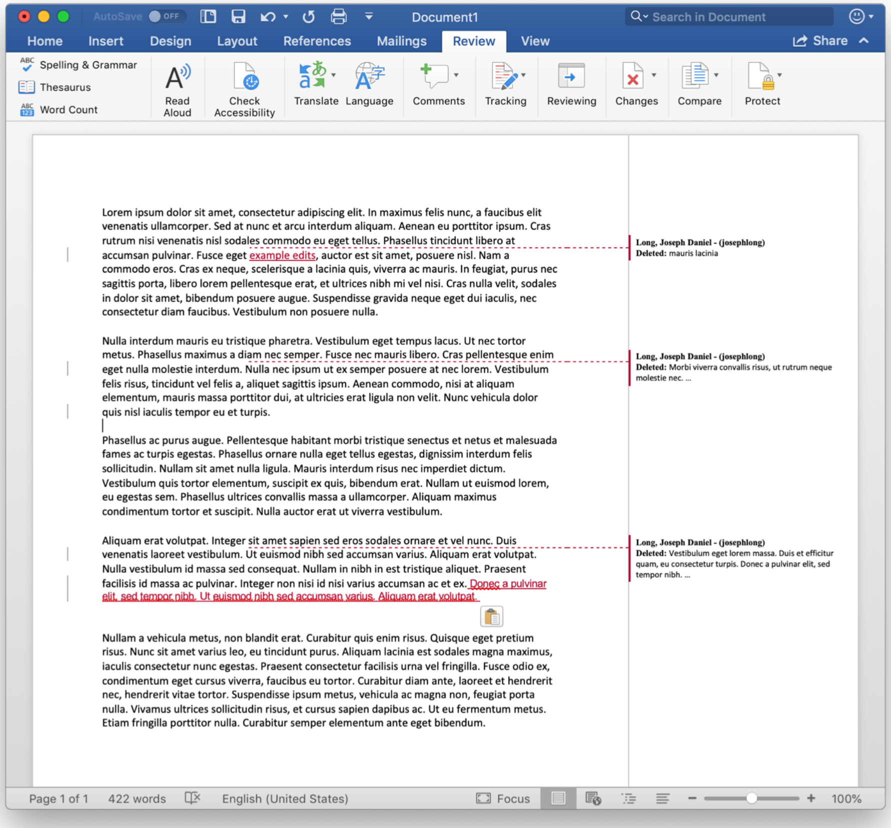
git and GitHub for research

Joseph Long

Code Coffee

November 13, 2018

What is version control?



Lorem ipsum dolor sit amet, consectetur adipiscing elit. In maximus felis nunc, a faucibus elit venenatis ullamcorper. Sed at nunc et arcu interdum aliquam. Aenean eu porttitor ipsum. Cras rutrum nisi venenatis nisl sodales commodo eu eget tellus. Phasellus tincidunt libero at accumsan pulvinar. Fusce eget example edits, auctor est sit amet, posuere nisl. Nam a commodo eros. Cras ex neque, scelerisque a lacinia quis, viverra ac mauris. In feugiat, purus nec sagittis porta, libero lorem pellentesque erat, et ultrices nibh mi vel nisi. Cras nulla velit, sodales in dolor sit amet, bibendum posuere augue. Suspendisse gravida neque eget dui iaculis, nec consectetur diam faucibus. Vestibulum non posuere nulla.

Nulla interdum mauris eu tristique pharetra. Vestibulum eget tempus lacus. Ut nec tortor metus. Phasellus maximus a diam nec semper. Fusce nec mauris libero. Cras pellentesque enim eget nulla molestie interdum. Nulla nec ipsum ut ex semper posuere at nec lorem. Vestibulum felis risus, tincidunt vel felis a, aliquet sagittis ipsum. Aenean commodo, nisi at aliquam elementum, mauris massa porttitor dui, at ultricies erat ligula non velit. Nunc vehicula dolor quis nisl iaculis tempor eu et turpis.

Phasellus ac purus augue. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Phasellus ornare nulla eget tellus egestas, dignissim interdum felis sollicitudin. Nullam sit amet nulla ligula. Mauris interdum risus nec imperdiet dictum. Vestibulum quis tortor elementum, suscipit ex quis, bibendum erat. Nullam ut euismod lorem, eu egestas sem. Phasellus ultrices convallis massa a ullamcorper. Aliquam maximus condimentum tortor et suscipit. Nulla auctor erat ut viverra vestibulum.

Aliquam erat volutpat. Integer sit amet sapien sed eros sodales ornare et vel nunc. Duis venenatis laoreet vestibulum. Ut euismod nibh sed accumsan varius. Aliquam erat volutpat. Nulla vestibulum id massa sed consequat. Nullam in nibh in est tristique aliquet. Praesent facilisis id massa ac pulvinar. Integer non nisi id nisi varius accumsan ac et ex. Donec a pulvinar elit, sed tempor nibh. Ut euismod nibh sed accumsan varius. Aliquam erat volutpat.

Nullam a vehicula metus, non blandit erat. Curabitur quis enim risus. Quisque eget pretium risus. Nunc sit amet varius leo, eu tincidunt purus. Aliquam lacinia est sodales magna maximus, iaculis consectetur nunc egestas. Praesent consectetur facilisis urna vel fringilla. Fusce odio ex, condimentum eget cursus viverra, faucibus eu tortor. Curabitur diam ante, laoreet et hendrerit nec, hendrerit vitae tortor. Suspendisse ipsum metus, vehicula ac magna non, feugiat porta nulla. Vivamus ultrices sollicitudin risus, et cursus sapien dapibus ac. Ut eu fermentum metus. Etiam fringilla porttitor nulla. Curabitur semper elementum ante eget bibendum.

