# 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 command line Github for class notes
3. Other
    a. Aliasing in terminal
    b. Git reset
    c. Merge Conflicts

# Github Terms

# Git Fork

- Copies a repository into your own online repository space
- Completely **online** operation

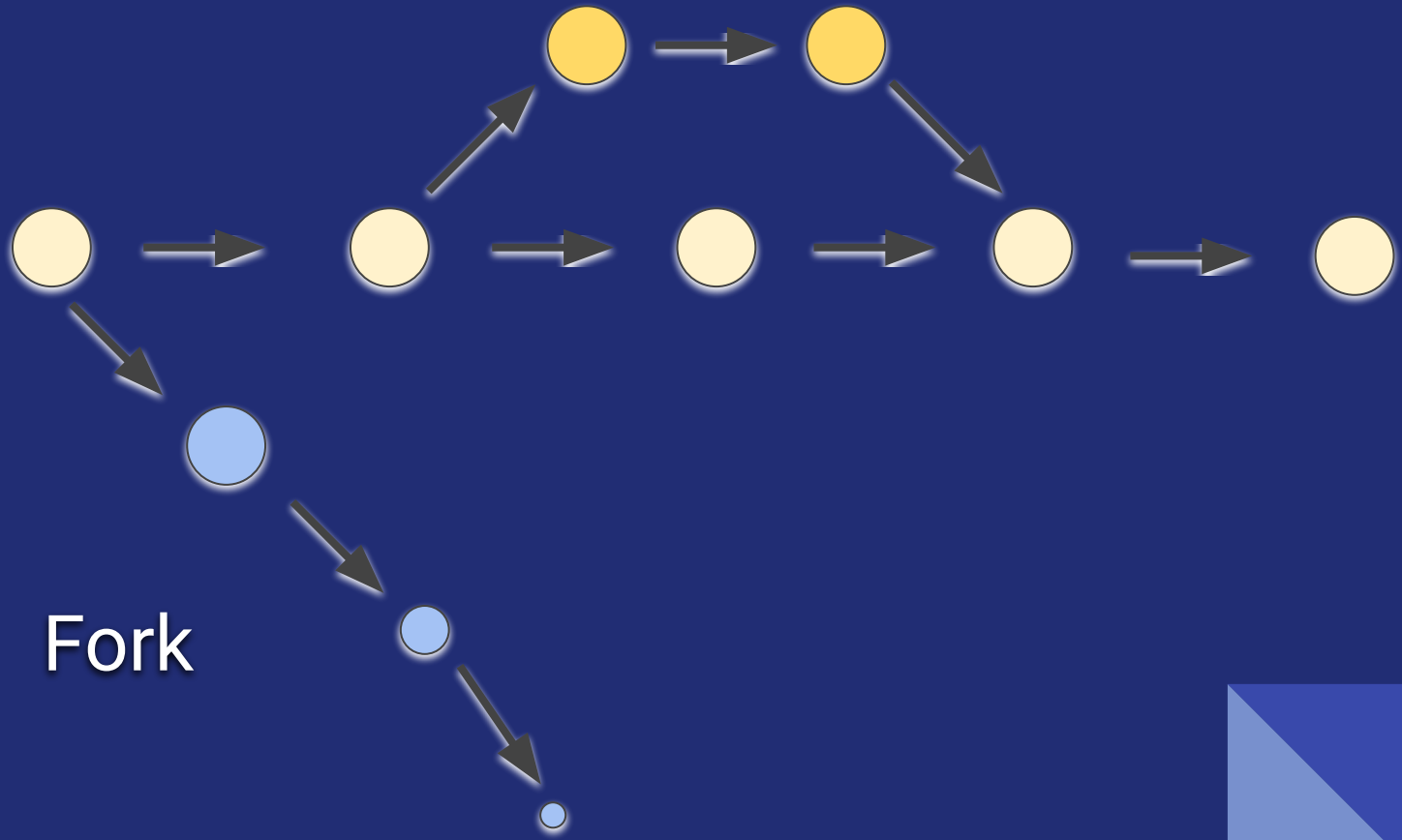
