Git And Github

Git

Git is a version control system that lets you keep track of all the modifications you make to your code.
This means if a new feature is causing some errors in your code, you can always roll back to a previous stable version.

But Git isn't any VCS, it's a distributed VCS which means that every collaborator of the project will have a history of the changes made on their local machine. It is distributed so you can access your code files from another computer – and so can other developers.

GitHub

GitHub is a widely used platform for version control that uses Git at its core.
It lets you host the remote version of your project from where all the collaborators can have access to it.

Differences between Git and GitHub
Git is a version control system that manages and keeps track of your code. GitHub, on the other hand, is a service that lets you host, share, and manage your code files on the internet.

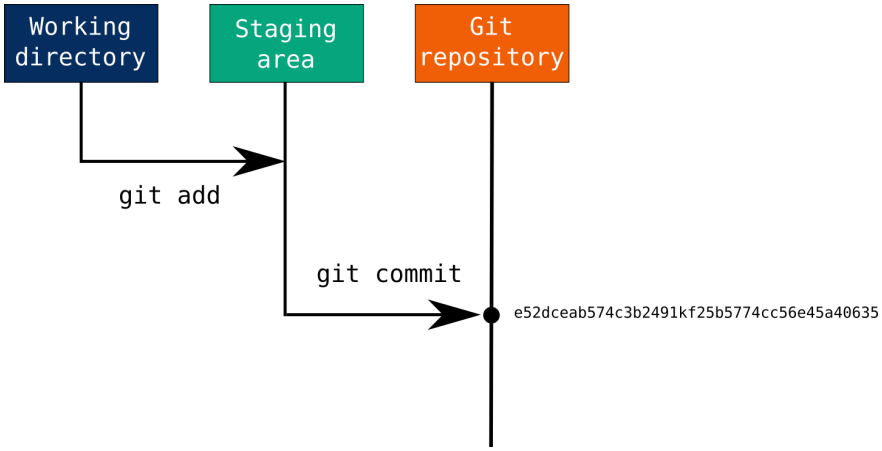
GitHub uses Git underneath, and lets you manage your Git repositories or folders easily on its platform.
So Git is the actual version control system and GitHub is the platform where you host your code.

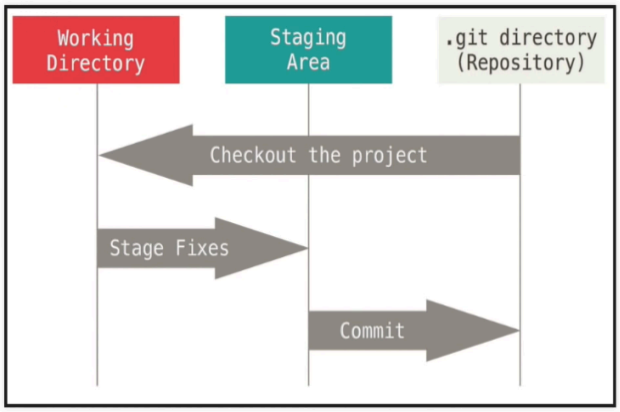
Git commands to type in terminal:

- git init (initializes repository (tells git to look for files in this folder))ONCE PER PROJECT
- git add . (stages files for a commit; the dot adds all files in the directory)
- git commit -m "comment" (creates a commit aka save. This is a save command in git; save often and frequently to track your changes)
- the -m "message" and -m :message commits to the master branch, e.g. the main branch
- git log (shows history of changes made by all contributing members)
- git status (shows you whats added ("staged") but not yet committed, and tells you if any files have changed since last commit)
- git commit -am "comment" (adds and commits in one step)
- git add -u (tells commit to remove files you've deleted)
- git push origin master (sends updates to github)
- git pull origin master (pulls updates from github)

Git Staging Environment
One of the core functions of Git is the concepts of the Staging Environment, and the Commit.

As you are working, you may be adding, editing and removing files. But whenever you hit a milestone or finish a part of the work, you should add the files to a Staging Environment.
Git has three main states that your files can reside in: modified, staged, and committed:
Modified means that you have changed the file but have not committed it to your database yet.
Staged means that you have marked a modified file in its current version to go into your next commit snapshot.
Staged files are files that are ready to be committed to the repository you are working on.





For more commands and what they do refer to [github cheat sheet]