Effective Software Development and Version Control

Jennifer Helsby, Eric Potash

Computation for Public Policy

Lecture 5: January 19, 2016

computationforpolicy.github.io

Announcements

- Do look at the readings
- Computing takes time outside of class to get comfortable with, it involves trial and error and can be frustrating
- Homework 1 will get back on Friday
- Homework 2 out on Thursday
- Sign up for (and look at) Piazza!

Today

- What is good code?
- Collaboratively writing code
- Version control with `git` and Github.com

Principles of Good Software

- Simplicity
- Clarity
- Generality
- Automation

Code Clarity: Comments

- Comments should add information
    - Unnecessary comment:

    ```
    i += 1  # Add 1 to i
    ```

- Comments should explain why choices were made if not obvious

Code Clarity: Comments

- Well named variables and functions document themselves

```
num_to_grade = num_assignments * num_students
```

- Docstrings
  - Comments at the top of e.g. a function definition that describe the function and its inputs and outputs

Docstring: Example

```
def complex(real=0.0, imag=0.0):
    """Form a complex number.

    Keyword arguments:
    real -- the real part (default 0.0)
    imag -- the imaginary part (default 0.0)
    """
    if imag == 0.0 and real == 0.0:
        return complex_zero
    ...
```

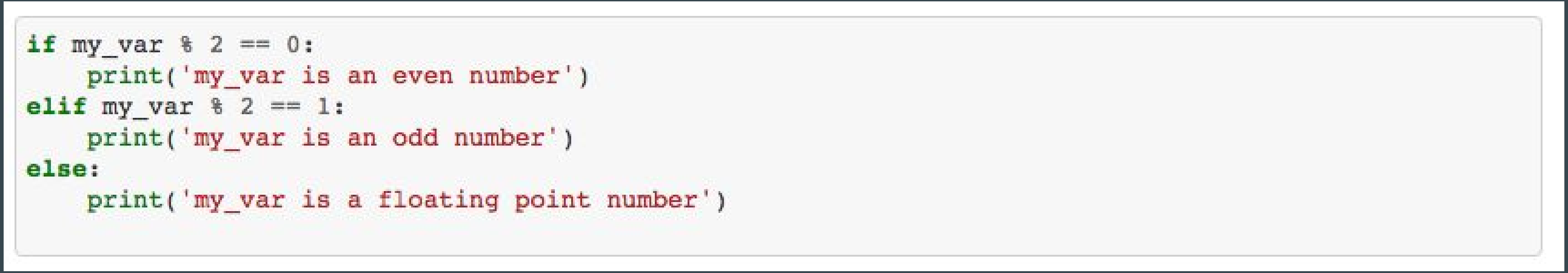
Code Clarity: Length

- Shorter codes are not always better!
- Better to have slightly longer code that is clearer
- Avoid unnecessary abstraction

Code Simplicity and Generality

- Keep it simple: functions should ideally do only one thing
- Avoid repetition in code
- The more complex the code, the more likely bugs will be introduced

Catching Errors, Bugs and Testing

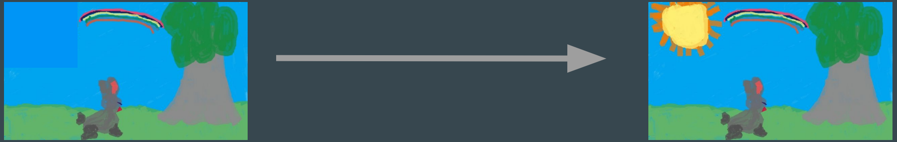


```
if my_var % 2 == 0:
    print('my_var is an even number')
elif my_var % 2 == 1:
    print('my_var is an odd number')
else:
    print('my_var is a floating point number')
```

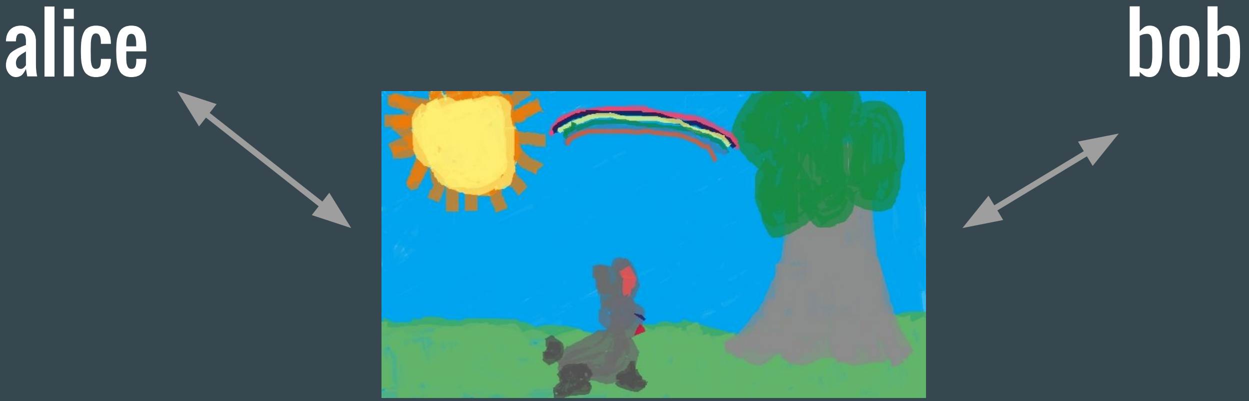
- Make programs more robust by validating inputs (and providing useful feedback)

```
if type(my_var) != ‘int’:

    print(‘Warning: Unexpected type for my_var’)
```

Version Control



Collaboration



Git Version Control System

Git Version Control System

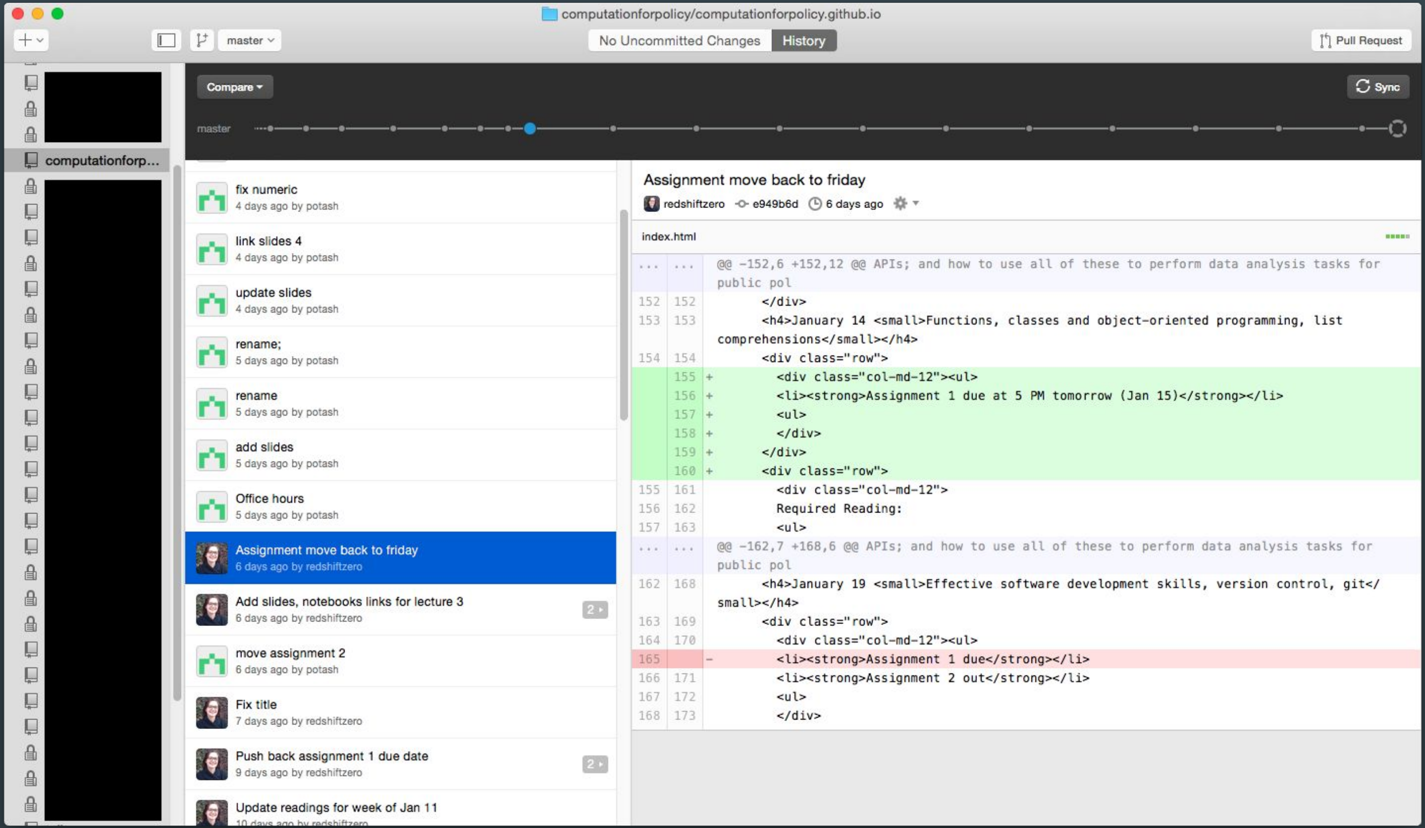
THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

