Introduction to Version Control with Git & Github

Valentina Staneva
Senior Data Scientist, eScience Institute

UNIVERSITY *of* WASHINGTON

Objectives

- Understand the advantages of using Git & Github
- Learn to track changes in code
- Understand the phases of commiting code
- Learn how to explore code history
- Learn about strategies for collaborating with others

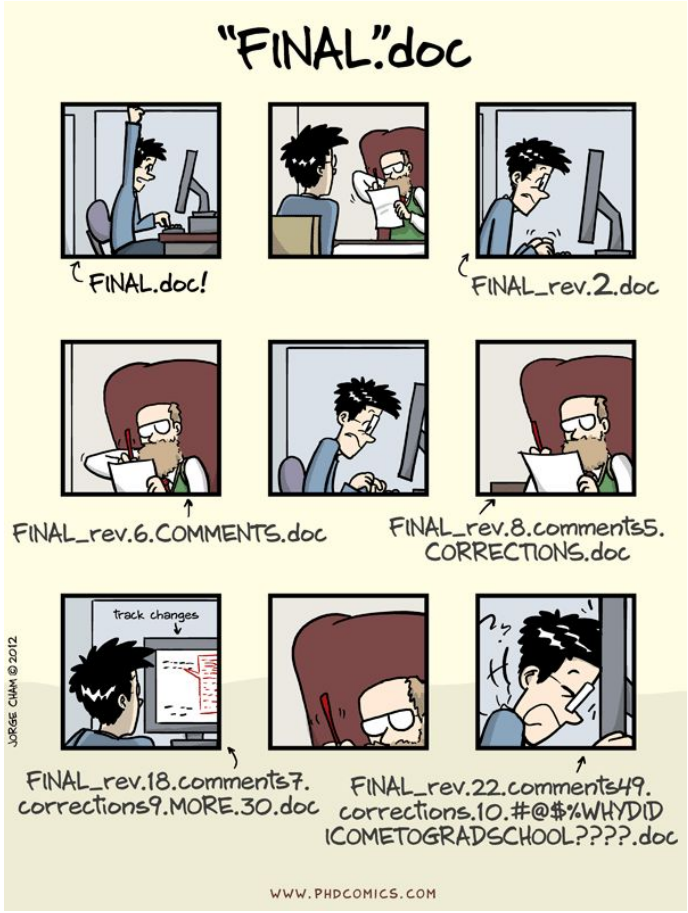
What is your experience with git:

A. I have not used git/Github before
B. I have used only git/Github on my own projects
C. I have collaborated with other using git & Github

Why Git & Github?

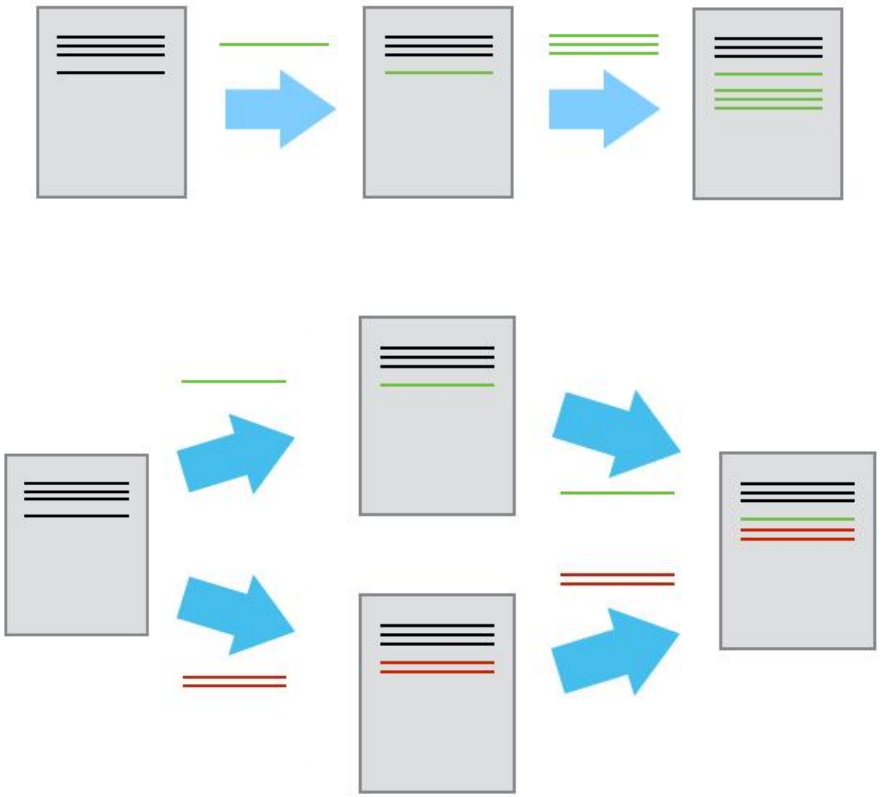
Goals:

