Android Development with Kotlin

Publish App

What is Version Control?

Version control is like a savings program for your project. By tracking and logging the changes you make to your file or file sets over time, a version-control system gives you the power to review or even restore earlier versions. Version control takes snapshots of every revision to your project. You can then access these versions to compare or restore them as needed.

What is GIT?

First developed back in 2005, Git is an extremely popular version control system that is at the heart of a wide variety of high-profile projects. Git is installed and maintained on your local system (rather than in the cloud) and gives you a self-contained record of your ongoing programming versions. It can be used completely exclusive of any cloud-hosting service — you don't even need internet access, except to download it.

What is GitHub?

In the discussion of Git vs. GitHub, it's been said that GitHub is to Git what Facebook is to your actual face. What's that mean? Well, it means that while Facebook is kind of like an online face database (of sorts). GitHub is designed as a Git repository hosting service.

And what exactly is a Git repository hosting service? It's an online database that allows you to keep track of and share your Git version control projects outside of your local computer/server. Unlike Git, GitHub is exclusively cloud-based. Also unlike Git, GitHub is a for-profit service (although basic repository-hosting features are available at no cost to those who are willing to create a user profile, making GitHub a popular choice for open-source projects).

GIT Vs GitHub:

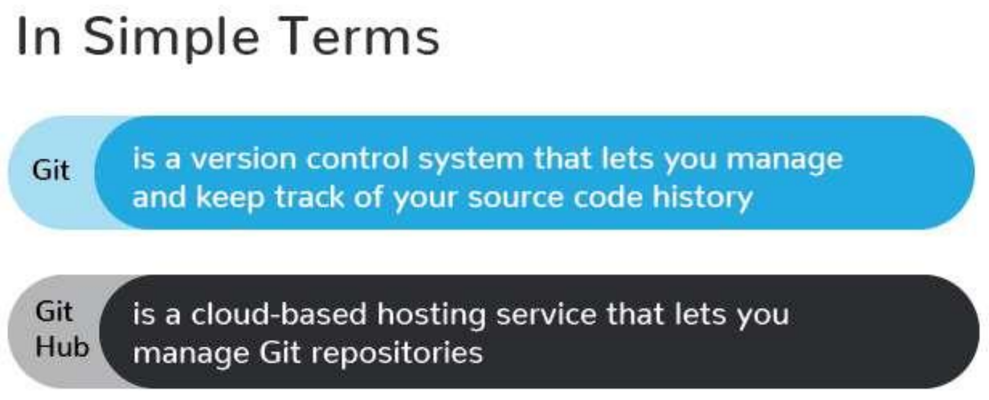

In Simple Terms
Git
is a version control system that lets you manage and keep track of your source code history
Git Hub
is a cloud-based hosting service that lets you manage Git repositories

Image src: https://blog.devmountain.com/wp-content/uploads/2019/07/Gitvs.Github-1c-750x321.jpg