https://xkcd.com/1597/

Git Repository

- “repo” for short
- Each project you work on will have a git repository
- Can store many types of data in a repository:
  - Code
  - Images
  - Folders
  - Text data
  - ...
- Can use version control software for many applications e.g. writing papers, presentations, collaboratively doing data analysis, etc.
- Used primarily for coding

Git GUI



[redacted]

computationforp...

[redacted]

fix numeric
4 days ago by potash

link slides 4
4 days ago by potash

update slides
4 days ago by potash

rename;
5 days ago by potash

rename
5 days ago by potash

add slides
5 days ago by potash

Office hours
5 days ago by potash

Assignment move back to friday
6 days ago by redshiftzero

Add slides, notebooks links for lecture 3
6 days ago by redshiftzero

move assignment 2
6 days ago by potash

Fix title
7 days ago by redshiftzero

Push back assignment 1 due date
9 days ago by redshiftzero

Update readings for week of Jan 11

Assignment move back to friday

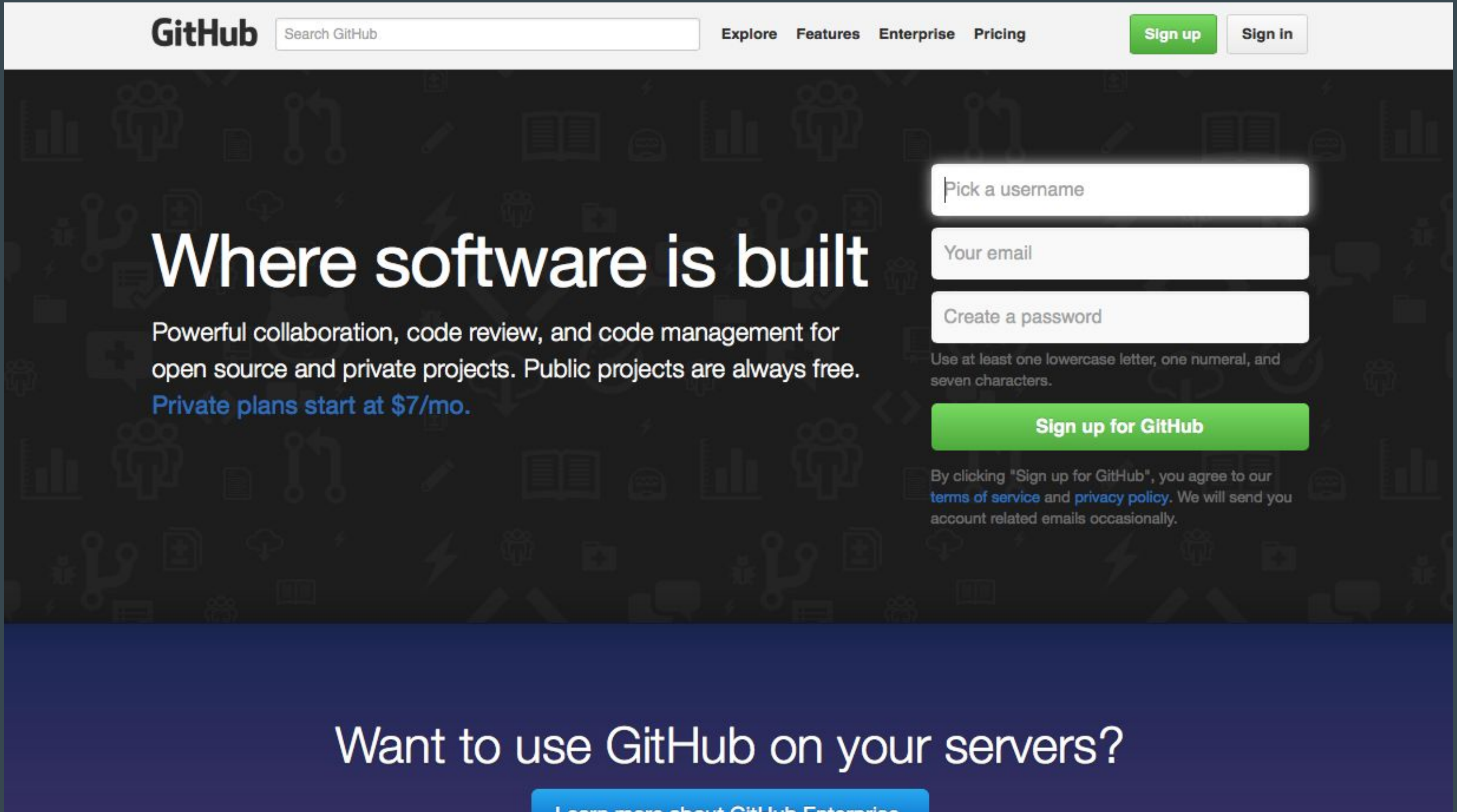
redshiftzero e949b6d 6 days ago

index.html

```
@@ -152,6 +152,12 @@ APIs; and how to use all of these to perform data analysis tasks for public pol
        </div>
        <h4>January 14 <small>Functions, classes and object-oriented programming, list comprehensions</small></h4>
        <div class="row">
+          <div class="col-md-12"><ul>
+          <li><strong>Assignment 1 due at 5 PM tomorrow (Jan 15)</strong></li>
+          <ul>
+          </div>
+        </div>
+        <div class="row">
          <div class="col-md-12">
          Required Reading:
          <ul>
@@ -162,7 +168,6 @@ APIs; and how to use all of these to perform data analysis tasks for public pol
        <h4>January 19 <small>Effective software development skills, version control, git</small></h4>
        <div class="row">
          <div class="col-md-12"><ul>
-          <li><strong>Assignment 1 due</strong></li>
          <li><strong>Assignment 2 out</strong></li>
          <ul>
          </div>
```

https://git-scm.com/downloads/guis

Github.com



Where software is built

Powerful collaboration, code review, and code management for open source and private projects. Public projects are always free. Private plans start at $7/mo.

Use at least one lowercase letter, one numeral, and seven characters.