- keep documents safe
- retrieve old versions
- share and collaborate with others

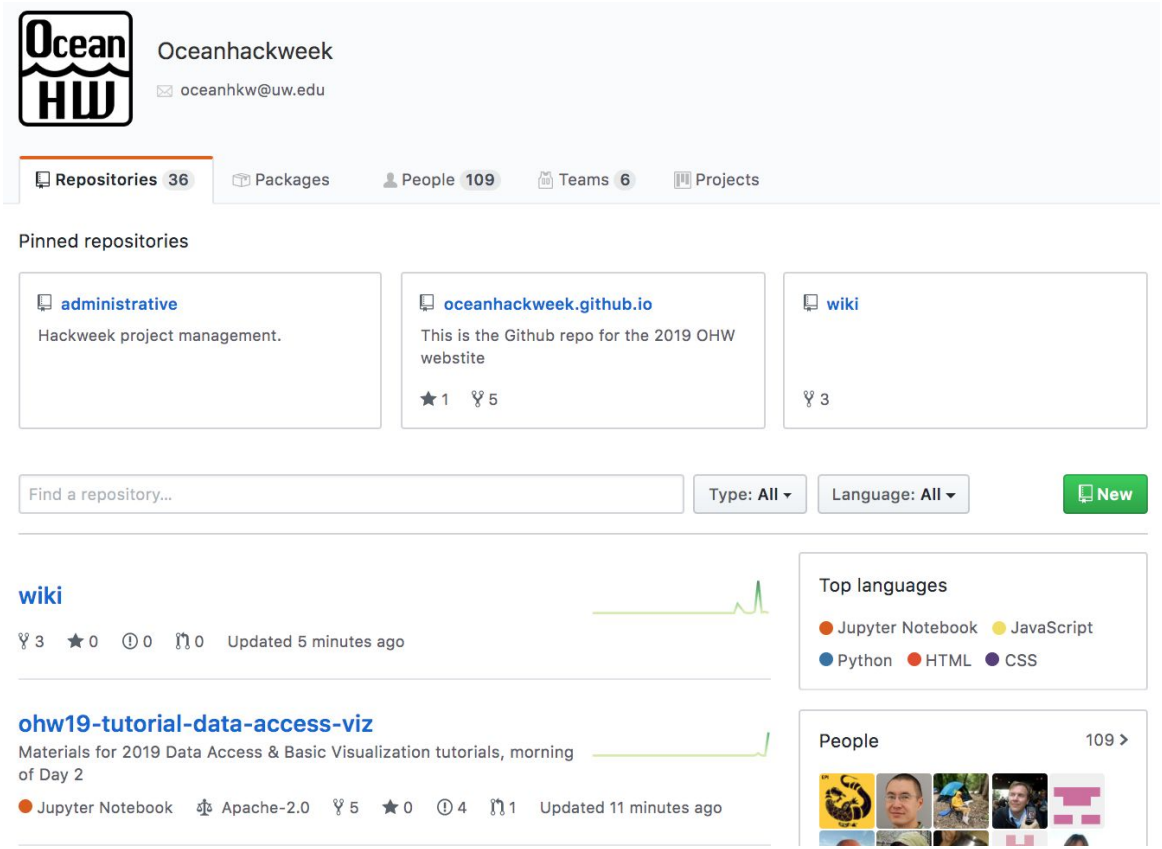


Git: local version control system

Saving only incremental changes



Github: web platform for version control and collaboration



[Oceanhackweek Organization]

