



GitHub

with the Class Repo

And an in-depth look of how to use Github while at Metis

Jeremy Chow

Outline:

1. Vocab
2. Using Github for class notes
3. Other
 - a. Aliasing in terminal
 - b. Git reset
 - c. Merge Conflicts

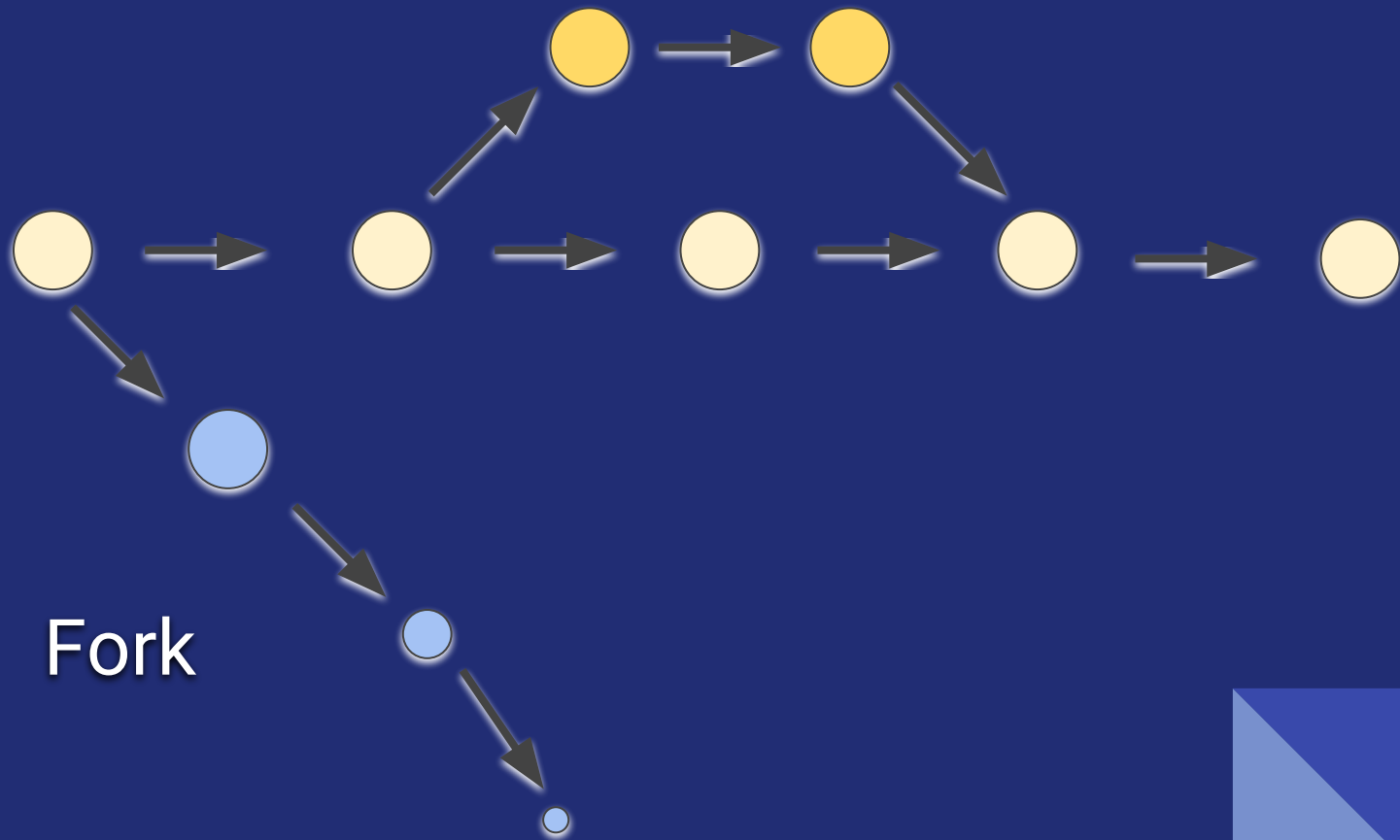


Github Vocabulary

Fork vs. Branch

- Branch is for team members
 - Intention to apply changes to operational branch
- Fork is for everyone else
- Can function the same way

Branch



Git Clone

- Does a Fetch and a Merge
- Fetch
 - Updates local records of online checkpoints
 - Doesn't change any files
- Merge
 - Brings online changes into your local directory



Github for Class Materials

Why are there issues pulling materials?

