**GIT**

Other Git commands link: <https://www.atlassian.com/git/glossary>

<https://phoenixnap.com/kb/how-to-enable-ssh-centos-7#ftoc-heading-1>

**What is Git and why it is used?**

Git is a **DevOps tool used for source code management**. It is a free and open-source version control system used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development.

IMPORTANT LINKS

1. LearnWebCode link

<https://www.youtube.com/user/LearnWebCode/playlists>

1. GIT (LearnWebCode link) video training

<https://www.youtube.com/watch?v=9GKpbI1siow&list=PLpcSpRrAaOarEpNz71TSfNVd0eQmsZSgN>

**GIT COMMANDS**

[joshnh/Git-Commands: A list of commonly used Git commands (github.com)](https://github.com/joshnh/Git-Commands)

**Getting & Creating Projects**

| **Command** | **Description** |
| --- | --- |
| Git config –global user.name “YourName” | Let Git know about your name to track changes made by you |
| Git config –global user.name “YourEmail@address.com” | Let Git know about your email address |
| git init | Initialize a local Git repository. From now on, Git will keep tracking anything that happens within the current directory. In other words, we are creating a brand new repository (or repo). This is how you create a repo. |
| git clone ssh://git@github.com/[username]/[repository-name].git | Create a local copy of a remote repository |

**Basic Snapshotting**

| **Command** | **Description** |
| --- | --- |
| git status | Check status |
| git add [file-name.txt] | Add a file to the staging area |
| git add -A | Add all new and changed files to the staging area |
| git commit -m "[commit message]" | Commit changes |
| git rm -r [file-name.txt] | Remove a file (or folder) |

**Branching & Merging**

| **Command** | **Description** |
| --- | --- |
| git branch | List branches (the asterisk denotes the current branch) |
| git branch -a | List all branches (local and remote) |
| git branch [branch name] | Create a new branch |
| git branch -d [branch name] | Delete a branch |
| git push origin --delete [branch name] | Delete a remote branch |
| git checkout -b [branch name] | Create a new branch and switch to it |
| git checkout -b [branch name] origin/[branch name] | Clone a remote branch and switch to it |
| git branch -m [old branch name] [new branch name] | Rename a local branch |
| git checkout [branch name] | Switch to a branch |
| git checkout - | Switch to the branch last checked out |
| git checkout -- [file-name.txt] | Discard changes to a file |
| git merge [branch name] | Merge a branch into the active branch |
| git merge [source branch] [target branch] | Merge a branch into a target branch |
| git stash | Stash changes in a dirty working directory |
| git stash clear | Remove all stashed entries |

**Sharing & Updating Projects**

| **Command** | **Description** |
| --- | --- |
| git push origin [branch name] | Push a branch to your remote repository. Origin stand for address of repo in github.com (exple origin: https://github.com/vidalgithub/travel.git) |
| git push -u origin [branch name] | Push changes to remote repository (and remember the branch) |
| git push | Push changes to remote repository (remembered branch) |
| git push origin --delete [branch name] | Delete a remote branch |
| git push --force or git push --force-with-lease | Force the push command |
| git reset | git reset. This gets you back to the situation in which you have replaced your old-and-incorrect commits with your new-and-improved copies—i.e., back to where you were before you ran git pull |
|  |  |
| git pull | Update local repository to the newest commit |
| git pull origin [branch name] | Pull changes from remote repository |
| git remote add origin ssh://git@github.com/[username]/[repository-name].git | Add a remote repository |
| Git remote -v | Git will show the server address (origin) of where it thinks you want to push to. This is the location for the starter (origin) file repo |
| git remote set-url origin ssh://git@github.com/[username]/[repository-name].git | Set a repository's origin branch to SSH |

**Inspection & Comparison**

| **Command** | **Description** |
| --- | --- |
| git log | View changes  (to view the commit history and make sure it’s what you want) |
| git log --summary | View changes (detailed) |
| git log --oneline | View changes (briefly) |
| git diff [source branch] [target branch] | Preview changes before merging |

1. GIT Tutorial 2

<https://www.youtube.com/watch?v=n-p1RUmdl9M&ab_channel=LearnWebCode>

A screenshot of a computer

Description automatically generated with low confidence

Graphical user interface

Description automatically generated with low confidence

A picture containing text, sign

Description automatically generated

Graphical user interface

Description automatically generated

TUTO 1

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Text

Description automatically generated



**What is the purpose of a readme file?**

The Readme file is often the first file which the users read. It is a text file **that contains the information for the user about the software, project, code, game, or it might contain instructions, help, or details about the patches or updates**.

A Readme file is a file that **describes the purpose of the repository and gives hints on what the code does, how to compile/use it, etc**. The . md extension stands for Markdown, which is a type of file that is both readable in plain text, but can also be easily converted to HTML to display special elements.

Does GitHub require a readme file?

A good README is for others to understand what our code includes, and why it's noteworthy. **A README file is also essential to retreive a project - on GitHub** but also in browsers (e.g. Google).

What does README do in GitHub?

README (as the name suggests: “read me”) is the first file one should read when starting a new project. It's **a set of useful information about a project and a kind of manual**. It is the first file Github or any Git hosting site will show when someone opens your repository.