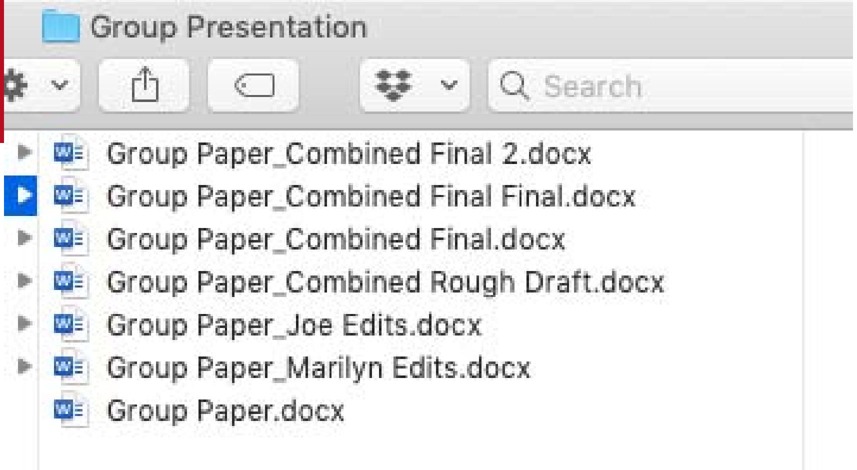
Goals for today:

- Use Git/Github to version control your code
- Contribute to group code efforts
- Clone a public code repository and submit your suggested revisions

Why version control?


Group Presentation

Group Paper_Combined Final 2.docx
Group Paper_Combined Final Final.docx
Group Paper_Combined Final.docx
Group Paper_Combined Rough Draft.docx
Group Paper_Joe Edits.docx
Group Paper_Marilyn Edits.docx
Group Paper.docx

- Your folders look like this -->
- Add 1 feature, break everything "time machine"
- Informative comments and differences between versions
- Try a new idea out without losing place
- Collaborative coding: I work on section, you work on other
- Reproducible code with a DOI (citations)
- Open code that collaborators can use and improve

git vs GitHub

- Software
- Installed locally
- Version control tool

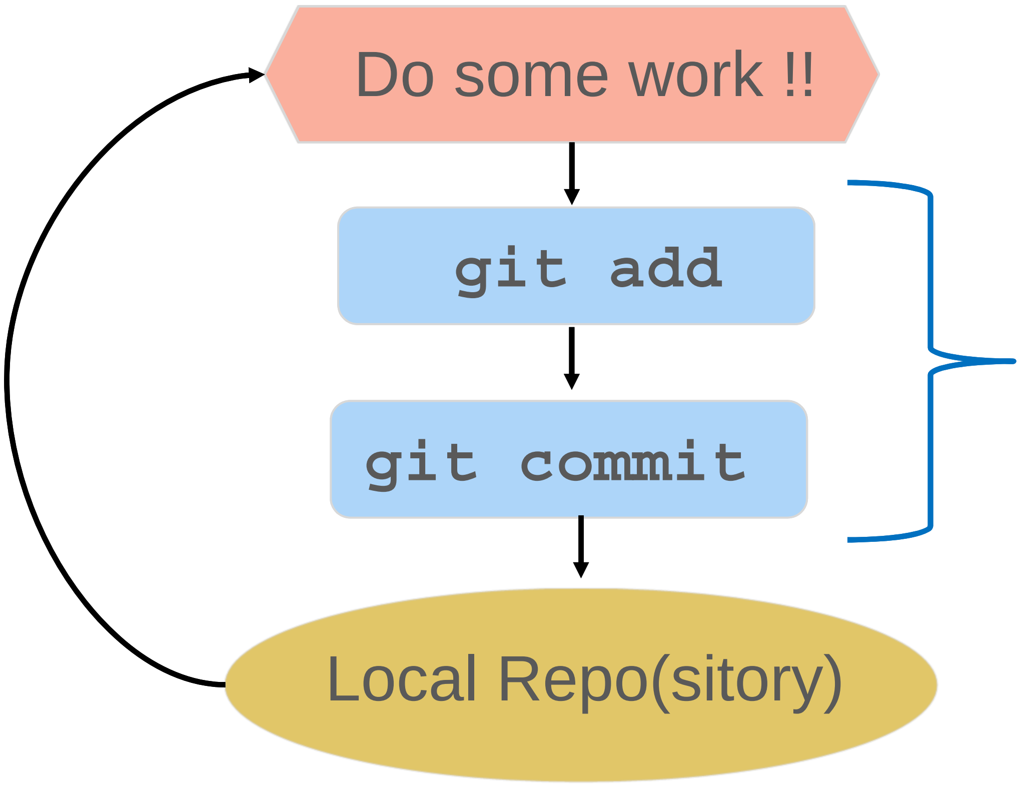
Do I have to use Github to use Git? **No**

- Service
- Hosted on web
- Online space to hold git repository
- One of many options (bitbucket, gitlab), but by far most common

General Idea

“Git allows groups of people to work on the same documents at the same time, and without stepping on each other’s toes.”

Typical workflow: Local computer



Do some work !!

`git add`

`git commit`

Local Repo(sitory)

Analogous to a file save.
Snapshot of all files
in directory.

Commits have more info:

- Date/time stamp
- Commit message
- Unique identifier

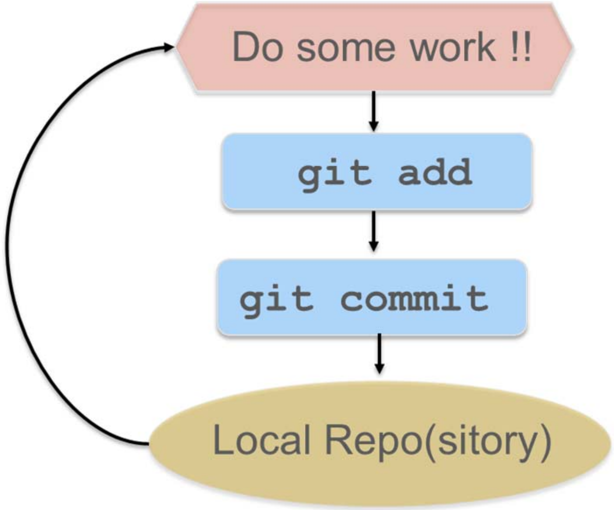
New terminology

- **Commit:** Snapshot of files at a given time
  Records the time, “track changes” from previous commit, and a unique identifier
  “I just made a new commit” (noun) or
  “I committed my code” (verb)

  Repository: shortened to “repo”
  Collection of all files and history of the files (commits)
  Can be on your local machine or remote server (github)

