Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

GitHub is the largest host of source code in the world, and has been owned by *Microsoft* since 2018

For Windows, you can use Git bash, which comes included in Git for Windows. For Mac and Linux you can use the built-in terminal.

README.md file describes the repository (recommended for all repositories):

git –version

git config --global user.name "username"

git config --global user.email "email"

Use global to set the username and e-mail for **every repository** on your computer.

If you want to set the username/e-mail for just the current repo, you can remove global

git init

git status

we will use the --short option to see the changes in a more compact way:

git status --short

Note: Short status flags are:

* ?? - Untracked files
* A - Files added to stage
* M - Modified files
* D - Deleted files

git diff HEAD

git diff - - staged

git diff filename

git diff –cached filename

git rest filename (exit from stage)

git checkout - - filename (return file to last commit)

In Git, a branch is a new/separate version of the main repository.

git branch

git branch -d branchname (delete a branch)

git branch newbranchname

git checkout newbranch (entering to new branch)

Using the -b option on checkout will create a new branch, and move to it, if it does not exist

git merge newbranchname (merges newbranch with current)

git rm file (removes file from file system and git)

git clone <https://github.com/jadijadi/linuxandlife.git>

git push origin master (Push local updates to the main project)

git pull origin master (Pull the latest version of the project to a local copy)

git remote add origin https://....

Git show commit\_id

Git tag -a v2.0 -m 'comment'

Git tag -a v1.8 -m d2265db3d2 (tag previous commits)

git tag (shows tags)

git tag -l "v\*" shoes tags which start with v

git show v1.8

git push origin v1.8 //pushing tags to origin

git push origin –tags

git checkout v1.8

git log (To view the history of commits for a repository)

git help --all

git commit -help

Track new files 🡪 Change files 🡪 stage 🡪 commit

git add -A 🡪 change files 🡪 git add -A 🡪 git commit -m 'comments go here'

As you are working, you may be adding, editing and removing files. But whenever you hit a milestone or finish a part of the work, you should add the files to a Staging Environment.

**Staged** files are files that are ready to be **committed** to the repository you are working on.

Since we have finished our work, we are ready move from stage to commit for our repo. Git considers each commit change point or "save point". It is a point in the project you can go back to if you find a bug, or want to make a change.

adding custom files

* page1.htm
* page2.htm
* page3.htm
* git add 'page\*'
* git add -A = git add --all

Sometimes, when you make small changes, using the staging environment seems like a waste of time. It is possible to commit changes directly, skipping the staging environment. The -a option will automatically stage every changed, already tracked file.

git commit -a -m "Updated index.html with a new line"

Skipping the Staging Environment is not generally recommended.

Skipping the stage step can sometimes make you include unwanted changes