Long, Joseph Daniel - (josephlong)
Deleted: mauris lacinia

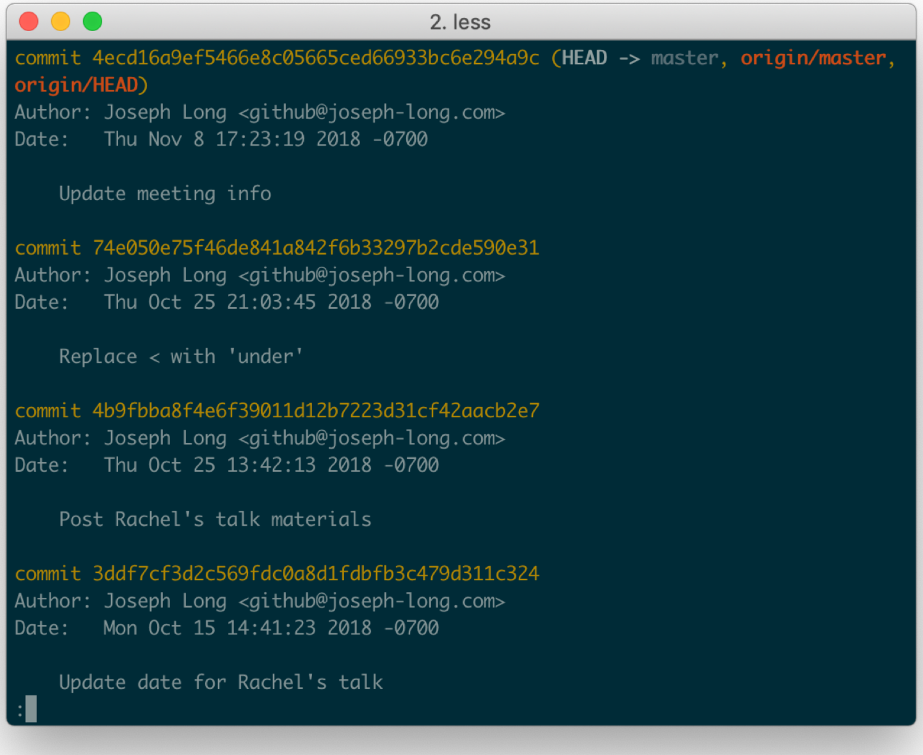
Long, Joseph Daniel - (josephlong)
Deleted: Morbi viverra convallis risus, ut rutrum neque molestie nec. ...

Long, Joseph Daniel - (josephlong)
Deleted: Vestibulum eget lorem massa. Duis et efficitur quam, eu consectetur turpis. Donec a pulvinar elit, sed tempor nibh. ...

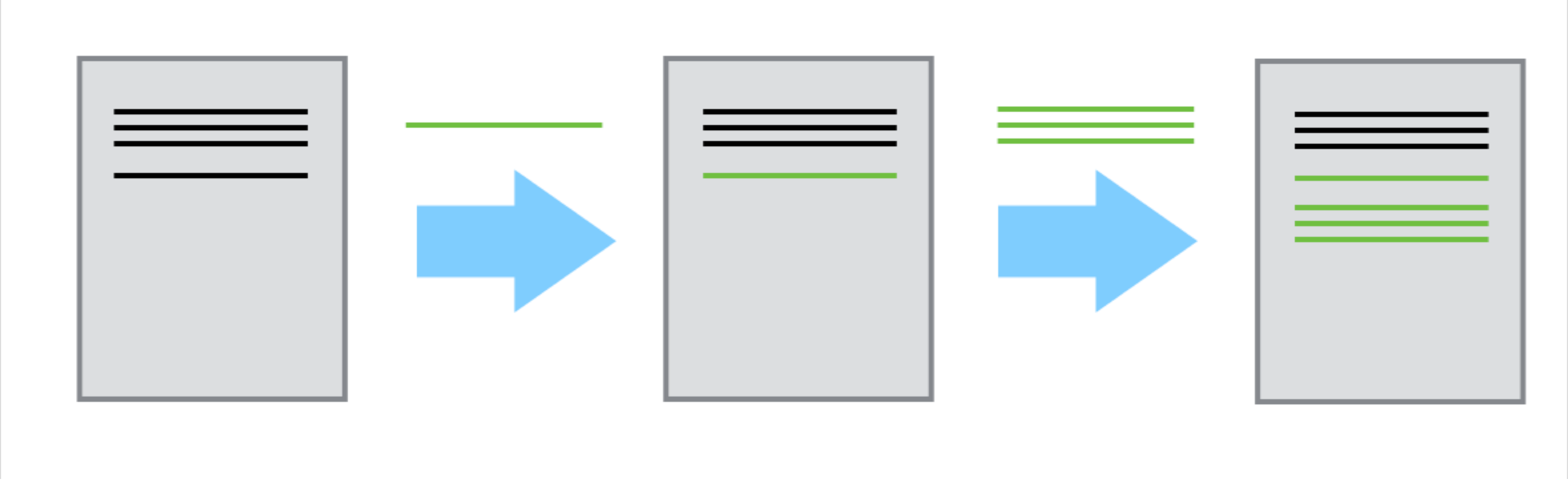
- Think of “track changes” for your source code
  - Or LaTeX documents, or…
- Make checkpoints (called *commits*) that store known good versions
- Allows you to *merge* changes from other sources (i.e. collaborators)

git

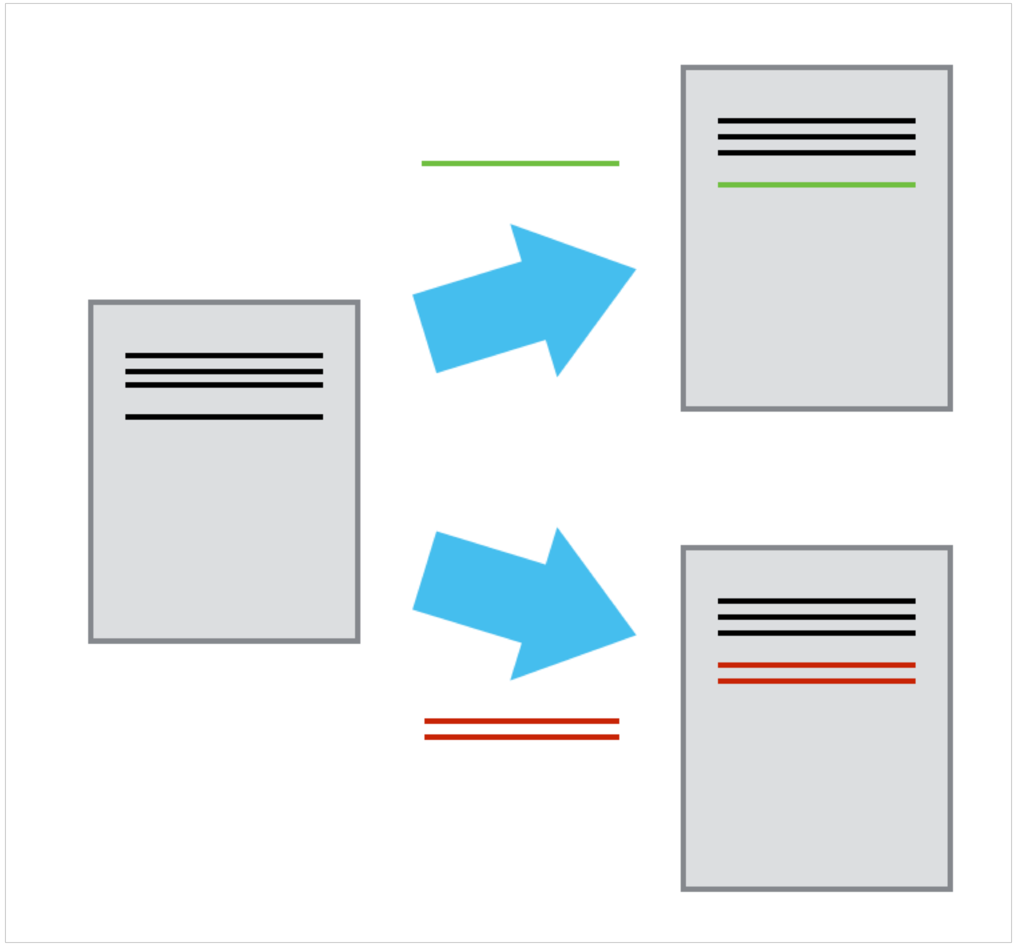
- Invented by Linus Torvalds (that Linux guy)
- Terminal (command-line) tool to make commits, perform merges, etc.
- Instead of tracking changes to a document (file), git tracks changes to a *repository* (folder of files)



