

Git Commands

- **git config**

- Usage: `git config --global user.name "[name]"`

- Usage: `git config --global user.email "[email address]"`

- This command sets the author name and email address respectively to be used with your commits.

- **git init**

- Usage: `git init [repository name]`

- This command is used to start a new repository.

- **git clone**

- Usage: `git clone [url]`

- This command is used to obtain a repository from an existing URL.

- **git add**

- Usage: `git add [file]`

- This command adds a file to the staging area.

- Usage: `git add *`

- This command adds one or more to the staging area.

- Usage: `git add .`

- This command add the all the files present in the directory.

- Usage: `git add -p`

- This command add the file, after approving/giving the permission.

- **git commit**

- Usage: **git commit -m "[Type in the commit message]"**

- This command records or snapshots the file permanently in the version history.

- Usage: **git commit -a**

- This command commits any files you've added with the git add command and also commits any files you've changed since then.

- Usage: **git commit --amend**

- This command is used to override/overwrite the previous commit that you have made, use of this is only when you have to change/edit the previous commit. {avoid amending commits that have already been made public.}

- **git diff** {<https://man7.org/linux/man-pages/man1/diff.1.html>}

- diff is used to find differences between two files. On its own, it's a bit hard to use; instead, use it with diff -u to find lines which differ in two files:

- Usage: **git diff**

- This command shows the file differences which are not yet staged.

- Usage: **git diff --staged**

- This command shows the differences between the files in the staging area and the latest version present.

- Usage: **git diff [first branch] [second branch]**

- This command shows the differences between the two branches mentioned.

- Usage: **git diff --staged**

- This return the staged differences on the repo.

- Usage: **git diff filename1.extension filename.extension**

- Returns all the differences in the code on the command window.

- Usage: **git diff -u**

- diff -u is used to compare two files and give you a detailed report ..i.e comparison is done line by line.

- Usage: **git diff -u filename1.extension filename.extension**

- Returns all the detailed differences in the code after comparing the above files on the command line.

- Usage: **git diff -u filename.extension filename1.extension > filename3.diff**

- This function will detect the changes that are there in the files (filename, filename1) and store the changes made in them in filename3.diff.

- **git reset**

- Usage: **git reset [file]**

- This command unstages the file, but it preserves the file contents.

- Usage: **git reset [commit]**

- This command undoes all the commits after the specified commit and preserves the changes locally.

- Usage: **git reset --hard [commit]**

- This command discards all history and goes back to the specified commit.

- **git rebase**

- Usage: **git rebase [branch]**

- This command is used when you want to keep a linear commit history..i.e it moves the current branch on top of the branch name mentioned on the command.

- Usage: **git rebase -i**

- This command is used to pop up a screen where upon you can select various options for merging/commit.

- **git status**

- **Usage: git status**

- This command lists all the files that have to be committed.

- **git revert**

- **Usage: git revert HEAD**

- This command is used to revert back the previous commit, by creating a commit that is exactly opposite to the mentioned HEAD commit..i.e if in your previous commit there's a error(line), then in the new commit created the error line would be removed.

- **git rm**

- **Usage: git rm [file]**

- This command deletes the file from your working directory and stages the deletion.

- **git log**

- **Usage: git log**

- This command is used to list the version history for the current branch.

- **Usage: git log -follow[file]**

- This command lists version history for a file, including the renaming of files also.

- **Usage: git log -p**

- This command list all the deatiled info of the logs for the current branch.

- **Usage: git log --stat**

- This command show all the statistics for the commits done on the cuurent branch.

- **Usage: git log --graph --oneline**

- This command shows a summarized view of the commit hostory for a repo.

- **git show**

- Usage: **git show [commit (id)]**

- This command shows the metadata and content changes of the specified commit.

- **git mv**

- Usage: **git mv filename1.ext filename2.ext**

- This command renames the file filename1 as filename2 and also can perform the moving of the files from one directory to other.

- **git tag**

- Usage: **git tag [commitID]**

- This command is used to give tags to the specified commit.

- **git branch**

- Usage: **git branch**

- This command lists all the local branches in the current repository.

- Usage: **git branch [branch name]**

- This command creates a new branch.

- Usage: **git branch -d [branch name]**

- This command deletes the feature branch.

- Usage: **git branch -D [branch name]**

- This command forcibly deletes the feature branch.

- **git checkout**

- **Usage: git checkout/reset [filename]**

- This command reverts back all the changes to the previous commit on the current branch.

- **Usage: git checkout [branch name]**

- This command is used to switch from one branch to another.

- **Usage: git checkout -b [branch name]**

- This command creates a new branch and also switches to it.

- **git merge**

- **Usage: git merge [branch name]**

- This command merges the specified branch's history into the current branch.

- **Usage: git merge --abort**

- If there are merge conflicts (meaning files are incompatible), --abort can be used to abort the merge action...i.e reverts back to the previous commit of the master branch.

- **git remote**

- **Usage: git remote**

- This command lists remote repos.

- **Usage: git remote add [variable name] [Remote Server Link]**

- This command is used to connect your local repository to the remote server.

- **Usage: git remote show origin**

- This command is used to display all the info of the local and the remote repo.

- **Usage: git remote show [name]**

- This command describes a single remote repo.

- **Usage: git remote update**

- This command fetches the most up-to-date objects.

- **Usage: git remote -v**

- This command lists remote repos verbosely.

- **git push**

- **Usage: git push [variable name] master**

- This command sends the committed changes of master branch to your remote repository.

- **Usage: git push [variable name] [branch]**

- This command sends the branch commits to your remote repository.

- **Usage: git push -f**

- This command forcefully sends the branch commits to your remote repository.

- **Usage: git push --all [variable name]**

- This command pushes all branches to your remote repository.

- **Usage: git push [variable name] :[branch name]**

- This command deletes a branch on your remote repository.

- **git pull**

- **Usage: git pull [Repository Link]**

- This command fetches and merges changes on the remote server to your working directory.

- **git stash**

- **Usage: git stash save**

- This command temporarily stores all the modified tracked files.

- **Usage: git stash pop**

- This command restores the most recently stashed files.

- **Usage:** `git stash list`

- This command lists all stashed changesets.

- **Usage:** `git stash drop`

- This command discards the most recently stashed changeset.

- **cat filename.extension**

- Returns/Shows all the value inside the file ..on the command window.

- **Patch** {<https://man7.org/linux/man-pages/man1/patch.1.html>}

- Patch is useful for applying file differences. See the below example, which compares two files. The comparison is saved as a .diff file, which is then patched to the original file

- **patch filename1.extension < filename2.diff**

- Patches all the differences that has been made on filename2.ext to filename1.ext.

****git:** an open source, distributed version-control system

****GitHub:** a platform for hosting and collaborating on Git repositories

****commit:** a Git object, a snapshot of your entire repository compressed into a SHA

****branch:** a lightweight movable pointer to a commit

****clone:** a local version of a repository, including all commits and branches

****remote:** a common repository on GitHub that all team member use to exchange their changes

****fork:** a copy of a repository on GitHub owned by a different user

****pull request:** a place to compare and discuss the differences introduced on a branch with reviews, comments, integrated tests, and more

****HEAD:** representing your current working directory, the HEAD pointer can be moved to different branches, tags, or commits when using `git checkout`.

****fetch:** is the command that tells your local git to retrieve the latest meta-data info from the origin.

****continuous integration:** is basically a third party app to deploy your code. (travis-ci)

**** touch filename.extension** creates a new file in the command window.

****vimdiff filename.ext filename1.ext** command highlights the words that were changed in a file instead of working line by line.

****Diagram on how stuff works:**