Commits

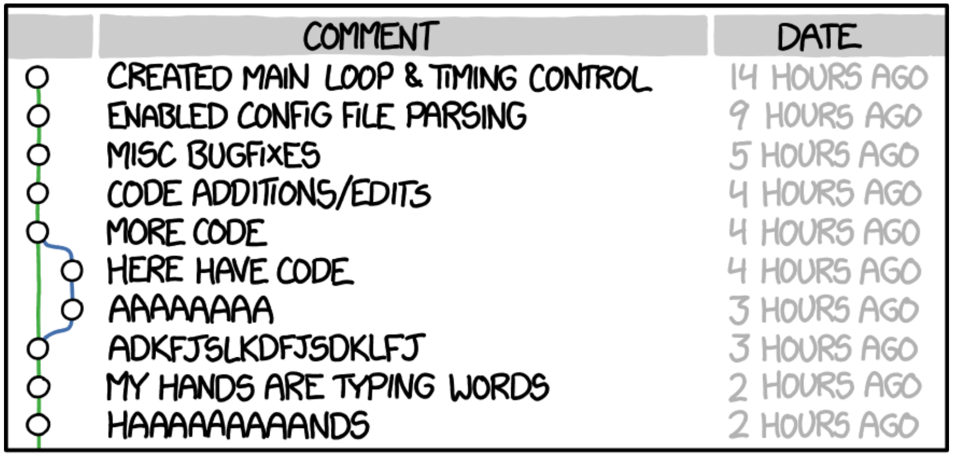
- Commits are “cheap”. Do them often.
- You provide a one-line message stating what you did.



```
git add .
git commit –m “Changed precip matrix to array”
git add .
git commit –m “Fixed precip var name”
```

Commit Messages

- Commit messages are critical for yourself and other people. Make them useful!
- Use them to roll back changes



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

https://xkcd.com/1296/

Initial Setup

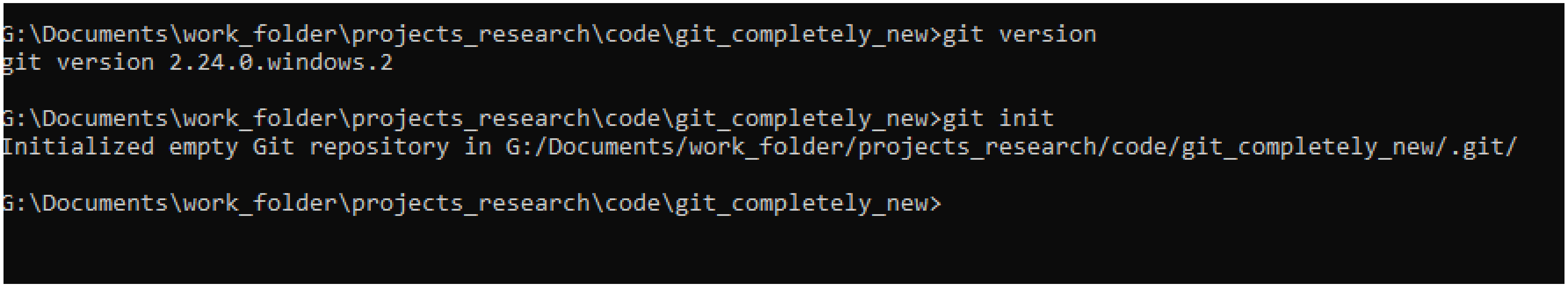
- Define yourself as a user. Only have to do this once.
- You can use different name/email for a specific project by leaving off the --global

```
git version

git config --global user.email "you@example.com"

git config --global user.name “Your git username"
```

Our first repository

- Create a new directory on your computer
- Open cmd line (Windows) or terminal (Linux) and move to folder
- Create a new repository for git to track

```
git init
```



```
G:\Documents\work_folder\projects_research\code\git_completely_new>git version
git version 2.24.0.windows.2

G:\Documents\work_folder\projects_research\code\git_completely_new>git init
Initialized empty Git repository in G:/Documents/work_folder/projects_research/code/git_completely_new/.git/

G:\Documents\work_folder\projects_research\code\git_completely_new>
```

Stage and make the initial commit

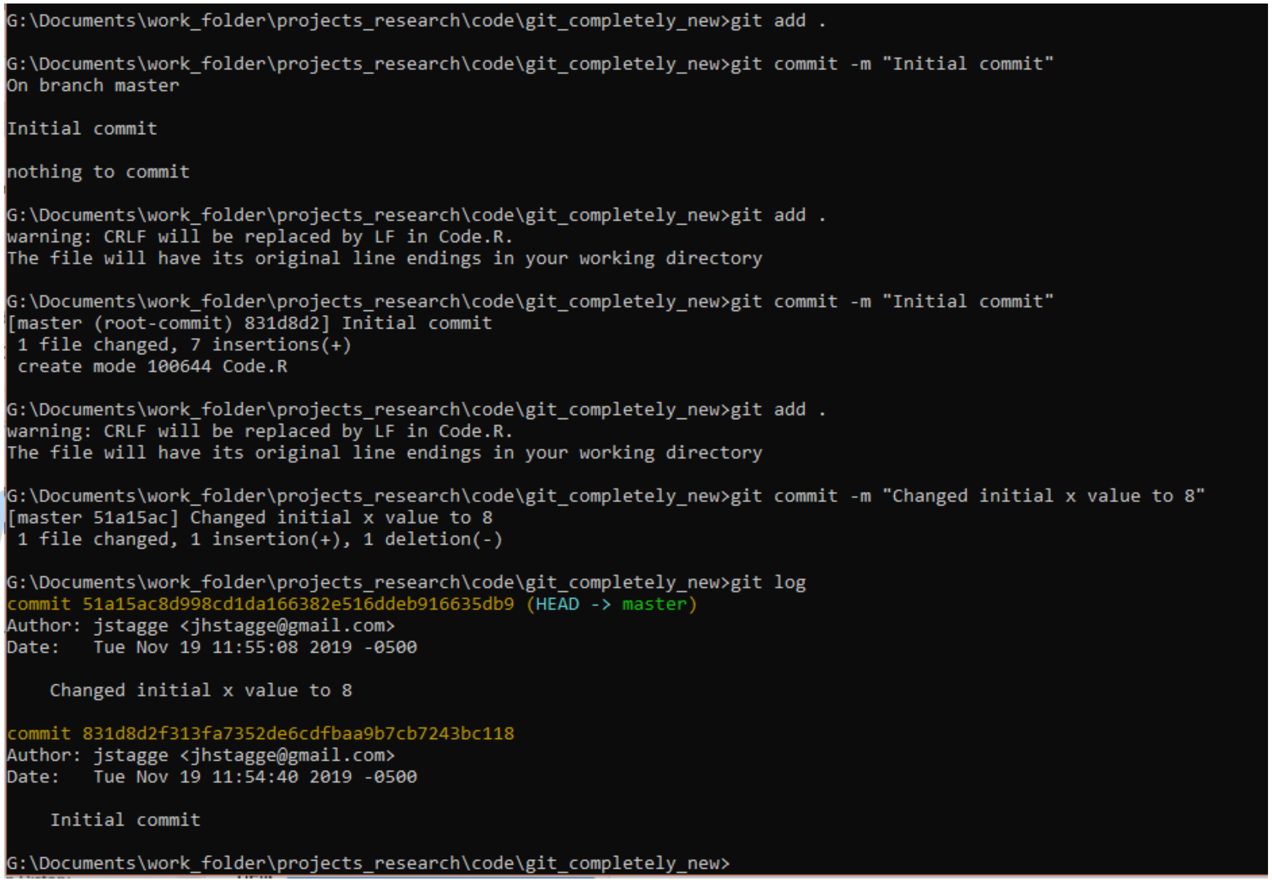
- Create a new file: **`code1.R`**
- Add some lines of code to this file
- Add and commit this change (with informative message!)

```
git add .

git commit –m “Initial commit”
```