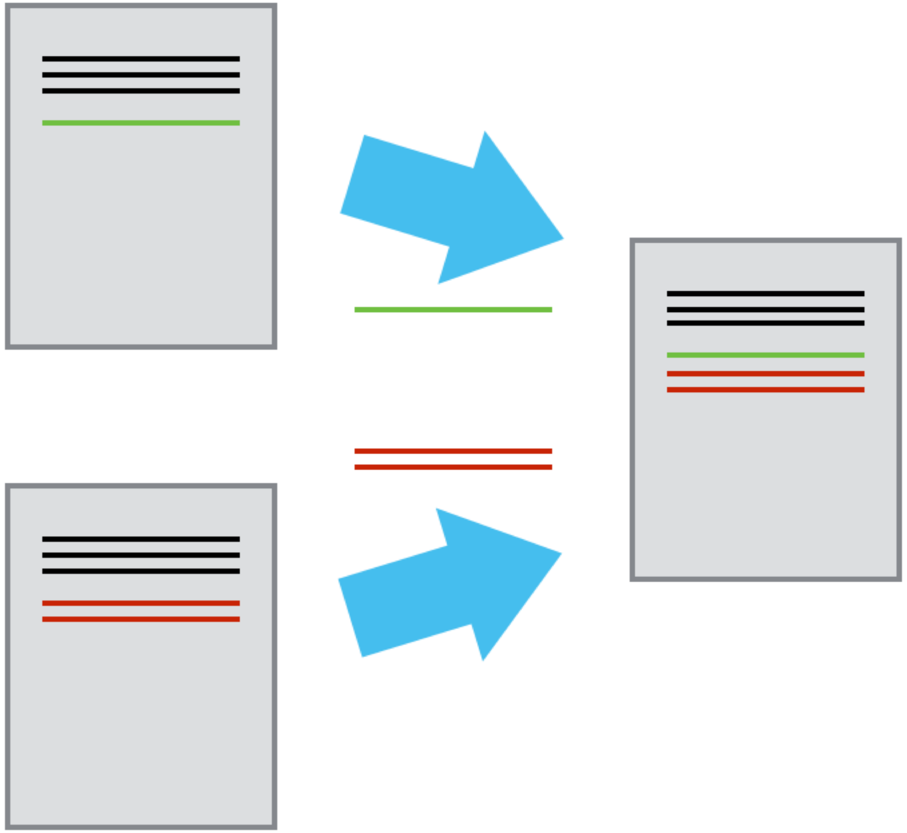
```
commit 4ecd16a9ef5466e8c05665ced66933bc6e294a9c (HEAD -> master, origin/master,
origin/HEAD)
Author: Joseph Long <github@joseph-long.com>
Date:   Thu Nov 8 17:23:19 2018 -0700

    Update meeting info

commit 74e050e75f46de841a842f6b33297b2cde590e31
Author: Joseph Long <github@joseph-long.com>
Date:   Thu Oct 25 21:03:45 2018 -0700

    Replace < with 'under'

commit 4b9fbba8f4e6f39011d12b7223d31cf42aacb2e7
Author: Joseph Long <github@joseph-long.com>
Date:   Thu Oct 25 13:42:13 2018 -0700

    Post Rachel's talk materials

commit 3ddf7cf3d2c569fdc0a8d1fdbfb3c479d311c324
Author: Joseph Long <github@joseph-long.com>
Date:   Mon Oct 15 14:41:23 2018 -0700

    Update date for Rachel's talk
:
```



Figures by the Software Carpentry Foundation + contributors
via http://swcarpentry.github.io/git-novice/01-basics/index.html used under CC BY 4.0

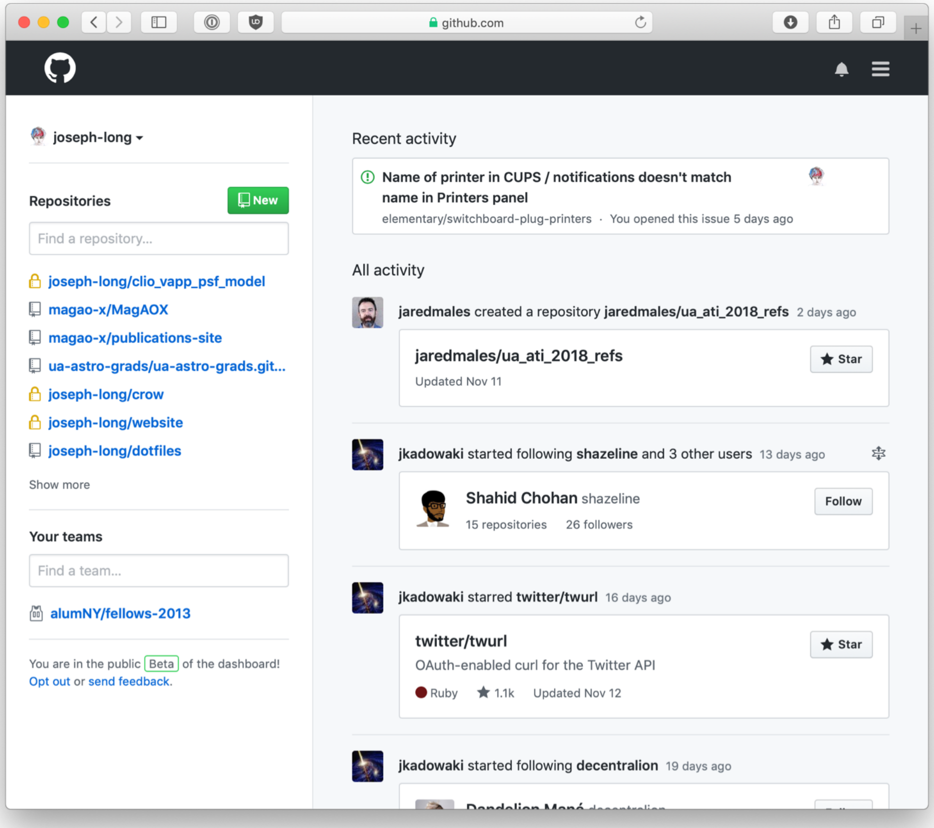


Figures by the Software Carpentry Foundation + contributors
via http://swcarpentry.github.io/git-novice/01-basics/index.html used under CC BY 4.0