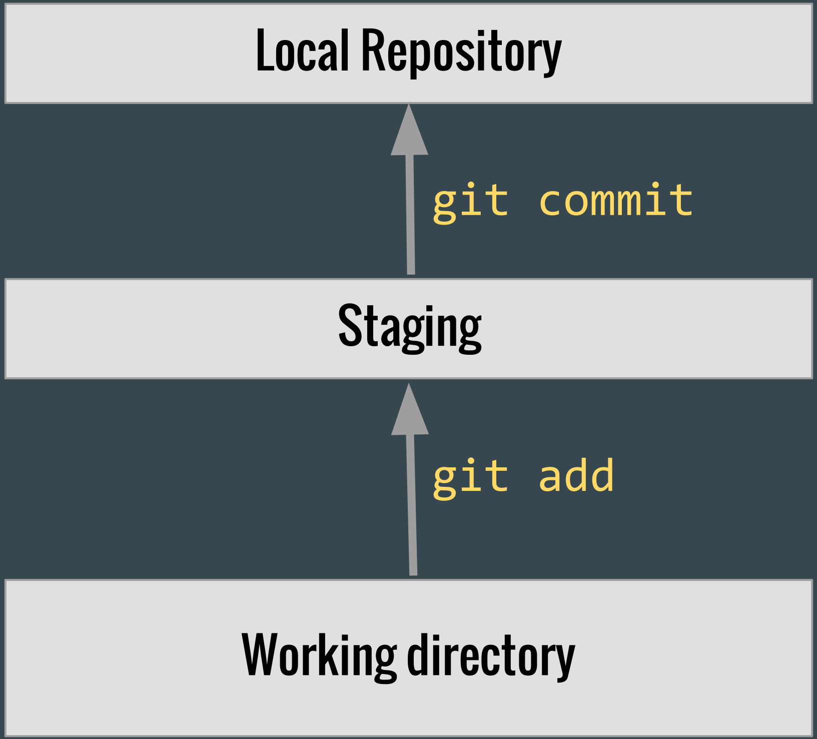
By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We will send you account related emails occasionally.

Want to use GitHub on your servers?

Github.com

- Git repositories can be hosted on Github.com
- Can sync everything with Github.com
- Used by many open source software projects to collaboratively develop software

Using git locally



git commit is a
set of changes

Basic Git Commands

https://try.github.io/levels/1/challenges/1

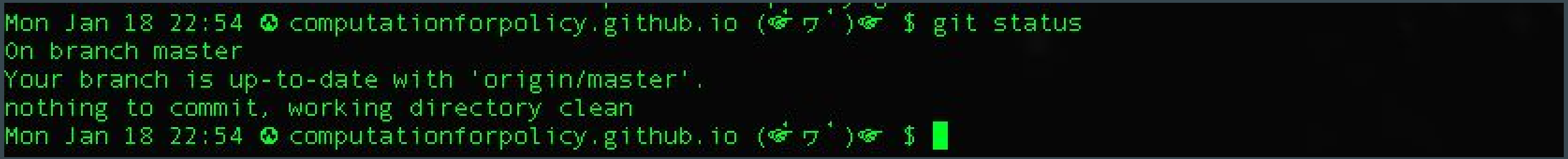
Make new repository

```
mkdir examplerepo  # make new folder for my new project

cd examplerepo  # go to that directory

git init  # initialize a git repo here
```

tutorial
started
here

git status

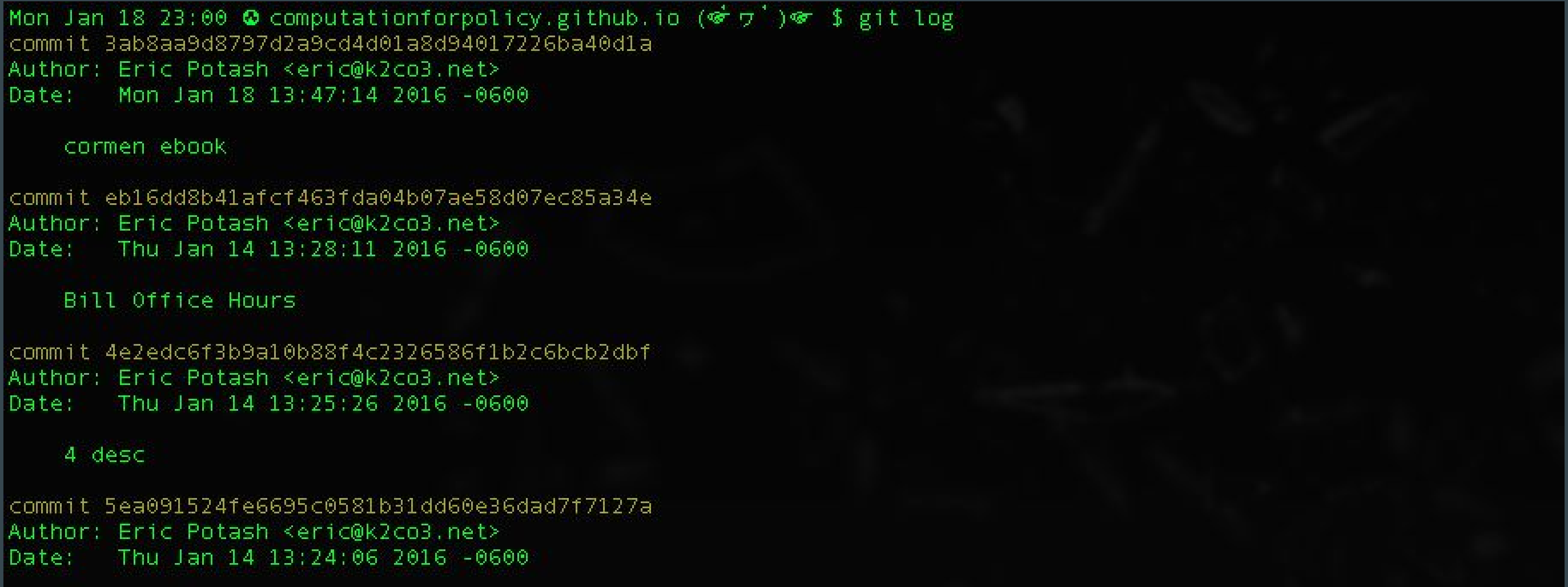
- See what the current status of your working directory is



```
Mon Jan 18 22:54 ♻ computationforpolicy.github.io (☞ﾟヮﾟ)☞ $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
Mon Jan 18 22:54 ♻ computationforpolicy.github.io (☞ﾟヮﾟ)☞ $ 
```

git log

- See the commit history of your project



```
Mon Jan 18 23:00 ♻ computationforpolicy.github.io (☞ﾟヮﾟ)☞ $ git log
commit 3ab8aa9d8797d2a9cd4d01a8d94017226ba40d1a
Author: Eric Potash <eric@k2co3.net>
Date:   Mon Jan 18 13:47:14 2016 -0600

    cormen ebook

commit eb16dd8b41afcf463fda04b07ae58d07ec85a34e
Author: Eric Potash <eric@k2co3.net>
Date:   Thu Jan 14 13:28:11 2016 -0600

    Bill Office Hours

commit 4e2edc6f3b9a10b88f4c2326586f1b2c6bcb2dbf
Author: Eric Potash <eric@k2co3.net>
Date:   Thu Jan 14 13:25:26 2016 -0600

    4 desc

commit 5ea091524fe6695c0581b31dd60e36dad7f7127a
Author: Eric Potash <eric@k2co3.net>
Date:   Thu Jan 14 13:24:06 2016 -0600
```

`git add`

- Tell git to start tracking them

```
git add myfile.txt
```

- Tell git to stage some files for the next commit

`git rm`

- Tell git to stop tracking some files

```
git rm myfile.txt
```

git reset

- Unstage a file

```
git reset myfile.txt
```

`git commit`

- Commit changes to the current branch

```
git commit -m “My descriptive commit message”
```

That’s everything you need for using git for version control on your local machine.

Review

Edit files with your text editor: nano, vi, emacs, Sublime text, etc.