- Make another change to your file, add, and commit.

Checking the progress

- git log
- git status



```
G:\Documents\work_folder\projects_research\code\git_completely_new>git add .

G:\Documents\work_folder\projects_research\code\git_completely_new>git commit -m "Initial commit"
On branch master

Initial commit

nothing to commit

G:\Documents\work_folder\projects_research\code\git_completely_new>git add .
warning: CRLF will be replaced by LF in Code.R.
The file will have its original line endings in your working directory

G:\Documents\work_folder\projects_research\code\git_completely_new>git commit -m "Initial commit"
[master (root-commit) 831d8d2] Initial commit
 1 file changed, 7 insertions(+)
 create mode 100644 Code.R

G:\Documents\work_folder\projects_research\code\git_completely_new>git add .
warning: CRLF will be replaced by LF in Code.R.
The file will have its original line endings in your working directory

G:\Documents\work_folder\projects_research\code\git_completely_new>git commit -m "Changed initial x value to 8"
[master 51a15ac] Changed initial x value to 8
 1 file changed, 1 insertion(+), 1 deletion(-)

G:\Documents\work_folder\projects_research\code\git_completely_new>git log
commit 51a15ac8d998cd1da166382e516ddeb916635db9 (HEAD -> master)
Author: jstagge <jhstagge@gmail.com>
Date:   Tue Nov 19 11:55:08 2019 -0500

    Changed initial x value to 8

commit 831d8d2f313fa7352de6cdfbaa9b7cb7243bc118
Author: jstagge <jhstagge@gmail.com>
Date:   Tue Nov 19 11:54:40 2019 -0500

    Initial commit

G:\Documents\work_folder\projects_research\code\git_completely_new>
```

Reverting back to previous version

- You can revert to a previous version using the reset command:

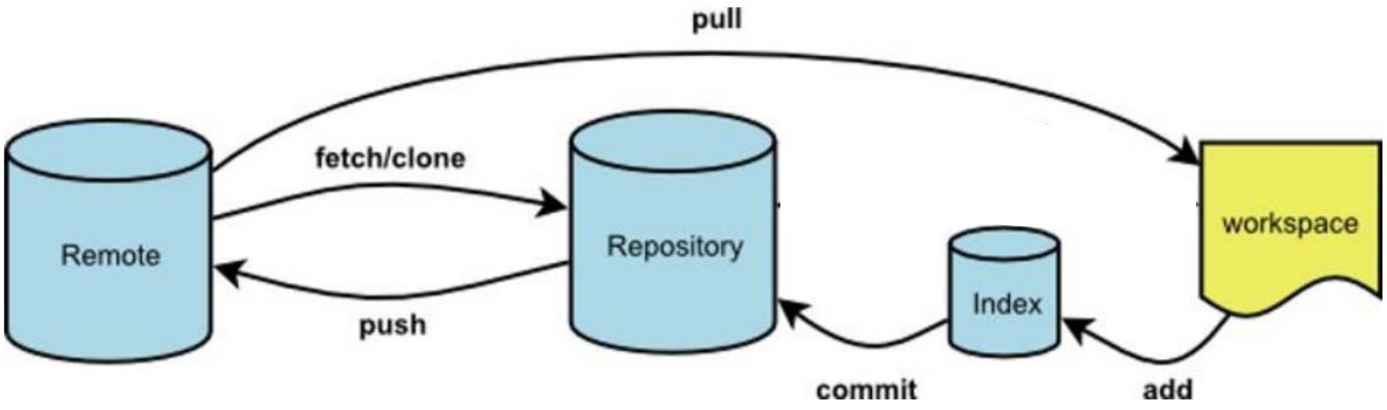
```
git reset --hard <hash>
```

- Careful – reset will delete the recent work completely
- There is another command <revert>, which is slightly softer, but more complex. Honestly, it’s beyond my ability.

Workflow: incorporating Github

Github website

Your personal computer



New Terminology

- **Cloning:** copying a repository from a remote server

- **Pull**: download new commits from remote repository (Github)
- **Push**: adding local changes to the remote repository (Github)

- **Branch:** all commits live on a branch. Could be 1 or 1,000 Main branch is called the master branch.

Typical personal workflow with Github

- Do some work

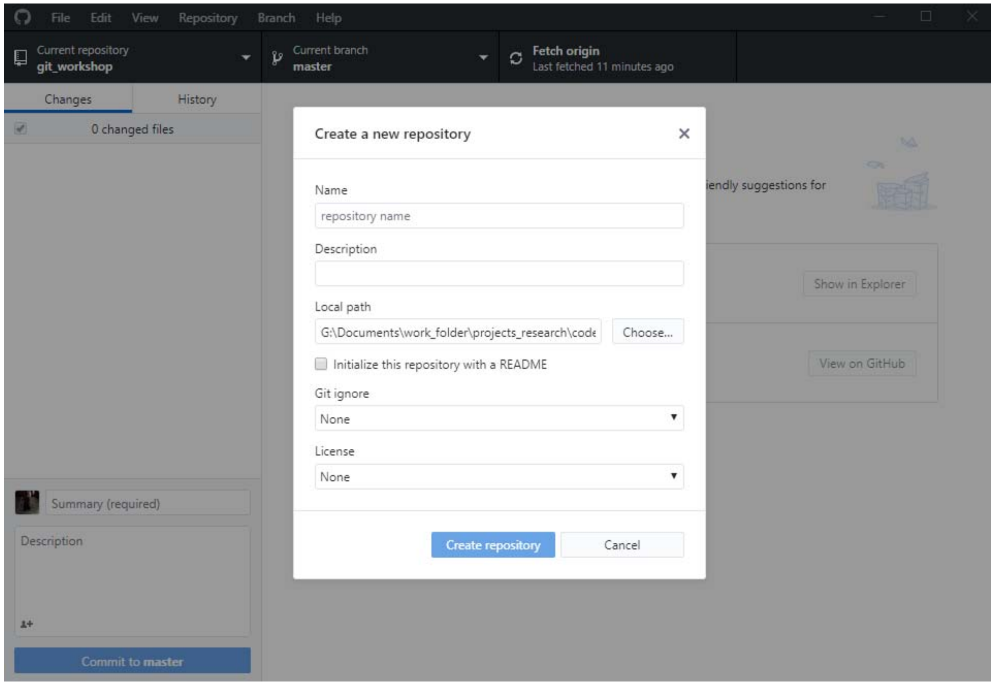
```
    git add .

    git commit –m “Message goes here”
```