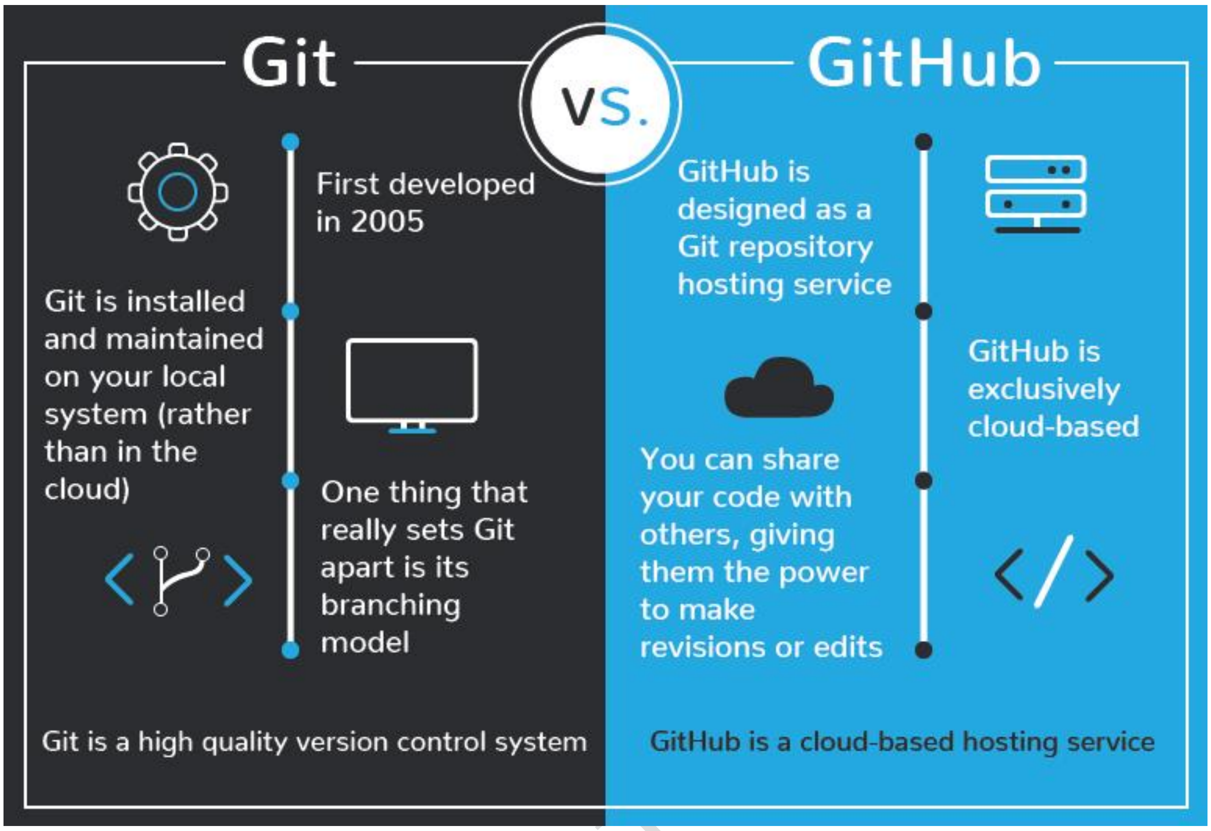
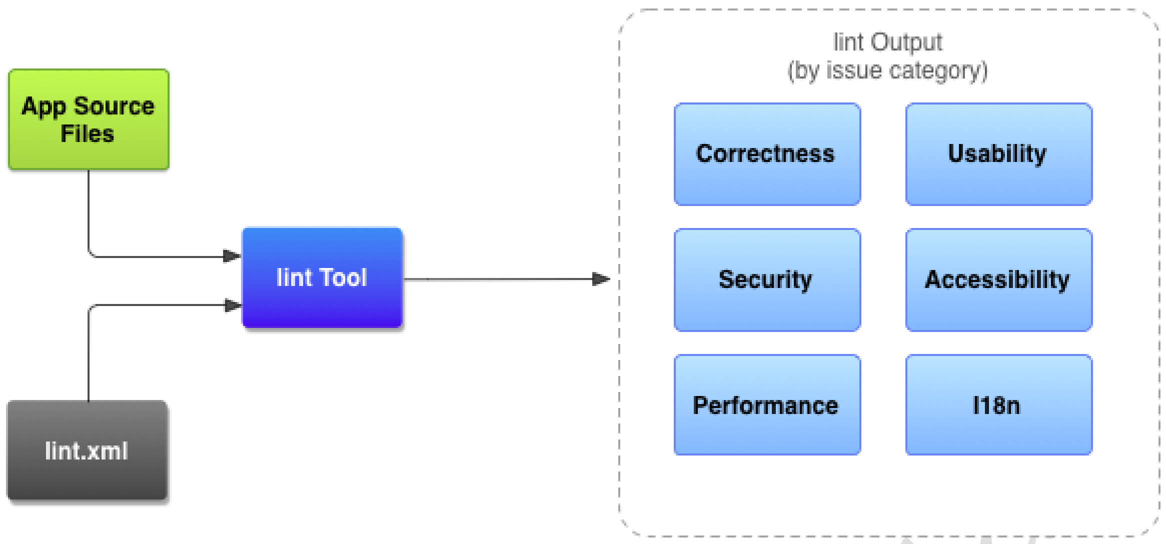