# Git Clone

- 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

# 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



Fork

# 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

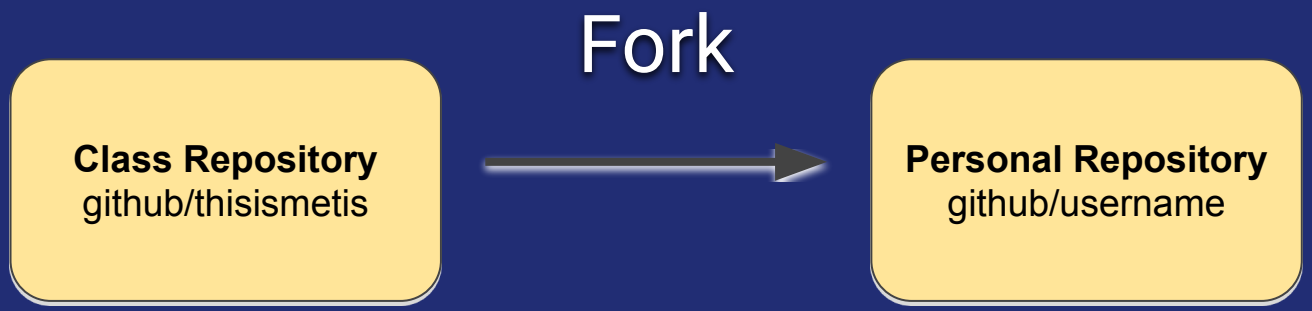
**Personal Repository**
github/username

Local

**Local Directory**
~/path-to-notes

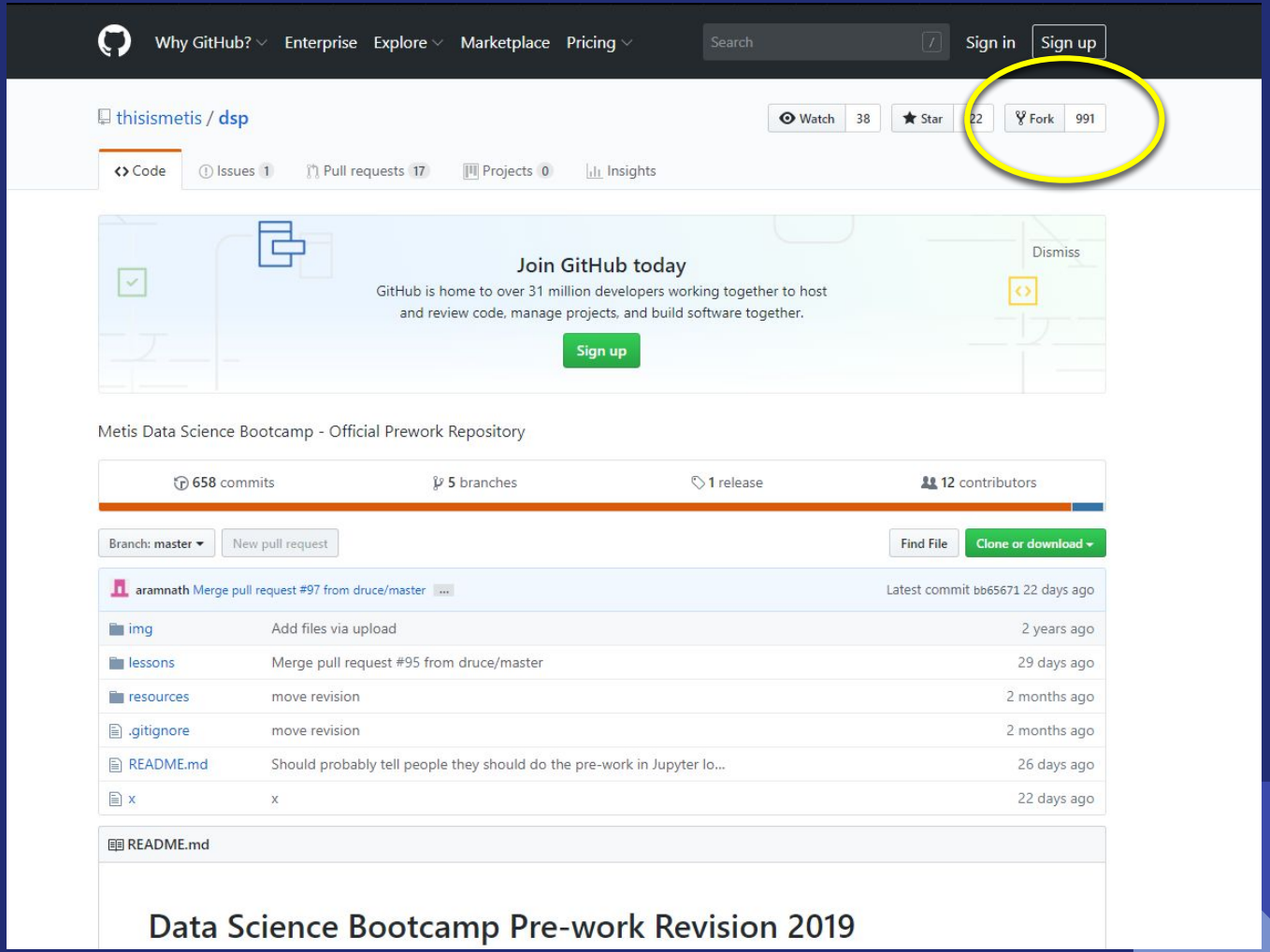
# Do once:

Online




Local

**Local Directory**
~/path-to-notes


Why GitHub?
Enterprise
Explore
Marketplace
Pricing
Search
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

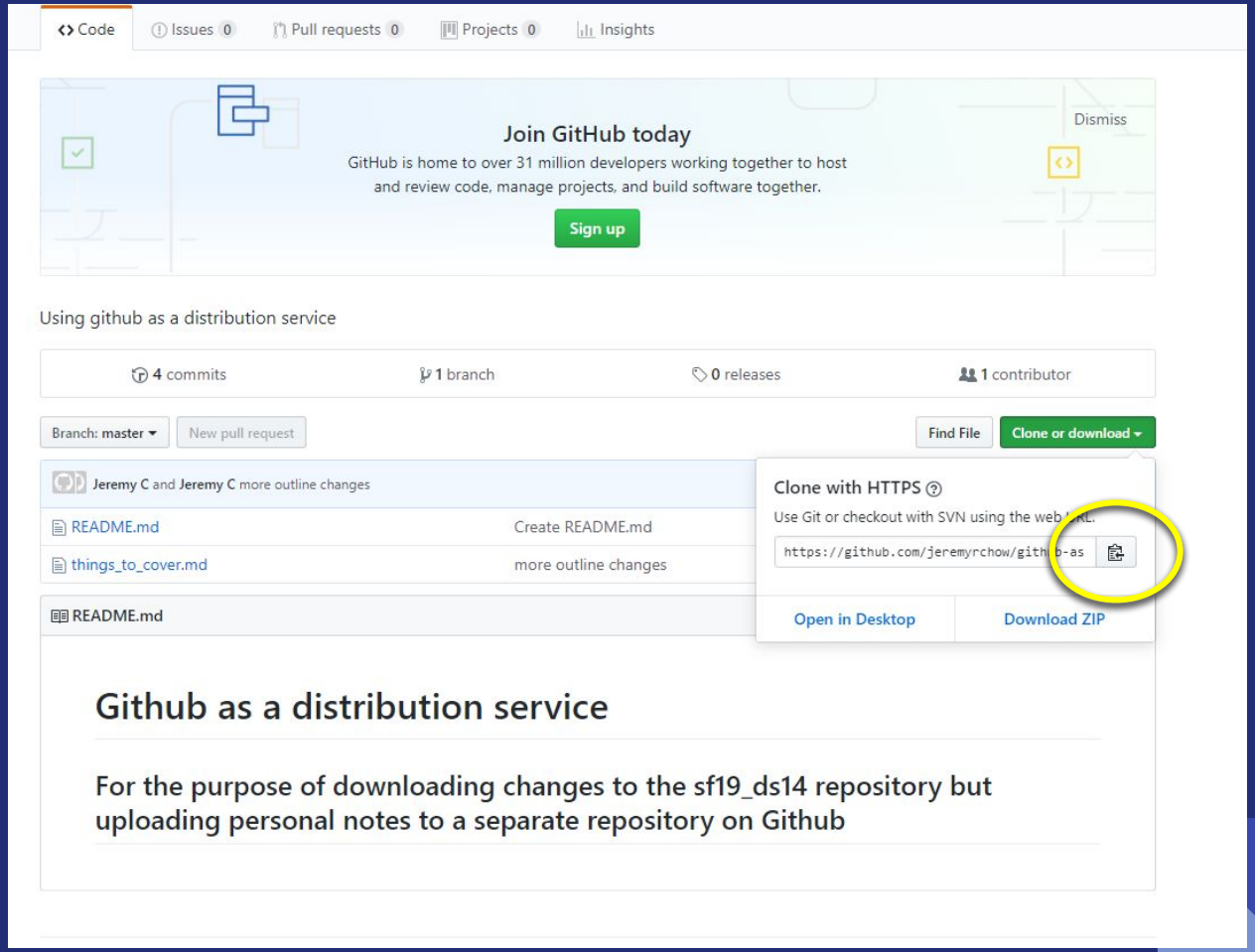
**Class Repository**
github/thisismetis

**Personal Repository**
github/username

Clone

Local

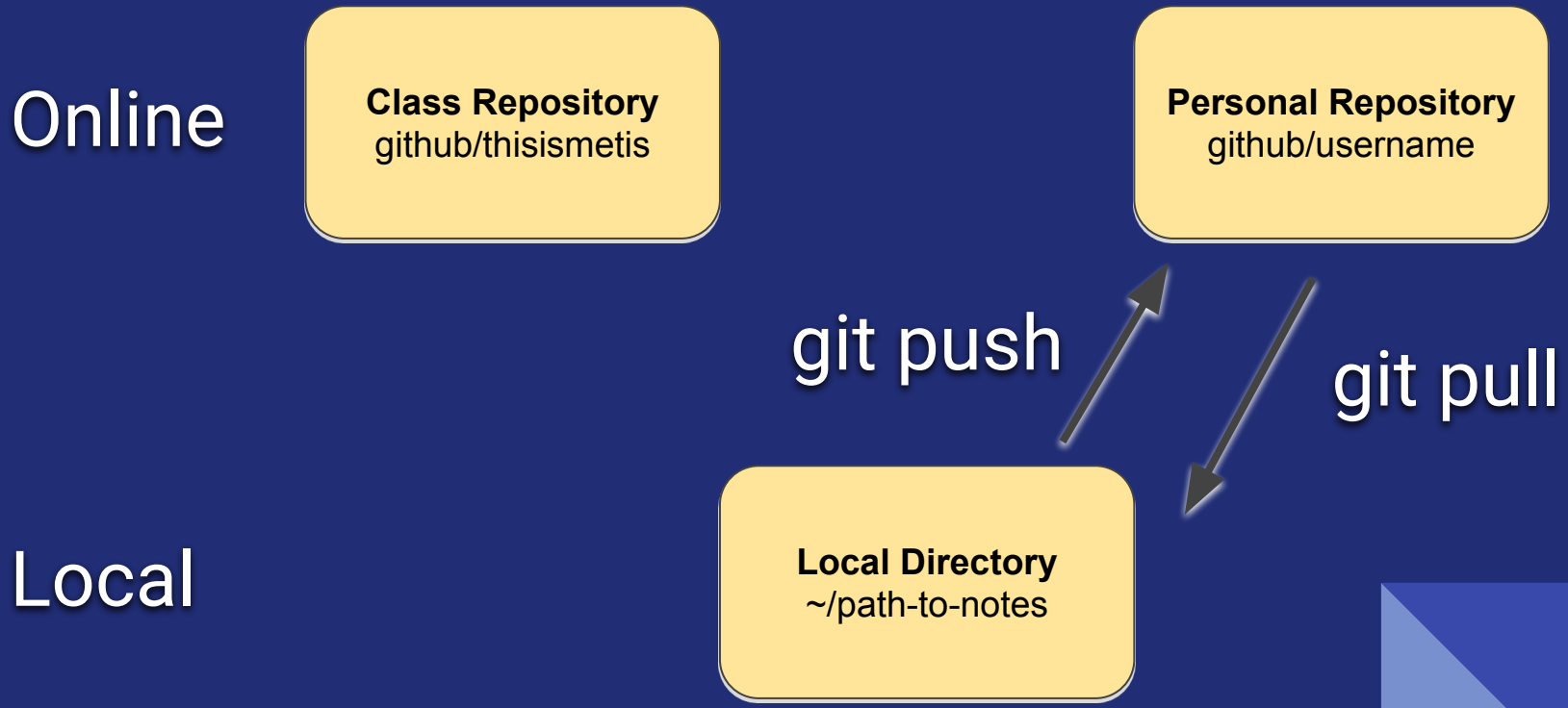
**Local Directory**
~/path-to-notes


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
Clone with HTTPS
Use Git or checkout with SVN using the web URL.
https://github.com/jeremyrchow/github-as
Open in Desktop
Download ZIP
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 by command line:

- Navigate to where you want to store notes (using terminal `cd`)
- git clone [your_repo_URL]

# Options Available After Cloning



# Not Available After Cloning

Online

**Class Repository**
github/thisismetis

**Personal Repository**
github/username

No link yet - add ‘upstream’