Stage changes: `git add filename`

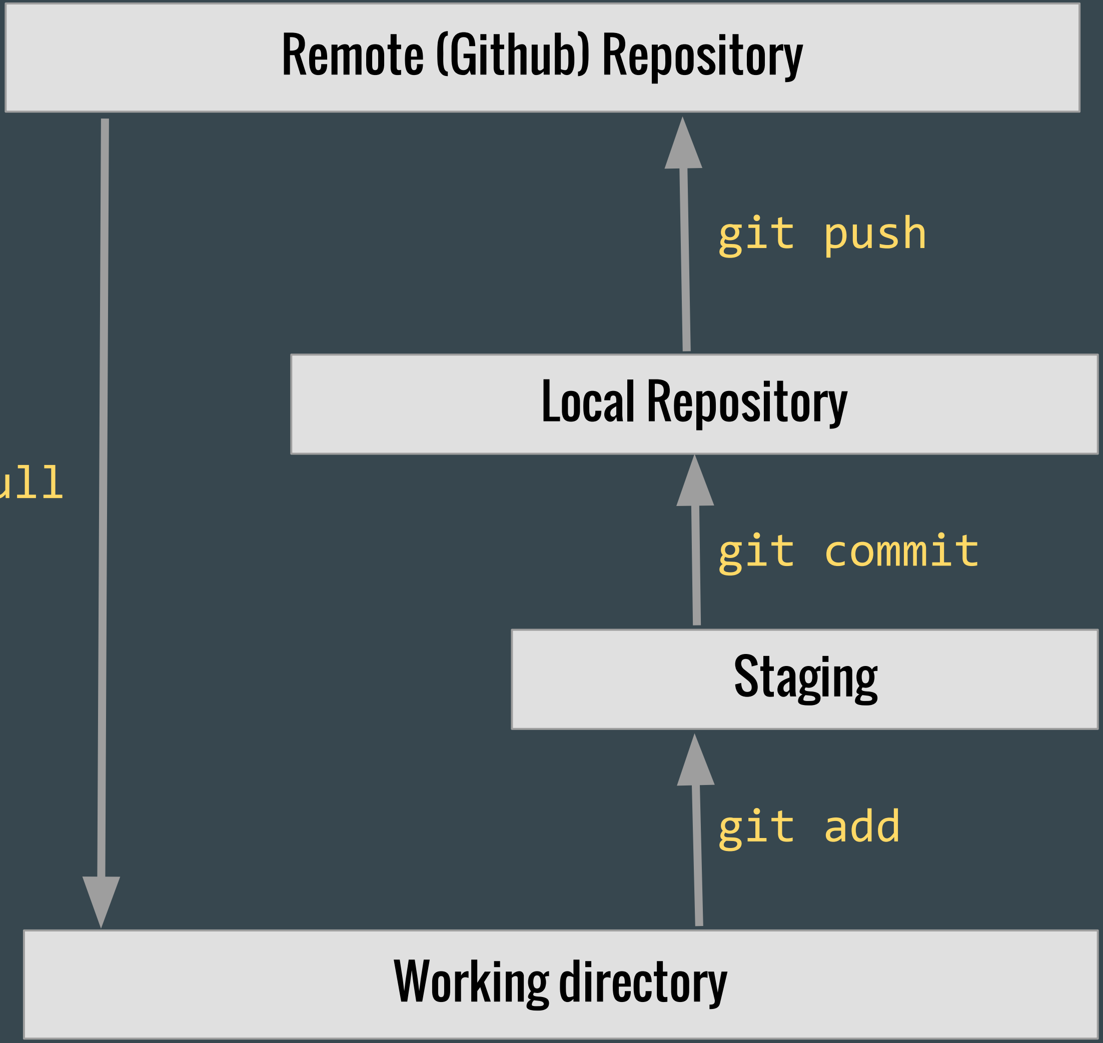
Check changes: `git status`

Commit changes: `git commit -m “Message here”`

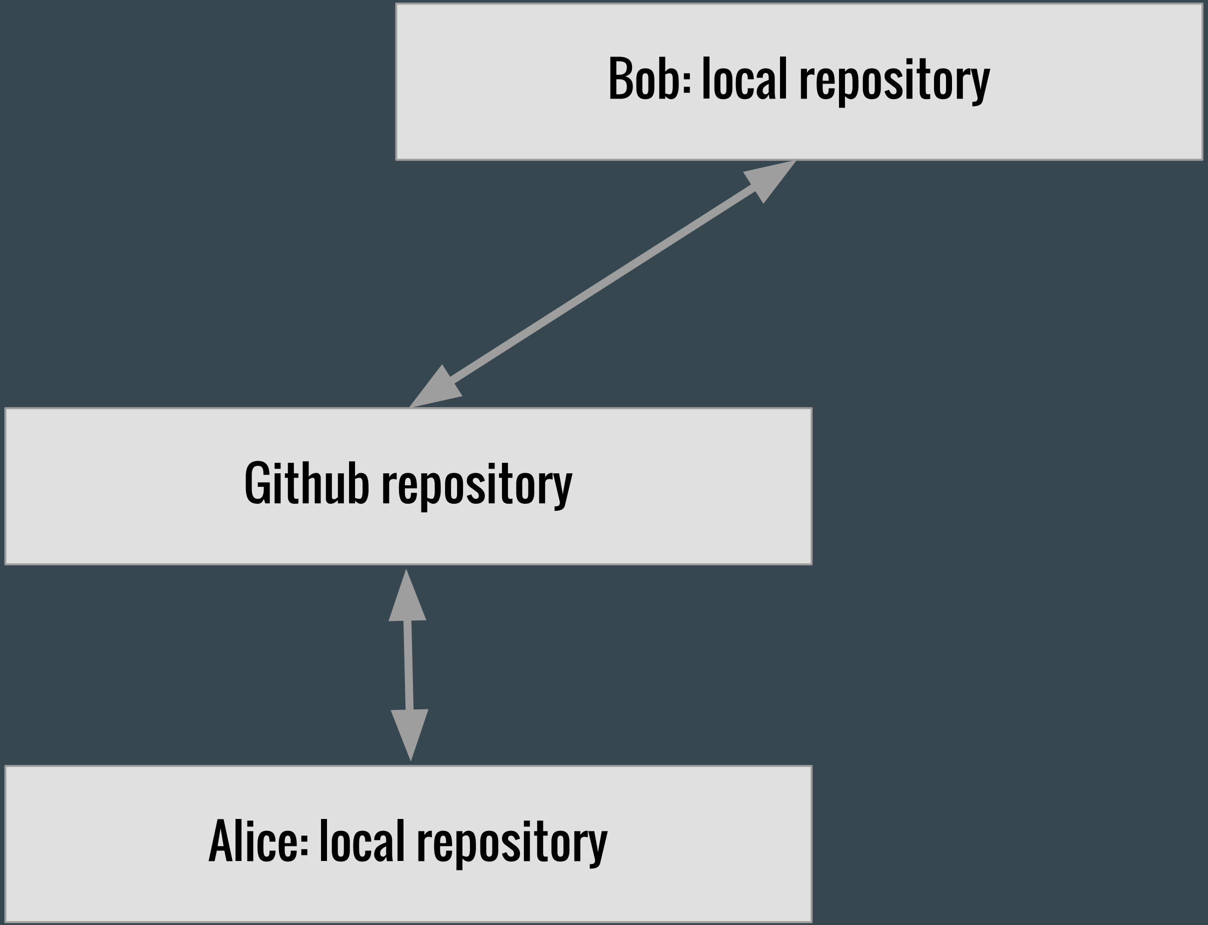
Important

- Almost nothing is ever deleted because the entire history is stored
- Can revert to any point in the history

