Intro to Git

October 3, 2018

While you wait...

- Take a Git cheatsheet

Matt Jansen

Data Analyst
Digital Research Services

Christina Cortland

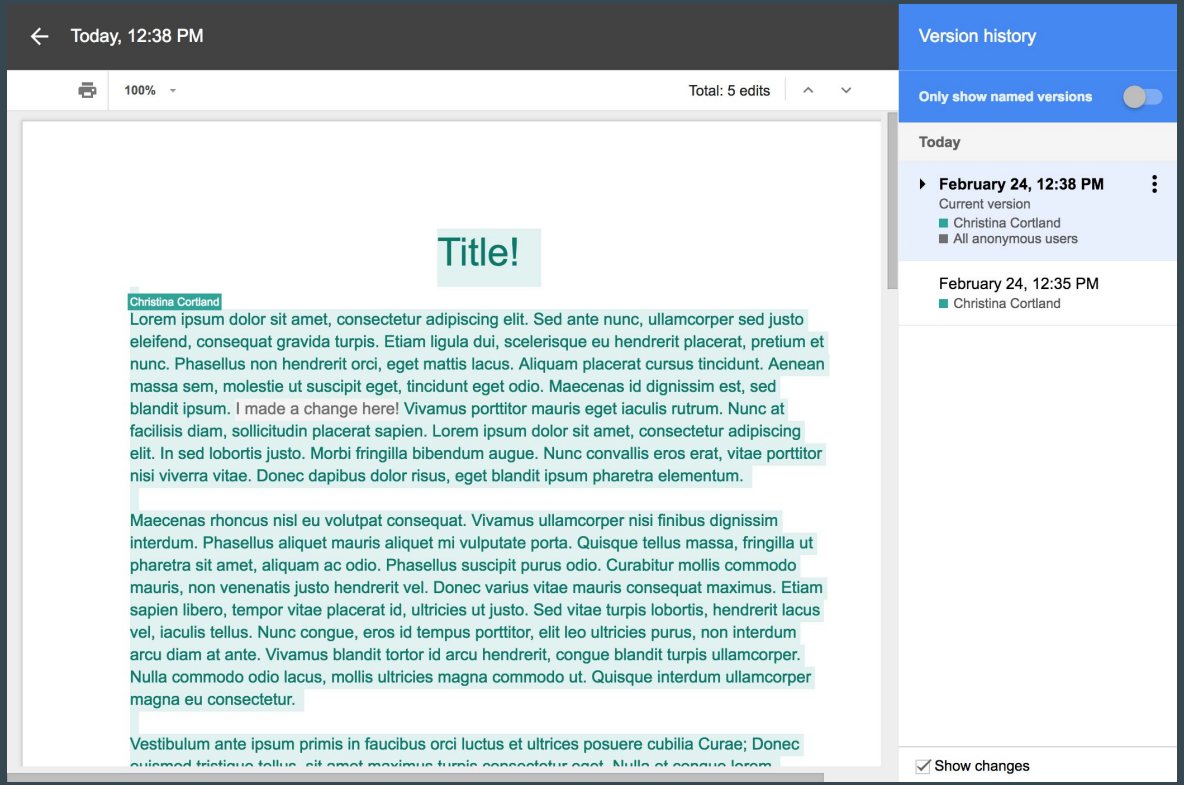
Carolina Academic Library Associate
Library & Information Technology

Overview

- ➔ What is Git?
- ➔ Setup!
- ➔ Using Git!
- ➔ Using GitHub!
- ➔ Practice Git with GitHub Pages!
- ➔ Wrap-Up

Git is a version control system.

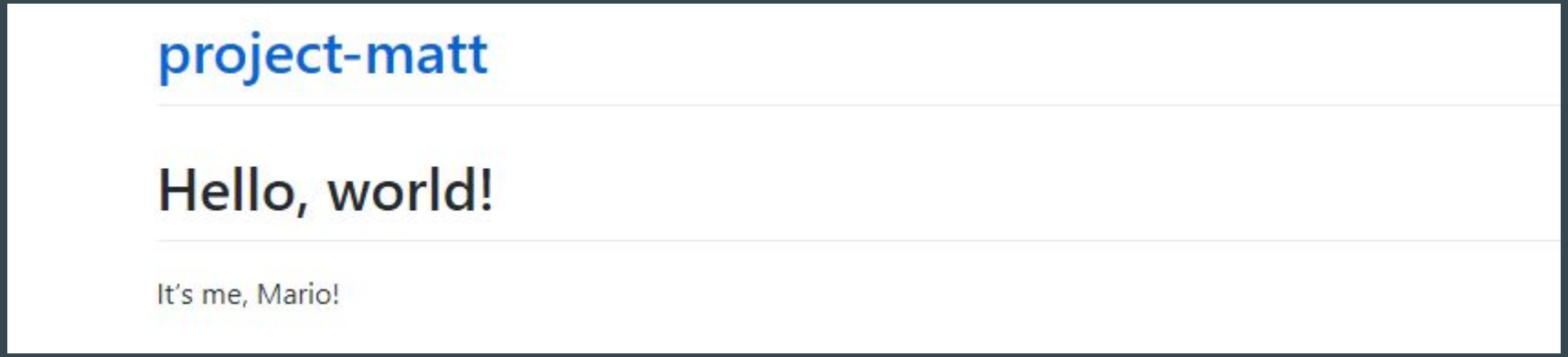
Git is a version control system.



Why use version control?

https://xkcd.com/1459/

What you're going to make



Git Setup!

Setup Overview

Install Git

Configure git username and email

Setup: Install Git

Download Git: https://git-scm.com/download

Leave default settings EXCEPT:

Windows
Default editor: “Use the Nano editor by default” Terminal emulator: “Use Windows’ default console window”

Note: These settings can be changed later if needed. (Official installation documentation)

Setup Overview

~~Install Git~~

Configure git username and email

Setup: Configure git username and email

```
$ git config --global user.name "Mona Lisa"

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

$ git config user.name
Mona Lisa

$ git config user.email
email@example.com
```

Using Git!

Glossary

git repository = place where history of changes to your project files is stored

Glossary

git repository = place where history of changes to your project files is stored

local environment = the computer you're using for this workshop

remote environment = other computers, or servers in the cloud

Glossary

git repository = place where history of changes to your project files is stored

local environment = the computer you're using for this workshop

remote environment = other computers, or servers in the cloud

commit = n. named snapshot of changes to a repository; v. to create a commit

Glossary

git repository = place where history of changes to your project files is stored

local environment = the computer you're using for this workshop

remote environment = other computers, or servers in the cloud

commit = n. named snapshot of changes to a repository; v. to create a commit

branch = named version of the repository

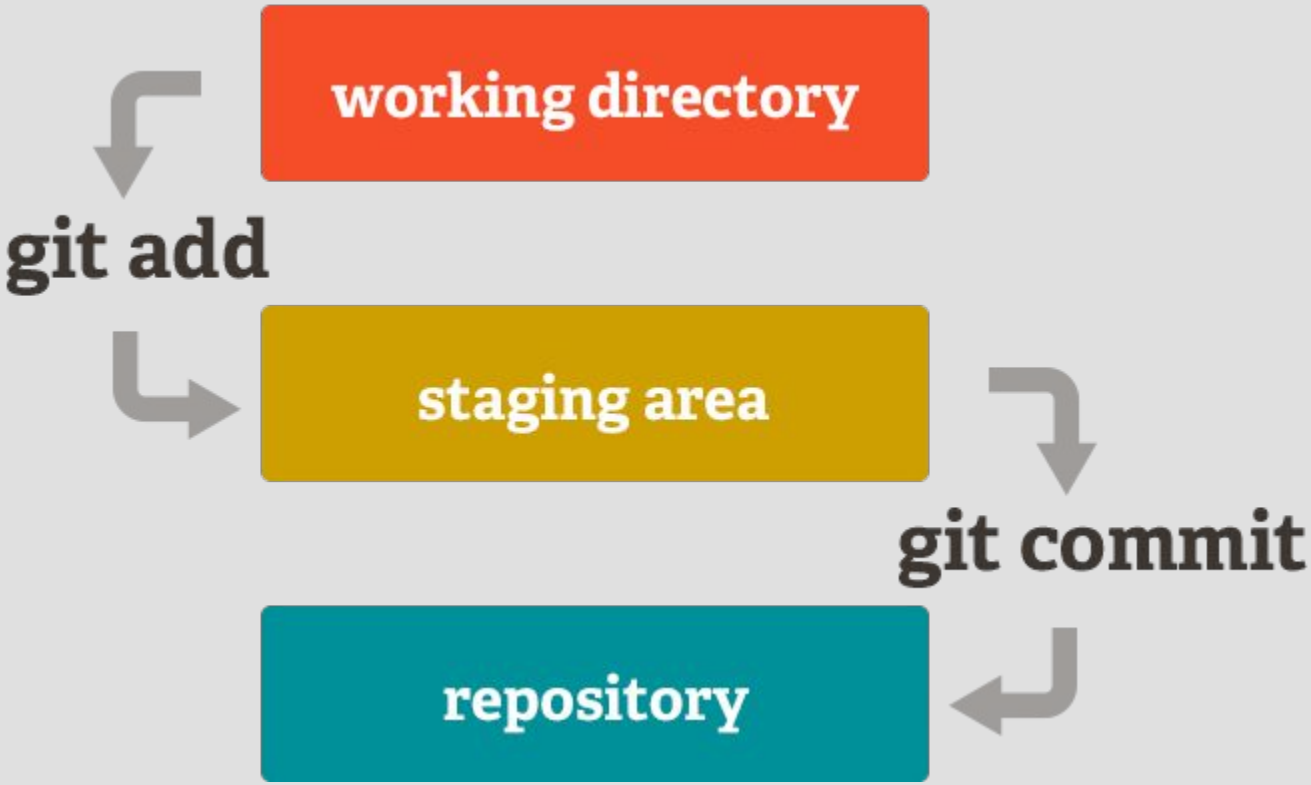
git init Turns Any Folder Into a Git Repository

```
$ pwd

$ ls

$ cd Documents

$ mkdir project-<yourname>

$ ls

$ cd project-<yourname>

$ git init
Initialized empty Git repository in
C:/Users/mtjansen/Documents/project-matt/.git/
```

git status Tells You What's Changed

```
$ git status
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
```

Git Workflow



https://git-scm.com/about/staging-area

Let’s Change Something

```
$ touch index.md
$ git status
On branch master
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)

      index.md

nothing added to commit but untracked files present (use "git add" to track)
```