Local

**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

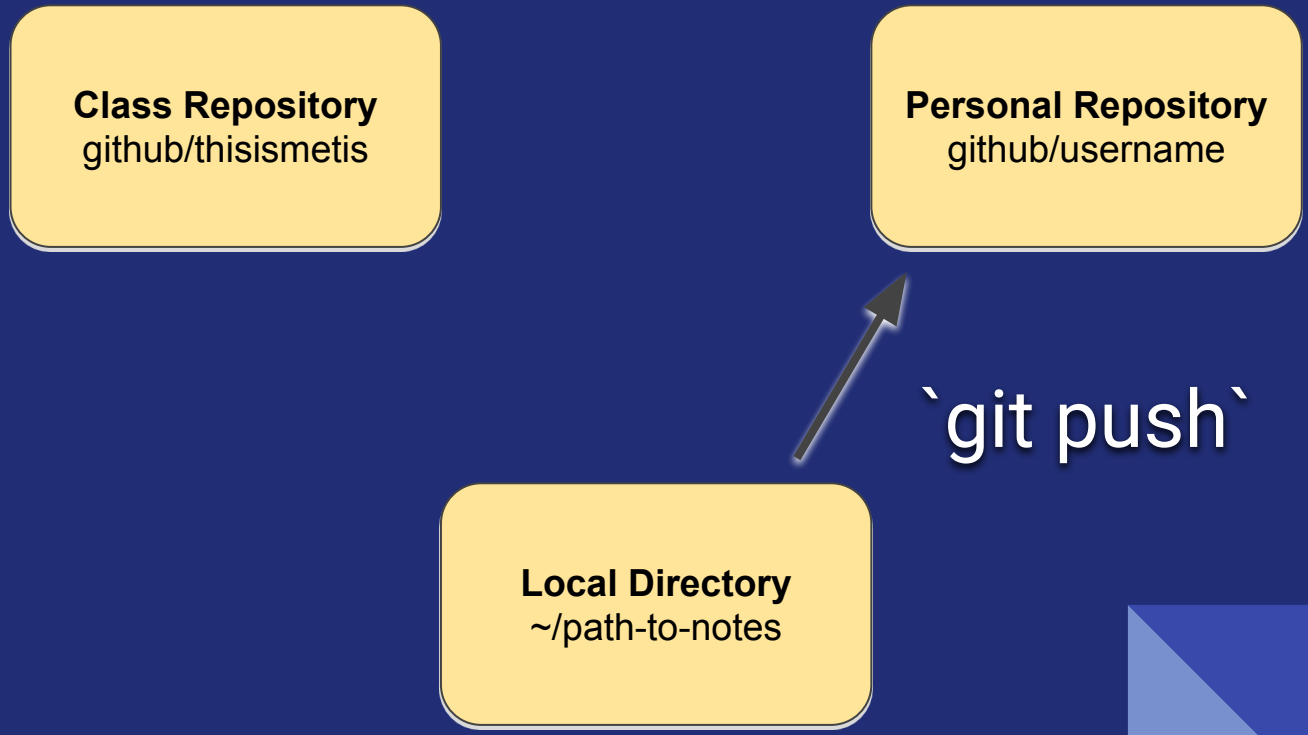
Local

**Local Directory**
~/path-to-notes

# Often:

Online

Local



# Oh no!

Online

**Class Repository**
github/thisismetis

**Personal Repository**
github/username

Local

# Only if catastrophe:

Online

**Class Repository**
github/thisismetis

**Personal Repository**
github/username

`git clone` or
`git reset --hard`

Local

**Local Directory**
~/path-to-notes

