PHYS 6260 — Introduction to Github and Python

Announcements:

- HW1: due this Friday (already posted on canvas)

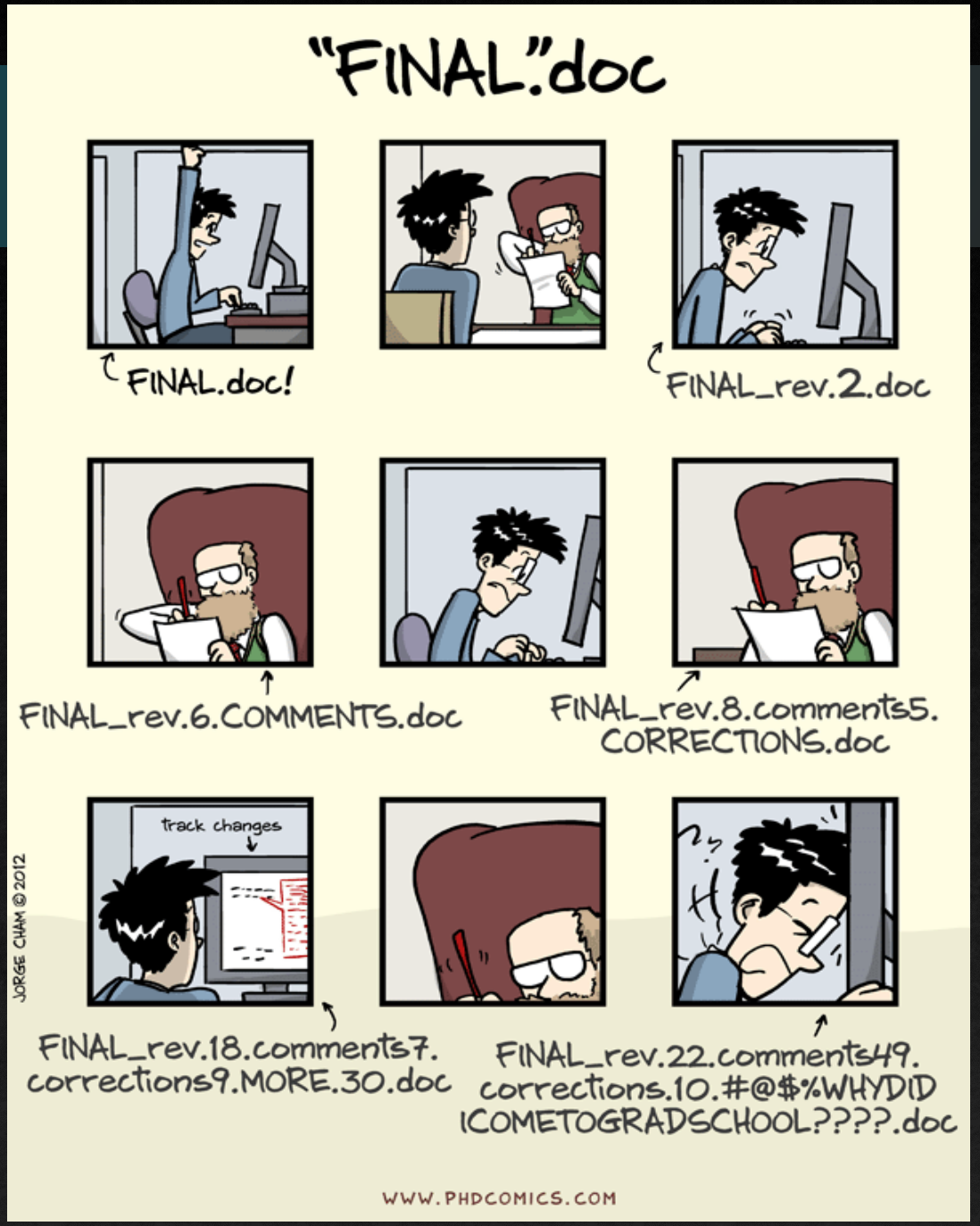