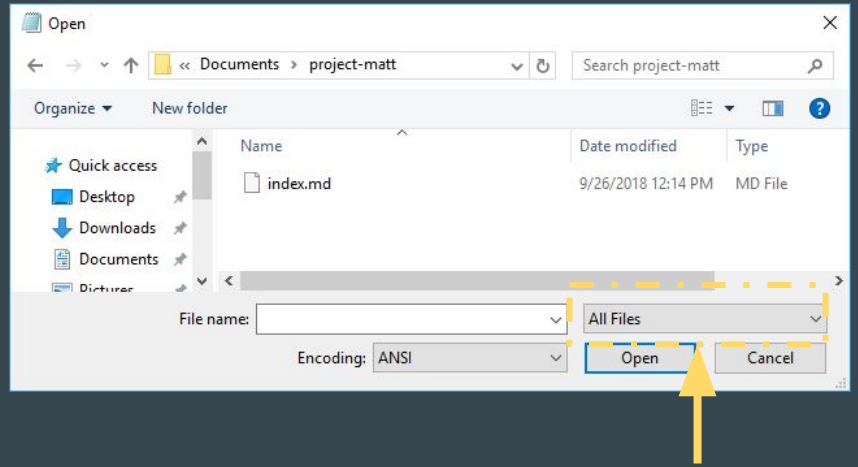
git add Adds Changes to Your Staging Area

```
$ git add index.md

$ git status
On branch master
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   index.md
```

Let's Change Another Thing

Add to index.md:

```
# Hello, world!
```



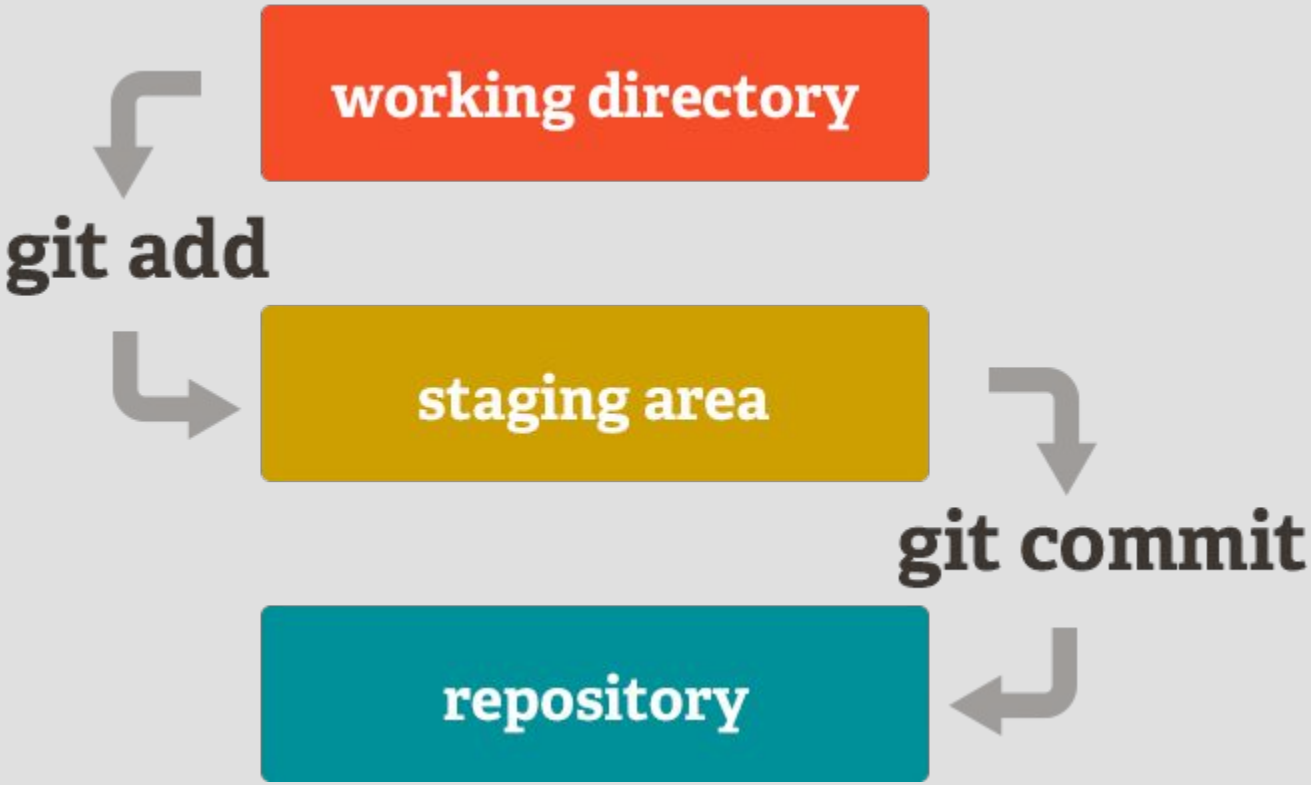
git add Adds Changes to Your Staging Area

```
$ git status
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached <file> <file>..." to unstage)
    new file:   index.md

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:   index.md
```

git add Adds Changes to Your Staging Area

```
$ git add index.md

$ git status
On branch master
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   index.md
```

Git Workflow



https://git-scm.com/about/staging-area

git commit Saves the Changes with Commentary

```
$ git commit -m 'Add index.md'
Counting objects: 3, done.
[master d6443d2] I added newfile.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 newfile.txt

$ git status
On branch master
nothing to commit, working tree clean
```

git log Shows Your Local History of Commits

```
$ git log
commit a93520b0a05575204c4985376ec011a861dea372 (HEAD -> master,
origin/master)
Author:  Matt Jansen <mtjansen@email.unc.edu>
Date:   Wed Sep 26 12:18:07 2018 -0400

    Add index.md
```

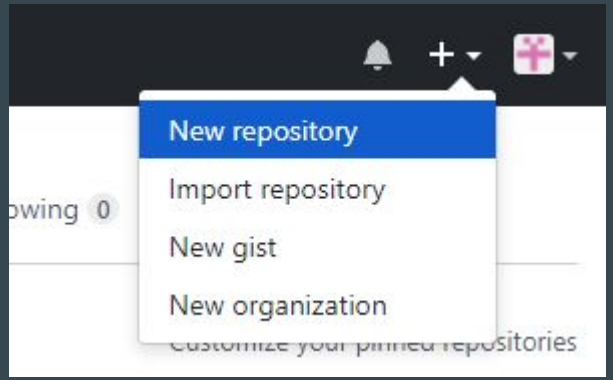
GitHub

Create a GitHub account

or

Sign in to your GitHub account

Create a new repository on GitHub



Create a new repository

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

Owner: mtjansen / Repository name: project-matt

Great repository names are short and memorable. Need inspiration? How about **literate-octo-parakeet**.

Description (optional)

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

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

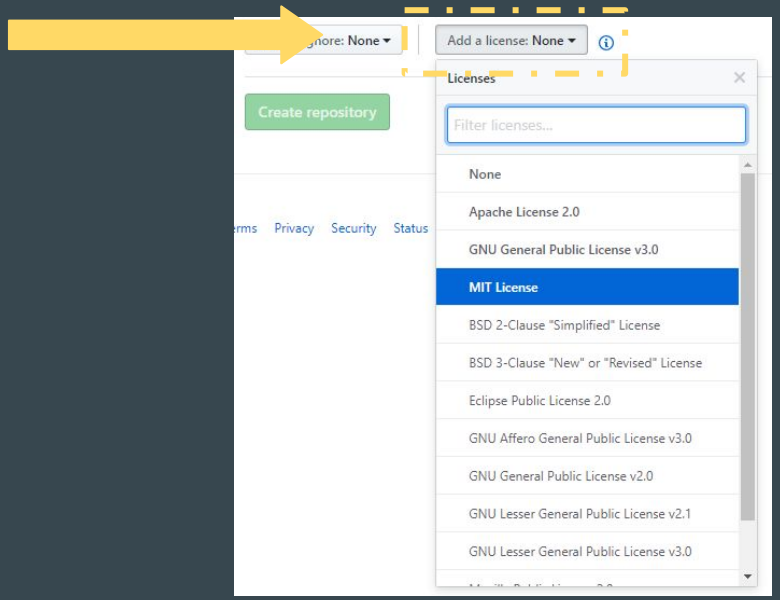
Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

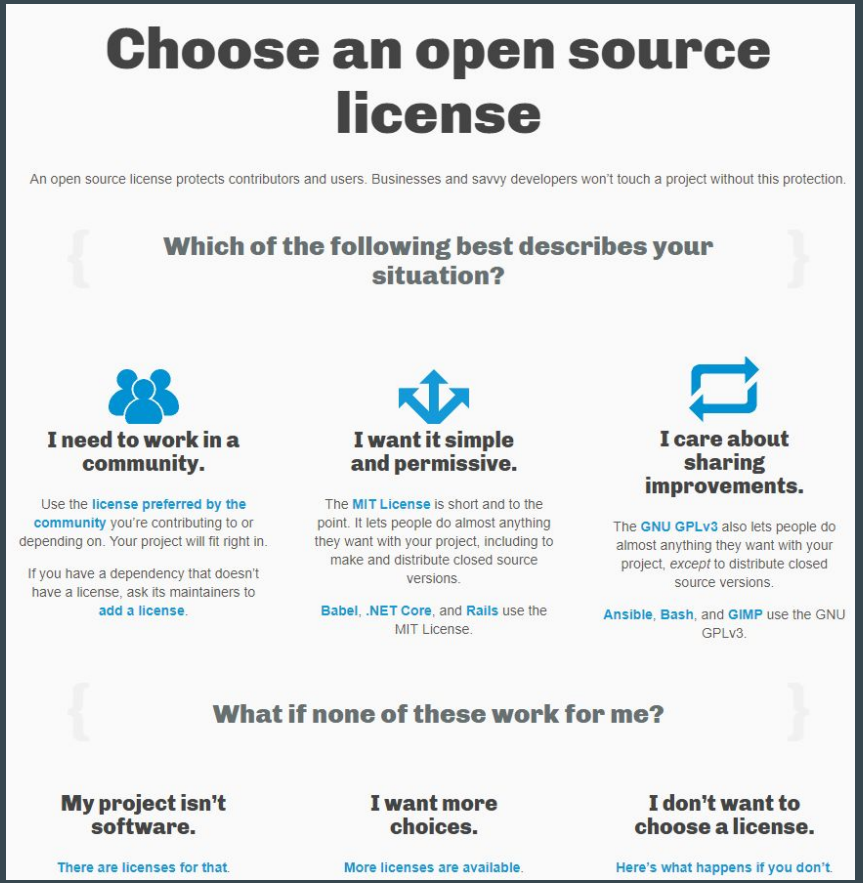
Add .gitignore: None | Add a license: None

Create repository

Choosing a License

Click ⓘ to read more about licenses.





git remote add origin Links our Local and Remote Repos

...or push an existing repository from the command line

```
git remote add origin https://github.com/mtjansen/project-matt.git
git push -u origin master
```