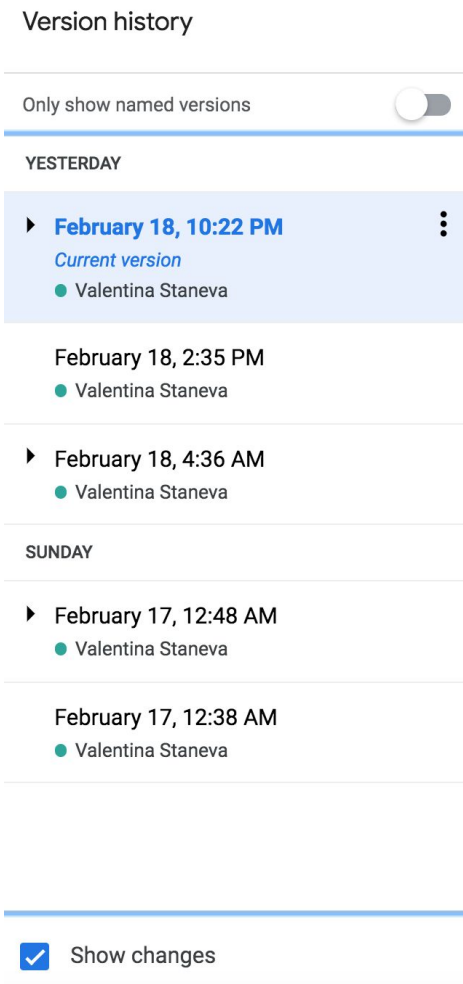
Comparison with cloud storage solutions

Google Docs/Drive, Dropbox

- all changes are visible
- versions are indexed by date
- hard to parse and revert to the right version
- some limits on versions

Git & Github

- strict change revision
- works well for code, not so well for text
- versions are named by you
- no limits on versions