Lecture Outline

We will cover these topics

- Basics to Github
- Python packages
- In-class problems

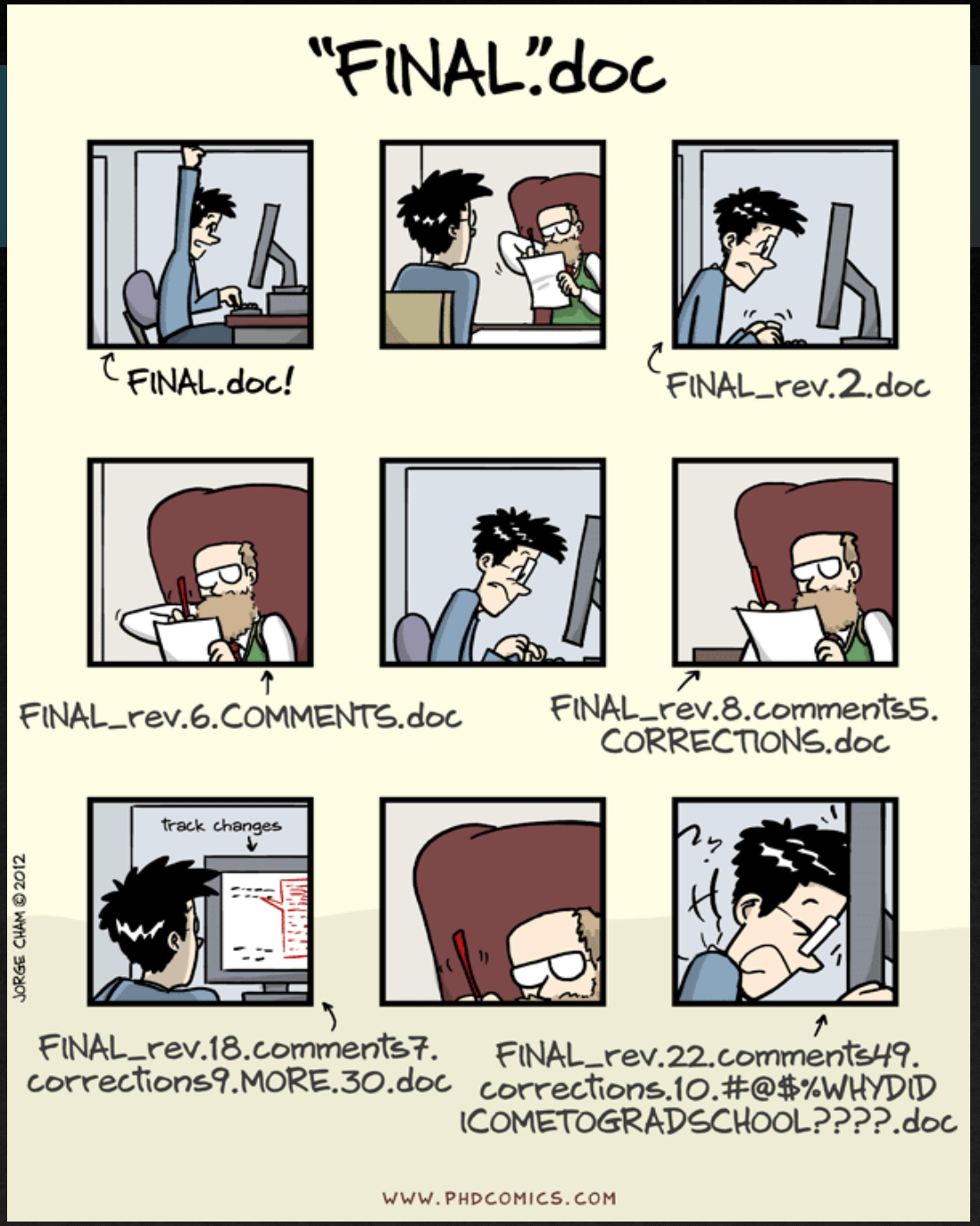
Why Version control?