Git
vs.
GitHub
First developed in 2005
Git is installed and maintained on your local system (rather than in the cloud)
One thing that really sets Git apart is its branching model
Git is a high quality version control system
GitHub is designed as a Git repository hosting service
GitHub is exclusively cloud-based
You can share your code with others, giving them the power to make revisions or edits
GitHub is a cloud-based hosting service

Image src: https://blog.devmountain.com/wp-content/uploads/2019/07/Gitvs.Github-1a.jpg

Inspect Code:

Android Studio provides a code scanning tool called **lint** that can help you to identify and correct problems with the structural quality of your code without your having to execute the app or write test cases. Each problem detected by the tool is reported with a description message and a severity level, so that you can quickly prioritize the critical improvements that need to be made. Also, you can lower the severity level of a problem to ignore issues that are not relevant to your project, or raise the severity level to highlight specific problems.

The lint tool checks your Android project source files for potential bugs and optimization improvements for correctness, security, performance, usability, accessibility, and internationalization. When using Android Studio, configured lint and IDE inspections run whenever you build your app. However, you can manually run inspections or run lint from the command line.



R8 and optimising app:

To make your app as small as possible, you should enable shrinking in your release build to remove unused code and resources. When enabling shrinking, you also benefit from obfuscation, which shortens the names of your app's classes and members, and optimization, which applies more aggressive strategies to further reduce the size of your app. This page describes how R8 performs these compile-time tasks for your project and how you can customize them.

Enabling optimisation, shrinking, obfuscation:

```
android {
    buildTypes {
        release {
            // Enables code shrinking, obfuscation, and optimization
for only
            // your project's release build type.
            minifyEnabled true

            // Enables resource shrinking, which is performed by the
            // Android Gradle plugin.
            shrinkResources true

            // Includes the default ProGuard rules files that are
packaged with
            // the Android Gradle plugin. To learn more, go to the
section about
            // R8 configuration files.
            proguardFiles getDefaultProguardFile(
                    'proguard-android-optimize.txt'),
                    'proguard-rules.pro'
            }
        }
        ...
}
```

Signing your app:

Android requires that all APKs be digitally signed with a certificate before they are installed on a device or updated. If you use Android App Bundles, you need to sign only your app bundle before you upload it to the Play Console, and app signing by Google Play takes care of the rest. However, you can also manually sign your app for upload to Google Play and other app stores.

Apk Analyzer:

Android Studio includes an APK Analyzer that provides immediate insight into the composition of your APK after the build process completes. Using the APK Analyzer can reduce the time you spend debugging issues with DEX files and resources within your app, and help reduce your APK size. It's also available from the command line with apkanalyzer.

With the APK Analyzer, you can accomplish the following:

- View the absolute and relative size of files in the APK, such as the DEX and Android resource files.
- Understand the composition of DEX files.
- Quickly view the final versions of files in the APK, such as the AndroidManifest.xml file.
- Perform a side-by-side comparison of two APKs.
- There are three ways to access the APK Analyzer when a project is open:
- Drag an APK into the Editor window of Android Studio.
- Switch to the Project perspective in the Project window and then double-click the APK in the default build/output/apks/ directory.
- Select Build > Analyze APK in the menu bar and then select your APK.

All the concepts have been covered in details in the lecture videos. Feel free to explore more using the below links:

Shrinking and Optimising App: https://developer.android.com/studio/build/shrink-code

App Signing: https://developer.android.com/studio/publish/app-signing

Apk Analyzer: https://developer.android.com/studio/build/apk-analyzer

Google play console: https://developer.android.com/distribute/console