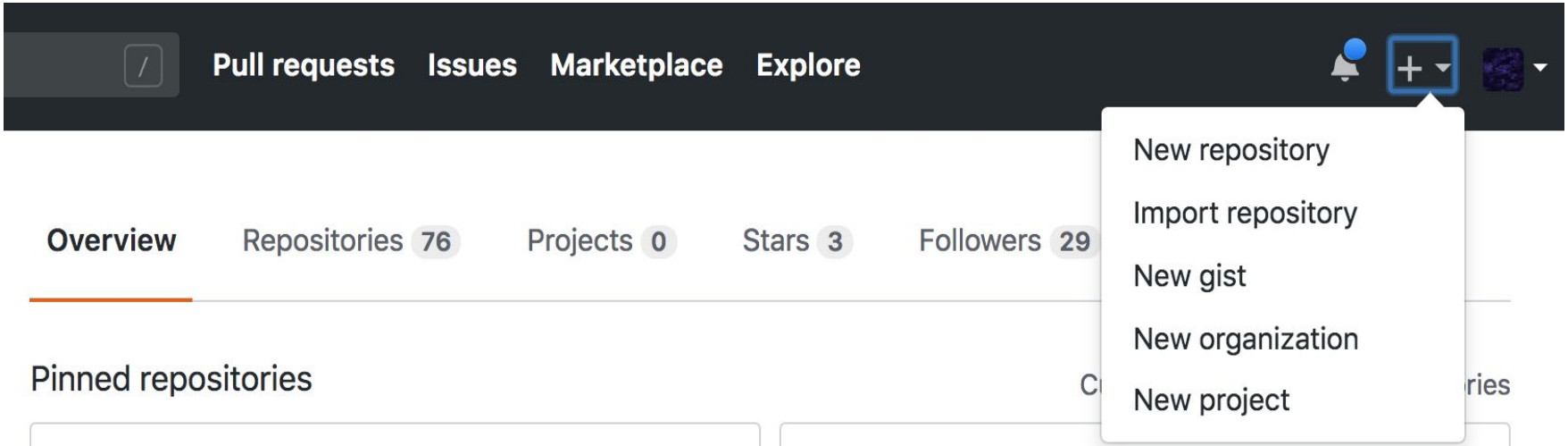
Prerequisites

1. You have git installed on your computer
   - https://carpentries.github.io/workshop-template/#git

2. You have created an account on `www.github.com`

3. You have a working bash terminal
   - https://swcarpentry.github.io/shell-novice/setup.html

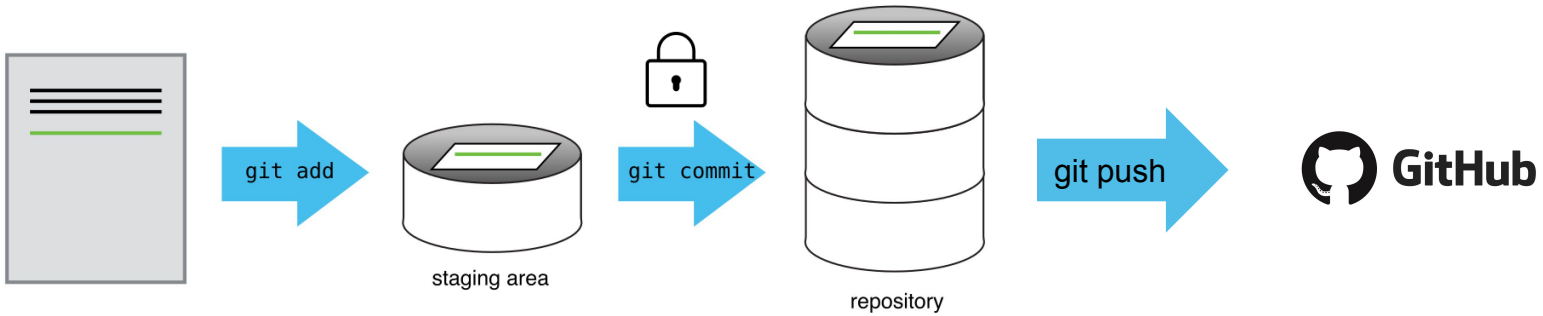
Getting Started



Scenario 1: You work by yourself, publishing on Github.

1. Make changes
2. Add files
3. Commit version
4. Publish on Github
5. Repeat

```
git diff
git add
git commit -m "message"
git push origin master
```



Quiz Time

True/False?

To publish a file on the Github website you can commit it with:

```
➢ git commit filename
```

False: you cannot commit individual files, you commit all staged files in your repository.

Quiz Time

What is the output of the file after the following steps:

1. Write: 'First line' in README.md
2. `git commit -m "adding first line to README.md"`
3. `git push origin`

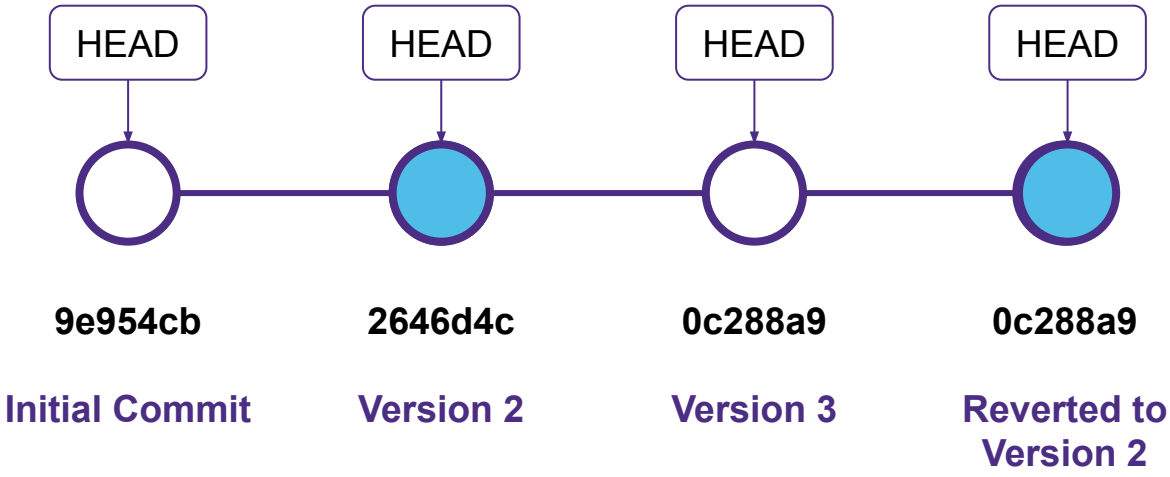
A: First line

B: Blank

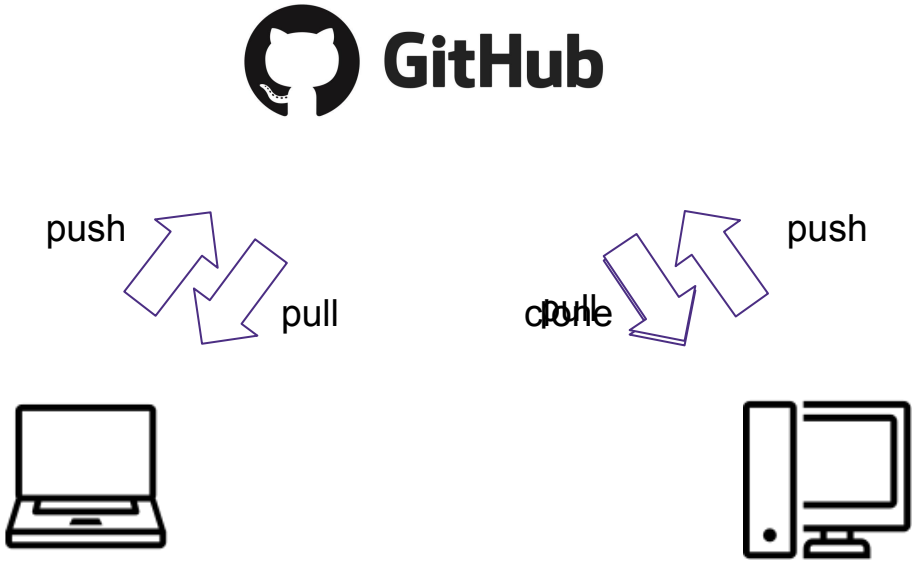
We need to `git add README.md` to stage the changes for the commit.

Exploring History

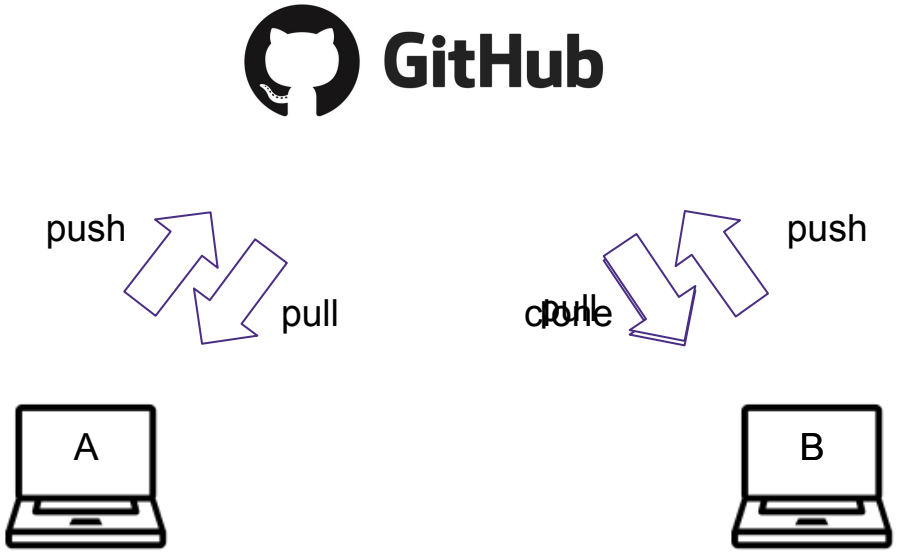
- looking at the commit history — `git log --oneline`
- checking out old versions — `git checkout version_id`
- retrieving old files — `git checkout filename`
- reverting a commit — `git revert HEAD`