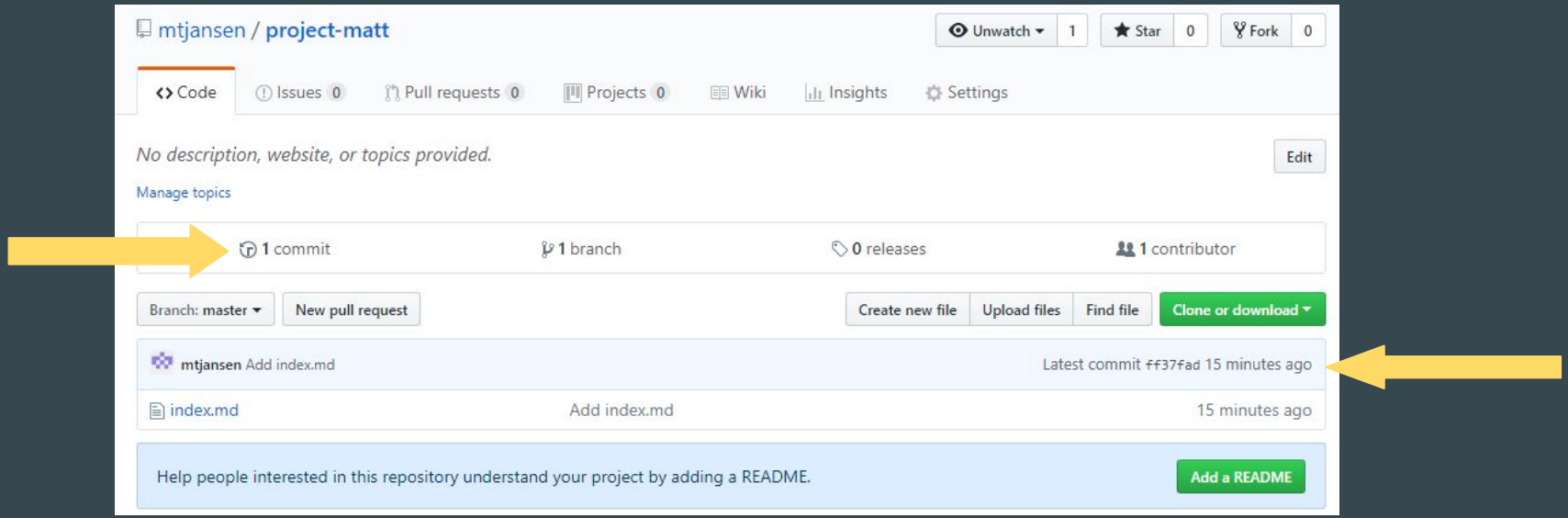
```
$ git remote add origin https://github.com/mtjansen/project-matt.git
```

(Shift-Ins pastes in Git Bash)

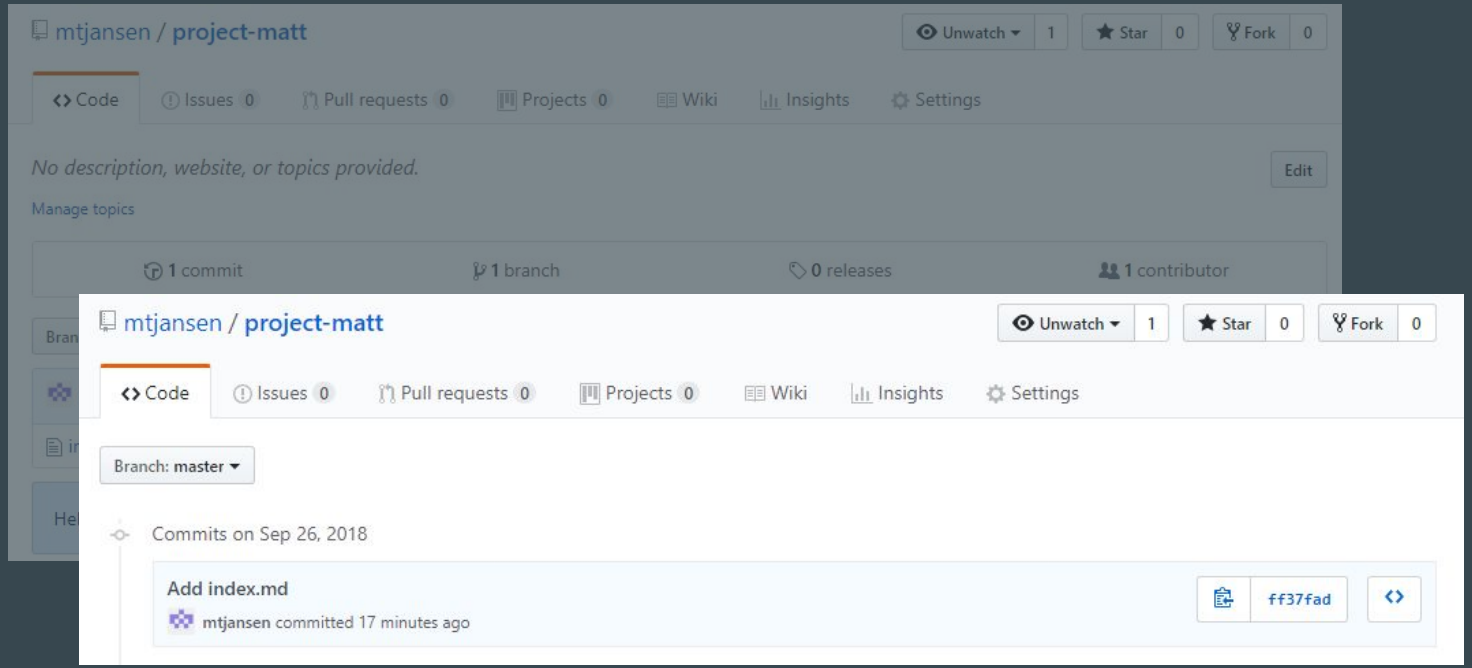
git push Sends Your Commits Back to GitHub

```
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 230 bytes | 230.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/mtjansen/project-matt.git
 * [new branch]      master -> master
```

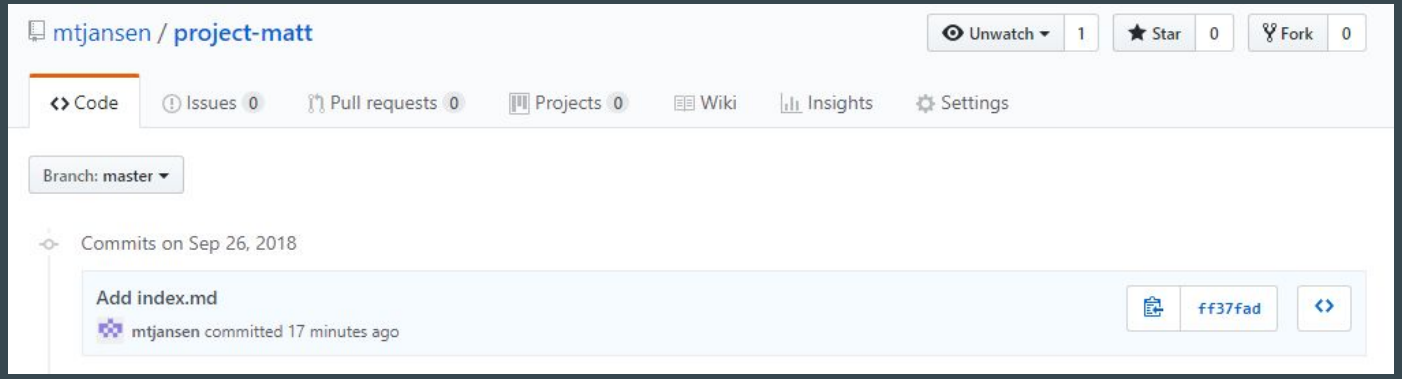
git push Sends Your Commits Back to GitHub