- Repeat regularly and ad infinitum. Like doing a save.
- When you’ve made significant progress, e.g. Version 1.2
  Push the code to the Github website

```
      git push
```

Let’s create a repository with Github Desktop

- File -> Create a new repository
- Local path
- README
- Ignore
- License
- Publish to Github



Current repository: git_workshop

Current branch: master

Create a new repository

Name: repository name

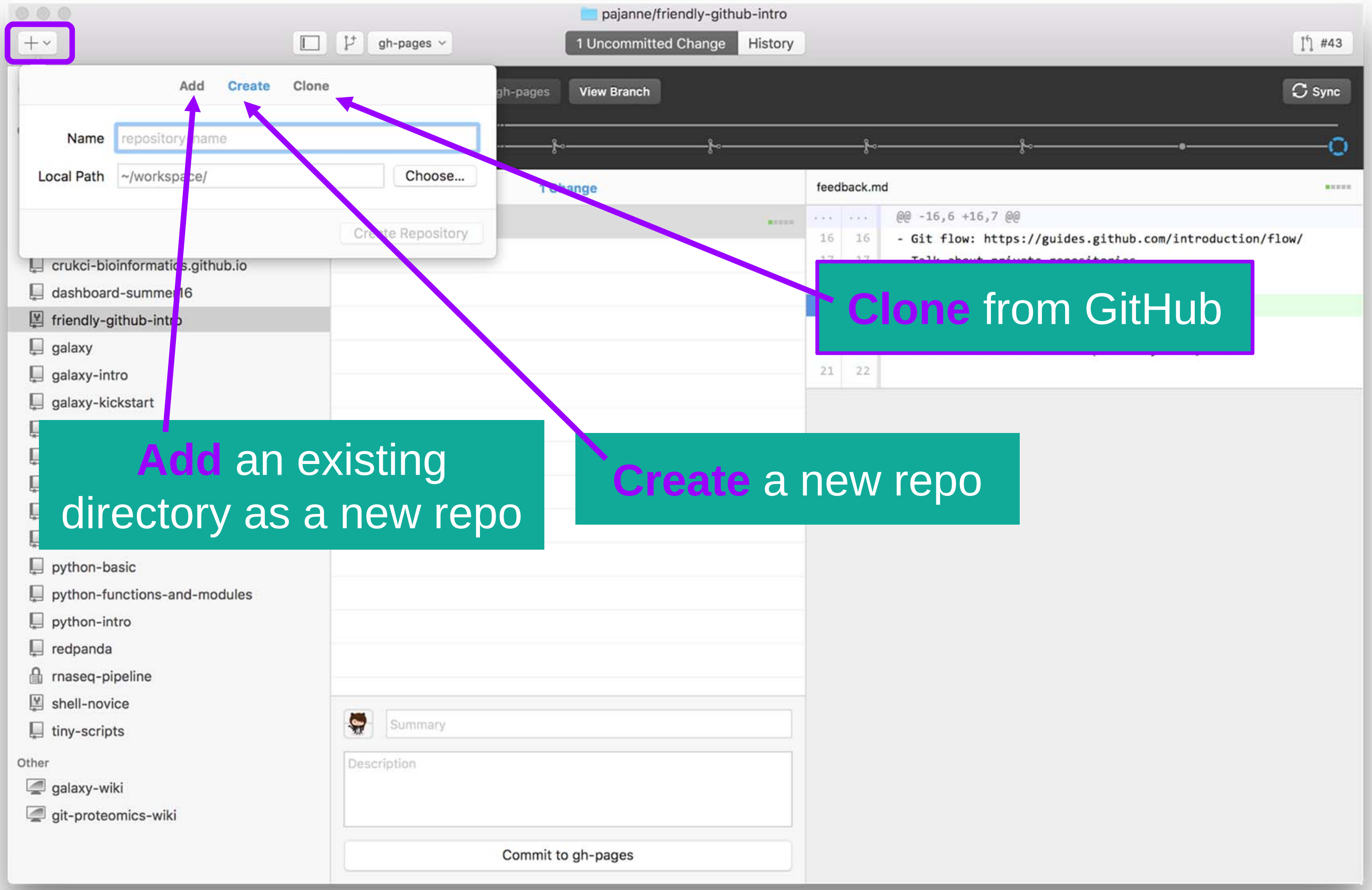
Description:

