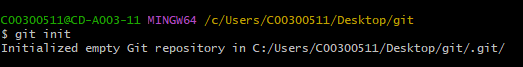
GitHub

Init

The way to start a project with git is to use git init. It starts a repository. There is a hidden directory called .git that initializes a repository. It stores all the objects and refs used in a git project. It is separated from the Git repository.

 There are a few uses of. git. To transform a directory to a git repository but using git init. When transforming a directory into a git repository one would use git init <directory>. To create a new repository, one would use git int –bare.



 Add

Git add is a command that changes the files, it also can add new files in your working directory. It put the files in the Git staging area. It gives the freedom to change and save files while you are working on a project. Git add gives you the ability to change as shape your commits and history.

There are a few different examples of git add. To stage a directory or file one would use git add<path>. To stage all files in a repository one would use git add. When interactively staging hunks of changes one would use git add –p.

Status

Git status is important as it outputs information without modifying or changing your local repository.

There are a few main ways to use this. Git status is how to display the base information. Git status –s is a condensed version of the information. Git status –v gives a more detailed (verbose) version of the information.

 Commit

Git commit creates a save point within the project. It can be used in different branches and can specify the changes to be made.

Different common uses for git commit. To start the commit process, one would use git commit, this will open the default text editor for you to create a commit message. Git commit –m “descriptive commit message” commit while including the message. Git commit –am “descriptive commit message” using this command will skip the staging part of the process. To change or replace a past commit with a new commit one would use git commit amend.

Remote

The command git remote tracks the set of remotes in your local repository.

The common commands for this include: git remote –v which takes the current remotes in the local repository and list them. Git remote ass [name] [URL] adds a new remote. Git remote remove [name] removes the remote.

Push

To upload the local branches to the corresponding remote branch one would use git push. Git push updated the remote branches with the local commits.

Some common uses for git push: git push –u origin [branch] takes one's local branch and adds an upstream tracking branch. Git push –-all , it push all branches. Git push –tag, this publishes tags that are not in a remote repository yet.

Branch

A branch is a subsection of a repository. It is used to build a feature to then commit to the main branch. It can be used in isolation from a project.

Checkout

Checkout lets you switch between different branches in the repository for revision. It updates the files in the directory to match the newest ones stored in that branch and records the new commits.

Merge

A merge in git hub is when one takes and isolated development and then integrate it into a branch.

Log

In git hub a log is used to review the past actions taken. It keeps track of what actions have been taken, by whom and when.

Clone

Git clone is used to create a copy of a repository or branch. It is used to make changes before committing it.

Some examples of git clone include: git clone [url] Downloads a clone of a repository. Git clone --mirror this clone over the repository without the ability to edit. To clone a single branch, one would use git clone –single-branch.

Pull

To update the local working branch and remote tracking branches one would use git pull. It is one of the 4 remote operations. Git pull is important to use every time you interact with the repository.

Ways one can use git pull. To update one's working branch git pull is used. This also updates all the remote tracking branches. Git pull – rebase rewrites history while updating the local working branch with commits from the remote. To fetch a particular remote tracking branch while using <refspec> one would use git pill –force. To fetch all the remotes, one would use git pull –all.

Stash

To stash in github is to keep the changes you are to make in isolation and to not commit them until later. It can also revert the changes back to the previous commit save point.

Rm

The command rm is used to remove files. It can remove one file or a collection. Its main function is to remove tracked files from the git index but can also remove them from the working directory and staging index.