Git is use for collaborating on code project.

Can make a check point so when you mess up you can go back to the checkpoint.

Modify, change, improve the code.

A tool to protect you and others from you and others.

Locally workflow

Useful for when you don’t have wifi.

Git init create a repository

3 states

Modify state: new changes that not yet saved in Git

Staged: current version of the file

Committed: safely saved by Git

Git status > check the status of the file

Git add index.html > copy the file and ready to be stored

Git commit -m “description”

Move a copy not the actual file.

Git log > check how many copy do you have

Branches, merge and merge conflict

Remote repository -> a copy of the work store in “the cloud”

Share with other

Accessible anywhere with internet

Add remote repository

Push the change to the repository

Git push -u origin master

After the first push, it should be

Git push

Git push tell git to upload all the changes

Don’t need to be done after every commit

It upload every commit

Branches - smaller bit of chunk, represent part of code

Allow us to work on code fixes and features without breaking the code

Fix and feature should alway start on a branch

“Master” branch should only contain clean code and ready for deployment (clean and functional)

Git branch <name> tells git to maintain a new copy of our code

Git checkout <branch> tell git to switch working folder to the branch name specifically

Git merge <name> = to combine 2 differents branches

Merge conflict = file has changed in both of the branches

You try to combine but git can’t determine what to keep

The topic today can save me so much time when I try to combine 2 css files in the last project. Collaboration can be easier with this. It help people to work on separate functions of the web and not broke the website and keep the change for people to see and decide to keep it or not.

My understanding on this is 4

Best part of Thanksgiving is I finally finish Isle 2 in Cuphead