- Orderly process
- Clear history of versions
  - Easy to revert errors
  - Automatic backup
- Easy to collaborate

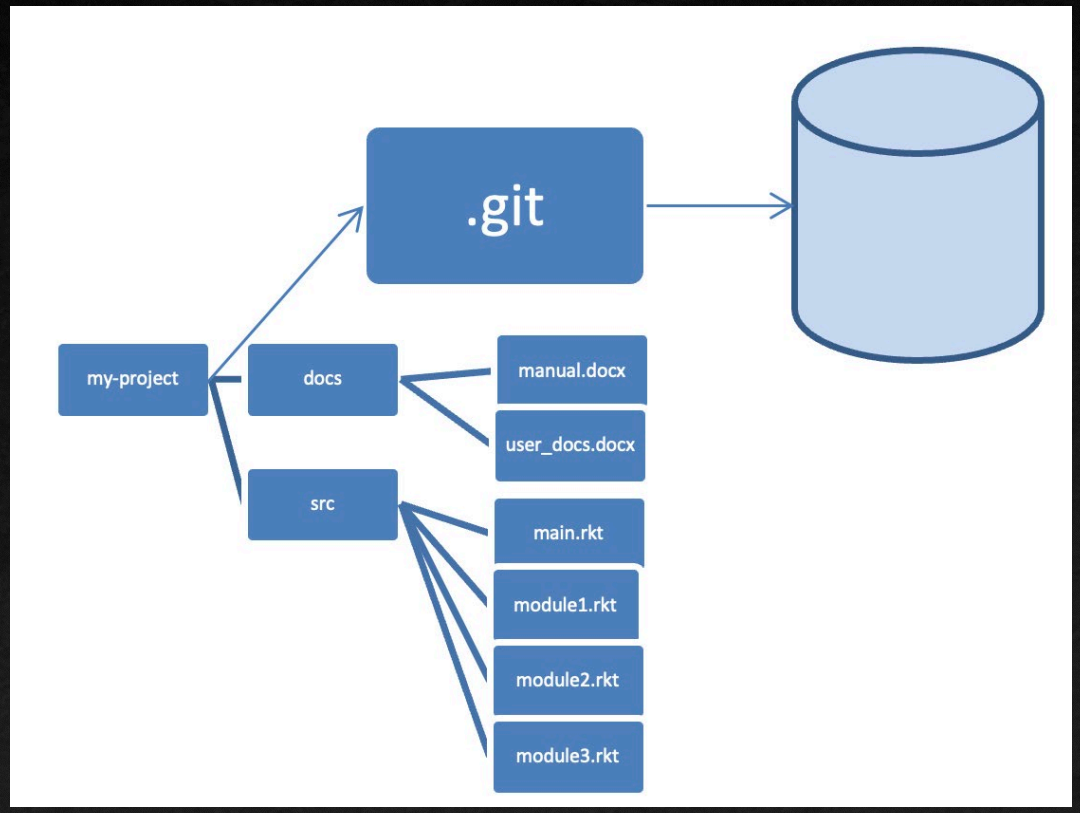


Git vs. Github

- Git:
  - Version control
    - => collaboration is more transparent
  - Multiple parallel branches
  - Able to handle large projects
- Github:
  - Uses Git version control
  - Free space for open-source projects
  - Web-interface for real-time collaboration