git push Sends Your Commits Back to GitHub

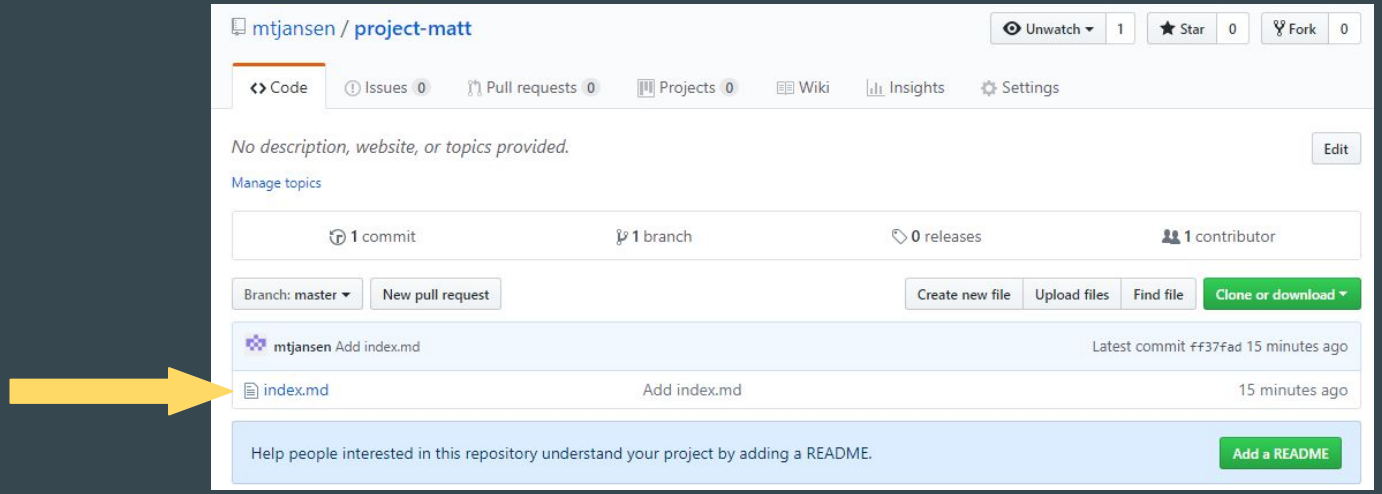


git log Shows Your Local History of Commits

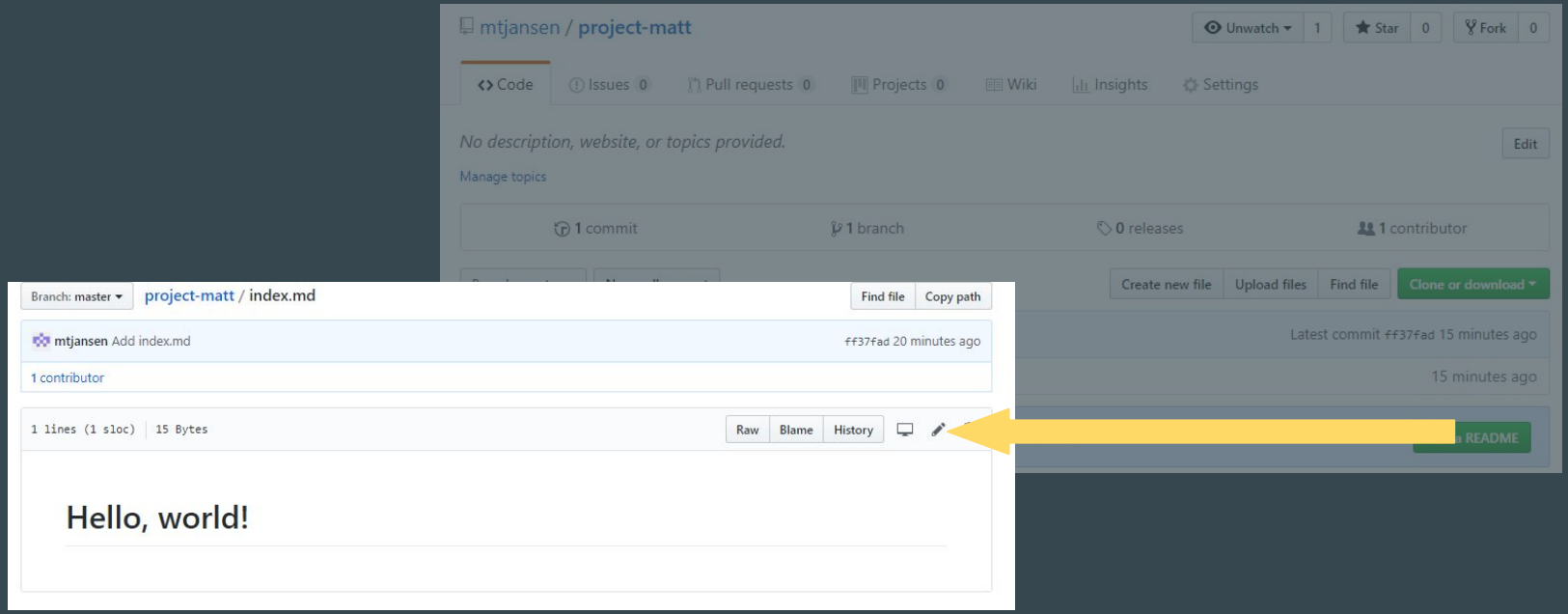



```
$ git log
commit a93520b0a05575204c4985376ec011a861dea372 (HEAD -> master,
origin/master)
Author: Matt Jansen <mtjansen@email.unc.edu>
Date:   Wed Sep 26 12:18:07 2018 -0400

    Add index.md
```

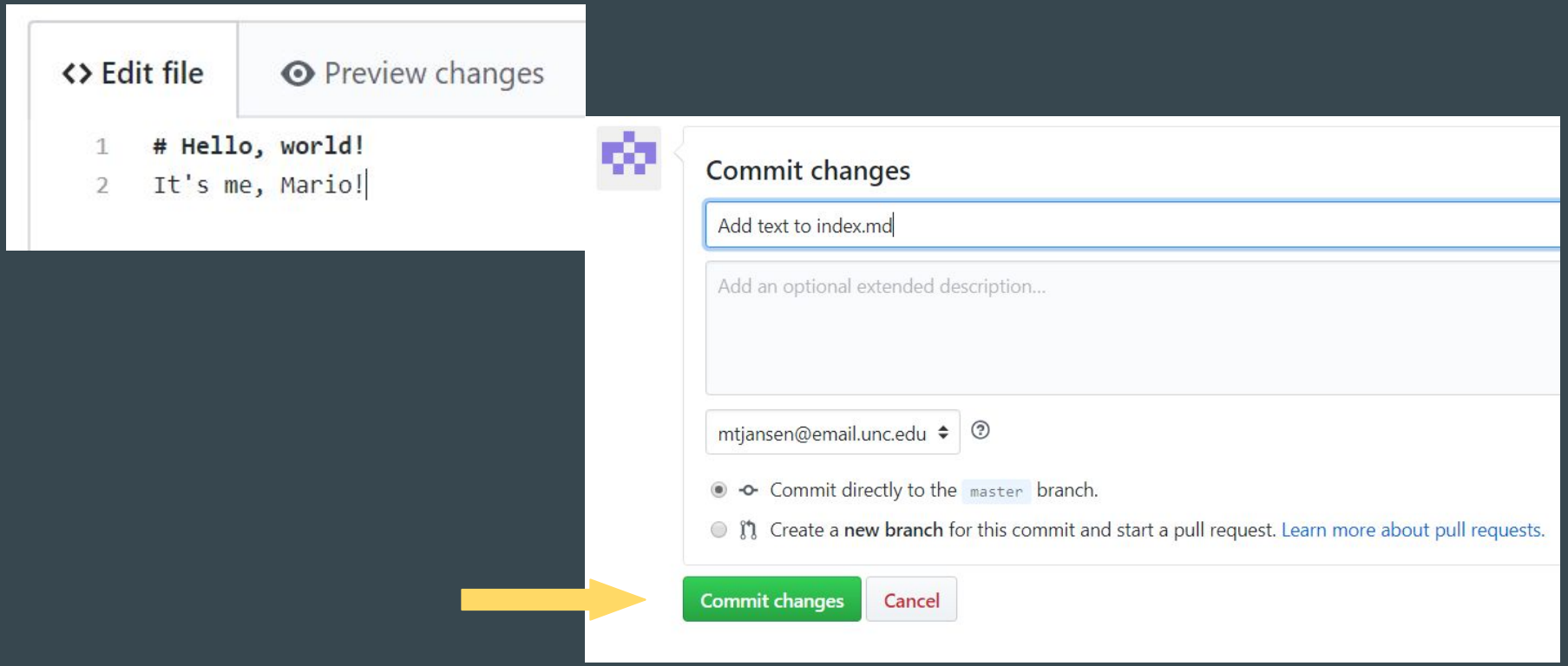
Let's Change Something Remotely



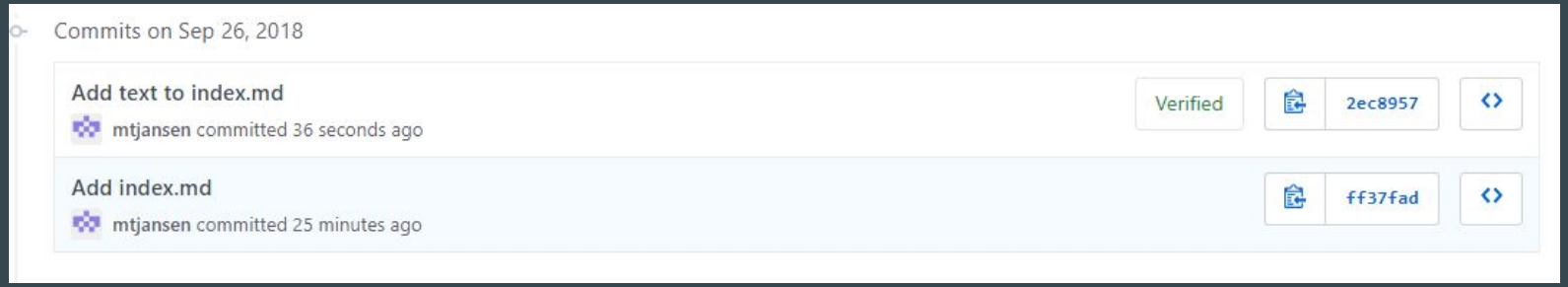
Let's Change Something Remotely



Let's Change Something Remotely



git pull Pulls Remote Changes into Your Local Repository




```
$ git log
commit a93520b0a05575204c4985376ec011a861dea372 (HEAD -> master,
origin/master)
Author: Matt Jansen <mtjansen@email.unc.edu>
Date:   Wed Sep 26 12:18:07 2018 -0400

    Add index.md
```

git pull Pulls Remote Changes into Your Local Repository

```
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/mtjansen/project-matt
 * branch            master     -> FETCH_HEAD
   ff37fad..2ec8957  master     -> origin/master
Updating ff37fad..2ec8957
Fast-forward
 index.md | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

git pull Pulls Remote Changes into Your Local Repository

```
$ git log
commit 2ec895774dbc69fd6bbfd16150287c7e20c443ca (HEAD -> master, origin/master)
Author: Matt Jansen <mtjansen@email.unc.edu>
Date:   Wed Sep 26 12:42:26 2018 -0400

    Add text to index.md

commit ff37fada75f8550430d637f2fc8c308926d7e06e
Author: Matt Jansen <mtjansen@email.unc.edu>
Date:   Wed Sep 26 12:18:07 2018 -0400

    Add index.md

$ git pull origin master
Already up-to-date.
```

Review!

```
git config user.email

git init

git status

git add filename

git commit -m "Something"