- Github is a version control platform
 - Meant for team contributions
 - Not intended for distribution
- Students cannot edit class repository
 - Every student wants to save their own notes
 - Requires **Forking**

File Structure

Online

Class Repository
github/thisismetis

Personal Repository
github/username

Local

Local Directory
~/path-to-notes

Do once:

Online

Class Repository
github/thisismetis

Fork



Personal Repository
github/username

Local

Local Directory
~/path-to-notes

[Why GitHub?](#)[Enterprise](#)[Explore](#)[Marketplace](#)[Pricing](#)[Sign in](#)[Sign up](#)[thisismetis](#) / [dsp](#)[Watch](#) 38[Star](#) 22[Fork](#) 991[Code](#)[Issues](#) 1[Pull requests](#) 17[Projects](#) 0[Insights](#)

Join GitHub today

GitHub is home to over 31 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)[Dismiss](#)

Metis Data Science Bootcamp - Official Prework Repository

[658 commits](#)[5 branches](#)[1 release](#)[12 contributors](#)Branch: [master](#)[New pull request](#)[Find File](#)[Clone or download](#)[aramnath](#) Merge pull request #97 from druce/master

Latest commit bb65671 22 days ago

[img](#)

Add files via upload

2 years ago

[lessons](#)

Merge pull request #95 from druce/master

29 days ago

[resources](#)

move revision

2 months ago

[.gitignore](#)

move revision

2 months ago

[README.md](#)

Should probably tell people they should do the pre-work in Jupyter lo...

26 days ago

[x](#)

x

22 days ago

[README.md](#)

Data Science Bootcamp Pre-work Revision 2019

Do once:

Online

Class Repository
github/thisismetis

Personal Repository
github/username



Local

Local Directory
~/path-to-notes

Clone



[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Insights](#)



Join GitHub today

GitHub is home to over 31 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)

[Dismiss](#)

Using github as a distribution service

[4 commits](#)


[1 branch](#)

[0 releases](#)

[1 contributor](#)


Branch: master [New pull request](#)

[Find File](#) [Clone or download](#)

 [Jeremy C and Jeremy C more outline changes](#)

[README.md](#) [Create README.md](#)

[things_to_cover.md](#) [more outline changes](#)


 [README.md](#)

Github as a distribution service

For the purpose of downloading changes to the sf19_ds14 repository but uploading personal notes to a separate repository on Github

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

<https://github.com/jeremyrchow/github-as> 

[Open in Desktop](#) [Download ZIP](#)

Clone by command line:

- Navigate to where you want to store notes
- `git clone [your_repo_URL]`

Options Available After Cloning

Online

Class Repository
github/thisismetis

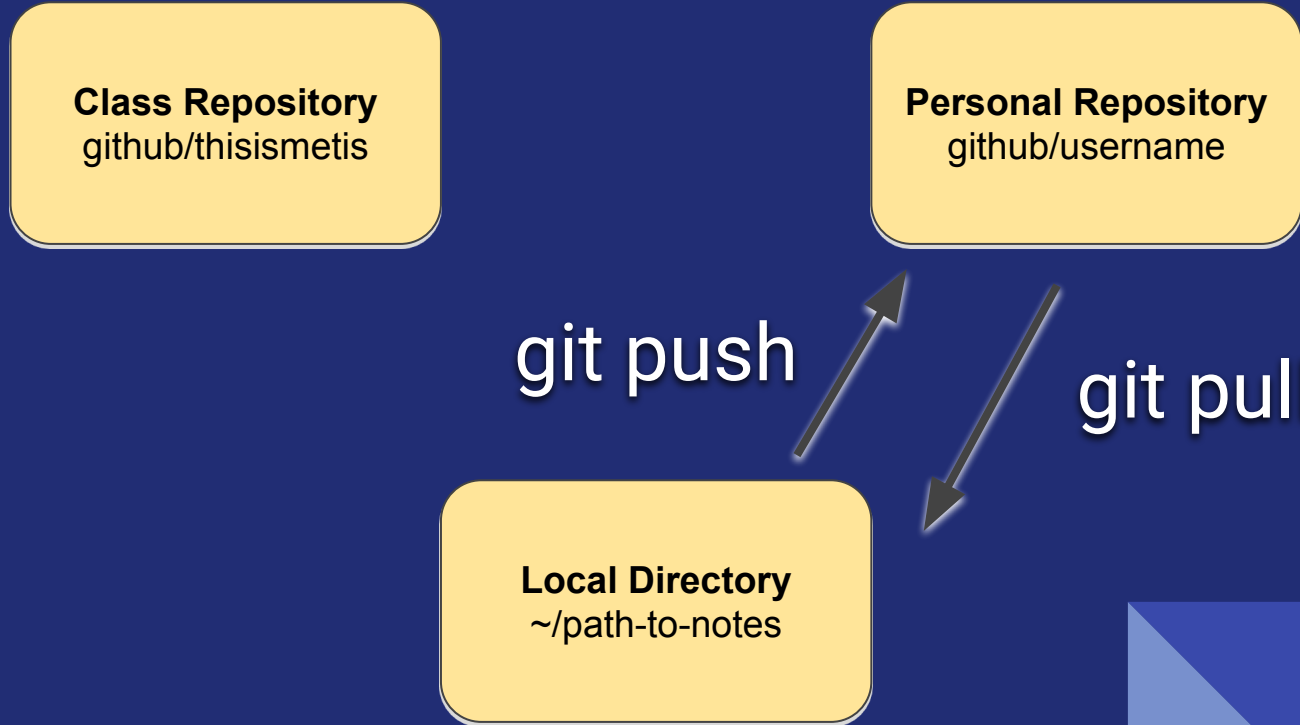
Personal Repository
github/username

Local

Local Directory
~/path-to-notes

git push

git pull



Not Available After Cloning

Online

Class Repository
github/thisismetis

Personal Repository
github/username

Local

No link yet

Local Directory
~/path-to-notes

Add upstream to your local directory

- ``git remote add upstream [thisismetis_repo]``
- ``git remote -v`` to verify
- Upstream points to Metis, origin points to you
 - Git push and pull default to origin

Online

Class Repository
github/thisismetis

Personal Repository
github/username

git pull upstream master

Local

Local Directory
~/path-to-notes



git pull upstream master

- Terminal uses git package

git pull upstream master

- Download updates online to your computer

git pull **upstream** master

- Points to repository set as 'upstream'
 - In our case -> thisismetis repo

git pull upstream **master**

- Specifies which branch to pull
- Master is default branch of every repo

New Files



Online

Class Repository
github/thisismetis

Personal Repository
github/username

Local

Local Directory
~/path-to-notes

Every morning:

Online

Class Repository
github/thisismetis

Personal Repository
github/username

git pull upstream master

Local

Local Directory
~/path-to-notes



Online

Class Repository
github/thisismetis

Personal Repository
github/username

Take notes



Local

Local Directory
~/path-to-notes

Often:

Online

Class Repository
github/thisismetis

Personal Repository
github/username

Local

Local Directory
~/path-to-notes

git push



```
graph BT; LD[Local Directory  
~/path-to-notes] -- "git push" --> PR[Personal Repository  
github/username];
```

The diagram illustrates the process of pushing code from a local directory to a personal repository on GitHub. It features three yellow rounded rectangular boxes. The 'Local Directory' box at the bottom center contains the text '~ / path - to - notes'. An arrow points from this box to the 'Personal Repository' box at the top right, which contains 'github / username'. The text 'git push' is placed to the right of the arrow. To the left of the 'Class Repository' box (top left, containing 'github / thisismetis') is the word 'Online'. To the left of the 'Local Directory' box is the word 'Local'. The background is dark blue with a geometric pattern of lighter blue triangles in the bottom right corner.

Oh no!

Online

Class Repository
github/thisismetis

Personal Repository
github/username

Local



Only if catastrophe:

Online

Class Repository
github/thisismetis

Personal Repository
github/username

Local

Local Directory
~/path-to-notes

git pull



```
graph TD; PR[Personal Repository  
github/username] -- "git pull" --> LD[Local Directory  
~/path-to-notes];
```

The diagram illustrates a workflow for pulling updates from a personal repository to a local directory. It features three yellow rounded rectangular boxes on a dark blue background. The 'Online' section at the top contains two boxes: 'Class Repository' (github/thisismetis) on the left and 'Personal Repository' (github/username) on the right. The 'Local' section at the bottom contains a box for 'Local Directory' (~/path-to-notes). A grey arrow points from the 'Personal Repository' box down to the 'Local Directory' box, with the text 'git pull' positioned to the right of the arrow. A decorative blue geometric shape is in the bottom right corner.

Daily Workflow

- In Morning: **git pull upstream master**
- While taking notes / end of day:
 - `Git status` to see files that have changed since last commit
 - Git add [files to be added]
 - Git commit -m "what did I add for this commit"
 - Git push



Other

Terminal Aliasing

What is Aliasing?