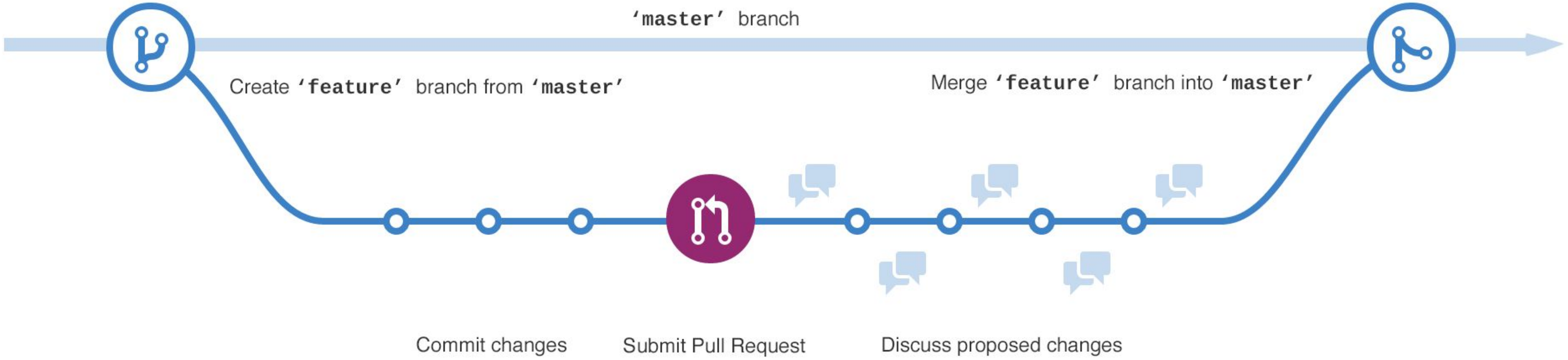
Using Github



Using Github Collaboratively



Branches


'master' branch
Create 'feature' branch from 'master'
Merge 'feature' branch into 'master'
Commit changes
Submit Pull Request
Discuss proposed changes

Connecting local repos with Github: Two options

1. Initialize repo locally, then create empty repo on Github and push local repo to Github:

   ```
   git remote add origin https://github.com/username/reponame.git
   ```

2. Initialize repo on Github, clone (copy) to local machine:

   ```
   git clone https://github.com/username/username.git
   ```

`git push`

- Push our local commits to our remote repository on Github

```
git push -u origin master
```

name of remote repo

branch name

`git pull`

- Pull down any changes that have been made:

```
git pull origin master
```

name of remote repo

branch name

.gitignore

- Tell git to never track certain files by putting them into .gitignore:

```
mysecretfile
```

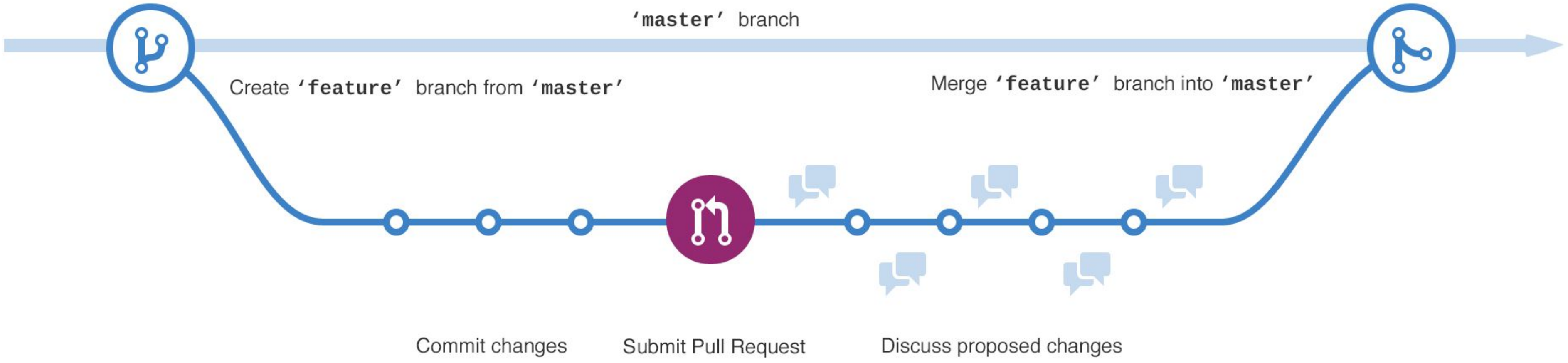
```
 *.pyc
```

- Files on Unix-based operating systems that begin with . are **hidden files** and can be shown by `ls -a`

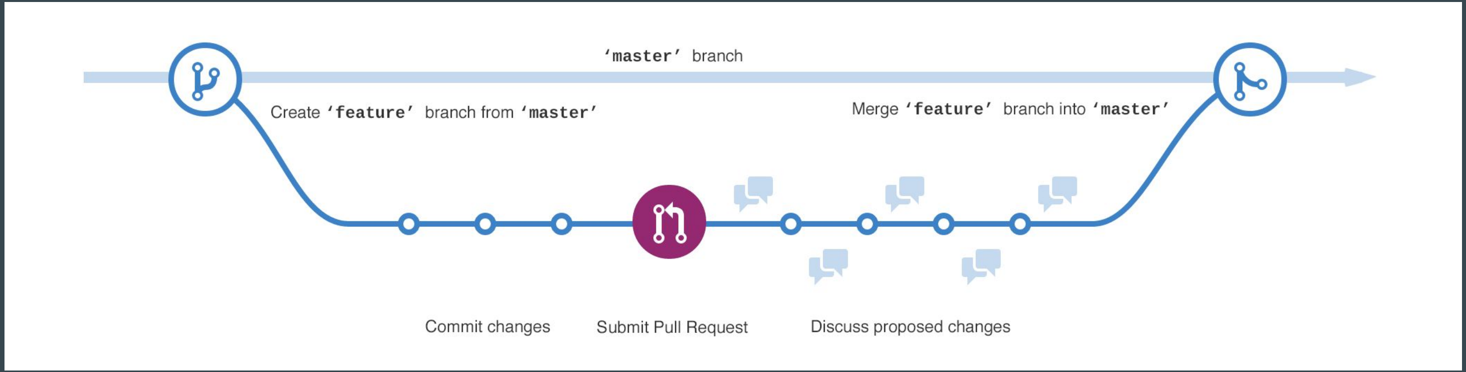
Important

- Every copy of the repo is a backup of the entire history
- Only need network for pulling and pushing

Using Branches


'master' branch
Create 'feature' branch from 'master'
Merge 'feature' branch into 'master'
Commit changes
Submit Pull Request
Discuss proposed changes

Branches


'master' branch
Create 'feature' branch from 'master'
Merge 'feature' branch into 'master'
Commit changes
Submit Pull Request
Discuss proposed changes

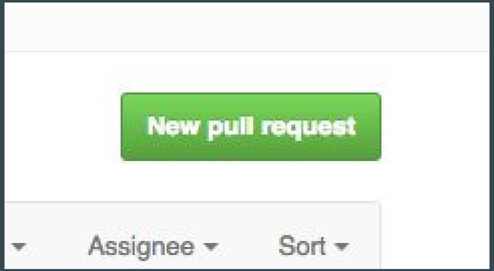
1. Make new branch: `git branch feature`
2. Switch to new branch: `git checkout feature`
3. Make some commits: `git commit ...`
4. Merge in feature branch:

   ```
   git checkout master
   ```

   ```
   git merge feature
   ```

5. Push everything to Github: `git push`

Pull request

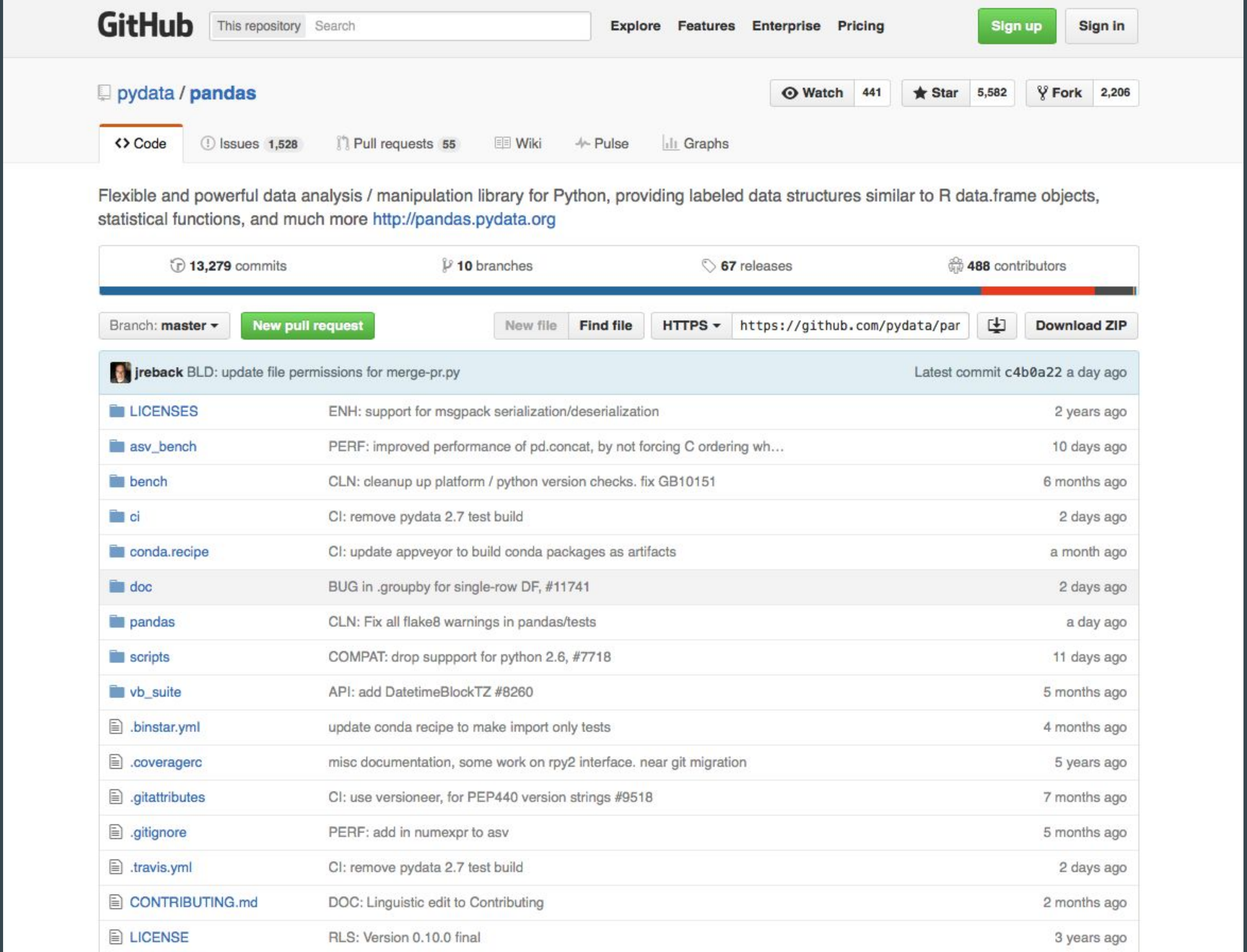


- Used instead of pushing directly to master
- Make some changes, e.g. add a new feature, in a branch and submit a pull request
- Maintainer reviews changes
- Discussion and further commits may occur
- When maintainer is happy with the new code, she accepts the pull request

Git nomenclature

- **HEAD**: Refers (points) to the current branch
- **origin**: Refers to the remote repo (Github)

Pandas



pydata / pandas

Watch 441 | Star 5,582 | Fork 2,206

Code | Issues 1,528 | Pull requests 55 | Wiki | Pulse | Graphs

Flexible and powerful data analysis / manipulation library for Python, providing labeled data structures similar to R data.frame objects, statistical functions, and much more http://pandas.pydata.org

13,279 commits | 10 branches | 67 releases | 488 contributors

Branch: master | New pull request | New file | Find file | HTTPS | https://github.com/pydata/par | Download ZIP

jreback BLD: update file permissions for merge-pr.py — Latest commit c4b0a22 a day ago

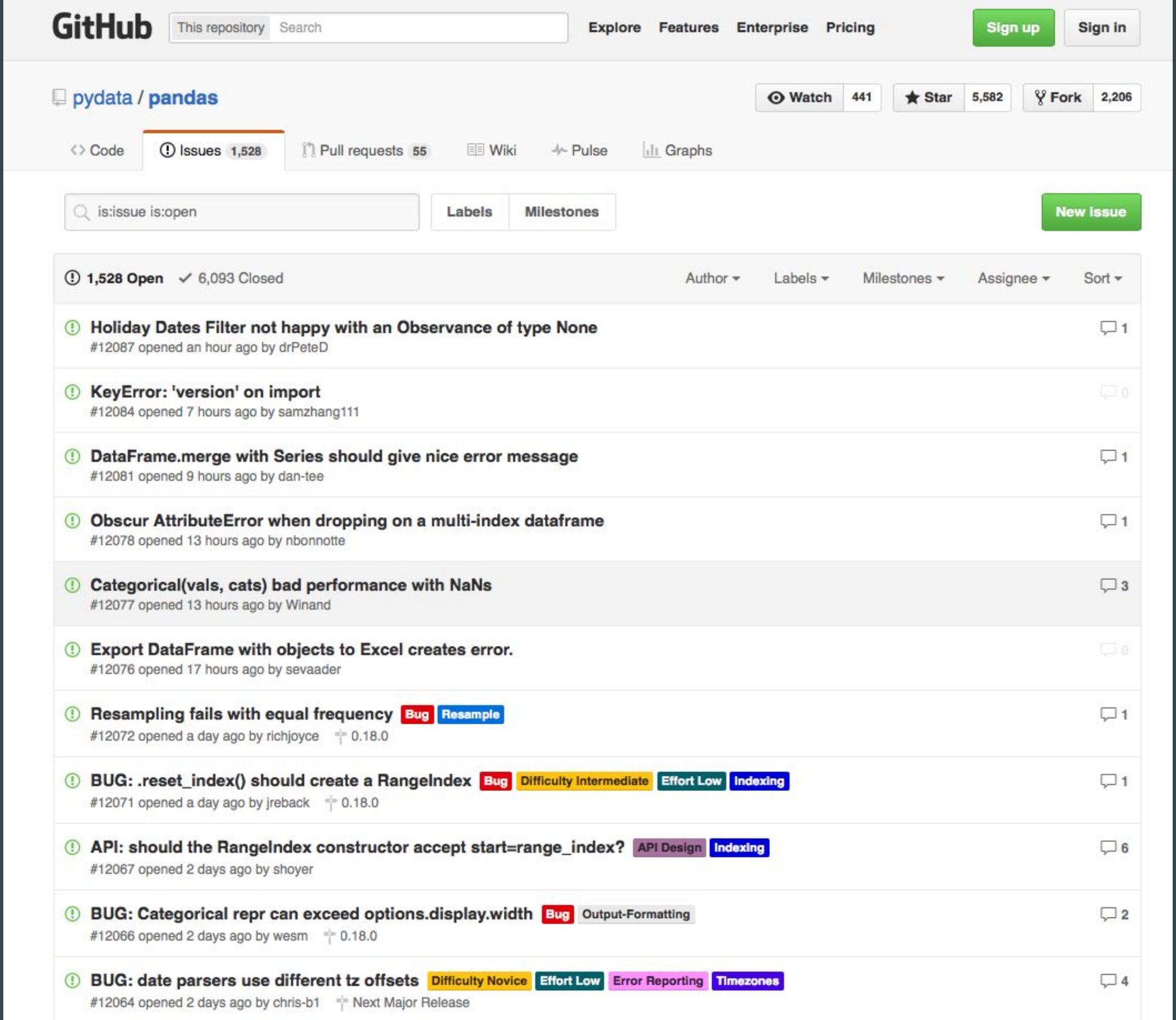
LICENSES	ENH: support for msgpack serialization/deserialization	2 years ago
asv_bench	PERF: improved performance of pd.concat, by not forcing C ordering wh…	10 days ago
bench	CLN: cleanup up platform / python version checks. fix GB10151	6 months ago
ci	CI: remove pydata 2.7 test build	2 days ago
conda.recipe	CI: update appveyor to build conda packages as artifacts	a month ago
doc	BUG in .groupby for single-row DF, #11741	2 days ago
pandas	CLN: Fix all flake8 warnings in pandas/tests	a day ago
scripts	COMPAT: drop suppport for python 2.6, #7718	11 days ago
vb_suite	API: add DatetimeBlockTZ #8260	5 months ago
.binstar.yml	update conda recipe to make import only tests	4 months ago
.coveragerc	misc documentation, some work on rpy2 interface. near git migration	5 years ago
.gitattributes	CI: use versioneer, for PEP440 version strings #9518	7 months ago
.gitignore	PERF: add in numexpr to asv	5 months ago
.travis.yml	CI: remove pydata 2.7 test build	2 days ago
CONTRIBUTING.md	DOC: Linguistic edit to Contributing	2 months ago
LICENSE	RLS: Version 0.10.0 final	3 years ago