# Git clone vs git reset

- `git pull` does NOT restore deleted files!
  - Pulls in differences between commits
- Use `git reset --hard` instead

# Daily Workflow

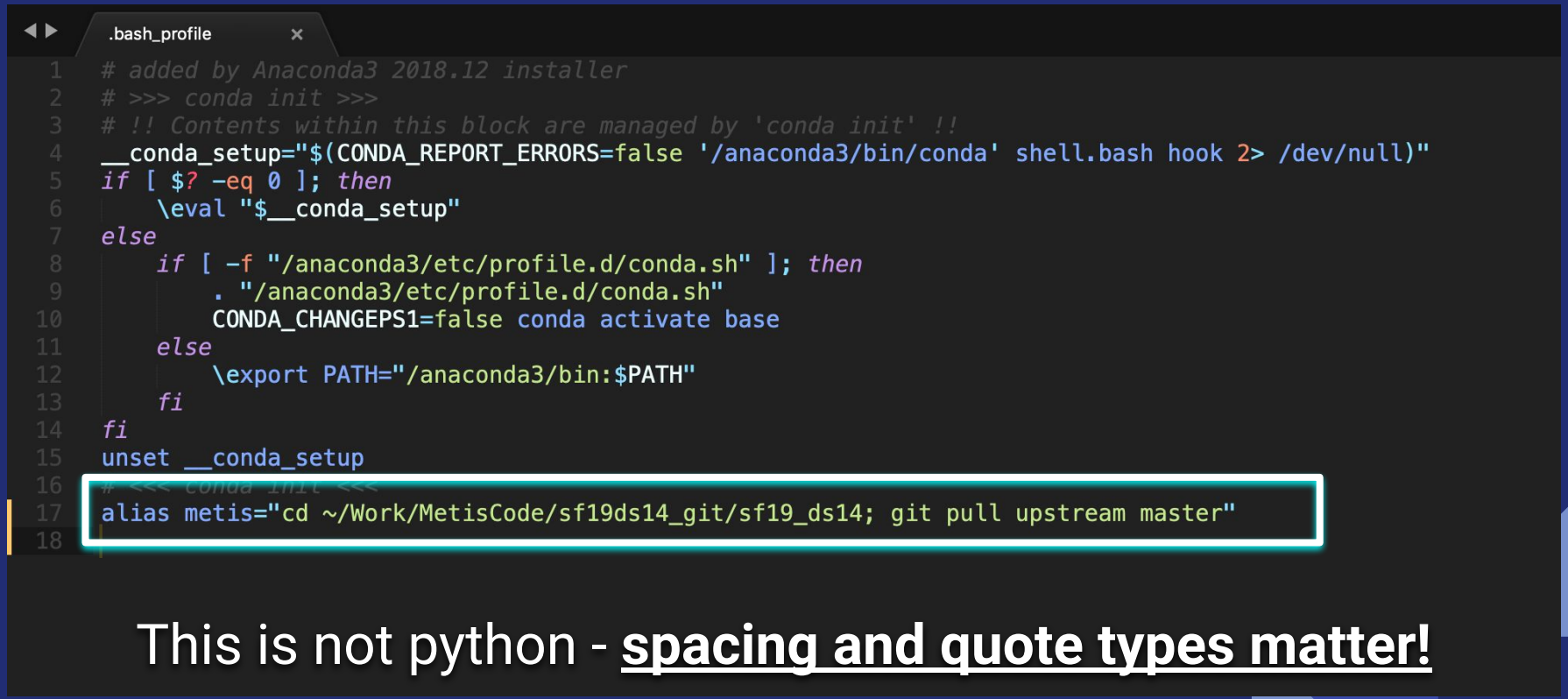
- In Morning: **git pull upstream master**
- When you want to store notes on Github:
  - `Git status` to see files that have changed since last commit
  - Git add [files to be added]
  - Git commit -m “what did I add for this commit”
  - Git push