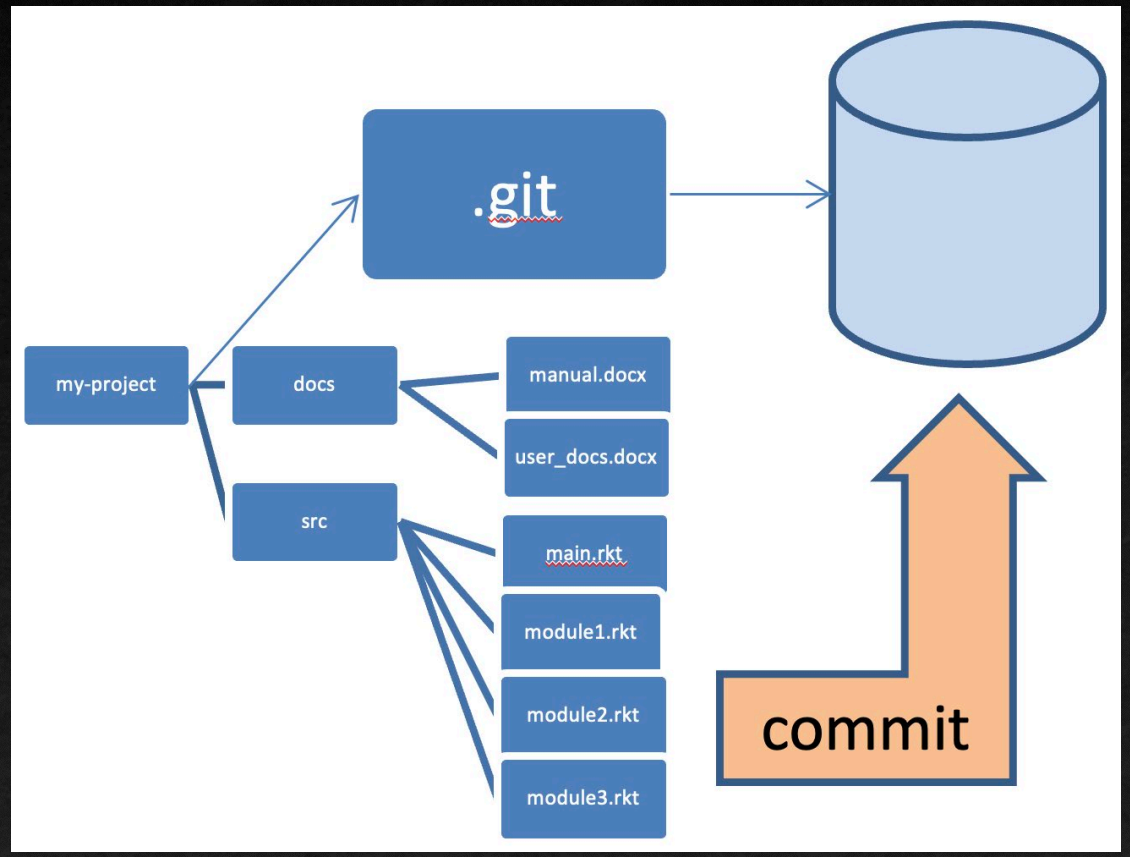


Git Client



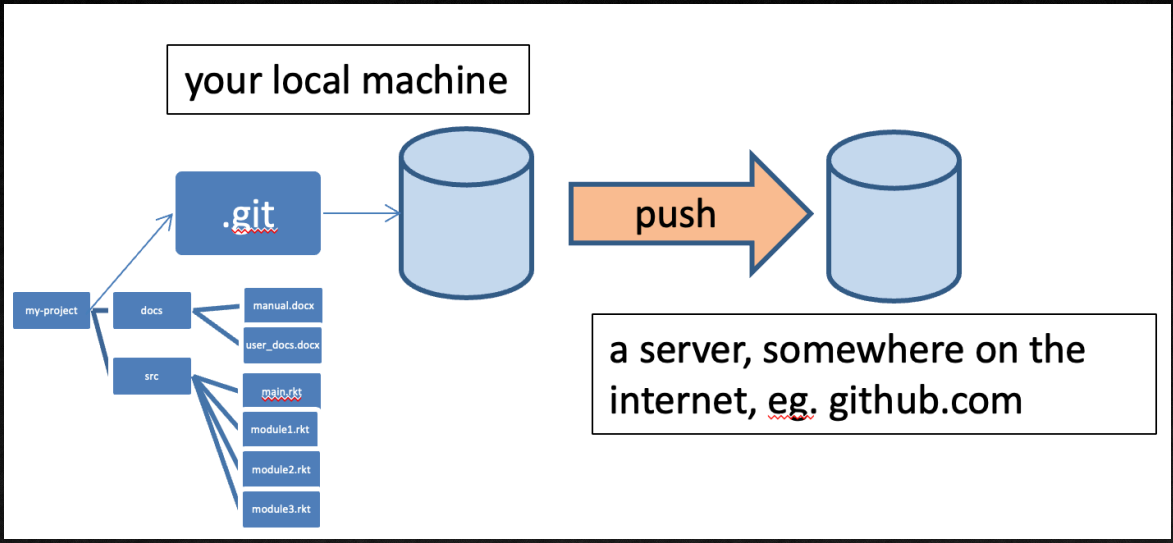
- The job of the git client is to manage the mini-filesystem
- You edit your local files directly
  - You can edit, add files, delete files, etc., using whatever tools you like
  - This doesn't change the mini-fs repo, so now your mini-fs is behind