Figures by the Software Carpentry Foundation + contributors
via http://swcarpentry.github.io/git-novice/01-basics/index.html used under CC BY 4.0

GitHub



github.com

joseph-long

Repositories New

Find a repository…

- joseph-long/clio_vapp_psf_model
- magao-x/MagAOX
- magao-x/publications-site
- ua-astro-grads/ua-astro-grads.git…
- joseph-long/crow
- joseph-long/website
- joseph-long/dotfiles

Show more

Your teams

Find a team…

alumNY/fellows-2013

You are in the public Beta of the dashboard! Opt out or send feedback.

Recent activity

Name of printer in CUPS / notifications doesn't match name in Printers panel
elementary/switchboard-plug-printers · You opened this issue 5 days ago

All activity

jaredmales created a repository jaredmales/ua_ati_2018_refs 2 days ago

jaredmales/ua_ati_2018_refs
Updated Nov 11 — ★ Star

jkadowaki started following shazeline and 3 other users 13 days ago

Shahid Chohan shazeline
15 repositories 26 followers — Follow

jkadowaki starred twitter/twurl 16 days ago

twitter/twurl
OAuth-enabled curl for the Twitter API
● Ruby ★ 1.1k Updated Nov 12 — ★ Star

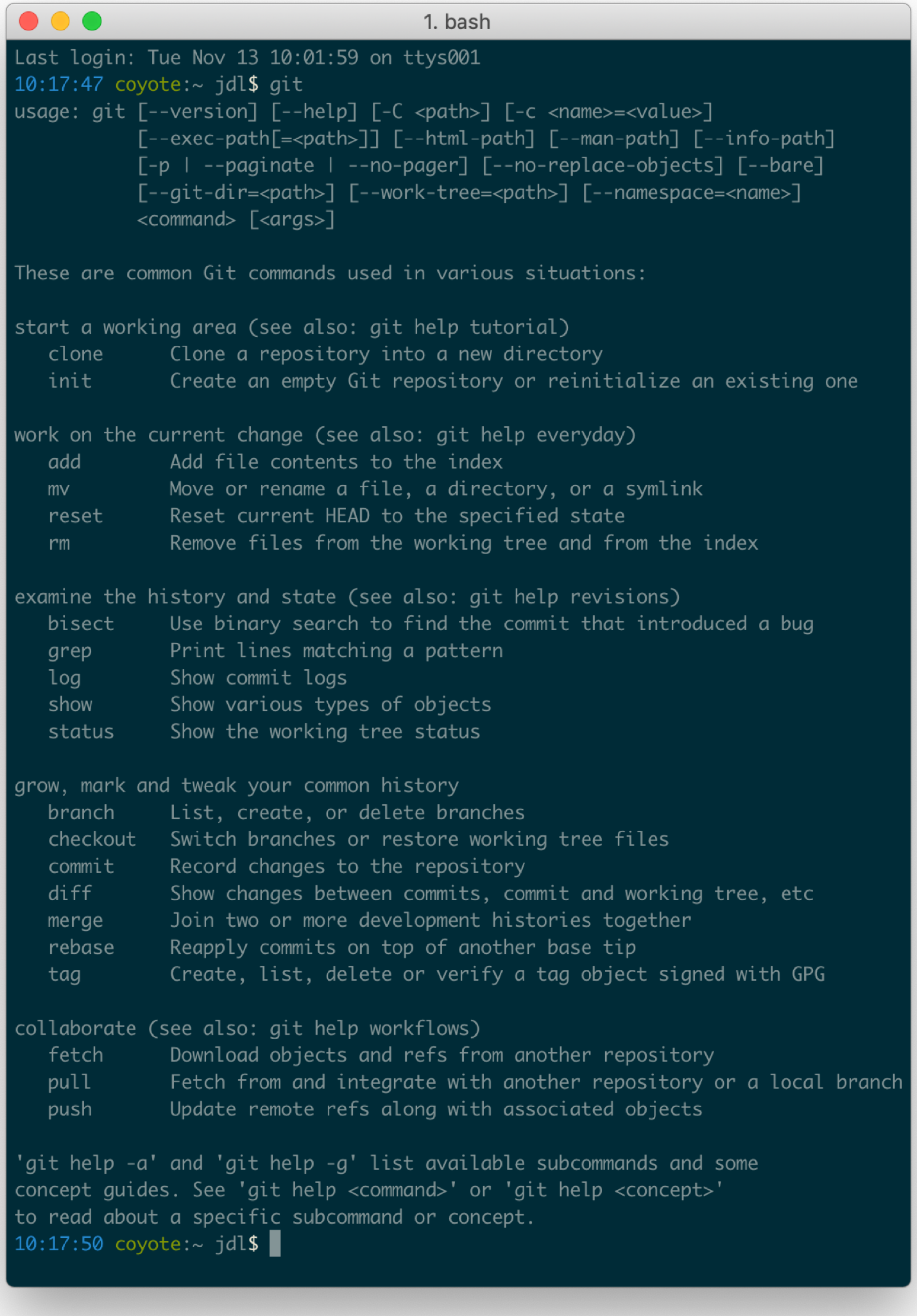
jkadowaki started following decentralion 19 days ago

