Get a Jumpstart on Collaboration and Code Review in GitHub

By Katherine "Kati" Michel

2017
djangocon.us
spokane

About ▾   Schedule ▾   Venue ▾   Speaking ▾   Sponsors ▾   News   Buy

Six days of talks, sprints, and tutorials by the community for the community.

August 13–18 | Hotel RL by Red Lion Spokane at the Park
2017   **Spokane, Washington**

github
SOCIAL CODING
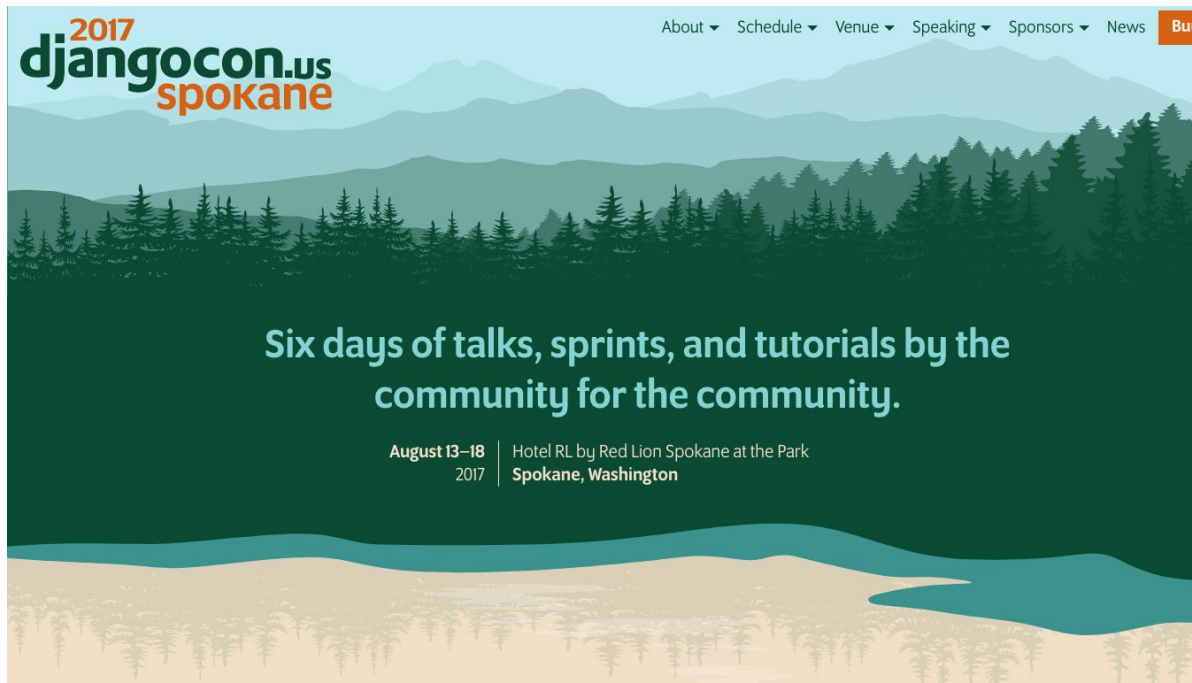
Welcome!

Goal

I want to teach you a process that will get you started collaborating and doing code review as quickly as possible

Announcing TacoFancy!

dan sinker @dansinker · 3 Nov 2013

Announcing tacofancy, a github repo for community-driven taco amazingness:

sinker/tacofancy

community-driven taco repo. stars stars stars. Contribute to tacofancy development by creating an account on GitHub.

github.com

16   19

The One Sentence That Motivated Me to Start Using GitHub

Are You New to Github But Want to Contribute?

No better time than the present to learn Git! Thankfully, this repo is pretty simple. Github makes some amazing guides that are a great starting point. Specifically, I'd recommend you start with the Understanding the Github Workflow guide, which is an awesome top-level explanation of how all this works. From there, pretty much everything you'd want to know to contribute to Tacofancy is contained in Github's very handy Forking Projects guide section. Beyond that, their help section is crazy helpful.