Local path: G:\Documents\work_folder\projects_research\code

□ Initialize this repository with a README

Git ignore: None

License: None

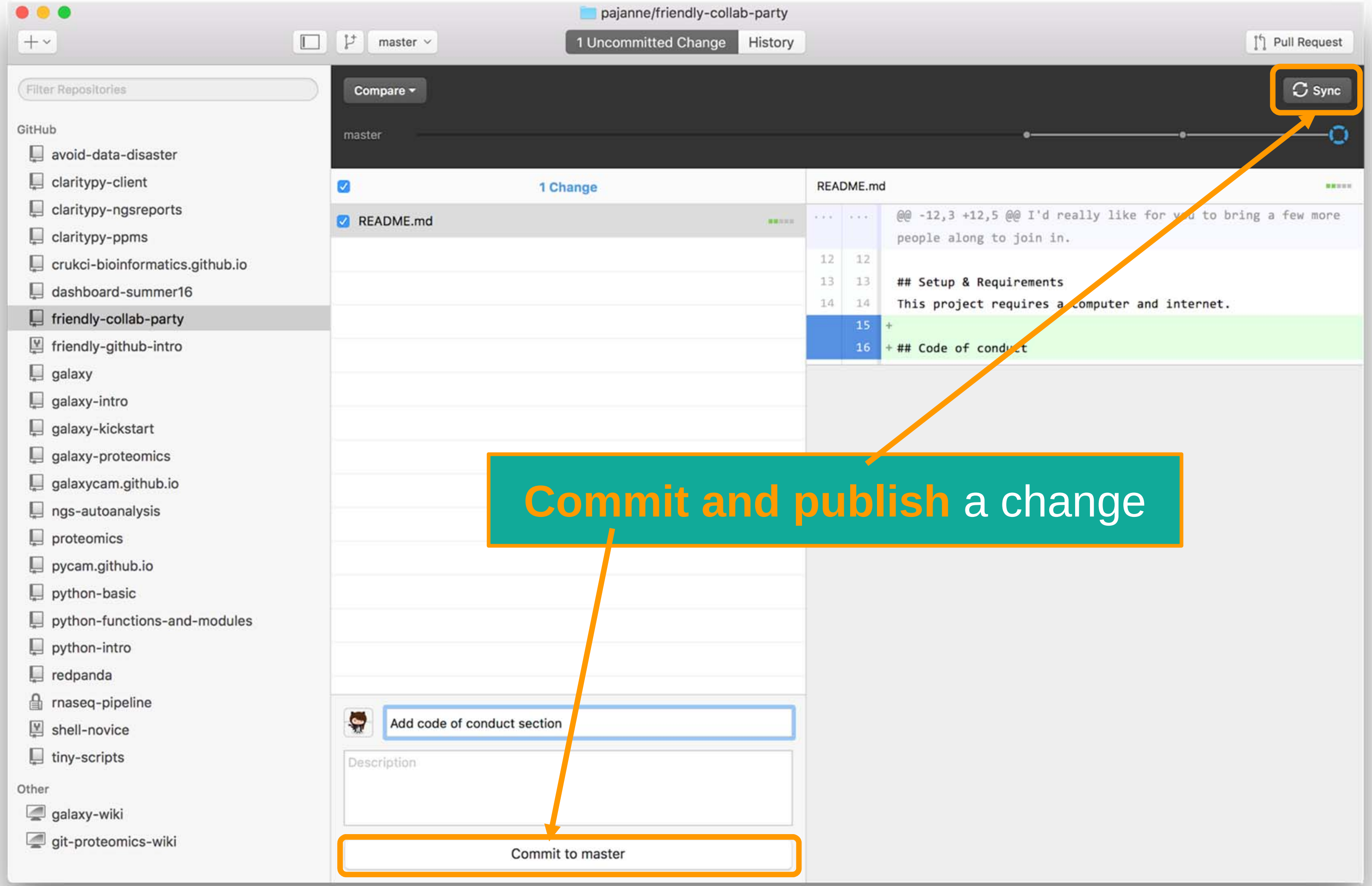


```
@@ -16,6 +16,7 @@
16  16  - Git flow: https://guides.github.com/introduction/flow/
```

Add an existing directory as a new repo

Create a new repo

Clone from GitHub



Commit and publish a change

GitHub

- avoid-data-disaster
- claritypy-client
- claritypy-ngsreports
- claritypy-ppms
- crukci-bioinformatics.github.io
- dashboard-summer16
- friendly-collab-party
- friendly-github-intro
- galaxy
- galaxy-intro
- galaxy-kickstart
- galaxy-proteomics
- galaxycam.github.io
- ngs-autoanalysis
- proteomics
- pycam.github.io
- python-basic
- python-functions-and-modules
- python-intro
- redpanda
- rnaseq-pipeline
- shell-novice
- tiny-scripts

Other

- galaxy-wiki
- git-proteomics-wiki

☑ README.md

README.md

```
@@ -12,3 +12,5 @@ I'd really like for you to bring a few more
people along to join in.

## Setup & Requirements
This project requires a computer and internet.
+
+## Code of conduct
```