Commit



- When you do a “commit”, you record all your local changes into the mini-fs.
- Nothing is ever over-written there, so everything you ever commit can be recovered.

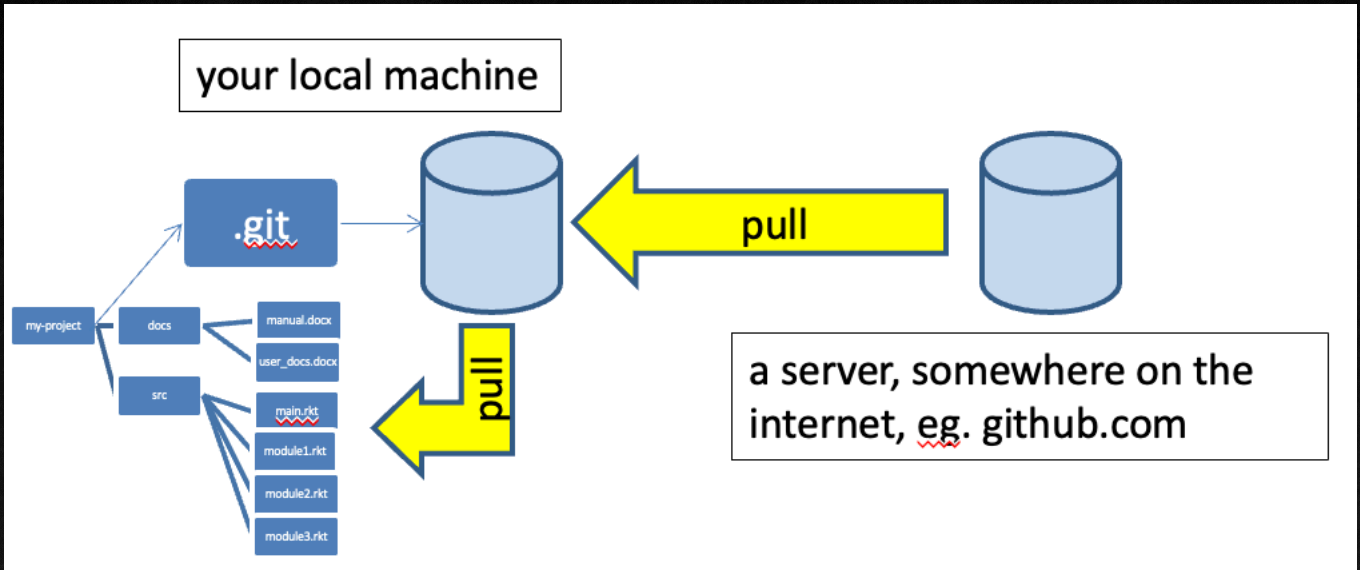
Synchronizing with the server (1)



`git push`	push your new branches and data to a remote repository

- At the end of each work session, you need to save your changes on the server. This is called a "push".
- Now all your data is backed up.
  - You can retrieve it, on your machine or some other machine.
  - We can retrieve it (that's how we collect homework)