Dandelion Mané decentralion

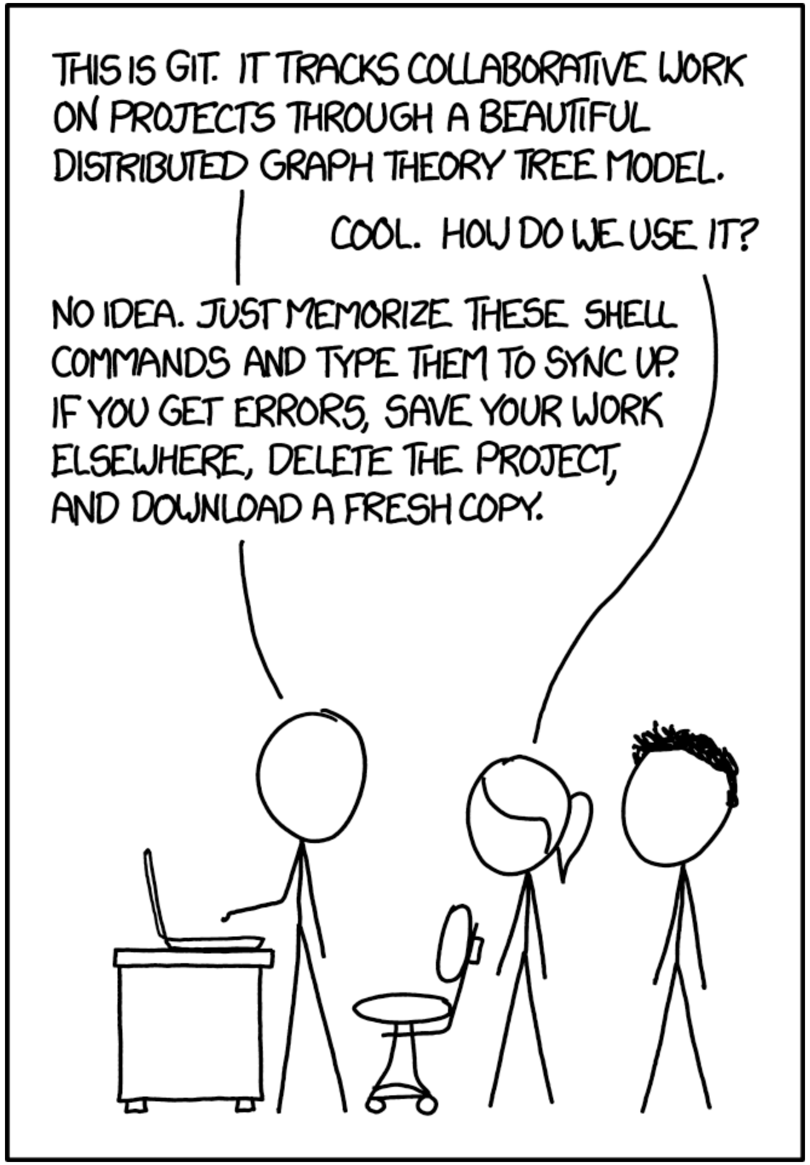
- Commercial service to store and share git repositories
  - free for open-source
- https://github.com/
- Includes issue tracker (i.e. bug reporting)
- Web-based change requests called *pull requests*
- Used by Astropy, numpy, scipy, millions more

Getting started with git

1. Open a terminal
2. Type `git`
3. Witness the overwhelming number of options



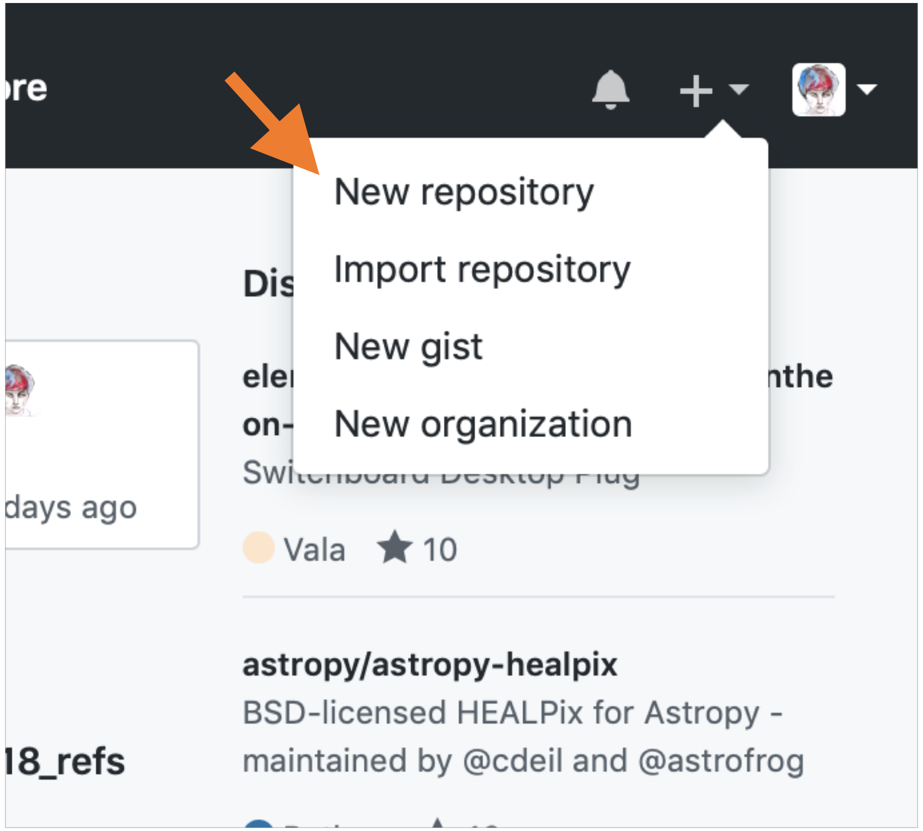
```
Last login: Tue Nov 13 10:01:59 on ttys001
10:17:47 coyote:~ jdl$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone      Clone a repository into a new directory
   init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
   add        Add file contents to the index
   mv         Move or rename a file, a directory, or a symlink
   reset      Reset current HEAD to the specified state
   rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
   bisect     Use binary search to find the commit that introduced a bug
   grep       Print lines matching a pattern
   log        Show commit logs
   show       Show various types of objects
   status     Show the working tree status

grow, mark and tweak your common history
   branch     List, create, or delete branches
   checkout   Switch branches or restore working tree files
   commit     Record changes to the repository
   diff       Show changes between commits, commit and working tree, etc
   merge      Join two or more development histories together
   rebase     Reapply commits on top of another base tip
   tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
   fetch      Download objects and refs from another repository
   pull       Fetch from and integrate with another repository or a local branch
   push       Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
10:17:50 coyote:~ jdl$
```


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” by Randall Munroe, CC BY-NC 2.5