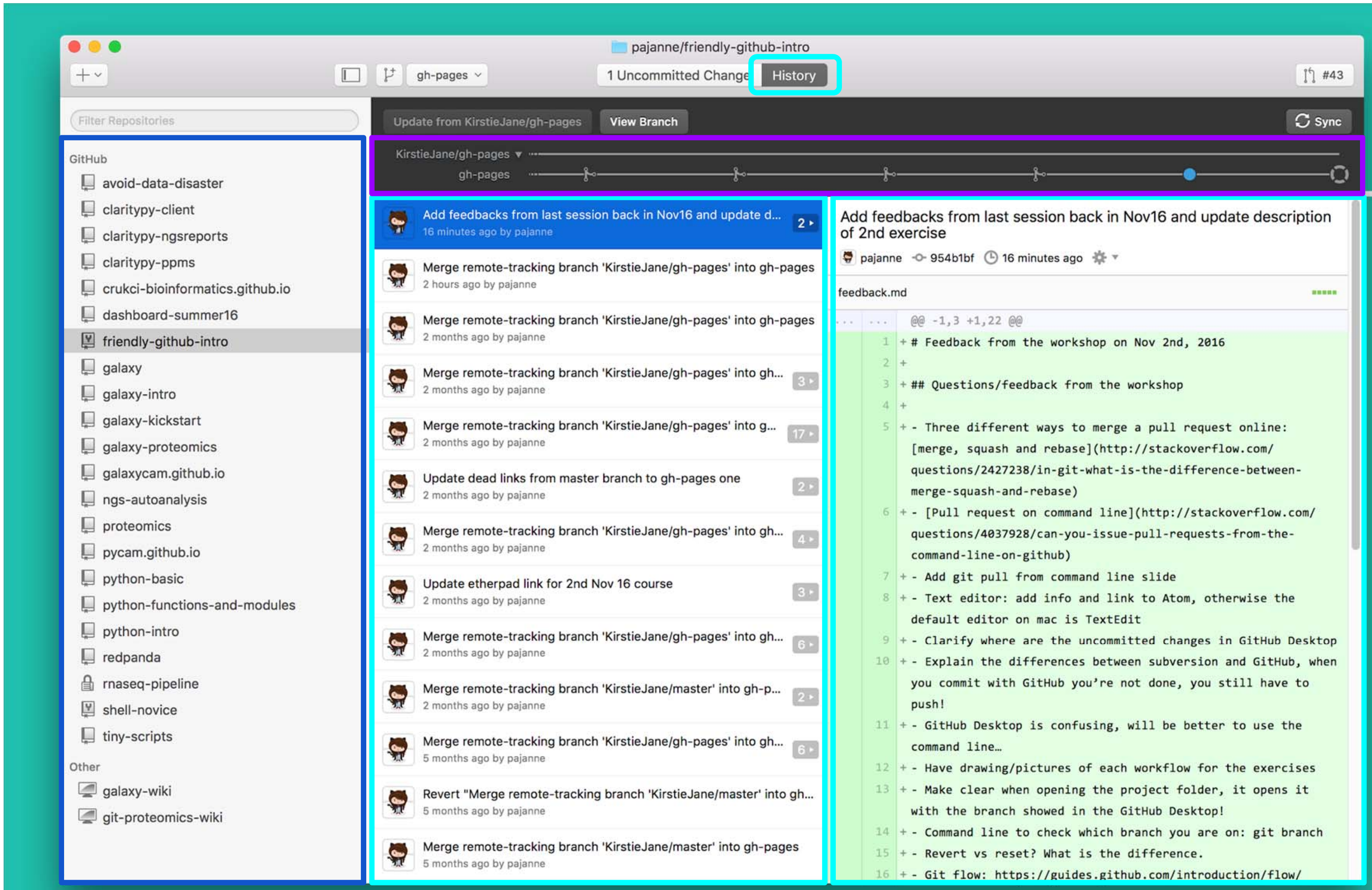
Add code of conduct section

Description

Commit to master

GitHub Desktop

History of commits



GitHub
- avoid-data-disaster
- claritypy-client
- claritypy-ngsreports
- claritypy-ppms
- crukci-bioinformatics.github.io
- dashboard-summer16
- friendly-github-intro
- galaxy
- galaxy-intro
- galaxy-kickstart
- galaxy-proteomics
- galaxycam.github.io
- ngs-autoanalysis
- proteomics
- pycam.github.io
- python-basic
- python-functions-and-modules
- python-intro
- redpanda
- rnaseq-pipeline
- shell-novice
- tiny-scripts

Other
- galaxy-wiki
- git-proteomics-wiki

Add feedbacks from last session back in Nov16 and update d... 2
16 minutes ago by pajanne

Merge remote-tracking branch 'KirstieJane/gh-pages' into gh-pages
2 hours ago by pajanne

Merge remote-tracking branch 'KirstieJane/gh-pages' into gh-pages
2 months ago by pajanne

Merge remote-tracking branch 'KirstieJane/gh-pages' into gh... 3
2 months ago by pajanne

Merge remote-tracking branch 'KirstieJane/gh-pages' into g... 17
2 months ago by pajanne

Update dead links from master branch to gh-pages one 2
2 months ago by pajanne

Merge remote-tracking branch 'KirstieJane/gh-pages' into gh... 4
2 months ago by pajanne

Update etherpad link for 2nd Nov 16 course 3
2 months ago by pajanne

Merge remote-tracking branch 'KirstieJane/gh-pages' into gh... 6
2 months ago by pajanne

Merge remote-tracking branch 'KirstieJane/master' into gh-p... 2
2 months ago by pajanne

Merge remote-tracking branch 'KirstieJane/gh-pages' into gh... 6
5 months ago by pajanne

Revert "Merge remote-tracking branch 'KirstieJane/master' into gh...
5 months ago by pajanne

Merge remote-tracking branch 'KirstieJane/master' into gh-pages
5 months ago by pajanne

Add feedbacks from last session back in Nov16 and update description of 2nd exercise

pajanne 954b1bf 16 minutes ago

feedback.md

```
@@ -1,3 +1,22 @@
+# Feedback from the workshop on Nov 2nd, 2016
+
+## Questions/feedback from the workshop
+
+- Three different ways to merge a pull request online:
  [merge, squash and rebase](http://stackoverflow.com/
  questions/2427238/in-git-what-is-the-difference-between-
  merge-squash-and-rebase)
+- [Pull request on command line](http://stackoverflow.com/
  questions/4037928/can-you-issue-pull-requests-from-the-
  command-line-on-github)
+- Add git pull from command line slide
+- Text editor: add info and link to Atom, otherwise the
  default editor on mac is TextEdit
+- Clarify where are the uncommitted changes in GitHub Desktop
+- Explain the differences between subversion and GitHub, when
  you commit with GitHub you're not done, you still have to
  push!
+- GitHub Desktop is confusing, will be better to use the
  command line…
+- Have drawing/pictures of each workflow for the exercises
+- Make clear when opening the project folder, it opens it
  with the branch showed in the GitHub Desktop!
+- Command line to check which branch you are on: git branch
+- Revert vs reset? What is the difference.
+- Git flow: https://guides.github.com/introduction/flow/
```

history

timeline

repositories

history of commits

changes per commit

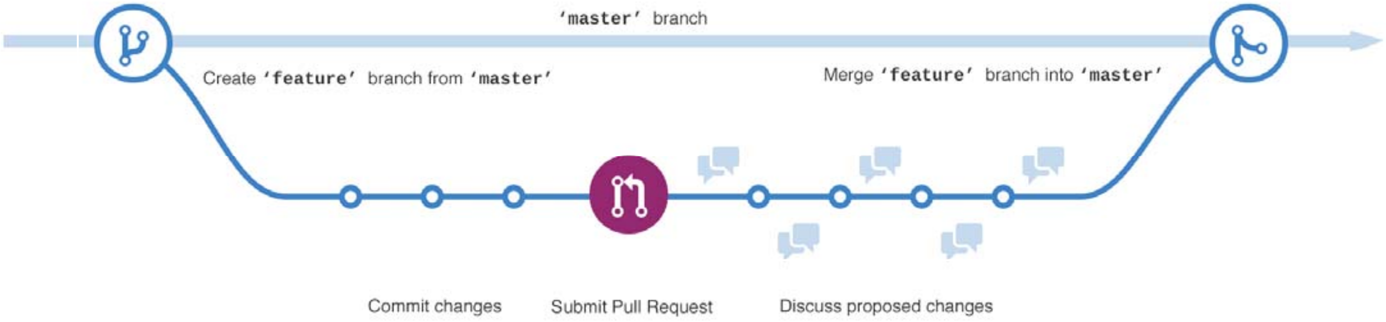
Using the command line

```
> git command arguments

> git clone repo

> git status
> git add changed_file
> git commit -m “message”
> git push
```

```
> git checkout -b cmd-line-branch
> git branch
> git push origin cmd-line-branch

> git checkout master
> git branch
> git merge cmd-line-branch
> git push

> git log
> git revert commit_id
> git push
```

Branches

- A branch is a parallel version of the main line of development (master)
- Used to:
  - Develop features
  - Fix bugs
  - Safely experiment with new ideas


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

Git/GitHub Terms (2)