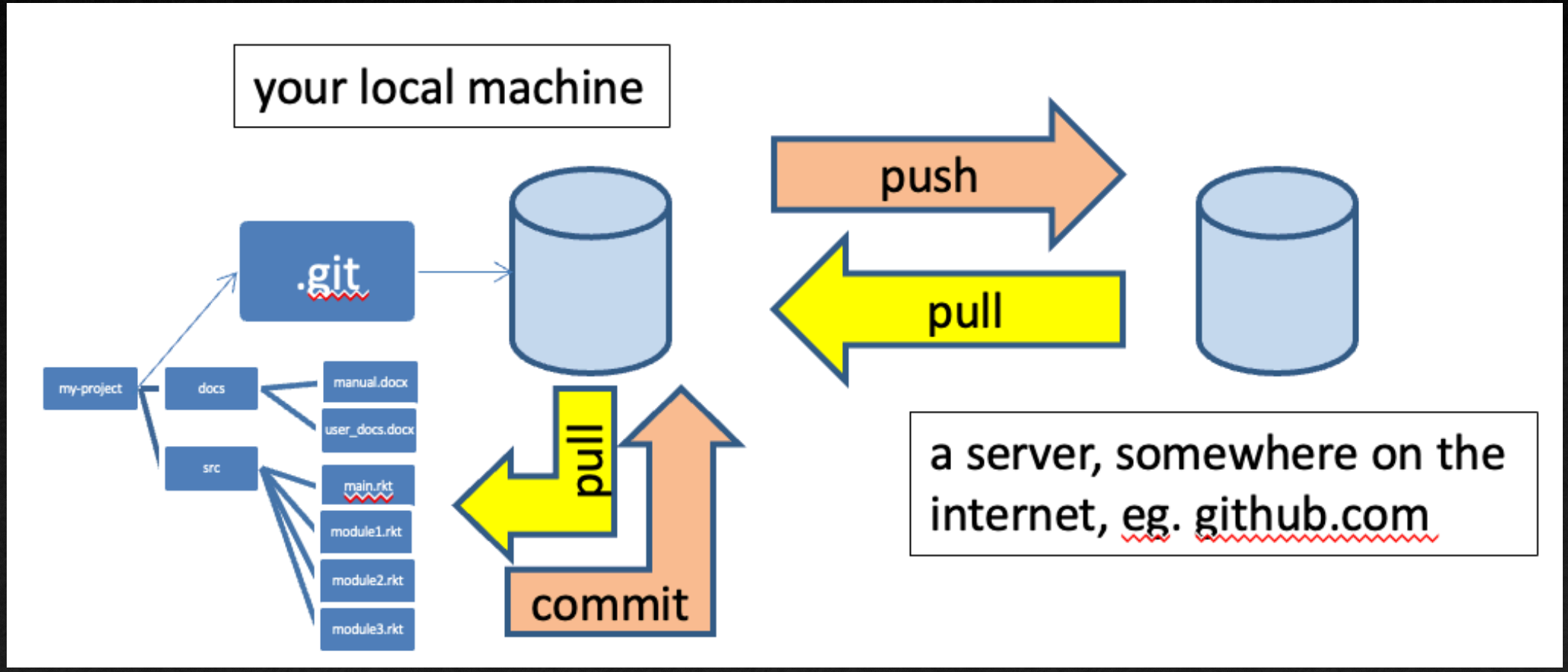
Synchronizing with the server (2)