My First Pull Request

DjangoCon US Website as an Example Project
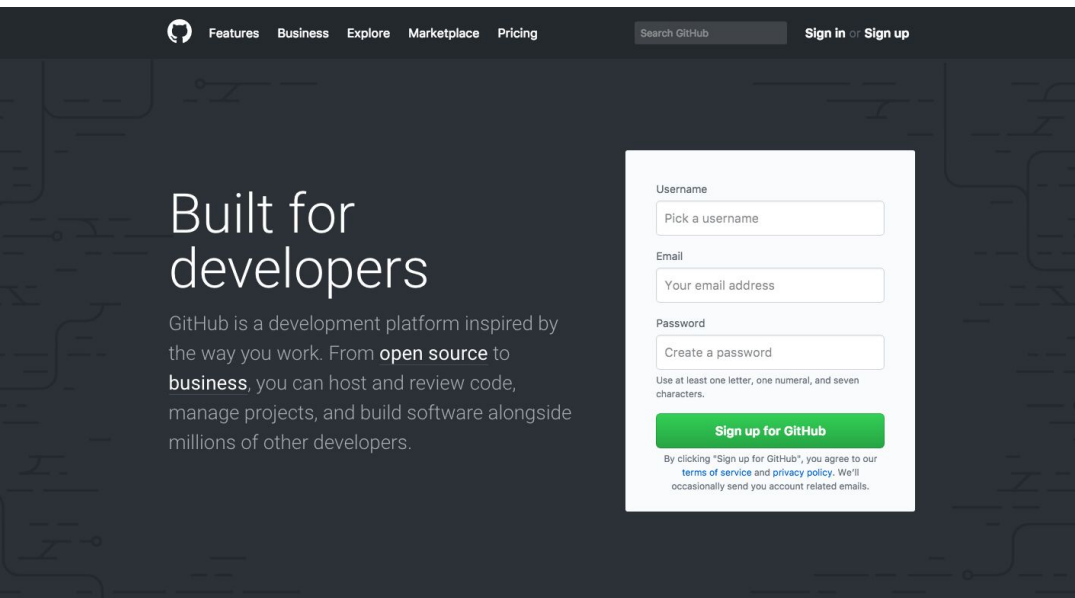
DjangoCon US Website Homepage
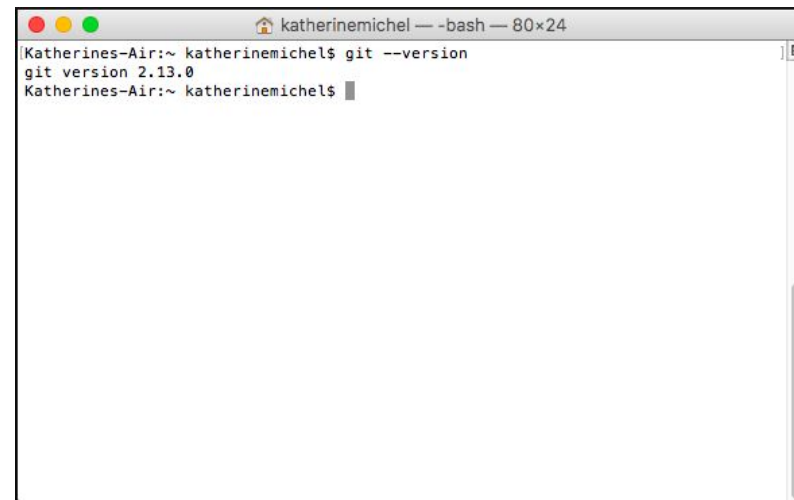

