the solution if you're using the Git repository

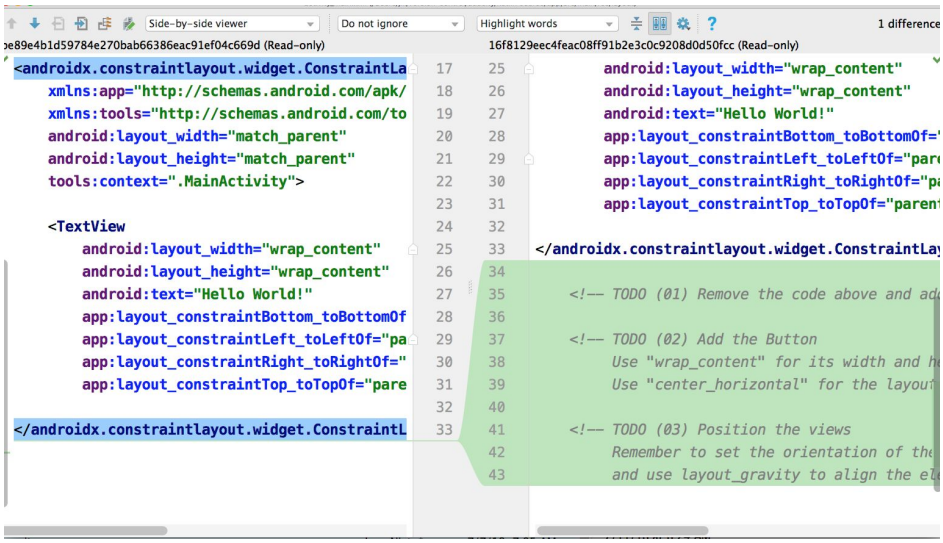
1. Click **Version Control** to open the Version Control pane with a list of branches.



2. Select a solution commit. For example, the Step 1 solution branch will be called "Step.01-Solution-*Followed-by-a-description*".

3. Press **Ctrl-D** on Windows, or **Cmd-D** on a macOs, to open a window showing the differences with the previous commit.

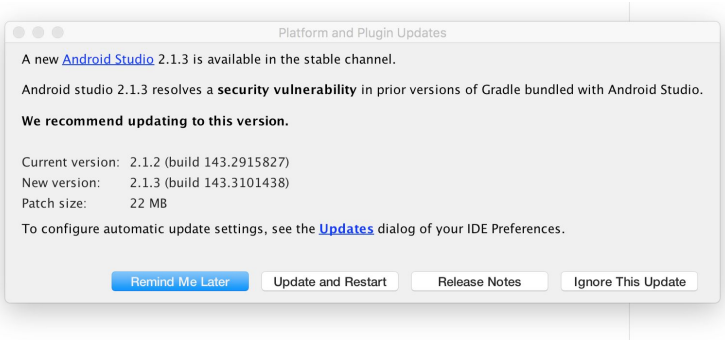
4. The diff window presents the two commits side-by-side and highlights the differences between them. You can see the code that was added to fulfill the TODOs.



Updates

Android Studio and plugin updates

Android Studio will alert you when new versions of Android Studio or its plugins are available. We recommend you always install the latest SDK, tools, and plugin versions.

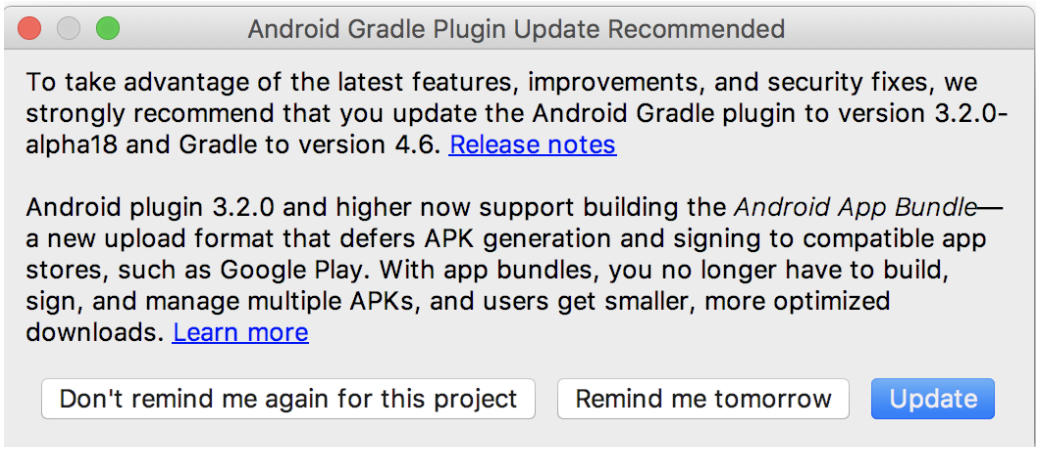


Gradle plugin updates

Android Studio will alert you when any of Udacity projects' Gradle version is out-of-date, when you open a project with an outdated version.

We recommend that you do NOT update the Gradle Plugin.

- If you choose to update to a newer version, there is no guarantee that your code will work as expected, although it typically will.



Dependencies updates

Android Studio alerts you when dependencies are out-of-date by highlighting them in the build.gradle file during editing.

```
dependencies {
    classpath "com.android.tools.build:gradle:$gradleVersion"
    classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
    classpath "android.arch.navigation:navigation-safe-args-gradle-plugin:$navigationV
```

We recommend that you do NOT update dependencies, unless absolutely necessary.

- Updating dependencies and Android versions can lead to issues and incompatibilities.
- If a code or dependency update is necessary, follow the course instructor's instructions.

Debugging

Enabling USB debugging

You can use an emulator for debugging, or a physical device.

In order to troubleshoot your app with a physical device, you must:

- Connect your device to your computer
- Enable USB debugging on your device.

You can find detailed instructions here.

Debugging in Android Studio

With Android Studio you can do the following and more to help you debug your app.

- Select a device or emulator to debug your app on.
- Set breakpoints at individual lines in your Java and Kotlin code.
- Examine variables and evaluate expressions at runtime.

The Android documentation contains detailed instructions on how to debug in Android Studio.

Getting Help

If you need more help, there are many excellent resources on the web for you to explore.

- **Android documentation:** The main access point for complete, up-to-date, and searchable Android documentation is https://developer.android.com. We encourage you to explore and search this resource.

- **Tutorials** are a great resource for learning new features or enhancing your Android knowledge. Search for a topic and "tutorial" to find them.

- **StackOverflow** is where the developer community at large, representing all skill levels, post questions and answers on a wide range of topics. If you're struggling with an issue, chances are someone else has already encountered and solved it.
- **Udacity courses:** And don't forget, you can find additional Android courses on udacity.com.