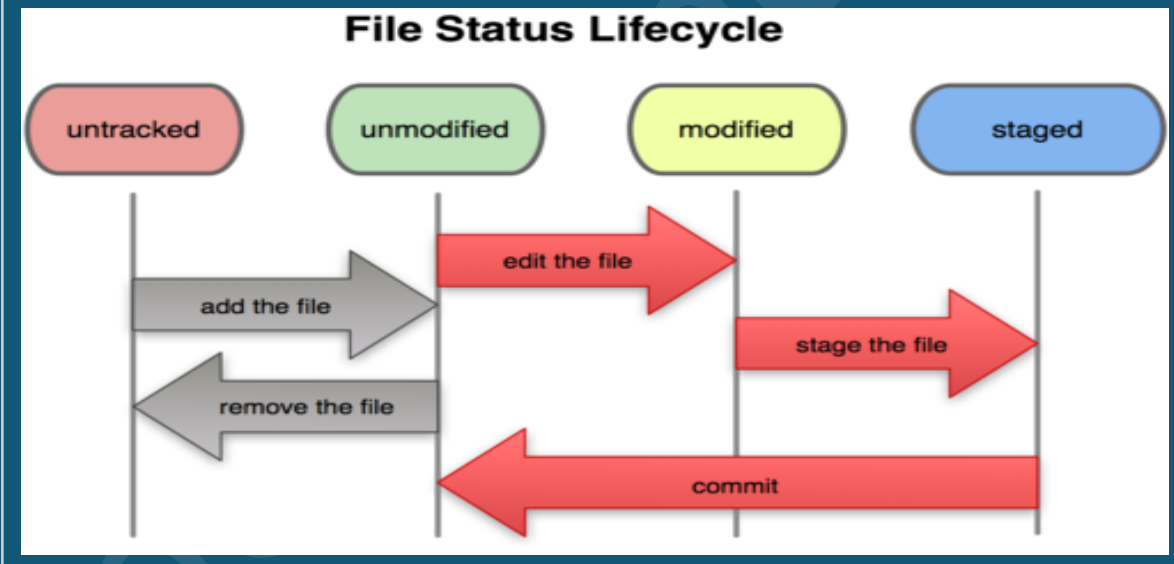
Git | GitHub Handbook

Command	Use
# How to install git in your machine? **# Installation for Windows / Mac O.S. / Linux / Unix Machine**	1. **Download Git setup** from the following url and install it in your machine. **https://git-scm.com/downloads** 2. Once Git is installed, **you can check the Git version** by following command. **git --version** 3. **If you want any help in Git**, use the following command. **git help**
# How to create a GitHub Account?	1. You can create a **GitHub account** by following the URL. **https://github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home**
# How to configure a GitHub account?	1. **Open GitBash in your local machine** and type following command. **git config user.name <your git username>** **(to set up your GitHub username)** ⇩ **e.g. git config --global user.name mrdeployment** **git config user.name** **(to check username is configured or not)** **git config --global user.email <your github registered email>** ⇩ **(to set up your GitHub registered email)** **e.g. git config --global user.email mrdeployment@myyahoo.com** **git config --global user.email** **(to check email is configured or not)**

Important Git | GitHub commands

git init	• This command is used to **initialize an empty repository.**
ls-lart	• This command is used to **display all hidden folders**.
git status	• This Git command used to **quickly check the status** of your Git repository. • It shows **which files have been modified**, **which files are staged for the next commit, and which files are untracked.** • It helps you to keep track of your changes and progress.



Few terminologies to know

Untracked : File which doesn't have any relation with git. It means you cannot track this file with the help of git.

Unmodified : It means Git says your **file is ready to commit.**

Modified : It means any **changes happens after staged or commit, can be re-pushed for commit**.

Staged : It is like a **plan where you want to push file for commit. Or** We can say, the file which is staged is ready to commit.

git add	• This command is used to **add a staged file for commit.** e.g. **git add index.html** (index.html is file which is to be added to commit)
git add -A	• This command is used to **add all staged files together to commit.** e.g. **git add -A**
git commit	• The command itself says the meaning, **It is used to commit files.** **Note** : When you do first commit, it will open vim editor , here type following message on top **"initial commit"** to show what changes you made.
Note : It is suggested that, keep using the git status command to know the changes.	
git commit -m "your message"	• This command is used to **commit without having/opening vim editor.** e.g. **git commit -m "my second commit"**
git chckout <filename>	• This command is used to **go back to your file's previous action. It works like an undo command.** e.g. **git checkout index.html**
git checkout -f	• This command is used to **go back to all of your file's previous actions together.** e.g. **git checkout -f**
git clean	• **Removes untracked files** from the working directory.
git log	• This command is used to **see what commit you have made.** • **In short, your commit details.** e.g. **git log**

git log -p -n	• This command is used to **filter your last "n" numbers of commits.** e.g. **git log -p -5** (it will show the last 5 commits details)
git log - - oneline	• This command is used to see all commits in one line.
git diff	• It will **compare the working tree with the staging area.** e.g. **git diff**
git diff - -staged	• It will **compare the staging area with the last commit.** **e.g. git diff - -staged**
git rm <filename> **git rm - -cached <filename>**	• This command **will remove files from the git stage as well from your local machine.** e.g. **git rm index.html** • This command **will remove files only from the git stage but remain available in your local machine.** e.g. **git rm - -cached index.html**
git status -s	• This command will **summarize the status of git.** • It is also called **short status**. M - It indicates your file has been **modified in the working tree.** M - It indicates your file has been **modified in the staging area.** MM - It indicates your file has been **modified in the staging area as well as the working tree.**
.gitignore	• This file can be created **to ignore specific files** while we commit anything or clone a repository. • e.g. **mylogs.log** - create this file and mention mylogs.log in the **.gitignore** file. When you commit anything, whatever you have mentioned in mylogs.log that files will be ignored.

	• Same way if you want to ignore all logs file, mention ***.log** Similarly, if you can ignore all python files by mentioning ***.py** If you want to ignore any specific directory, mention **mydirectory/**
git branch <branchname>	• This command is used to **create a new branch.** e.g. **git branch feature**
git checkout <branchname>	• If you want to **switch to another branch**, **git checkout** command is used. e.g. **git checkout main**
git merge <branchname>	• If you **want to merge any specific branch to the main branch, git merge** command is used. e.g. **git merge feature**
git branch	• This command is used to see, **currently you are in which branch.** **e.g. git branch** *** master** **feature** **It indicates, currently you are in master branch**
git branch -b <branchname>	• This command is used to **create a new branch and instantly switch to it without having the git checkout command.** e.g. **git branch -b newbranch**
git pull	• When you **update your local repository from remote. you can use this command.** e.g. **git pull**
git push	• This command is used in Git to **send your local commits to a**

	remote repository. **e.g. git push origin mybranch**
git stash	• When you want to **pause your current work without committing it**, you can use this command. e.g. **git stash** • **git stash** is handy for **when you need to switch tasks or branches without committing half-done work.**
git clone	• This command is used to clone (**copy) your remote repository to local machine or cloud machine.** **e.g.** **git clone https://github.com/asfaqshekh/DevOps-Master-4sure.git**
git reset	• This command is used to remove last commit and if it is having bug and push git head (pointer) to last to last commit (no bugs) e.g. **git reset - - hard <commitid>** Note : It is used in local machine in real environment practice
git revert	• This command will create a new commit by passing on the next commit (bugs) and function it properly. **e.g. git revert <commitid>** Note : This command is used in a remote repository in a real environment.