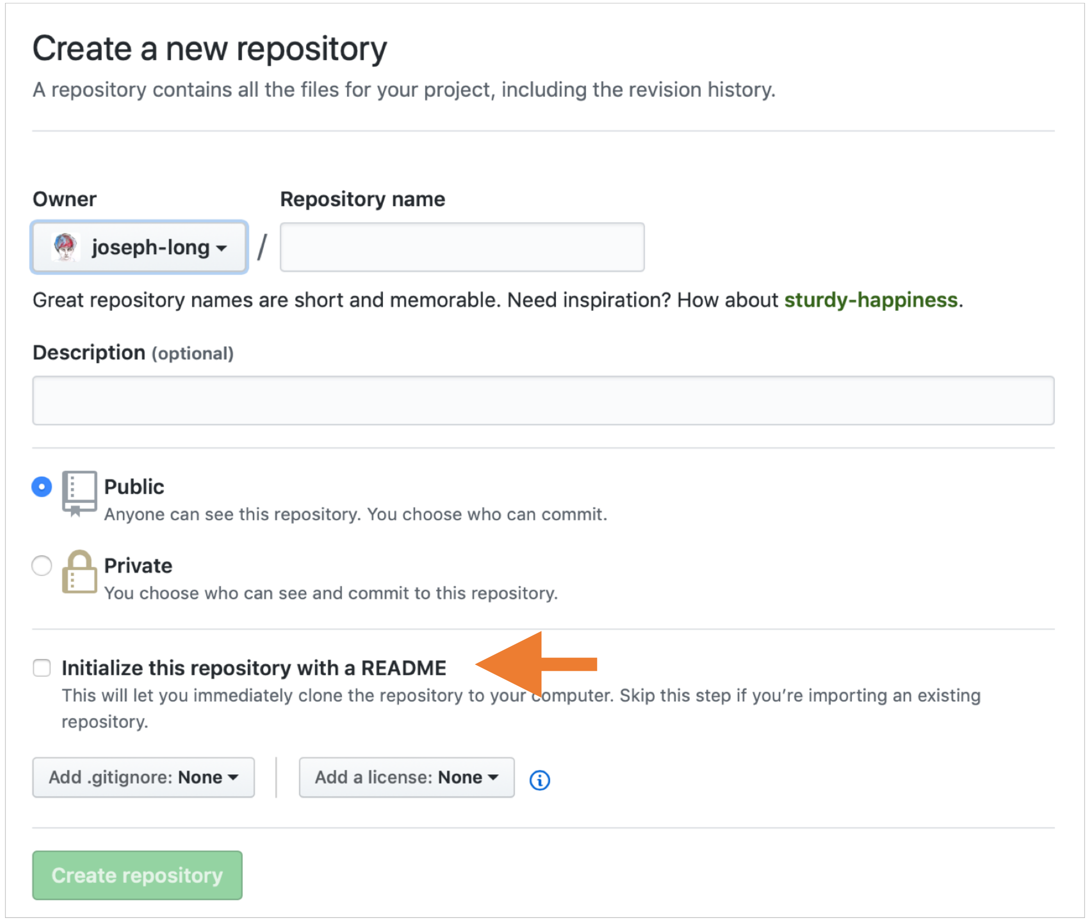
Getting started with ~~git~~ GitHub

- Log in to https://github.com/
- Click the "+" icon at top right



Getting started with ~~git~~ GitHub

- Log in to https://github.com/
- Click the "+" icon at top right
- Click "Initialize this repository with a README"
  - As the instructions say, you don't always want this.
  - You can initialize a repository yourself locally with `git init`, but then connecting up to GitHub is more complex



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner joseph-long / **Repository name**

Great repository names are short and memorable. Need inspiration? How about **sturdy-happiness**.

Description (optional)

● **Public**
Anyone can see this repository. You choose who can commit.

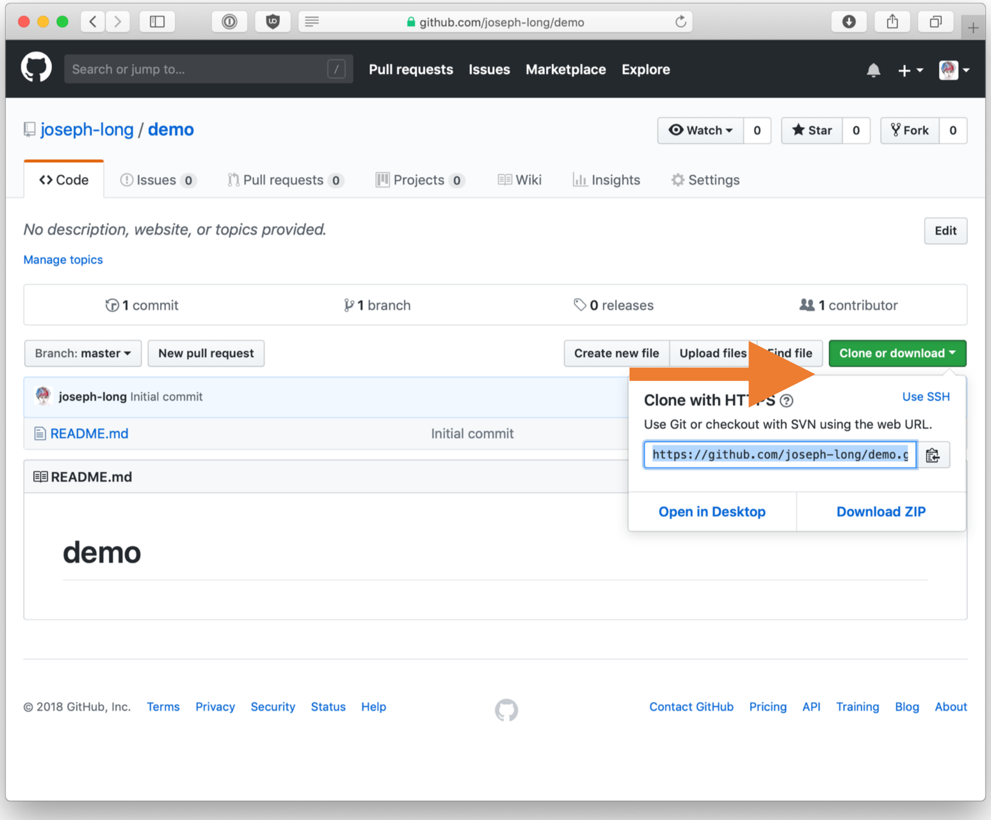
○ **Private**
You choose who can see and commit to this repository.

□ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None**

Make a local clone of your new repository

- We say *clone* and not copy because we get two things
  1. a copy of all the files
  2. a complete history of the repository
- Copy the repository URL from GitHub
- Open a terminal window and `git clone https://github.com/joseph-long/demo.git`
  - Well, modified for your username



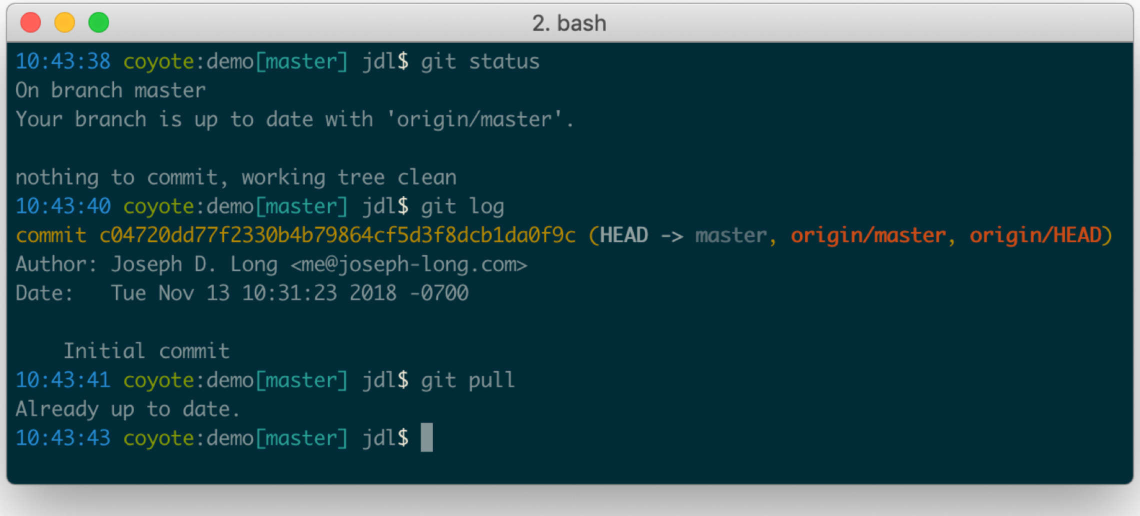
No description, website, or topics provided.

Use Git or checkout with SVN using the web URL.

`https://github.com/joseph-long/demo.g`

Getting started with ~~GitHub~~ git, actually

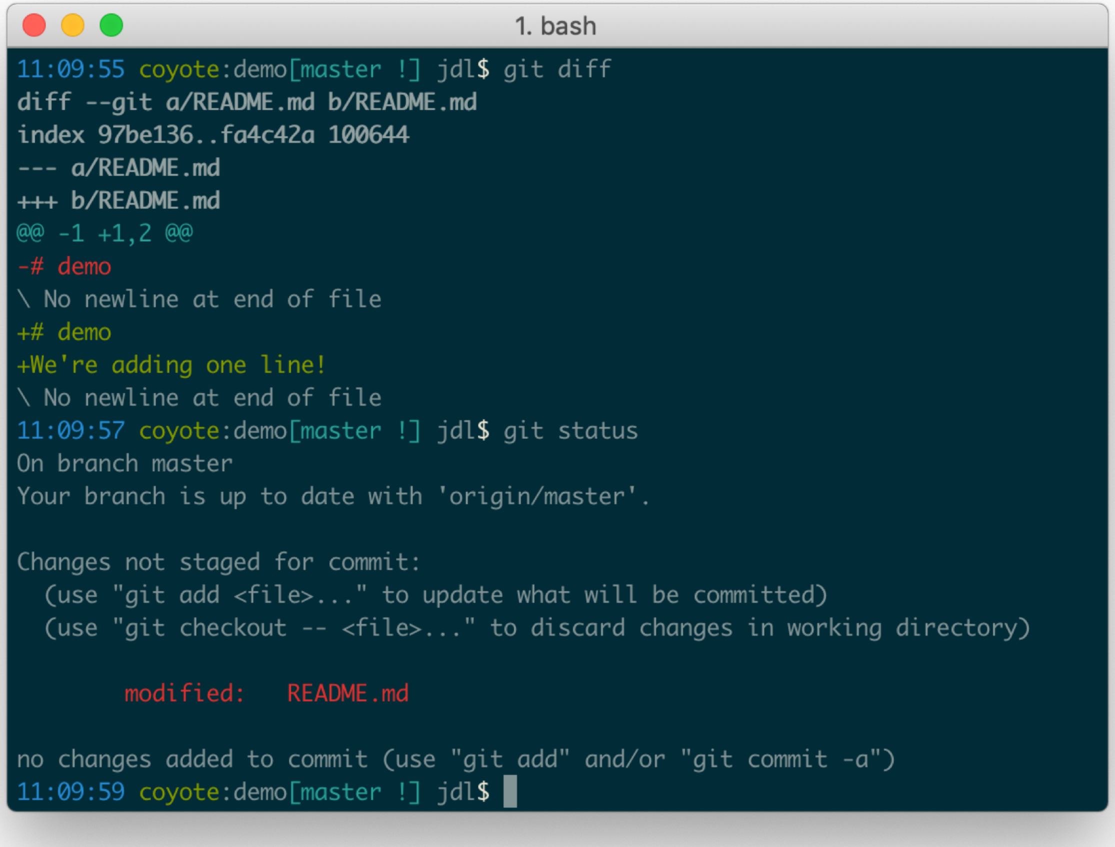
- Verify you have a new “demo” directory, “cd” into it
- Three commands you’ll use all the time
  - `git status`
  - `git log`
  - `git pull`
- One option you’ll use some of the time
  - `git [subcommand] --help`



```
10:43:38 coyote:demo[master] jdl$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
10:43:40 coyote:demo[master] jdl$ git log
commit c04720dd77f2330b4b79864cf5d3f8dcb1da0f9c (HEAD -> master, origin/master, origin/HEAD)
Author: Joseph D. Long <me@joseph-long.com>
Date:   Tue Nov 13 10:31:23 2018 -0700

    Initial commit
10:43:41 coyote:demo[master] jdl$ git pull
Already up to date.
10:43:43 coyote:demo[master] jdl$
```

