Gitting Started

Why do we use GitHub?

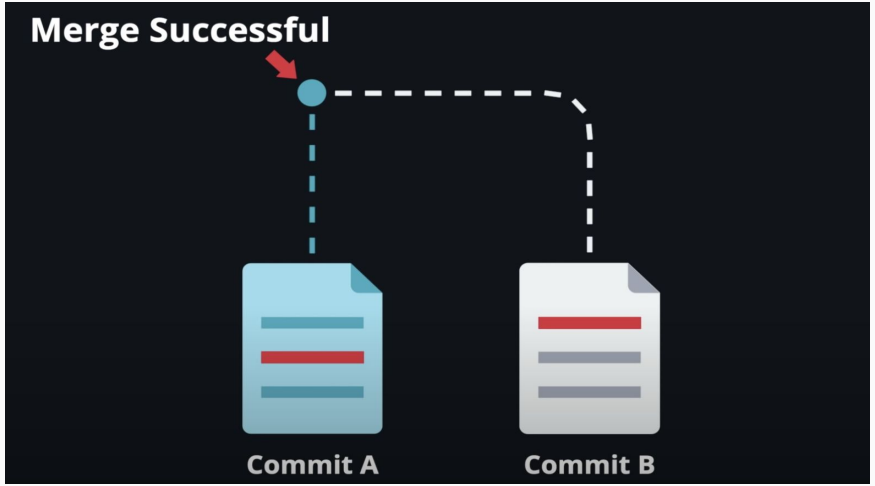
- Portfolio
- Collaboration
- Allows other people to see and implement your code!
- This is ***not*** a file hosting service!

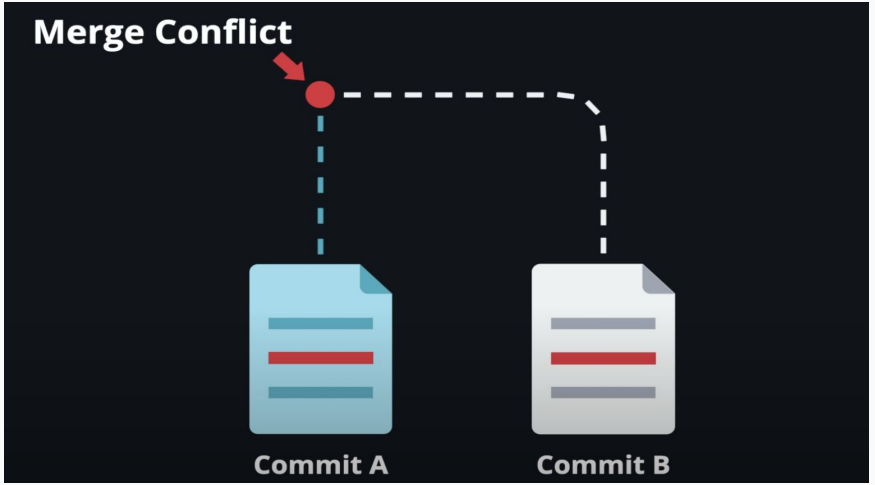
The importance of .gitignore files

- config files! We do not want anyone to be able to gain access to our API keys or passwords
- We have a limit to the amount of space GitHub can hold
- We will add a list of things to the lecture repo to add to your .gitignore!

It’s time to branch out!

- In group projects, branches allow us to work in parallel with our partners without risking any merge conflicts on our master branch
- What is a merge conflict?
- Pull requests!


Merge Successful
Commit A
Commit B


Merge Conflict
Commit A
Commit B

Time to get over commitment issues ☹️

- ADD AND COMMIT CONSTANTLY
- If your computer crashes and you have not added and committed your changes for multiple days, you could lose everything!
- Would you work on an essay for a week and only save it once?

Commit Messages 101

- Messages should be no more than 50 characters and explain *what* you did, not *how* you did it
- Examples of good commit messages:
  - “clean data”
  - “run descriptive stats”
- You’ve just scraped a bunch of data and formatted it properly into a dataframe. It’s time to commit! What does your message say?

Collaborating on Git!

Steps

1. ***One partner*** initializes a new repository with a README.md ***and*** a Python .gitignore
2. Add your partner(s) as collaborator(s)
3. Each partner clones down the repository (don’t fork!)
4. Create your own local branches
5. Add, commit, and push changes to your branch
6. When ready, create a pull request on GitHub to merge all files!

Time for a demo!

Things to watch out for

- Never work off the same Jupyter Notebook
- Be careful with how you name files - files with the same name will cause merge conflicts when we create pull requests to the master branch

What do we expect from you?

- A clean and annotated presentation Jupyter Notebook! This should not include any data collection or cleaning → anyone should be able to look at your code and be able to replicate your project with their own data.
  - Markdown cells for explanations/steps or lengthy comments
  - Commented code for short explanations
  - Docstrings!
- A descriptive and clear README.md
  - Think essay outline: the data, the question, the techniques, the findings, next steps
- A formal powerpoint presentation