git log

git remote add origin https://github.com/ username/repo-name.git

git push origin master

git pull origin master
```

https://xkcd.com/1597/

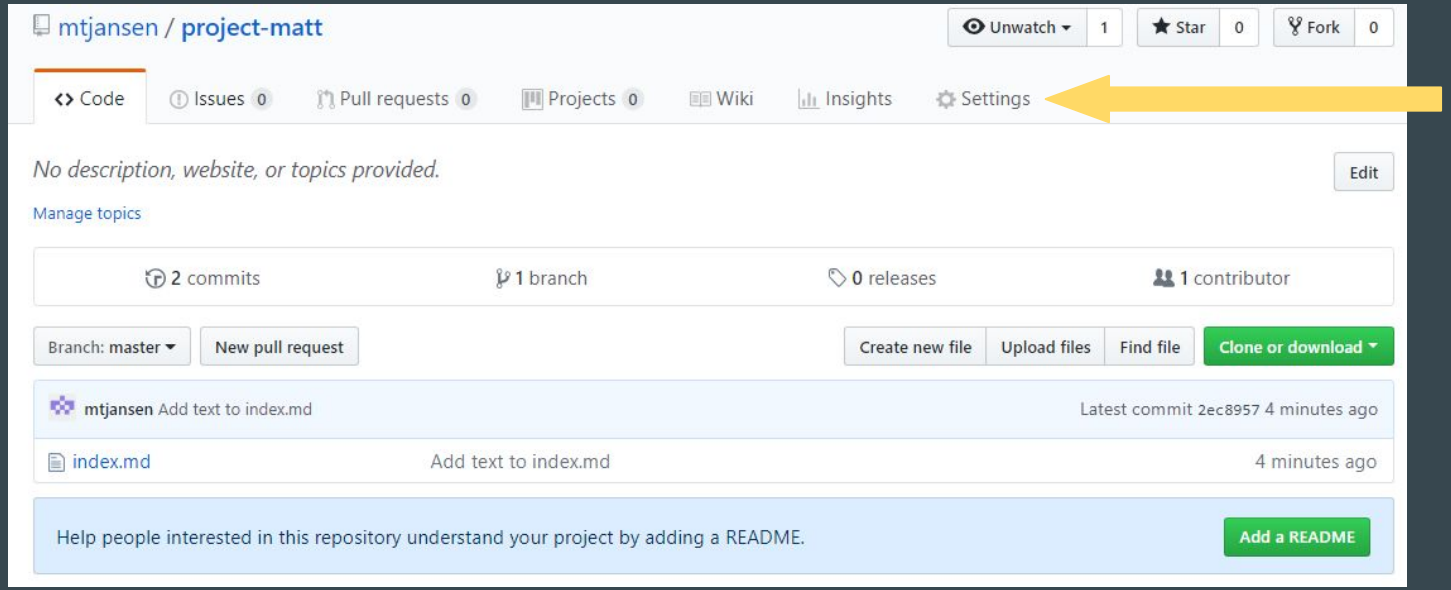
GitHub Pages!

GitHub Pages Is An Easy Way to Make a Collaborative Site

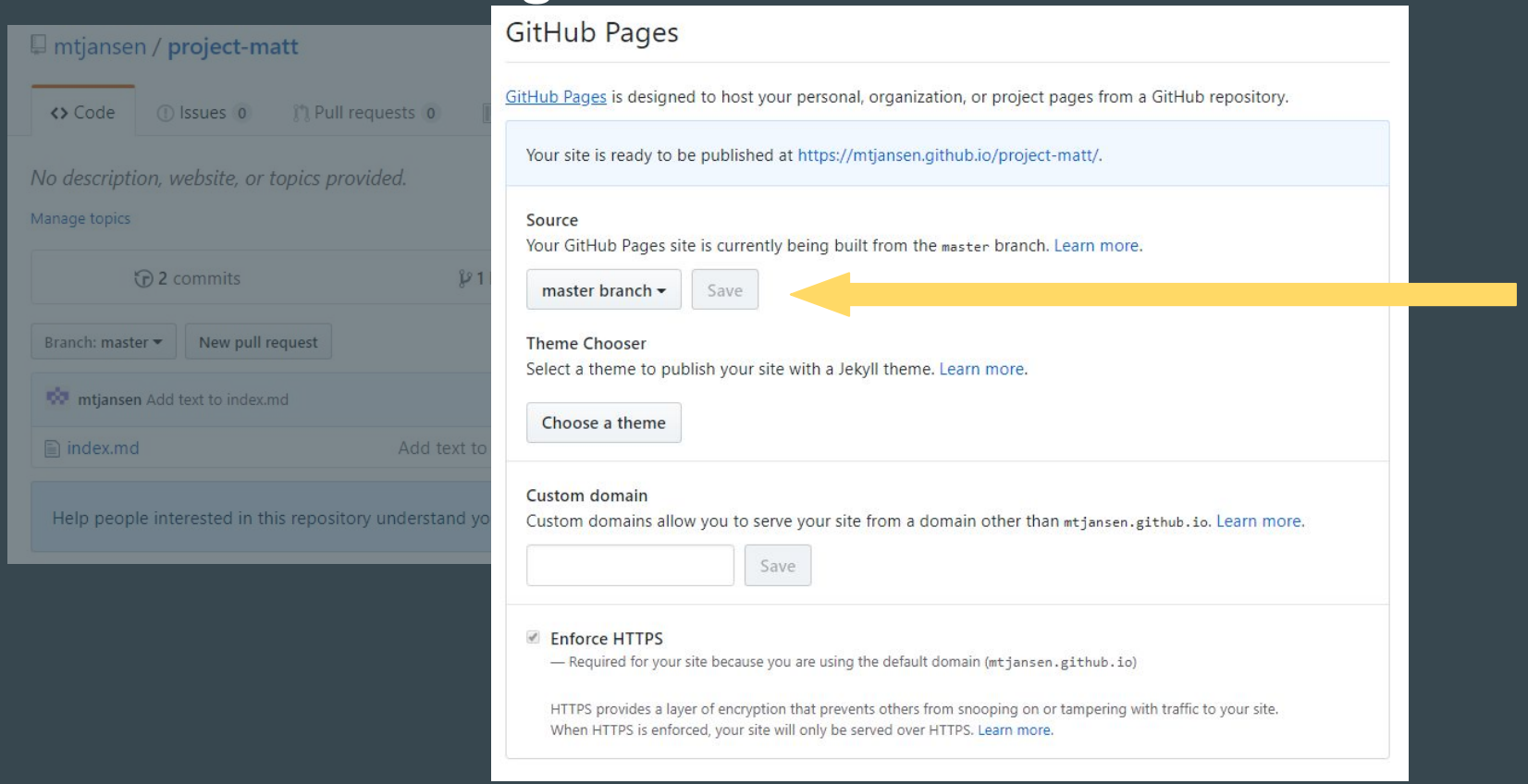
GitHub Pages allows anyone with a GitHub account to create a static website using:

- HTML or Markdown
- Git!

Create a GitHub Page

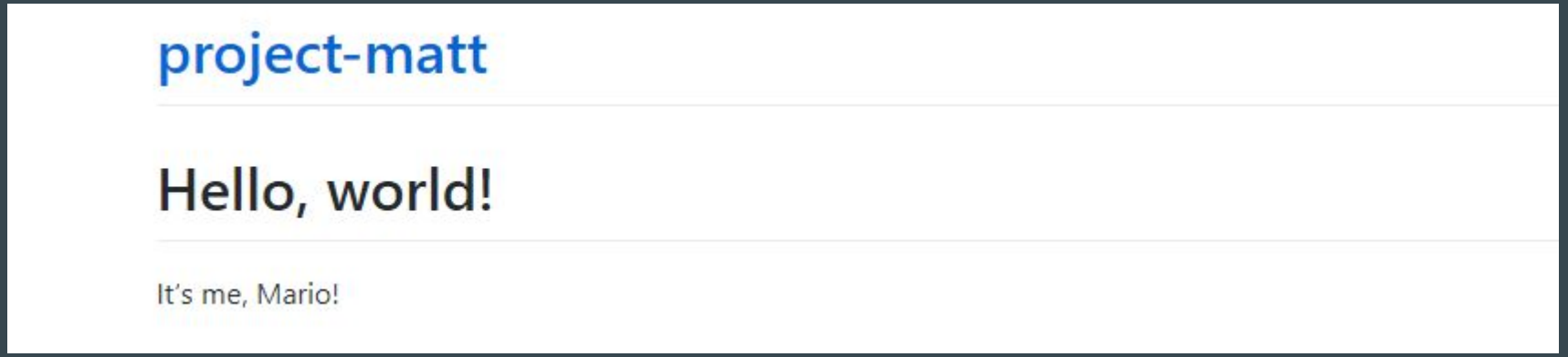


Create a GitHub Page



What We're Starting With...

https://*username*.github.io/project-*name*/



Copy an Existing Git Repo

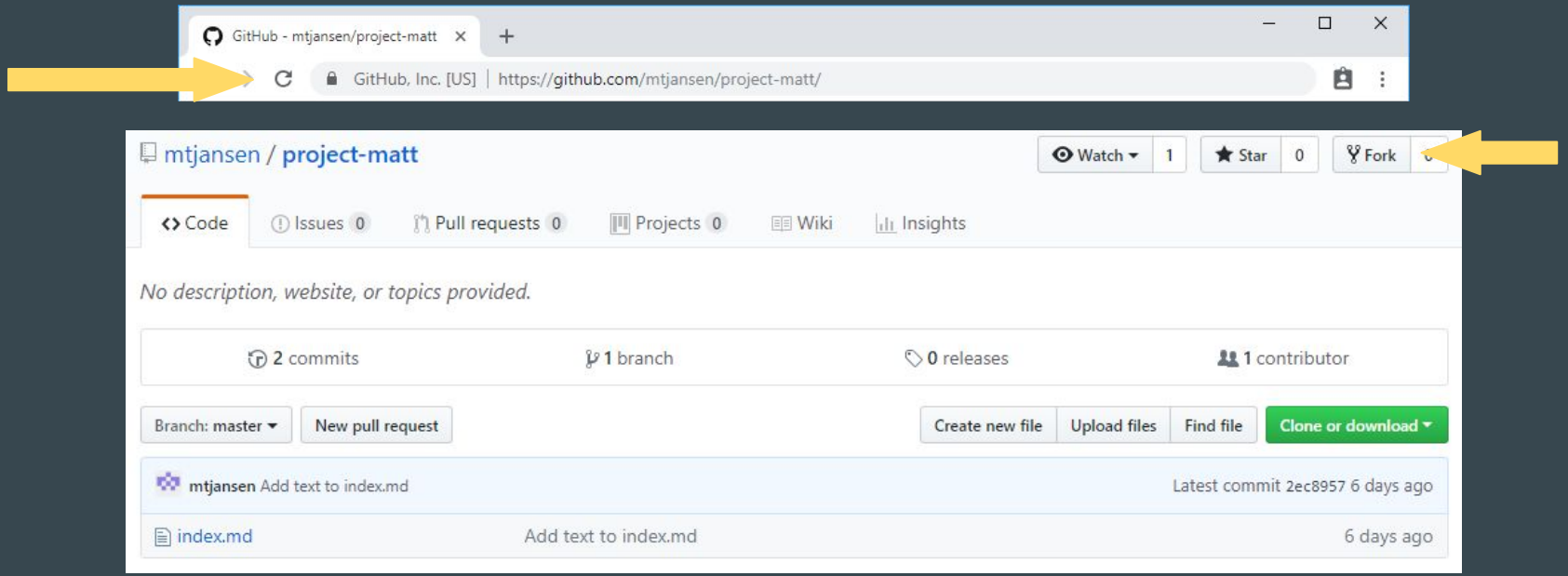
