

GIT & GITHUB

1 Definitions

- **Git.** A distributed version control system for source code, created by the famous Linus Torvalds. Source files are organized into file repositories.
- **GitHub.** A web based host (now owned by microsoft) for project repositories enabling version control using Git. GitHub is accessible via the web, a desktop app with a gui, and via the command line using Git commands. The command line is easy to use and is generally preferred.
- **Repository.** A logical grouping of files treated as a unit by Git. A Git repository can hypothetically exist anywhere; a GitHub repository is located at the GitHub host.
- **Local Repository.** A full copy of the Git repository on the user's local machine. Generally, the user manipulates files in the local repository and uses Git commands to update the Git repository.
- **Branch.** The Git repository maintains branches, which are copies of the one primary holy and official version of the files, the main branch, dubbed master. It is generally best to create secondary branches for maintaining your work, and subsequently merge thoroughly tested files from your working secondary branch back into the main branch.

2 Git Commands

- **git config.** Used to tie your local repository to your git account.
 - Type `git config --global user.name '<your user name>'`.
 - Type `git config --global user.email '<your email address>'`. Use the one associated with your github account.
- **git init.** Used to create an empty Git repository. Execution of this command establishes the working directory as a local repository. This is the command to use when there is an existing code base for the project.
- **git clone.** Used to clone a repository. This is used to create local repositories and is usually invoked `git clone <url of Git repository>`.
- **git status.** Reports the status of the local repository relative to the Git repository. If there are files residing in the local repository directory for which there is no corresponding file in the Git repository, this is reported. If a file has been added but not committed (see below), this is reported. Invoked via `git status`.
- **git add.** Adds a file to the local repository for tracking purposes. That is, simply locating a file in the local git repository directory is not enough for Git to know that this is a file to be tracked. Invoked via `git add <filename>` (**note:** `git add .` and `git add -A` both add all files in the local repository directory).
- **git commit.** The user's announcement to Git that the files in the local repository are ready to be pushed up to the Git repository. Invoked `git commit -m "<user generated message describing the update>"`.

- **git push.** This command causes the contents of the local repository to be pushed up to the Git repository. After this command is executed, changes made to the local repository are reflected in the Git repository, in other words, this is the command which syncs the files at the host with the local files to which the user has made changes. Invoked via `git push`.
- **git pull.** This command pulls the contents of the Git repository down to replace the contents of the local repository. Invoked via `git pull`.
- **git.** This command (i.e., `git`) displays a list of all git commands (there aren't many, but only a few are described here).
- **git branch.** Use this command to create branches via the command `git branch <branch_name>`. Creating a branch does not automatically switch your commits to the new branch! That is accomplished via the command `git checkout <branch_name>`.
- **git merge.** Once the files in the local repository have been massaged into the desired format and committed to their working branch of residence, the user may choose to merge that branch back into master. In this case, first issue a `git checkout master` command to make branch master the current branch. Then use the command `git merge <working_branch_name> -m '<a message indicating rationale for the merge>'`.

It must be noted here that all of the above commands, with the exception of `git push` and `git pull`, can be accomplished on one's own isolated computer. But the value of Git in coordinating the efforts of a team of analysts is only realized with a common repository for housing documents accessible by all. Often teams use Git in conjunction with GitHub, a private hosting service for remote repositories accessed via the internet. Git itself does not support the creation of remote repositories, and so a user must do so using GitHub (or some other hosting service) with the tools they provide.

3 Workflow using Git

- `git clone`.