https://github.com/pydata/pandas

Issues

Track bugs and feature requests

Especially useful for collaboration



Holiday Dates Filter not happy with an Observance of type None — 1
#12087 opened an hour ago by drPeteD

KeyError: 'version' on import — 0
#12084 opened 7 hours ago by samzhang111

DataFrame.merge with Series should give nice error message — 1
#12081 opened 9 hours ago by dan-tee

Obscur AttributeError when dropping on a multi-index dataframe — 1
#12078 opened 13 hours ago by nbonnotte

Categorical(vals, cats) bad performance with NaNs — 3
#12077 opened 13 hours ago by Winand

Export DataFrame with objects to Excel creates error. — 0
#12076 opened 17 hours ago by sevaader

Resampling fails with equal frequency [Bug] [Resample] — 1
#12072 opened a day ago by richjoyce 0.18.0

BUG: .reset_index() should create a RangeIndex [Bug] [Difficulty Intermediate] [Effort Low] [Indexing] — 1
#12071 opened a day ago by jreback 0.18.0

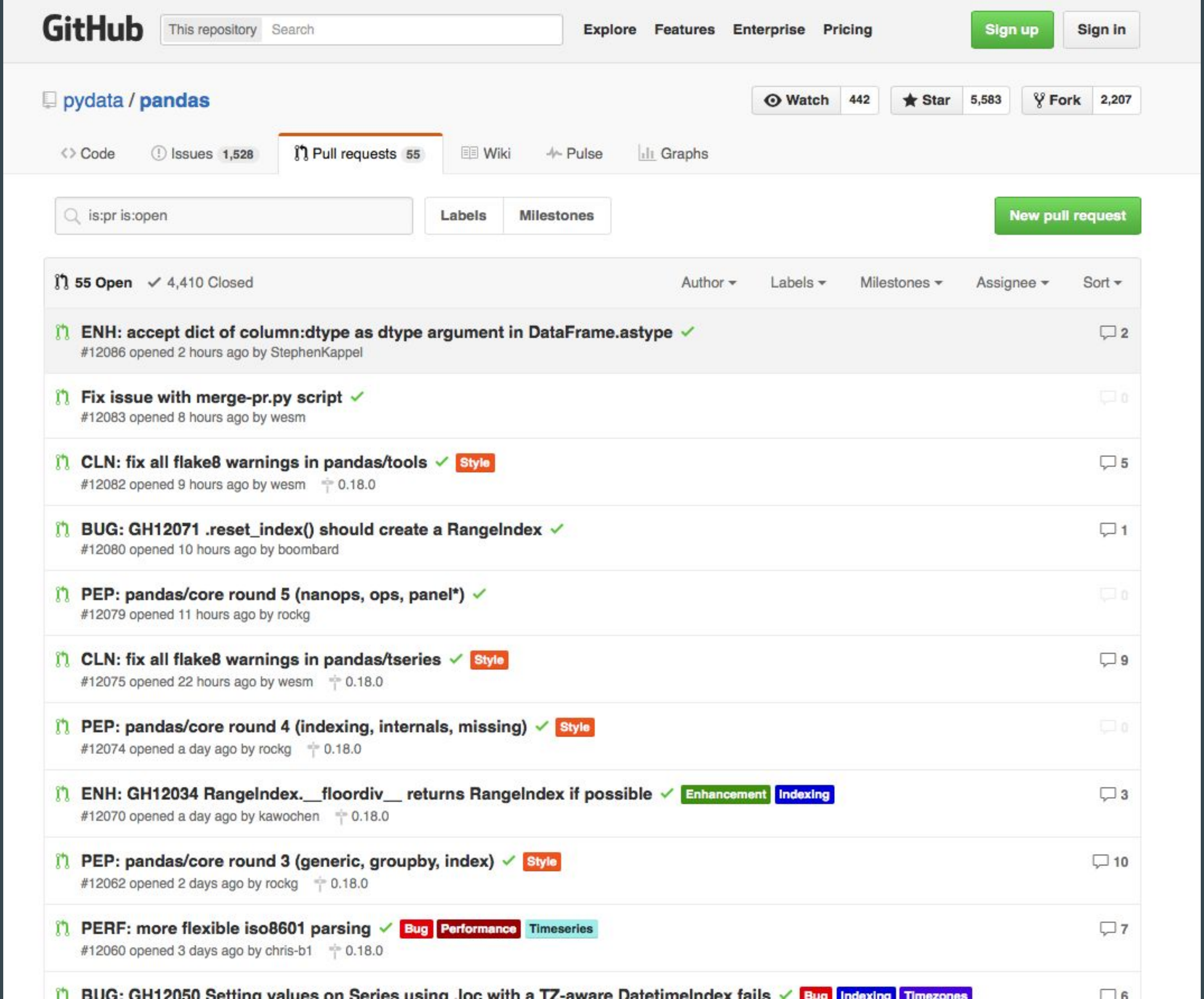
API: should the RangeIndex constructor accept start=range_index? [API Design] [Indexing] — 6
#12067 opened 2 days ago by shoyer

BUG: Categorical repr can exceed options.display.width [Bug] [Output-Formatting] — 2
#12066 opened 2 days ago by wesm 0.18.0

BUG: date parsers use different tz offsets [Difficulty Novice] [Effort Low] [Error Reporting] [Timezones] — 4
#12064 opened 2 days ago by chris-b1 Next Major Release