Editor:

- Fork neighbor's GitHub repo to create a remote copy you can edit
- Clone repo to have a local copy
- Modify, stage, commit. and push the change up to your GitHub repo
- Submit a pull request to your neighbor's repo

Repo Owner:

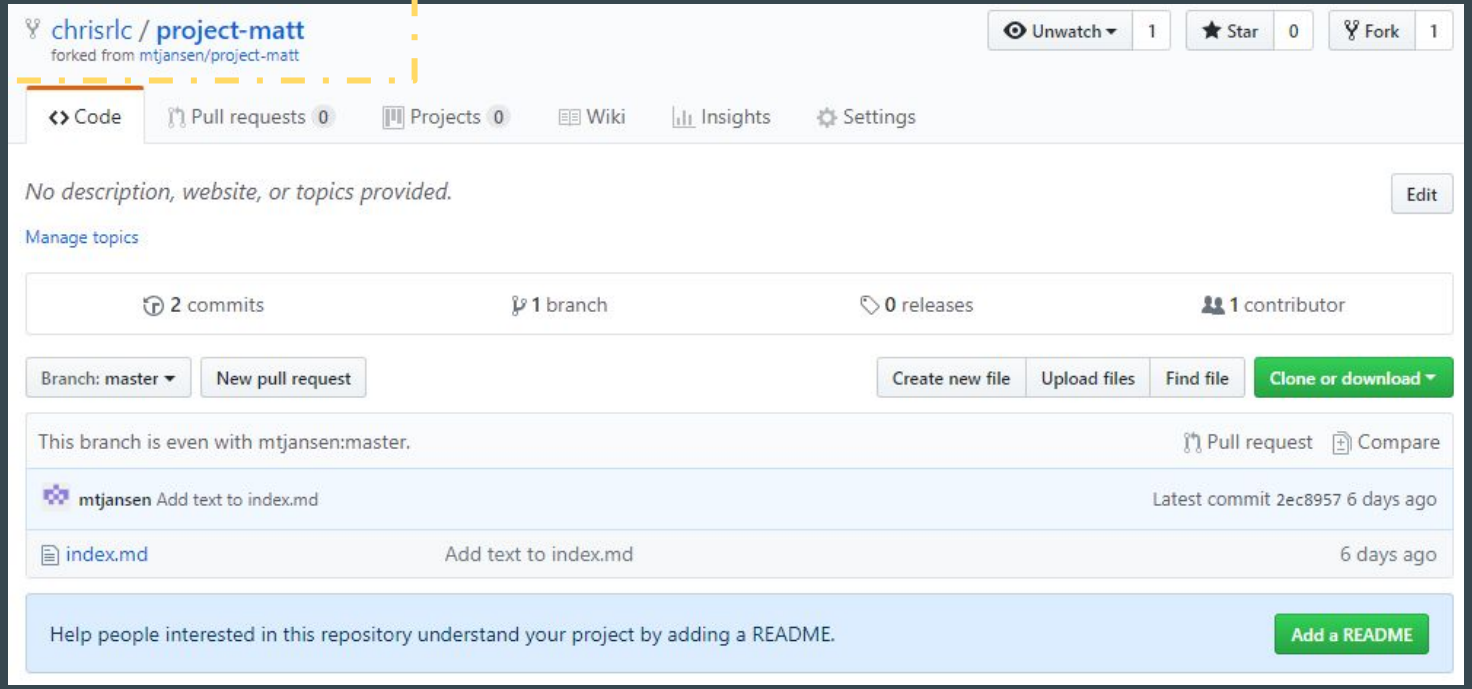
- Approve pull request and merge into original repo

GitHub Fork Copies Remote Projects on Github



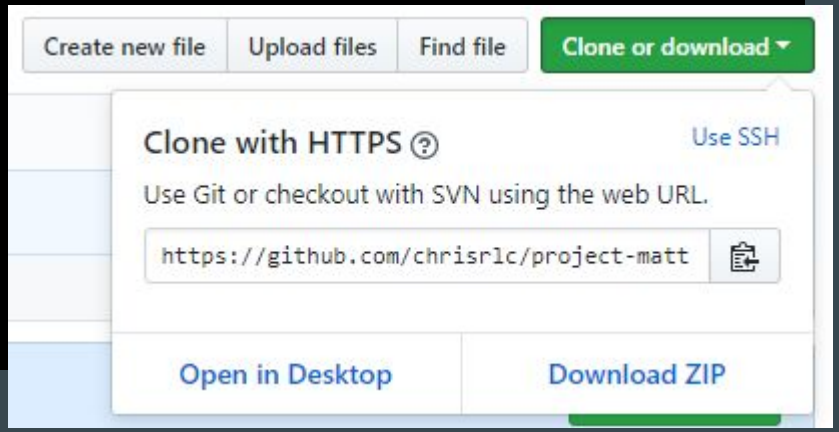
https://github.com/mtjansen/project-matt/

GitHub Fork Copies Remote Projects on Github



git clone Copies an Existing Repository to Your Computer

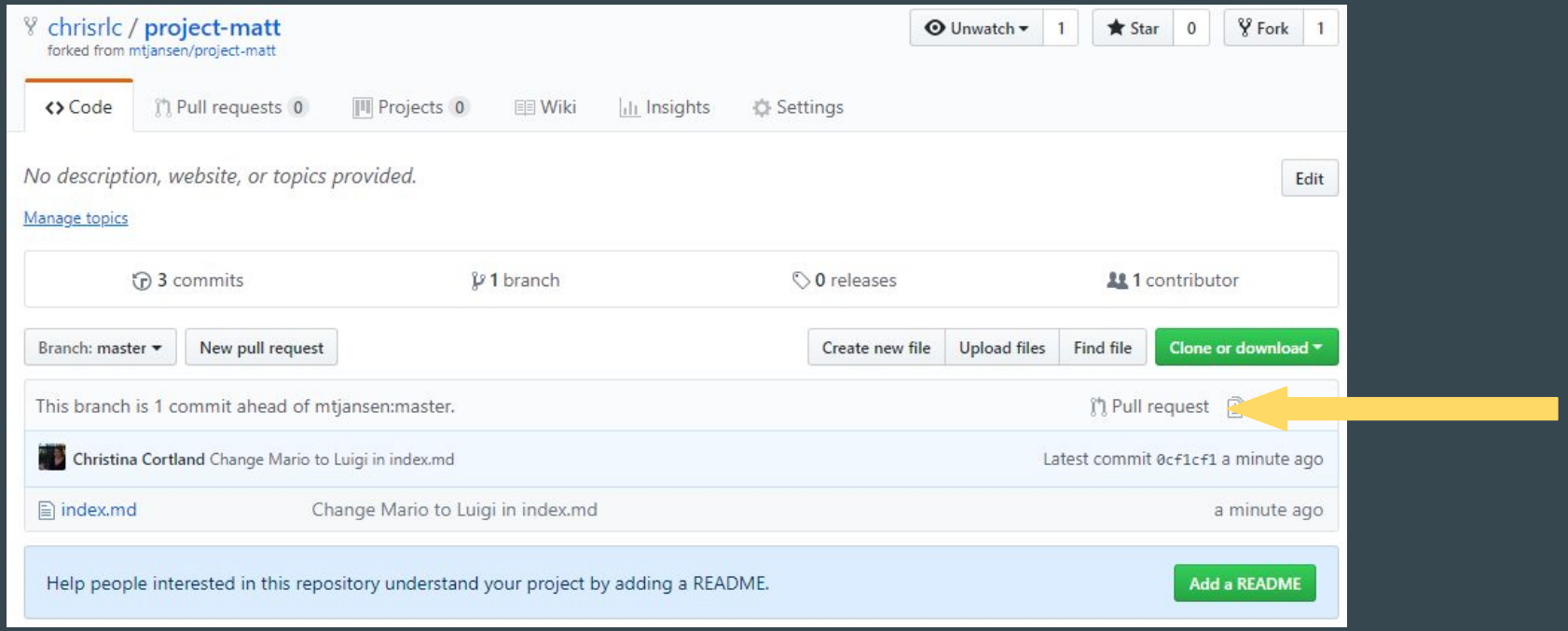
```
$ pwd