DjangoCon US Website Repo
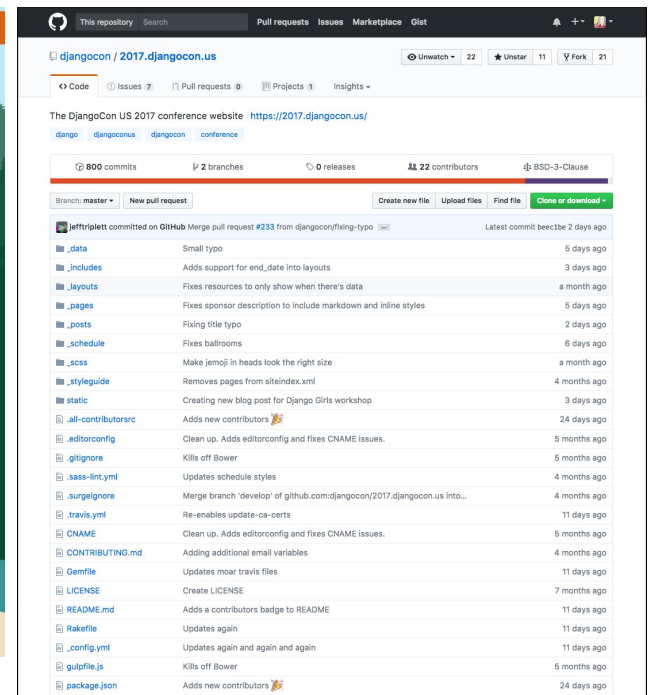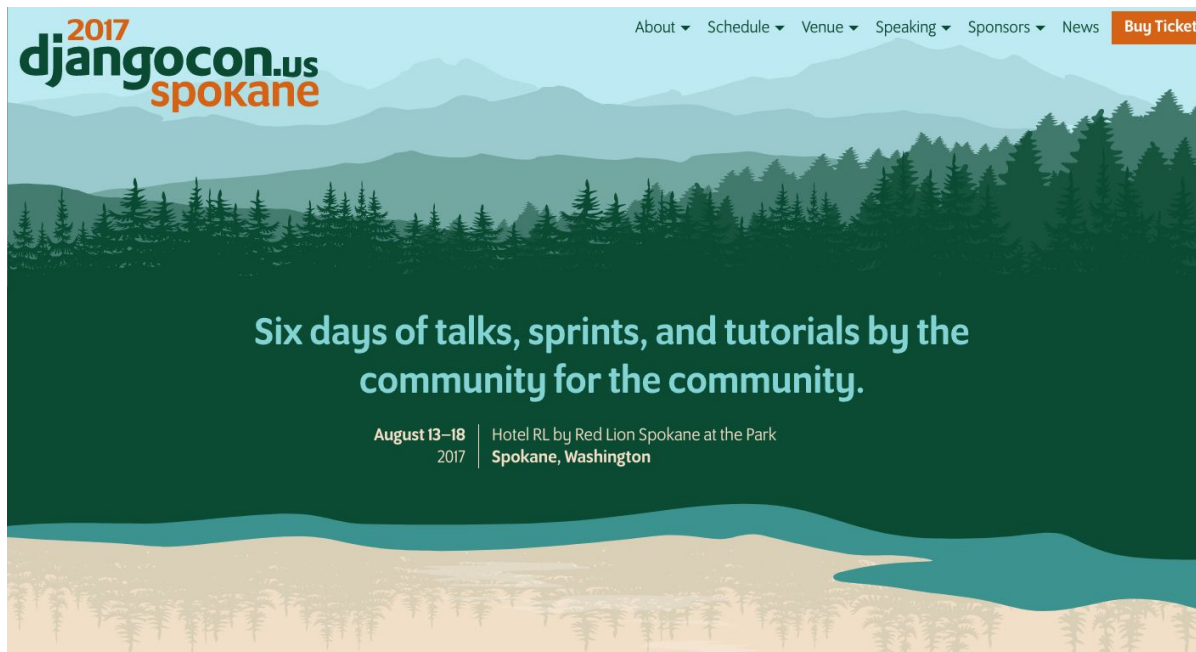
What are Git and GitHub?

