Git Course Summary

Initializing The Repository

- If you want to start a new repo locally: `git init`
- If you have a repo in github and you want to start working with it locally: `git clone <repo-url`

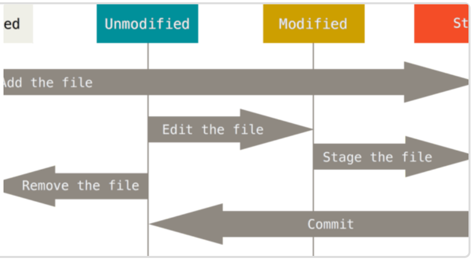
Committing Changes To The Repo

- Tracked vs Untracked Files

Git - Recording Changes to the Repository

At this point, you should have a bona fide Git repository on your local machine, and a checkout or working copy of all of its files in front of you. Typically, you'll want to start making changes and

https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository



- To check the current status of the repo: `git status`
- To get a short description of the status: `git status -s` or `git status --short`
- To add a file to the repo `git add <file-name>`
- To add a modified file to the staged files: `git add <file-name>`

- To make the commit `git commit -m "<commit-message>"`

Ignoring Files

- You can ignore files by adding them into `.gitignore` file.
- To add all the files with some extension like (ext), you can add the following line into `.gitignore` file: `*.ext`

Removing Files

- `git rm` removes the file from tracked files and remove it from the directory itself.
- To remove a file only from git: `git rm <file-name> --cach`

The History

- To get the history of the repo `git log`

EB: Working With Remotes

- to check the remote servers your repo is connected to, you can run `git remote`
- The default name of the remote for any cloned repo is `origin` ,however, you can change it.
- to show the urls of the remote, you can run: `git remote -v`
- Adding a remote by `git remote add <shortname> <url>`
- To pull the latest changes from the remote Repository: `git pull origin master`
- To push your changes to the remote repo `git push <remote-name> <branch-name>`
- Example: `git push origin main`

Branches

- to create a branch `git branch <branch-name>`
- to checkout to a branch: `git checkout branch <branch-name>`
- to remove a branch: `git branch -d <branch-name>`
- to create a branch and checkout to that branch in one command: `git checkout -b <branch-name>`
- To get the branches in the repo and to know in which branch you are in: `git branch`
- when there is a conflict and the merge is bending and you want to see which files are causing the conflict, you can run: `git status`