Scenario 2: You work on the same project from your laptop and your work desktop



Scenario 3: You work with a collaborator and both of you have access to the repo



Resolving Merge Conflicts

1. `git status`: we see which files have been modified
2. Open the file
3. Resolve the conflict manually
4. Add and commit the changes.
5. Move on.

Before you pull, make sure you have committed your changes!

Quiz Time

True/False?

1. Changes made to different files do not result in a merge conflict.

2. Changes made to different parts of the same file do not result in a merge conflict.

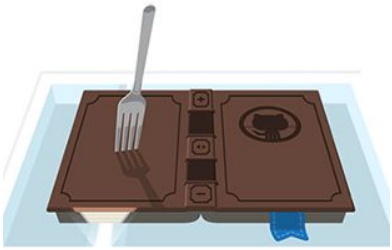
1. True.
2. True: (unless there are some formatting differences).

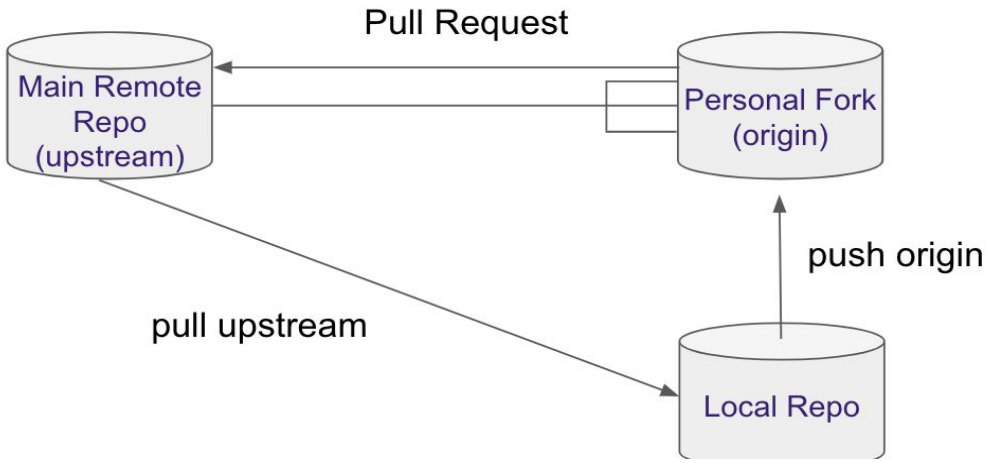
Scenario 4: You want to contribute to a repo for which you don't have permission

You cannot push to the repo without permissions!

Instead,

- you make a 'copy' of the central repo by forking it
- make the updates in the 'copy'
- request your changes to be pulled





Quiz Time

To periodically obtain the most recent changes from a public repository to a local repository, you use:

A: `git clone`

B: `git pull`

C: `fork`

B: `git pull`

we use `git clone` only the first time we create a local copy;

we `fork` to create a copy of a repo on Github (not locally).

Resources

https://swcarpentry.github.io/git-novice/

https://www.codecademy.com/learn/learn-git

https://www.atlassian.com/git/tutorials/what-is-version-control

https://guides.github.com/activities/hello-world/

https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

Quiz Time

What is the output of the file after the following steps:

1. Write: 'First line' in README.md
2. `Git add README.md`
3. Write: 'Second line'
4. `git commit -m "adding 2 changes to README.md"`
5. `git push origin`

A: First line

B: First Line

Second Line