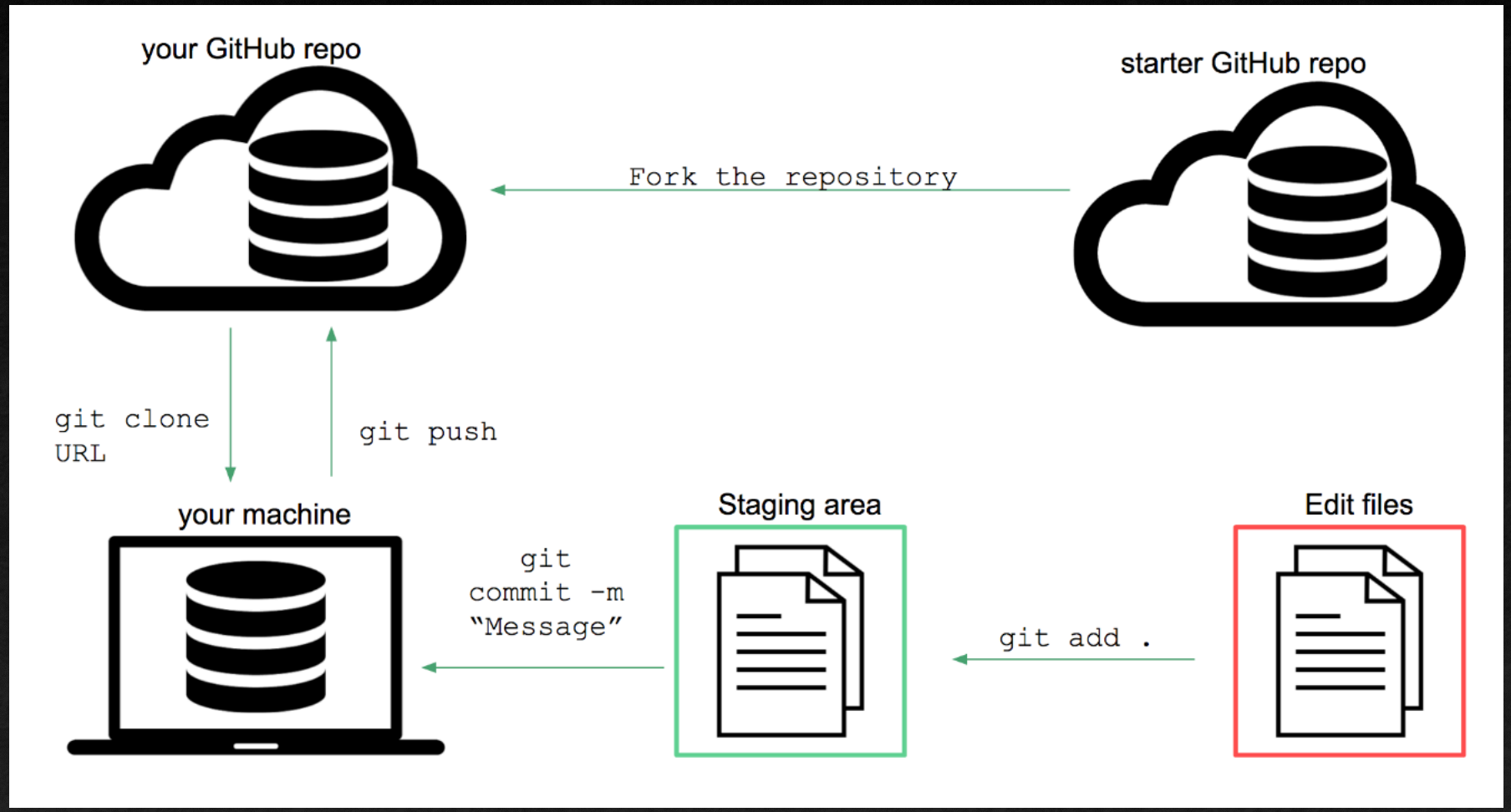
`git pull`	fetch from a remote repo and try to merge into the current branch

- To retrieve your data from the server, you do a "pull".
  - "pull" takes the data from the server and puts it both in your local mini-fs
- If your local file has changed, git will merge the changes if possible. If it can't figure out how to the merge, you will get an error message.

The whole picture (1)



The whole picture (2)



command	description
`git clone` ***`url [dir]`***	copy a git repository so you can add to it
`git add` ***`files`***	adds file contents to the staging area
`git commit`	records a snapshot of the staging area
`git status`	view the status of your files in the working directory and staging area
`git diff`	shows diff of what is staged and what is modified but unstaged
`git help` ***`[command]`***	get help info about a particular command
`git pull`	fetch from a remote repo and try to merge into the current branch
`git push`	push your new branches and data to a remote repository
others: `init`, `reset`, `branch`, `checkout`, `merge`, `log`, `tag`	

In-Class Problems

1. Canvas -> Assignments -> Homework 1
   - Accept invite to Github classroom
2. Go through python notebook on Canvas (02_InClass.ipynb) and complete three problems within the notebook