$ cd ..

$ git clone https://github.com/username/project-neighbor.git

$ ls

$ cd project-neighbor

$ ls
```



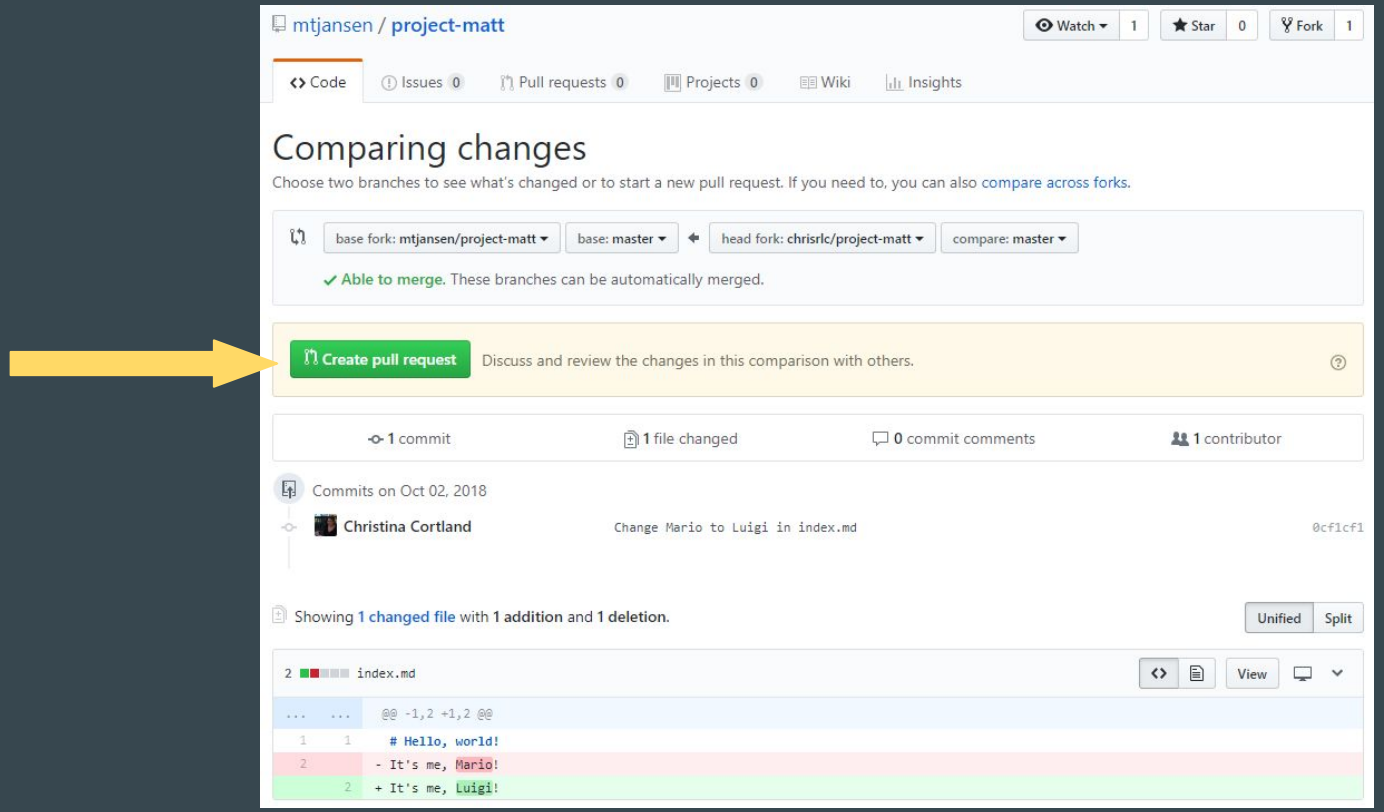
Exercise: Change Something

Stage the changes	`git add filename`
Commit the staged changes with a message	`git commit -m "Something"`
Push your changes up to your GitHub repo	`git push origin master`

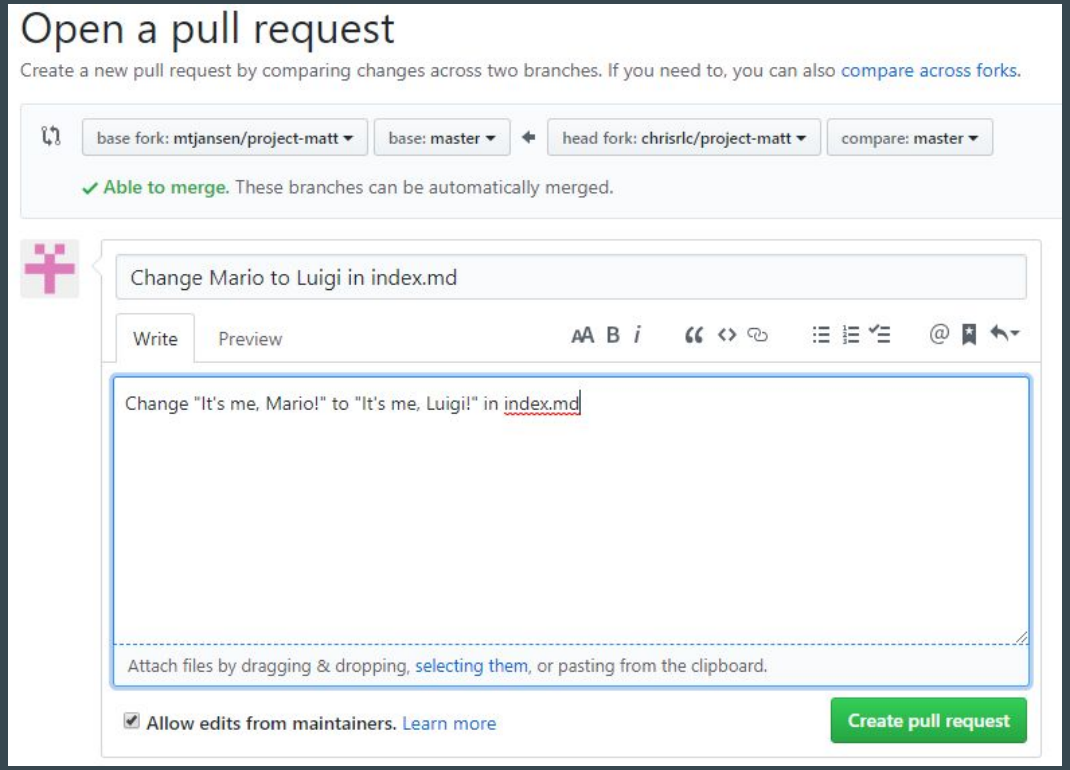
Pull Requests Propose Changes to a Repo



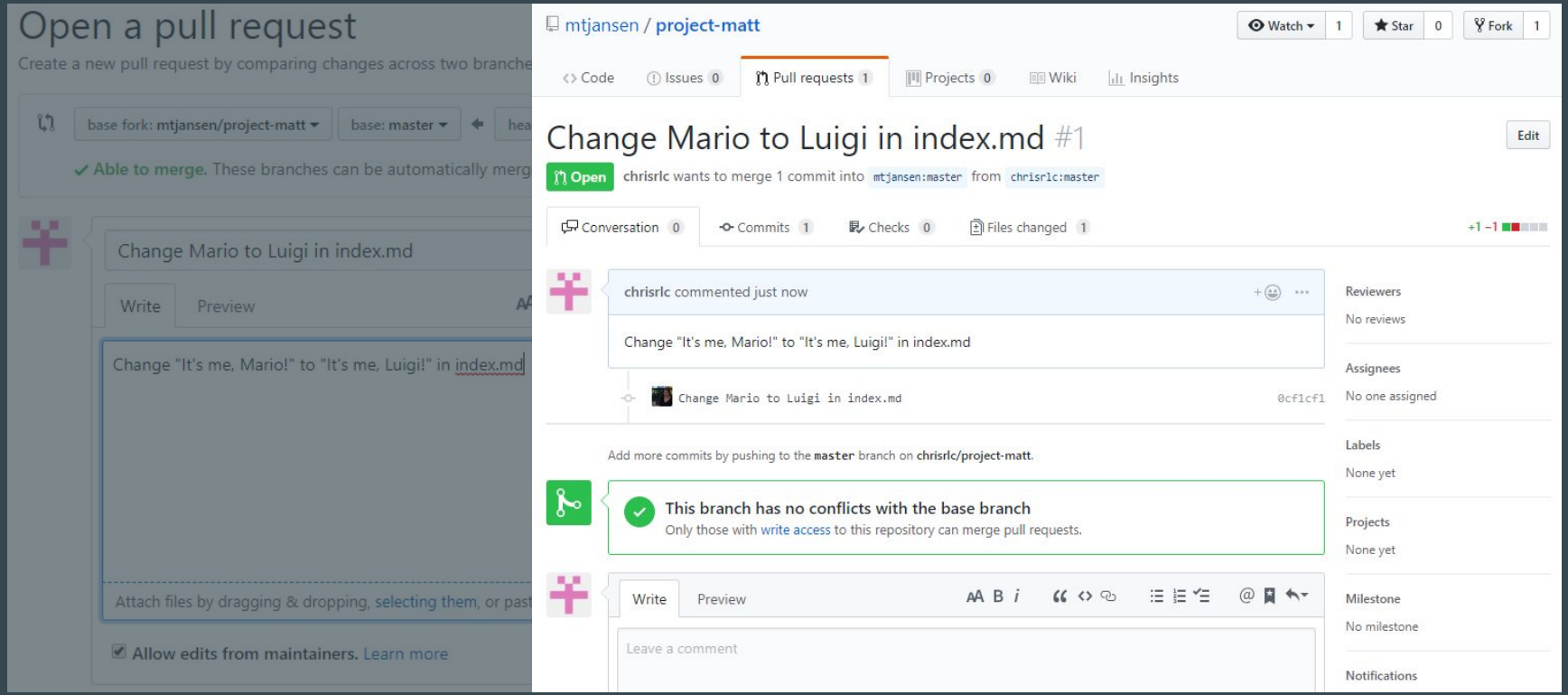
Pull Requests Propose Changes to a Repo