GitHub is a website built on the version control software Git.
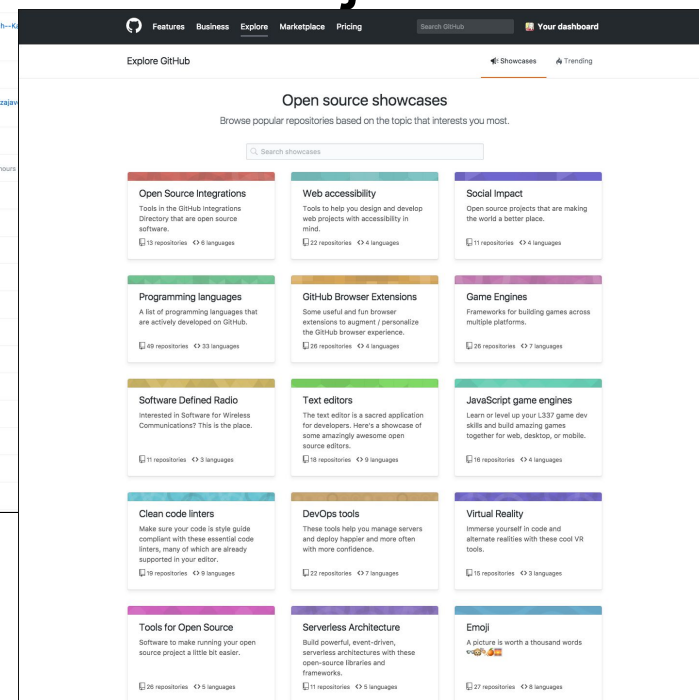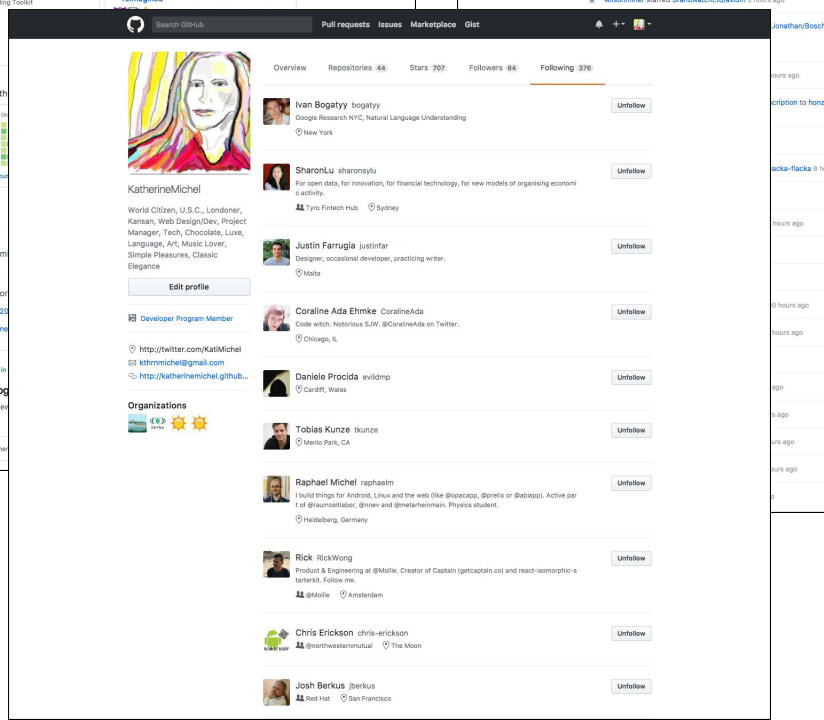
GitHub Website
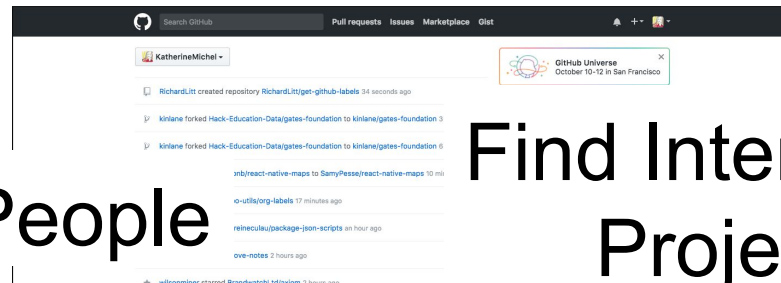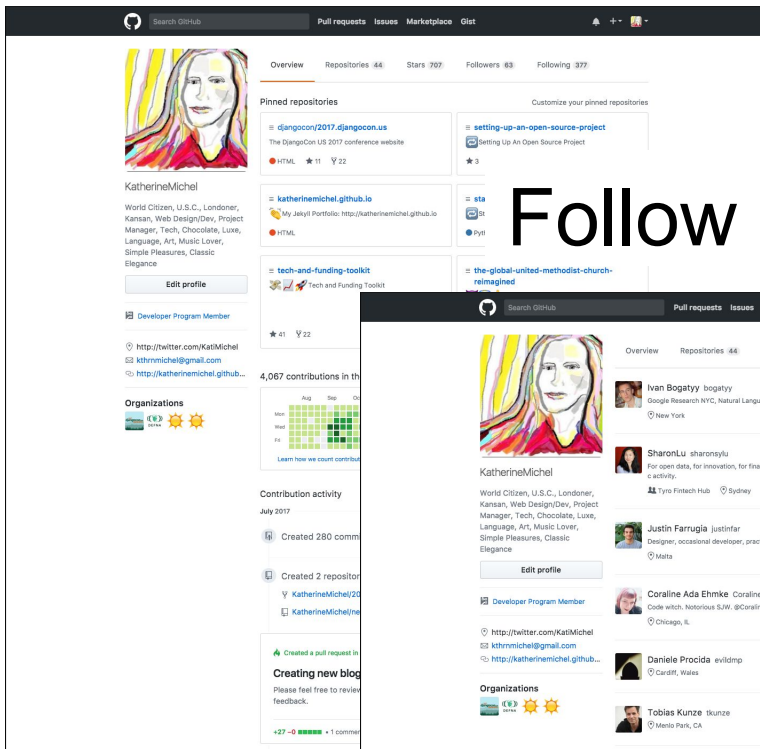
Git Command Line

