1.) Git means

Command

Basic

YouTube video

Git command

cd

cd /c

ls

cd git

ls

git config --global user.name"priya-787"

git config --global user.email saiwaghamode787@gmail.com

git clone https://github.com/priya-787/python.git

ls

cd python3

git add py3.txt

git commit -m " first commit" py3.txt

git push -u origin master

Give user name, password.....and refresh Page

1. **SETUP & INIT**

**git config-global username [first name last name]**

>>set name that is identifiable for credit when reviewing version history

**Git config-global usermail [valid email]**

>>> set an email address that will be associated with each history maker

**git config-global color UI auto**

>> set automatic command line for git for easy writing

**git init**

>>>initialize an existing repository as a git repository

**git clone [URL]**

>>> retrieve an entire repository from a hosted location via URL

**Git status**

>>displays the state of the working directory

**touch README.md**

**>>> to create file**

1. **STAGES & SNAPSHOTS**

**git branch**

>>> list your branch a will appear next to the currently active branch

**git branch [branch-name]**

>>create a new branch at current commit

**git checkout**

>> switch new another branch and check it out

**git merge**

>> merge the specified branches history into the current one

**git diff-staged**

>>>show modified files in working directory staged for your next committed

**git commit [descriptive message]**

>>commit your staged content as a new commit snapshot

1. **BRANCH & Merge**

**git branch**

>> show modified files in working directory staged for your next comment

**Git add [file]**

>>add the file as it looks wow to your next commit

**git reset [file]**

>>unstage a file while remaining the changes in the working directory

**git diff**

>> diff of what is staged but not yet committed

1. **INSPECT & COMPARE**

**git log**

>>show all commits in the current branch history

**git log branch\_B…branch\_A**

>>> show all commits that are in branch A that are not on branch B

**git log -- stat-M**

>>show all commit logs with an indication of any path that move

1. **REWRITE History**

**git rebase [branch]**

>>> apply any commit to the current branch ahead of specified one

**git reset-hard [commit]**

>>> show all commits that are in branch A that are not on branch B

1. **IGNORING PATTERNS**

**Logs/ notes/ pattern**

>>save files with desired patterns as git ignore with either direct string matches

**git config-global core excludes file [file]**

>>system-wide ignore pattern for all local repositories

1. **Temporary commits**

**git stash**

>>save modified and staged changes git

**git stash list**

>> list stack order of stashed file changes

**git stash pop**

>>write working from top of stash stack

**git log -- follow [life]**

>>show the commits that changes file even across rename

**git show [SHA]**

>>show any object in git in human readable format

1. **SHARE AND UPDATE**

**git remote add [alias] [URL]**

>>add git as alias

**git fetch [alias]**

>>fetch down all the branches from that git remote

**git merge [alias]/[branch]**

>>>merge a remote branch into your current branch to bring it up to date

**git push [alias] [branch]**

>>transmit local branch commit to the remote repository branch

**git pull**

>>fetch and merge any commit from the tracking remote

**9. TRACKING PATH CHANGE**

**git rm [file]**

>>delete the file from project and stage removal for commit

**git mv [existing path] [new path]**

>> change the existing file path and stage the move