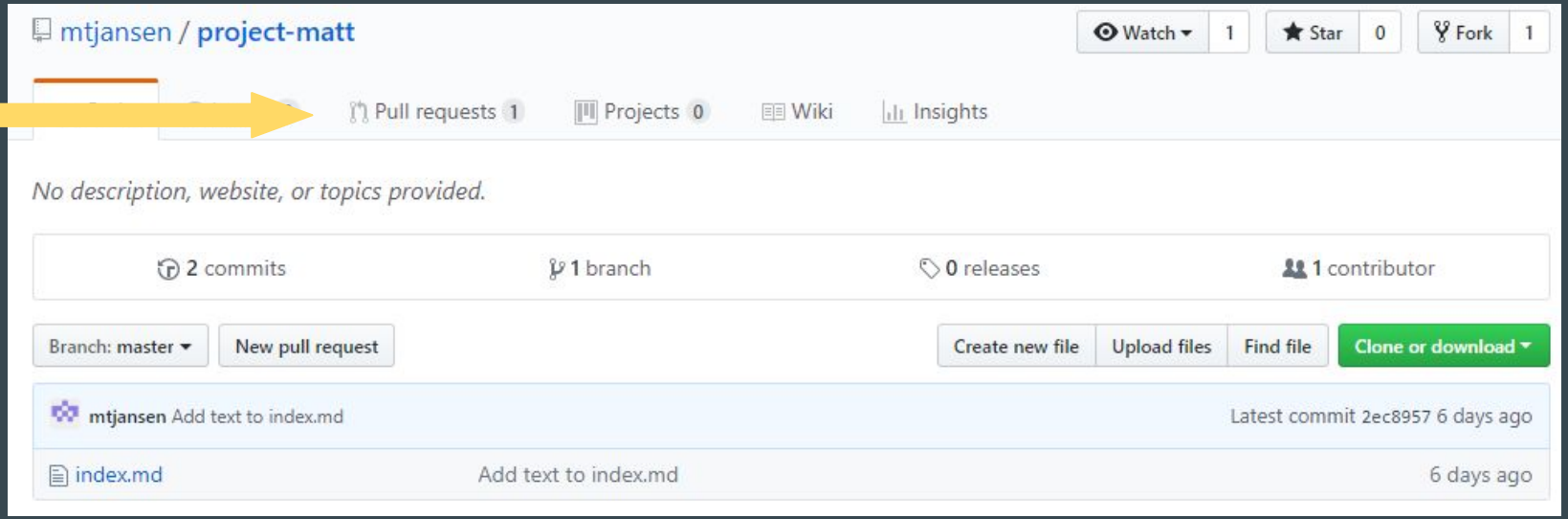
Pull Requests Propose Changes to a Repo



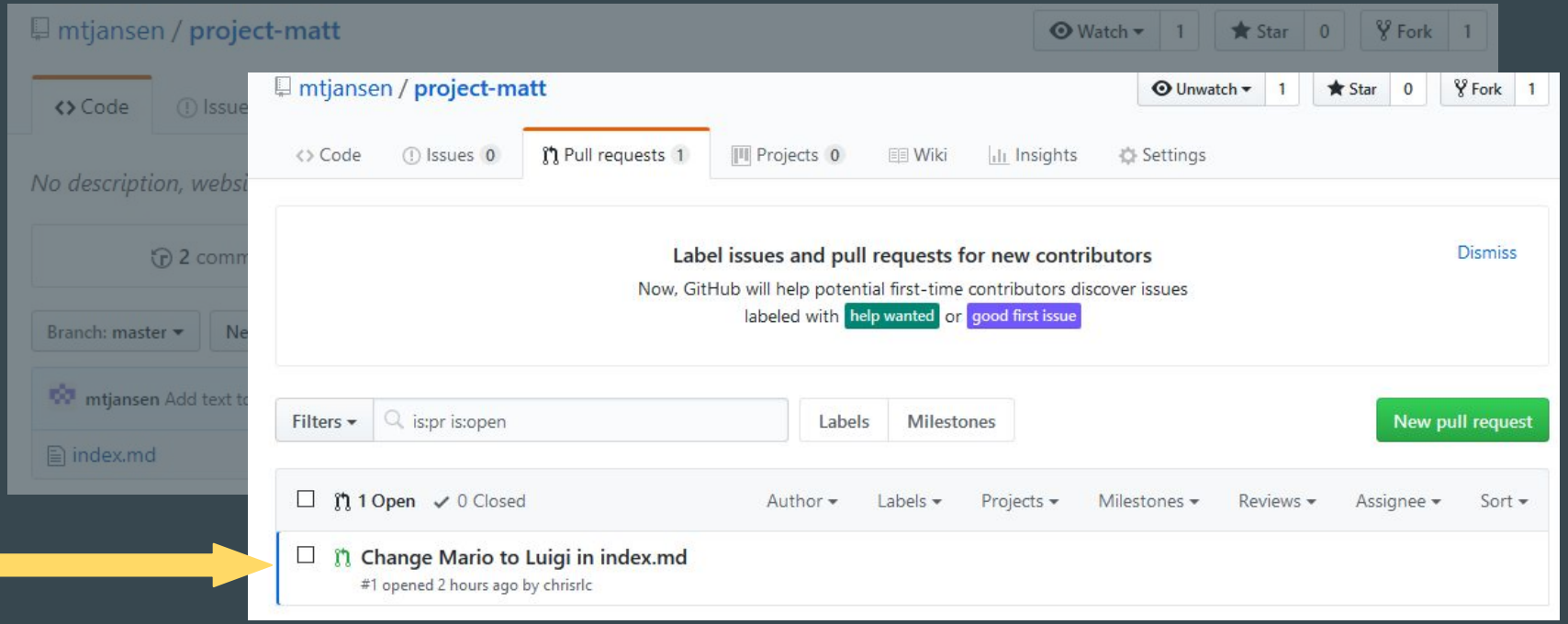
Pull Requests Propose Changes to a Repo



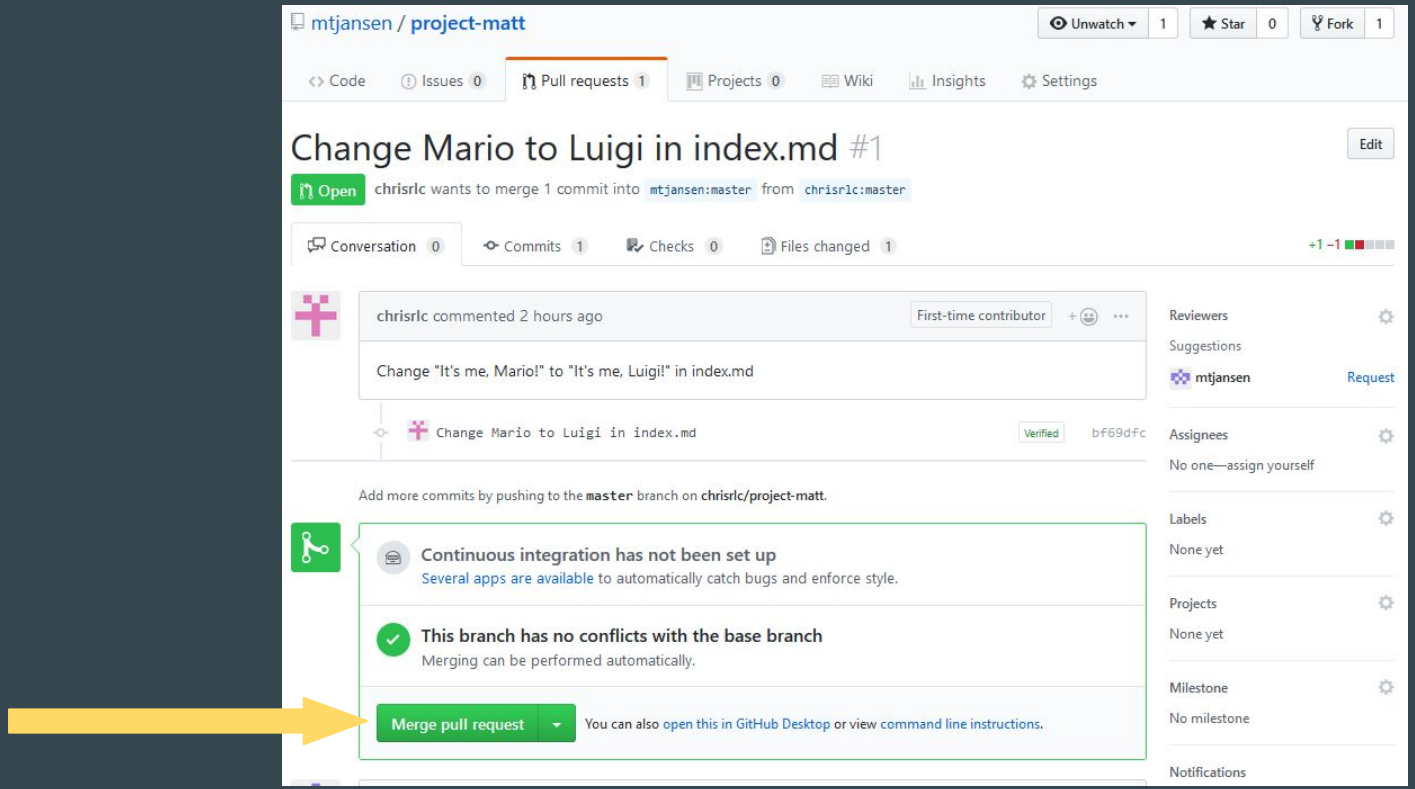
Merge Pull Request (Or Not)



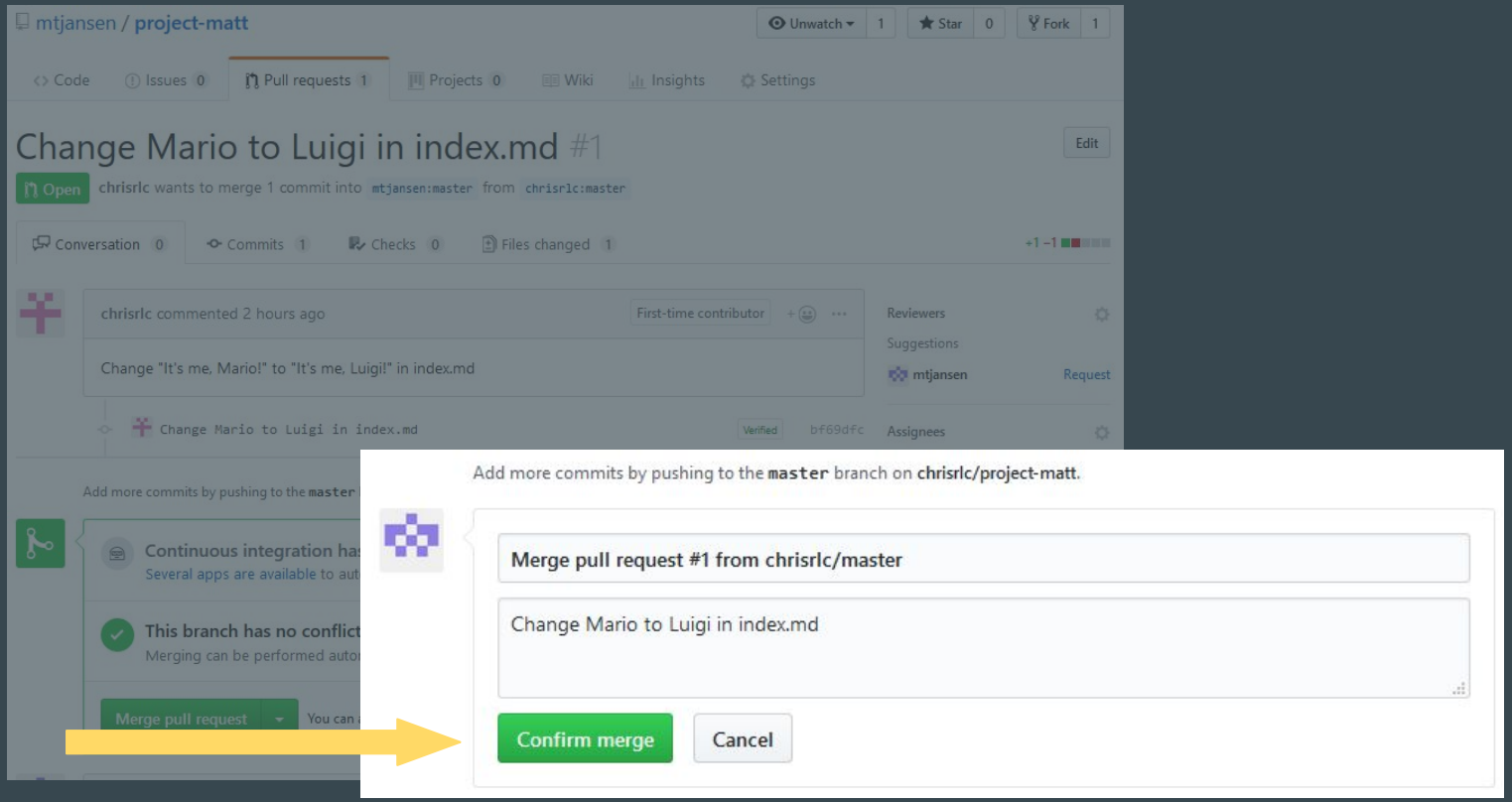
Merge Pull Request (Or Not)



Merge Pull Request (Or Not)

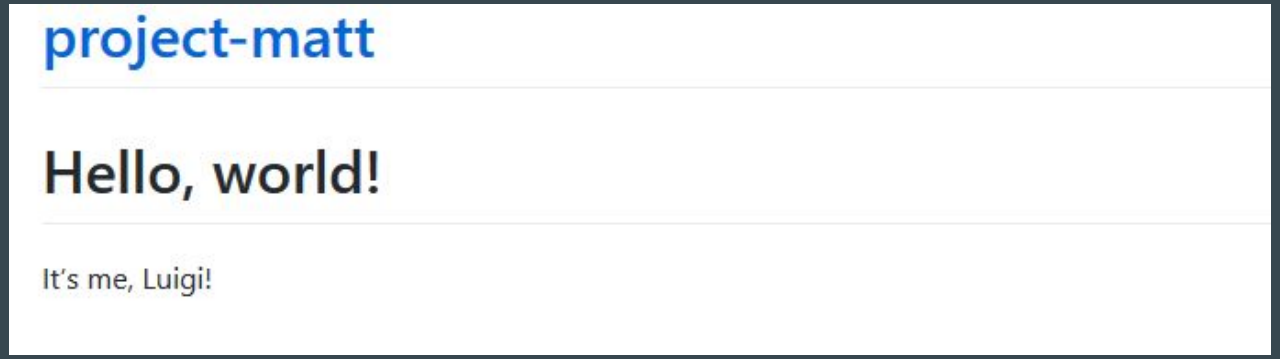


Merge Pull Request (Or Not)



Success!

https://*username*.github.io/project-*name*/



Feedback Survey:
http://unc.libsurveys.com/davishubfeedback