GitHub: Social Network

Make a User Profile

Follow Activity in Newsfeed
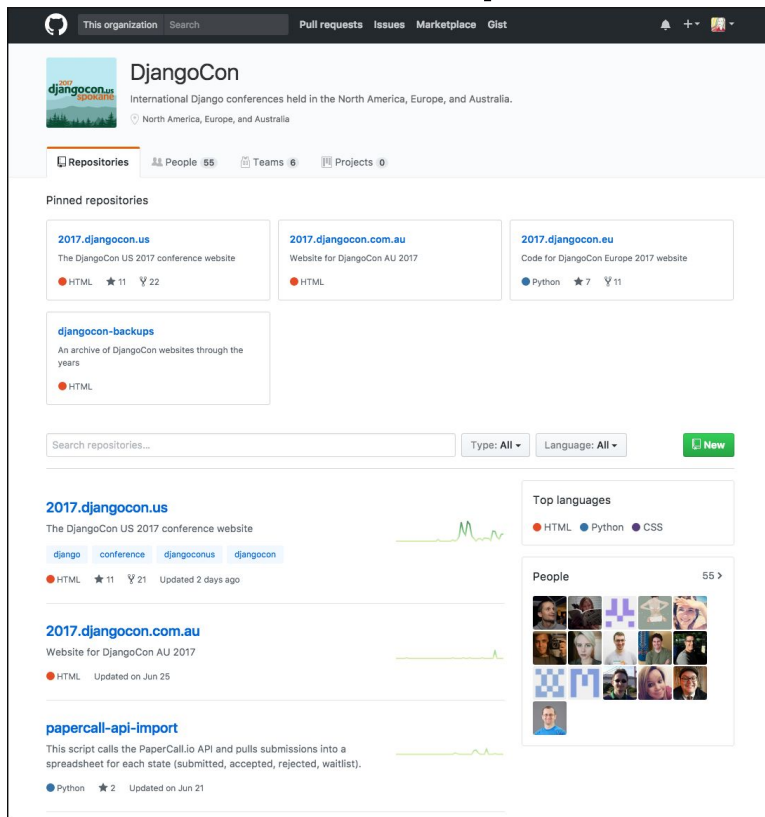
Follow People