https://github.com/pydata/pandas/issues

Pull Requests



pydata / pandas

55 Open ✓ 4,410 Closed

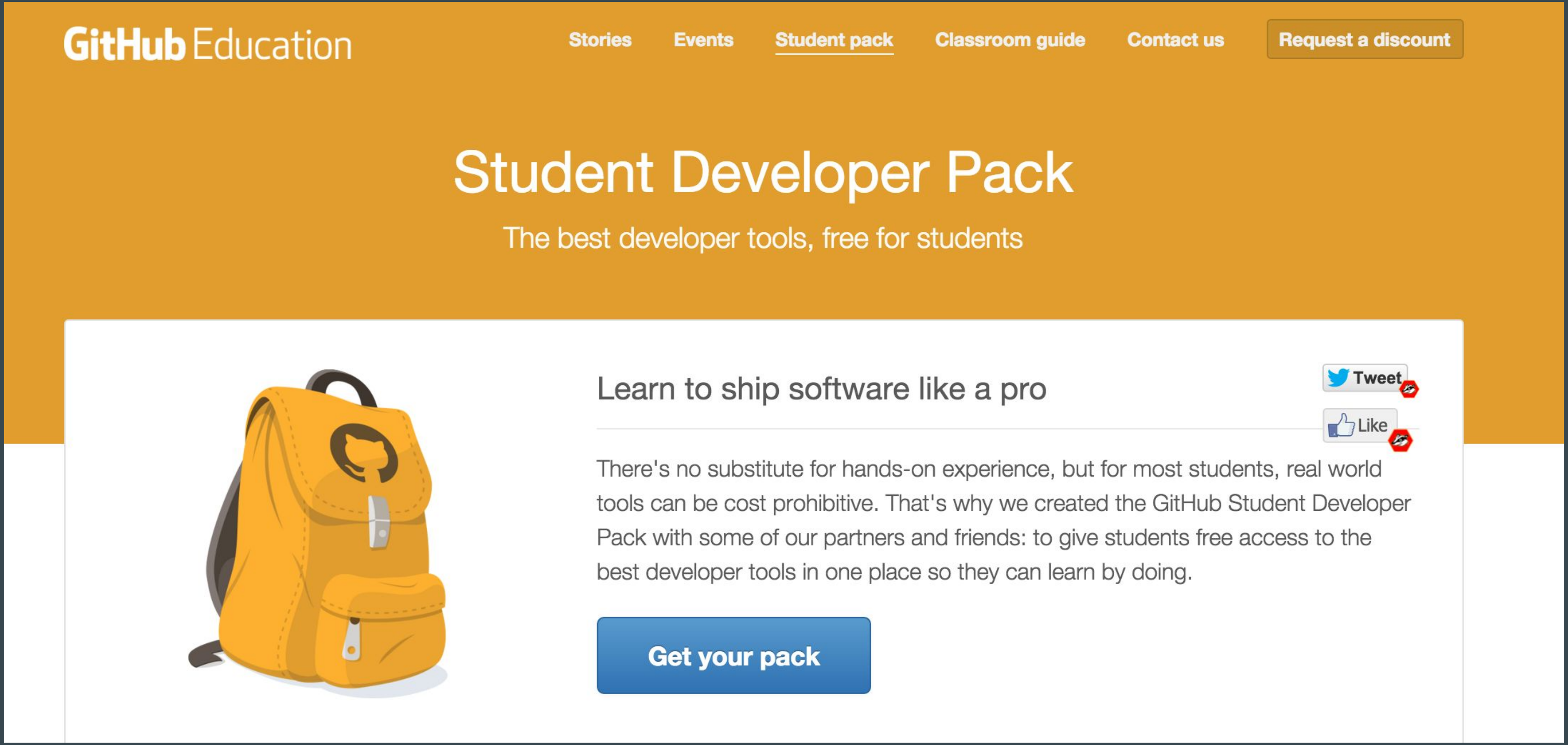
- ENH: accept dict of column:dtype as dtype argument in DataFrame.astype ✓ — 2
  #12086 opened 2 hours ago by StephenKappel
- Fix issue with merge-pr.py script ✓ — 0
  #12083 opened 8 hours ago by wesm
- CLN: fix all flake8 warnings in pandas/tools ✓ Style — 5
  #12082 opened 9 hours ago by wesm 0.18.0
- BUG: GH12071 .reset_index() should create a RangeIndex ✓ — 1
  #12080 opened 10 hours ago by boombard
- PEP: pandas/core round 5 (nanops, ops, panel*) ✓ — 0
  #12079 opened 11 hours ago by rockg
- CLN: fix all flake8 warnings in pandas/tseries ✓ Style — 9
  #12075 opened 22 hours ago by wesm 0.18.0
- PEP: pandas/core round 4 (indexing, internals, missing) ✓ Style — 0
  #12074 opened a day ago by rockg 0.18.0
- ENH: GH12034 RangeIndex.__floordiv__ returns RangeIndex if possible ✓ Enhancement Indexing — 3
  #12070 opened a day ago by kawochen 0.18.0
- PEP: pandas/core round 3 (generic, groupby, index) ✓ Style — 10
  #12062 opened 2 days ago by rockg 0.18.0
- PERF: more flexible iso8601 parsing ✓ Bug Performance Timeseries — 7
  #12060 opened 3 days ago by chris-b1 0.18.0
- BUG: GH12050 Setting values on Series using .loc with a TZ-aware DatetimeIndex fails ✓ Bug Indexing Timezones — 6

https://github.com/pydata/pandas/pulls

For Assignment 2: You will...

- Sign up for a student developer account on Github
- Make a private git repository on Github.com
- Add myself, Eric, and Bill as collaborators
- Use git to work on your homework

https://education.github.com/pack

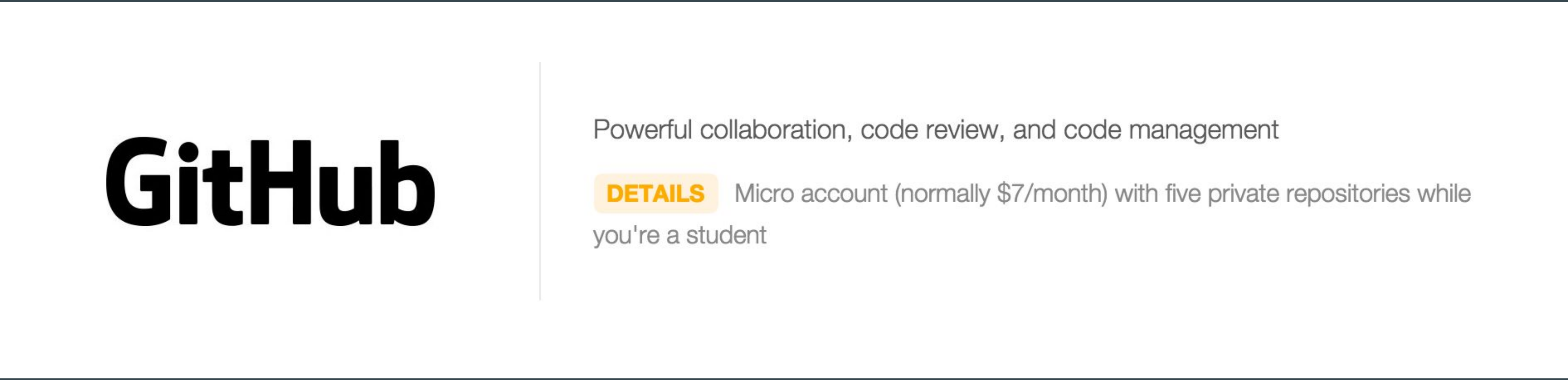


Student Developer Pack

The best developer tools, free for students

Learn to ship software like a pro

There's no substitute for hands-on experience, but for most students, real world tools can be cost prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.



GitHub

Powerful collaboration, code review, and code management

DETAILS Micro account (normally $7/month) with five private repositories while you're a student

Setting up git on your local machine

- Installed on the VM
- Instructions for getting your account set up with your machine:

  `https://help.github.com/articles/set-up-git/`

Commands to know

```
git init
git clone
git add
git status
git commit
git branch
```

```
git checkout
git merge
git push
git fetch
git pull
git log
```

Resources

- General programming advice:
  - “The Practice of Programming” by Kernighan and Pike
- Python style:
  - `https://www.python.org/dev/peps/pep-0008/`
  - `http://docs.python-guide.org/en/latest/writing/style/`
- Introduction to Github:
  - `https://guides.github.com/activities/hello-world/`
- Introduction to git:
  - `https://try.github.io/levels/1/challenges/1`

Resources

- General programming advice:
  - “The Practice of Programming” by Kernighan and Pike
- Python style:
  - `https://www.python.org/dev/peps/pep-0008/`
  - `http://docs.python-guide.org/en/latest/writing/style/`
- Introduction to Github:
  - `https://guides.github.com/activities/hello-world/`
- Introduction to git:
  - `https://try.github.io/levels/1/challenges/1`