

Git Course

ls - list files

mkdir - new directory

cd - to open files

rm - remove file

cd ../ - takes to previous repo

1. Create Repository

git init : to initialize the master git in the respective repo

2. git clone (link)

used to copy repos from the website to the system local directory.

3. git status

on branch master -is the default branch

Your branch is up-to-date with 'origin/master' - Because git-clone was used to copy this repository from another computer, this is telling us if our project is in sync with the one we copied from.

- `nothing to commit, working directory clean` – this is saying that there are no pending changes.

4. (a) git log

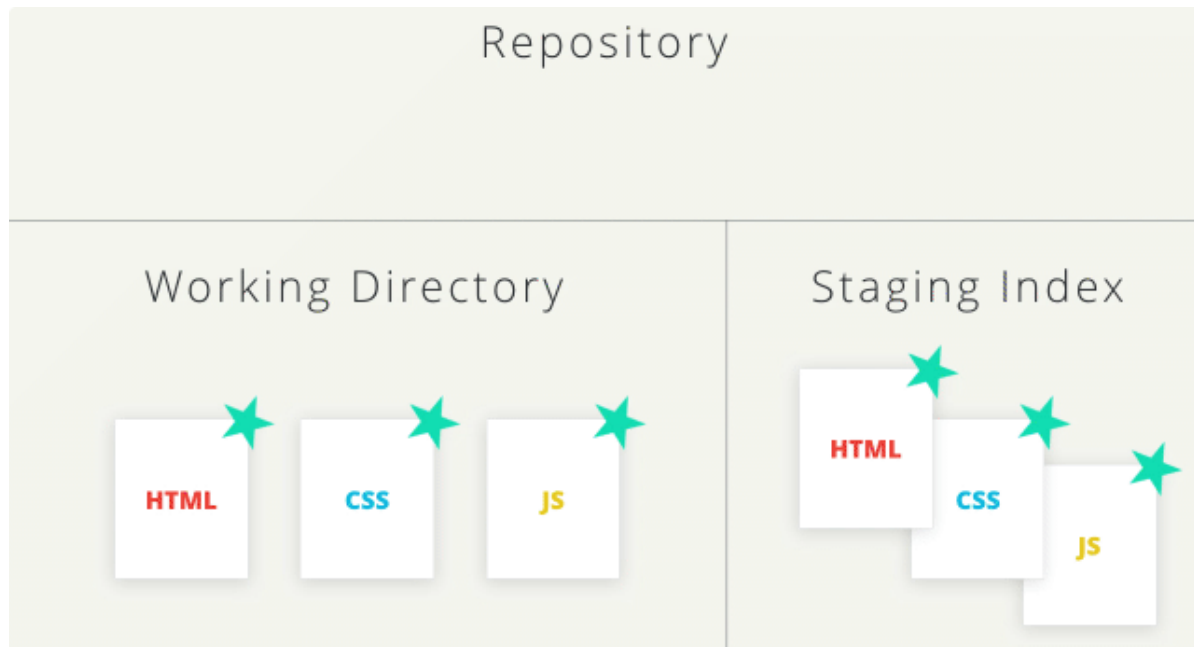
used to check for initial commits

Navigating The Log

If you're not used to a pager on the command line, navigating in **Less** can be a bit odd. Here are some helpful keys:

- to scroll **down**, press
 - `j` or `↓` to move *down* one line at a time
 - `d` to move by half the page screen

- **f** to move by a whole page screen
- to scroll **up**, press
 - **k** or **↑** to move *up* one line at a time
 - **u** to move by half the page screen
 - **b** to move by a whole page screen
- press **q** to **quit** out of the log (returns to the regular command prompt)
- (b) **git log --oneline**
displays the entire commit in one line
- (c) **git log --stat**
displays the files that have been changed in the commit
- (d) **git log -p**
this displays the exact what has been changed in all the respective commits.
to check before and that specific commit (command will *start at that commit*)
\$ git log -p fdf5493
it will *also* show all of the commits that were made *prior* to the supplied SHA.
- (e) **git show (SHA code)**
to check the specific commit
- (f) **git log --oneline --decorate --graph --all**
- - decorate displays all the tags , - - graph to display the branches , —all all branches.
- 4. **git add (file index)**
this takes the file from working directory to the staging index aka staging.



5. Commit

`git commit -m "Commit Name"`

6. Diff

`git diff`

used to check what changes are made that are not committed.

7. Touch

this creates a new file of any type

`touch mail.docx`

8. Git Ignore

`touch .gitignore`

this creates a ignore file

open this file in a regular file explorer and type in the name of the file which you don't want to track the versions > save > exit.

9. Tag

git tag -a v1.0

tagging a commit

to list all the tags

`git tag`

to display which tag is being used
`git log --decorate`

to delete a tag
git tag -d v1.0

provide SHA to tag that particular commit
\$ git tag -a v1.0 a87984

10. Branch

(i) `git branch`

lists all branches

(ii) Create a Branch

`git branch (branch name)`

(iii) Switch between branches

git checkout (branch_name)

(iv) **git log --oneline --decorate** shows what's the branch

(v) Delete branch

git branch -d sidebar

forced delete

git branch -D sidebar

(vi) Create and switch to new branch

\$ git checkout -b (new_branch_name)

`-b` is a flag to will create that new branch

11. Merge

git merge <name-of-branch-to-merge-in>

. When a merge is performed and fails, that is called a **merge conflict**. (*exact same line(s) are changed in separate branches.*)

12. Modifying The Last Commit

git commit --amend

rename your previous commit

13. **Revert**

git revert <SHA-of-commit-to-revert>

will undo the changes that were made by the provided commit ,creates a new commit to record the change