Making a change

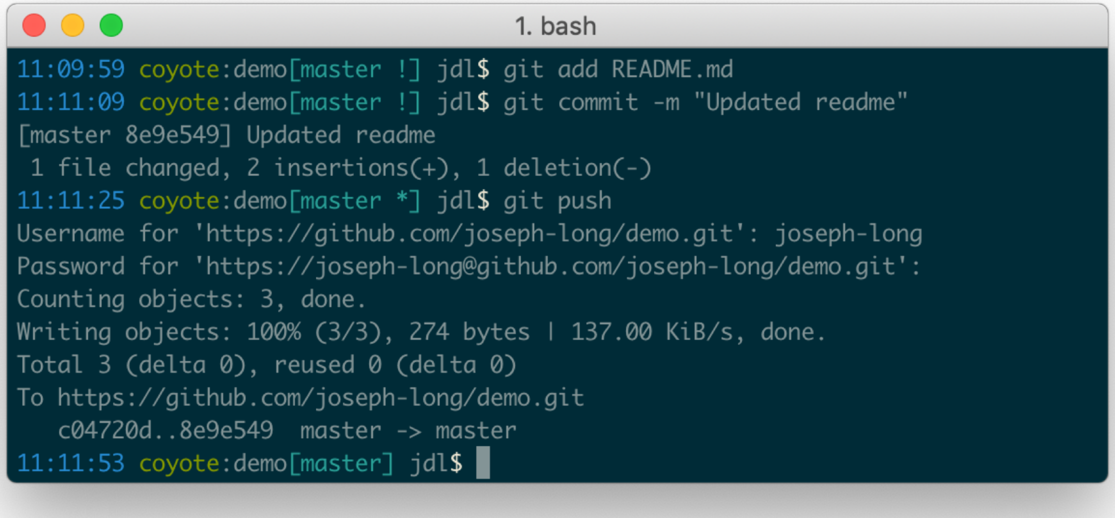
- Four **more** commands you'll use all the time
  - `git diff`
  - `git add`
  - `git commit`
  - `git push`
- We're going to edit the README.md file
- Use `git diff` to see just your changes
- After editing, `git status` again
  - The status message gives you hints…



```
11:09:55 coyote:demo[master !] jdl$ git diff
diff --git a/README.md b/README.md
index 97be136..fa4c42a 100644
--- a/README.md
+++ b/README.md
@@ -1 +1,2 @@
-# demo
\ No newline at end of file
+# demo
+We're adding one line!
\ No newline at end of file
11:09:57 coyote:demo[master !] jdl$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
11:09:59 coyote:demo[master !] jdl$
```

Making a change

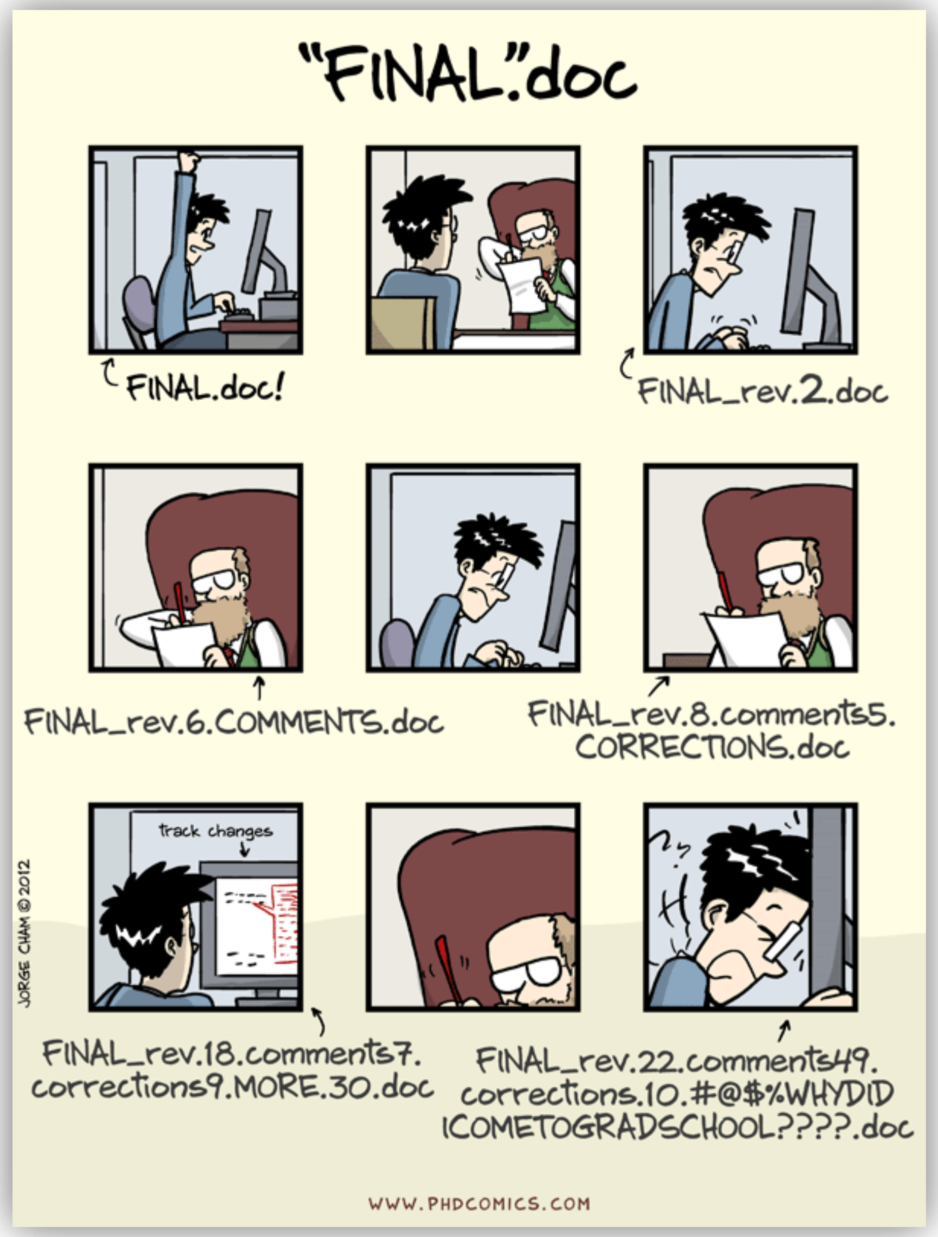
- A brief digression on the "staging area"
  - Or, when to `git add`
- After adding, `git commit` your changes
  - Forget something after you committed? `git commit --amend`
- And `git push` them
  - Git remembers where you cloned from, and will default to pushing changes there



```
11:09:59 coyote:demo[master !] jdl$ git add README.md
11:11:09 coyote:demo[master !] jdl$ git commit -m "Updated readme"
[master 8e9e549] Updated readme
 1 file changed, 2 insertions(+), 1 deletion(-)
11:11:25 coyote:demo[master *] jdl$ git push
Username for 'https://github.com/joseph-long/demo.git': joseph-long
Password for 'https://joseph-long@github.com/joseph-long/demo.git':
Counting objects: 3, done.
Writing objects: 100% (3/3), 274 bytes | 137.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/joseph-long/demo.git
   c04720d..8e9e549  master -> master
11:11:53 coyote:demo[master] jdl$ 
```

What would you like to talk about?

1. Using branches to experiment with different ways to evolve your code in parallel
2. Collaborating with others through GitHub Pull Requests
3. Resolving merge conflicts
4. Something else?


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

Additional resources

- Cheat sheet by GitLab
  - https://about.gitlab.com/images/press/git-cheat-sheet.pdf
- Hosting services
  - https://bitbucket.org
  - https://about.gitlab.com
- Free services for students
  - https://education.github.com/students
- Software Carpentry lessons
  - http://swcarpentry.github.io/git-novice/