Branch

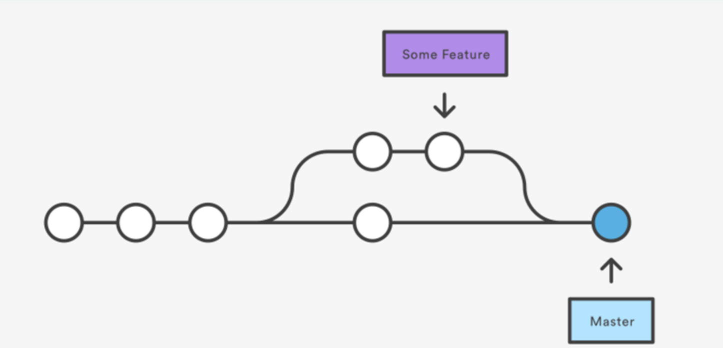
Copy of your project

Pull request

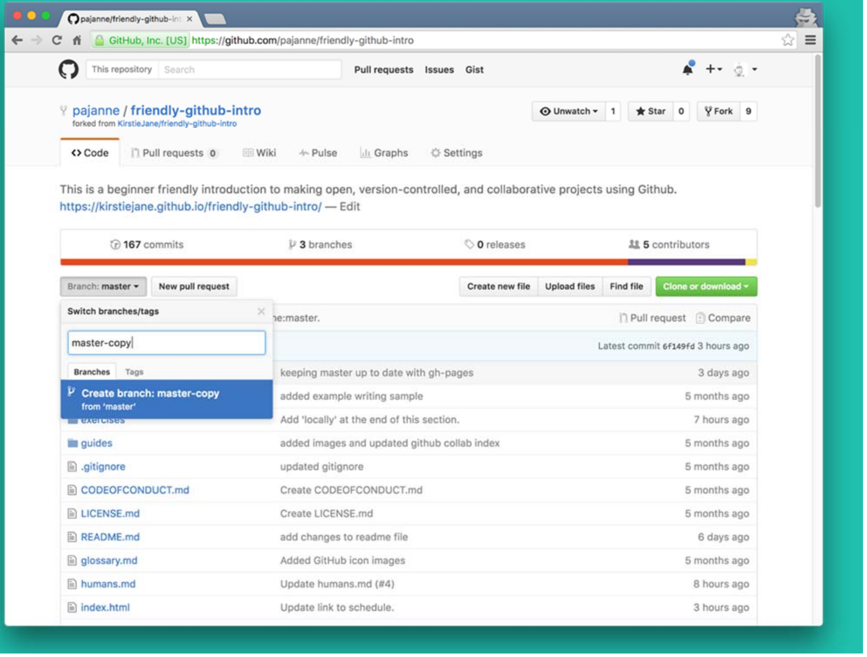
request to add your changes from a branch back into master

Merge

act of incorporating new changes (commits) from one branch to another



https://www.atlassian.com/git/tutorials/using-branches/git-merge

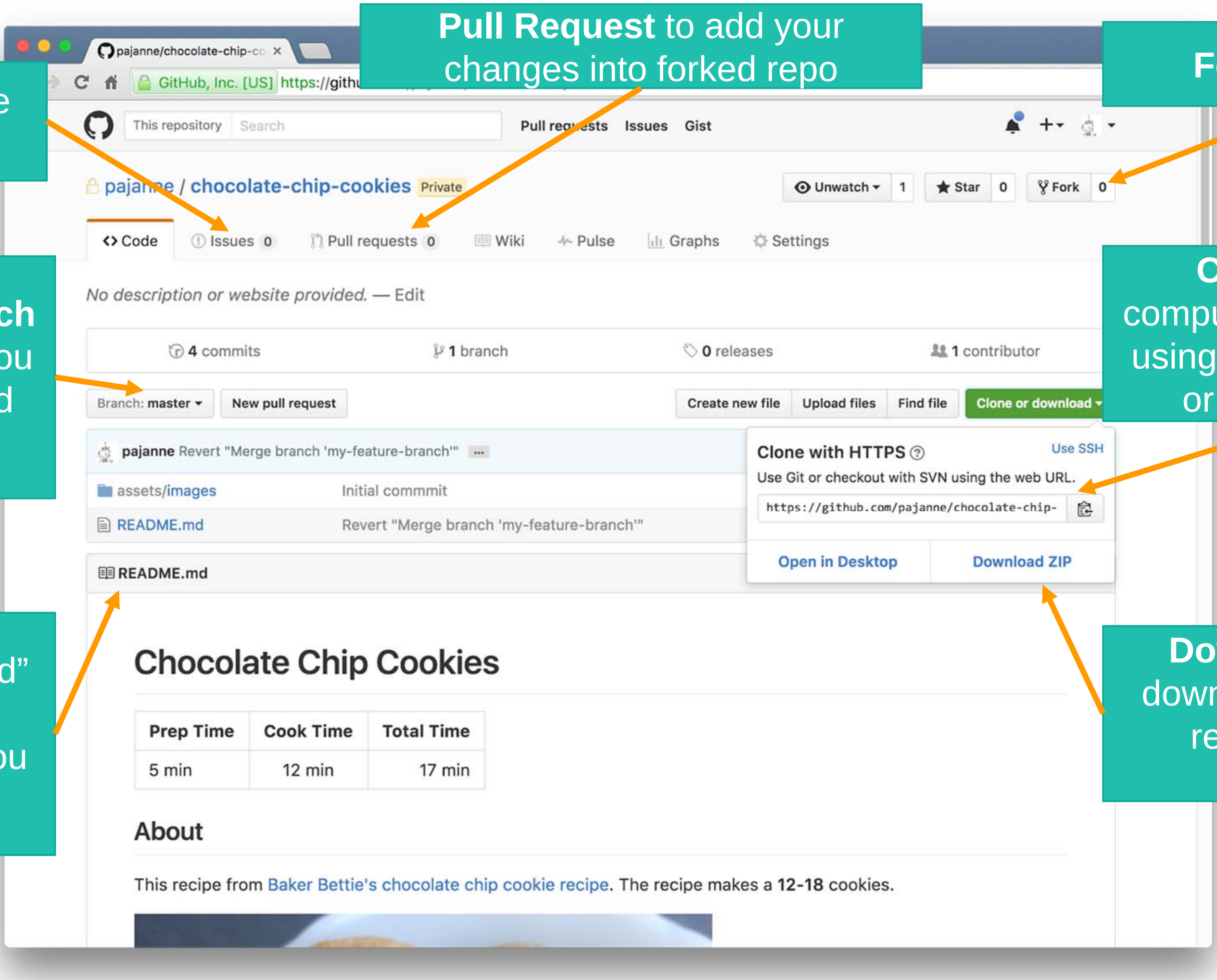


This is a beginner friendly introduction to making open, version-controlled, and collaborative projects using Github. https://kirstiejane.github.io/friendly-github-intro/ — Edit

ne:master.