- Can set up terminal shortcuts in `bash_profile`
 - May be called `bash_rc` if on linux
- Run multiple commands using one terminal command
- Run `'./bash_profile'` to update


```
.bash_profile x
1 # added by Anaconda3 2018.12 installer
2 # >>> conda init >>>
3 # !! Contents within this block are managed by 'conda init' !!
4 __conda_setup="$(CONDA_REPORT_ERRORS=false '/anaconda3/bin/conda' shell.bash hook 2> /dev/null)"
5 if [ $? -eq 0 ]; then
6     \eval "$__conda_setup"
7 else
8     if [ -f "/anaconda3/etc/profile.d/conda.sh" ]; then
9         . "/anaconda3/etc/profile.d/conda.sh"
10        CONDA_CHANGEPS1=false conda activate base
11    else
12        \export PATH="/anaconda3/bin:$PATH"
13    fi
14 fi
15 unset __conda_setup
16 # <<< conda init <<<
17 alias metis="cd ~/Work/MetisCode/sf19ds14_git/sf19_ds14; git pull upstream master"
18
```

This is not python - spacing and quote types matter!

Terminal Input/Output

```
[(base) bash-3.2$ metis
From https://github.com/thisismetis/sf19_ds14
* branch          master      -> FETCH_HEAD
Already up to date.
[(base) bash-3.2$ pwd
/Users/jchow/Work/MetisCode/sf19ds14_git/sf19_ds14
(base) bash-3.2$ █
```



Git reset

Git reset [--soft, --mixed, or --hard] [commit_id]

- --soft resets 'head', but changes will be staged
 - Reset 'path' of commits, but keep all working changes
- --mixed resets 'head' AND index to commit, but changes will not be staged
- --hard means reset local files as well

Merge Conflicts

Merge Conflicts

- If two users edit same file in different way
 - Easily reproducible by one person editing, one deleting same file
- Look for `< === head` in conflicting file(s)
- Manually delete / fix



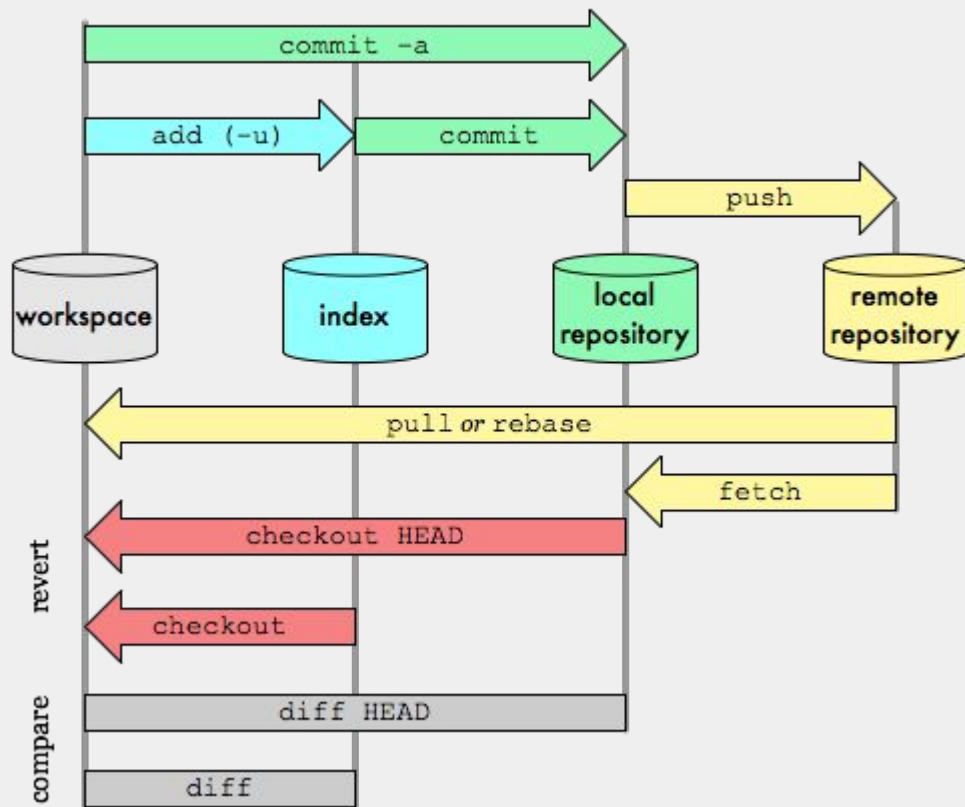
Thank you!

And thanks to the TA's!

Appendix

Git Data Transport Commands

<http://osteele.com>



Head vs. Index

- Head refers to the last commit or checkpoint your files are based on
 - When you make a new commit, Head points to that new commit
 - Points to the end of the branch you are working on

Head vs. Index

- Index refers to the changes from the head
- Equivalent to 'Stage'

Helpful Links

- <https://www.atlassian.com/git/tutorials/merging-vs-rebasing>
- <https://help.github.com/en/articles/resolving-a-merge-conflict-using-the-command-line>