Complete definitions for key Git & GitHub terminology

GIT WORKFLOW FUNDAMENTALS

• Working Directory - The working directory contains a single checkout of a single project version. If you checkout a branch (e.g. master) and are working on a specific commit (e.g. HEAD), your working directory is the "umbrella" word for all of your files and folders.

• Staging Area - The staging area contains files that will be part of the next commit, letting git know what modifications will be made in the file for the next commit. All of a project's commits are stored in the repository. They're also known as untracked files.

• Local Repo (head) - head is nothing more than a pointer to a commit object. Each head is given a name (branch name or tag name, etc). Every repository has a head called master by default. Any number of heads can be stored in a repository.

• Remote repo (master) - remote refers to a repository that is located in another location. The data on a hard drive is considered local. The content on GitHub's server is accessible from afar. origin = the default name of the remote GitHub repository that corresponds to the repo you're working on right now. master is the default name for a repository's first branch.

WORKING DIRECTORY STATES:

Staged - In git, a staging step allows you to keep making changes to the working directory while still allowing you to record changes in minor commits when you chose to interact with version control.

• Modified - Because Git just checked them out and you haven't updated anything, when you initially clone a repository, all of your files will be tracked and unmodified. Because you've altered them since your last commit, Git recognises them as modified as you edit them.

• Committed GIT COMMANDS:

• Git add - The first command in a chain of activities that tells Git to "store" a snapshot of the current project state into the commit history is git add. When run alone, git add moves pending changes from the working directory to the staging area.

• Git commit - a single modification to a file (or set of files). It's similar to saving a file, except that with Git, each save generates a unique ID that lets you to track which changes were made when and by whom.

• Git push - The git push command is used to transfer content from a local repository to a remote repository. Pushing is the process of sending commits from a local repository to a remote repository.

• Git fetch - git fetch is the command that tells your local git to retrieve the latest meta-data info from the original

• Git merge- Git's approach of bringing a forked history back together is called merging. The git merge command allows you to combine the several development lines established by git branch into a single branch.

• Git pull - The git pull command fetches and downloads content from a remote repository and then updates the local repository to match it. git pull will initiate a merging workflow once the content has been downloaded. A new merging commit will be made, and HEAD will be modified to point to it.