# 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` in home directory to update




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
  - ‘Head’ is end of the branch you are working on
- --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

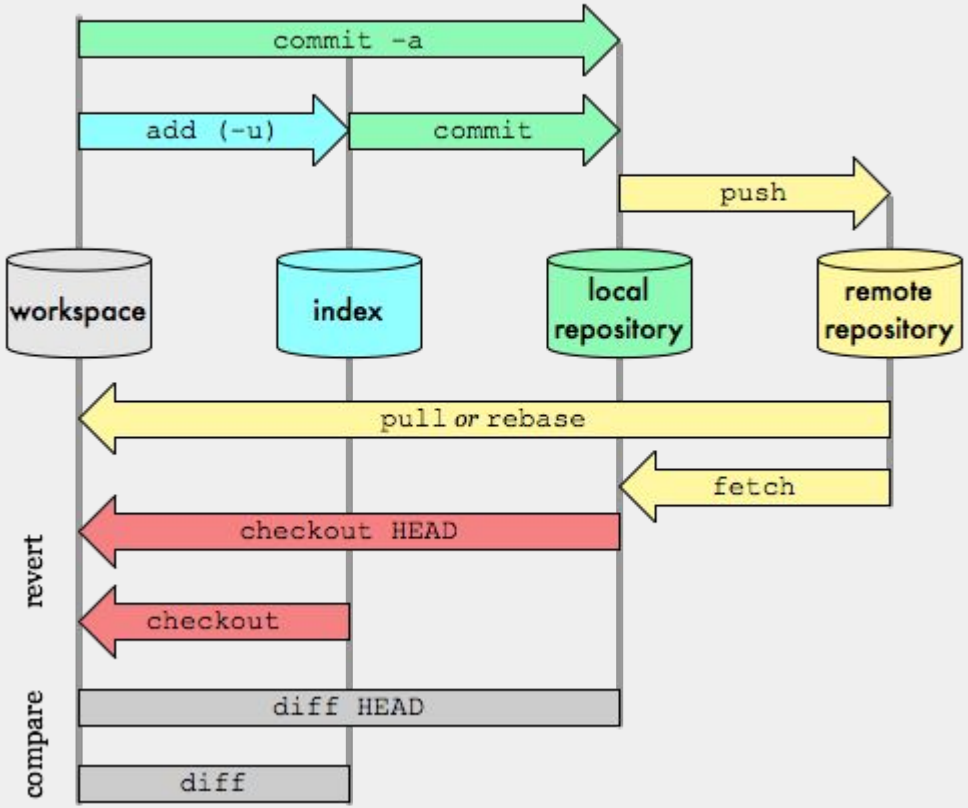
# Thank you!

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