Latest commit 6f149fd 3 hours ago

	keeping master up to date with gh-pages	3 days ago
	added example writing sample	5 months ago
exercises	Add 'locally' at the end of this section.	7 hours ago
guides	added images and updated github collab index	5 months ago
.gitignore	updated gitignore	5 months ago
CODEOFCONDUCT.md	Create CODEOFCONDUCT.md	5 months ago
LICENSE.md	Create LICENSE.md	5 months ago
README.md	add changes to readme file	6 days ago
glossary.md	Added GitHub icon images	5 months ago
humans.md	Update humans.md (#4)	8 hours ago
index.html	Update link to schedule.	3 hours ago



No description or website provided. — Edit

pajanne Revert "Merge branch 'my-feature-branch'"

assets/images	Initial commmit
README.md	Revert "Merge branch 'my-feature-branch'"

Use Git or checkout with SVN using the web URL.

```
https://github.com/pajanne/chocolate-chip-
```

Chocolate Chip Cookies

Prep Time	Cook Time	Total Time
5 min	12 min	17 min

About

This recipe from Baker Bettie's chocolate chip cookie recipe. The recipe makes a **12-18** cookies.

Create & manage “**Issues**”

Pull Request to add your changes into forked repo

Fork this repo

Your default **branch** is “master” - but you can switch or add branches here

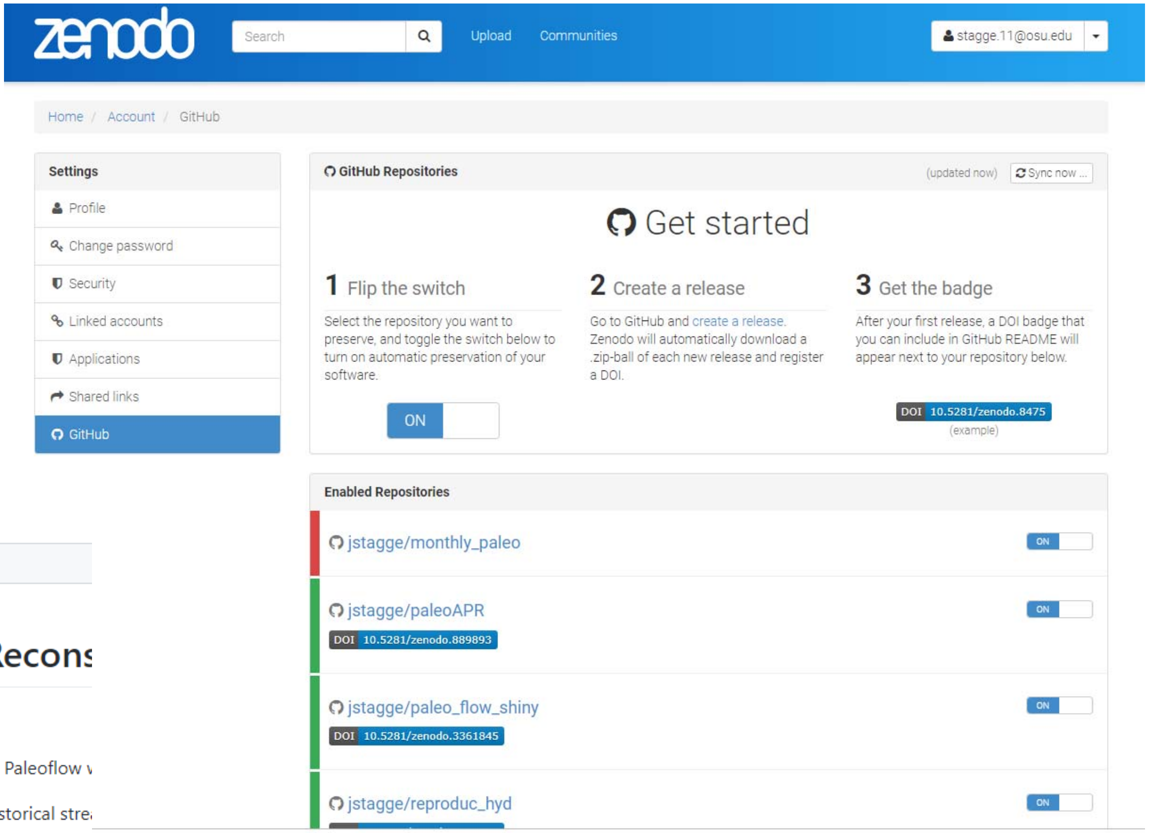
Clone to your computer from the web using GitHub Desktop or git command

The “README.md” is automatically rendered when you visit a repo

Download Zip to download the zipped repo onto your computer

Making your code citable

- Zenodo provides citeable DOI
- Repository or specific version
- Icon for Github



zenodo Search Upload Communities stagge.11@osu.edu

Home / Account / GitHub

Settings
- Profile
- Change password
- Security
- Linked accounts
- Applications
- Shared links
- GitHub

GitHub Repositories (updated now) Sync now ...

Get started

1 Flip the switch

Select the repository you want to preserve, and toggle the switch below to turn on automatic preservation of your software.

ON

2 Create a release

Go to GitHub and create a release. Zenodo will automatically download a .zip-ball of each new release and register a DOI.

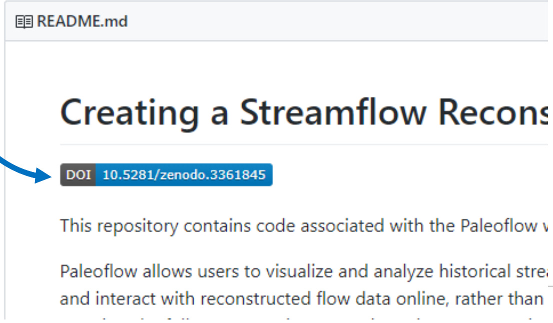
3 Get the badge

After your first release, a DOI badge that you can include in GitHub README will appear next to your repository below.

DOI 10.5281/zenodo.8475
(example)

Enabled Repositories

Repository	DOI	Switch
jstagge/monthly_paleo		ON
jstagge/paleoAPR	DOI 10.5281/zenodo.889893	ON
jstagge/paleo_flow_shiny	DOI 10.5281/zenodo.3361845	ON
jstagge/reproduc_hyd		ON



README.md

Creating a Streamflow Recons

DOI 10.5281/zenodo.3361845

This repository contains code associated with the Paleoflow v

Paleoflow allows users to visualize and analyze historical stre
and interact with reconstructed flow data online, rather than

Warning

- Dropbox

command	description
`git clone` ***url [dir]***	copy a Git repository so you can add to it
`git add` ***file***	adds file contents to the staging area
`git commit`	records a snapshot of the staging area
`git status`	view the status of your files in the working directory and staging area
`git diff`	shows diff of what is staged and what is modified but unstaged
`git help` ***[command]***	get help info about a particular command
`git pull`	fetch from a remote repo and try to merge into the current branch
`git push`	push your new branches and data to a remote repository
others: `init`, `reset`, `branch`, `checkout`, `merge`, `log`, `tag`	

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.