Find Interesting Projects

GitHub: Repositories

GitHub users work on code together

DjangoCon US Organization

Account Repo List

GitHub: Repositories

GitHub users work on code together

DjangoCon US Organization
　　　Account Repo List

GitHub: Repositories

GitHub users work on code together

DjangoCon US Organization Account Repo List

DjangoCon US Website Repo

Local Dev Environment

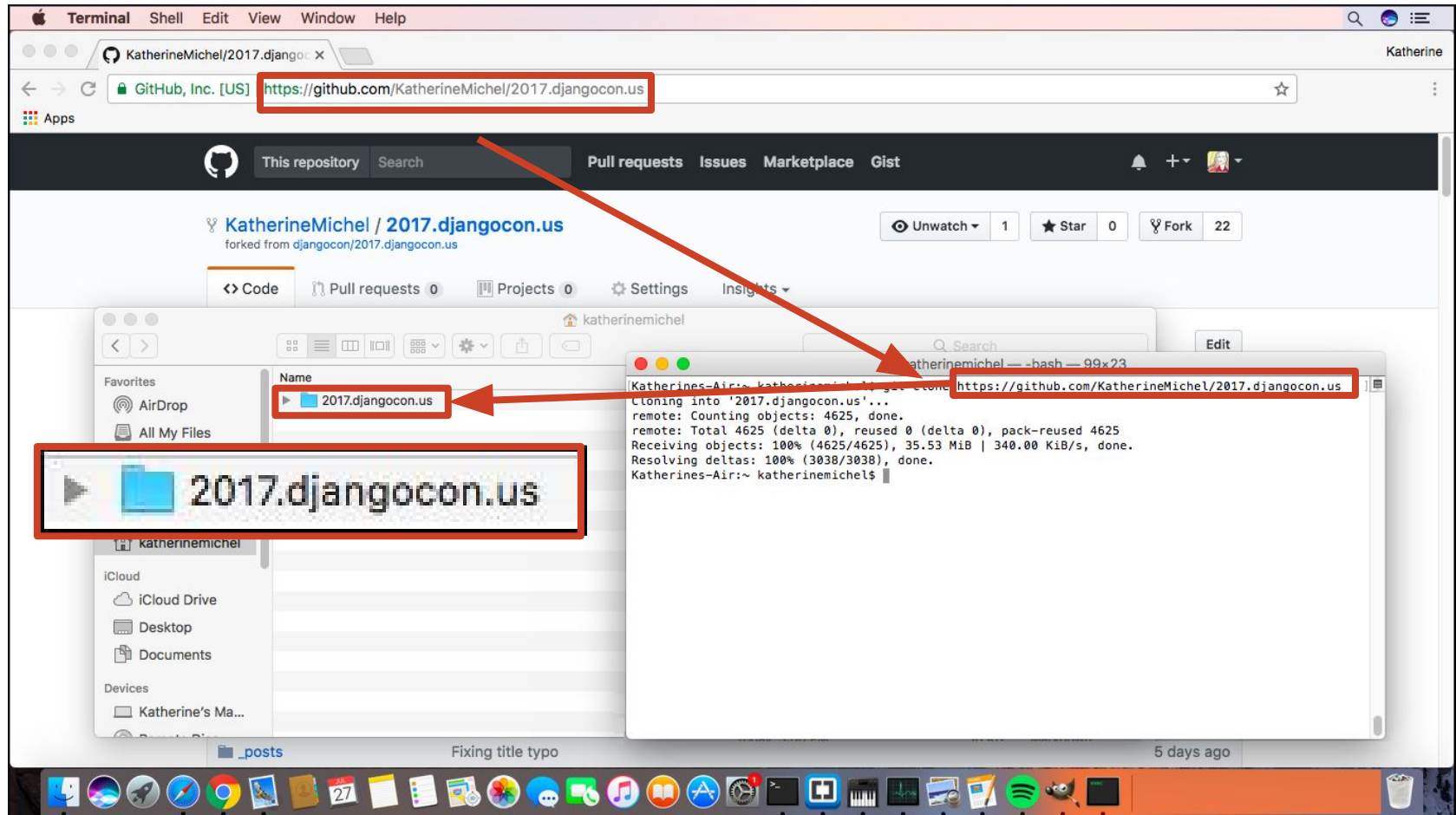# Local Dev Environment

$ git clone https://github.com/KatherineMichel/2017.djangocon.us

Local Dev Environment

$ git clone https://github.com/KatherineMichel/2017.djangocon.us
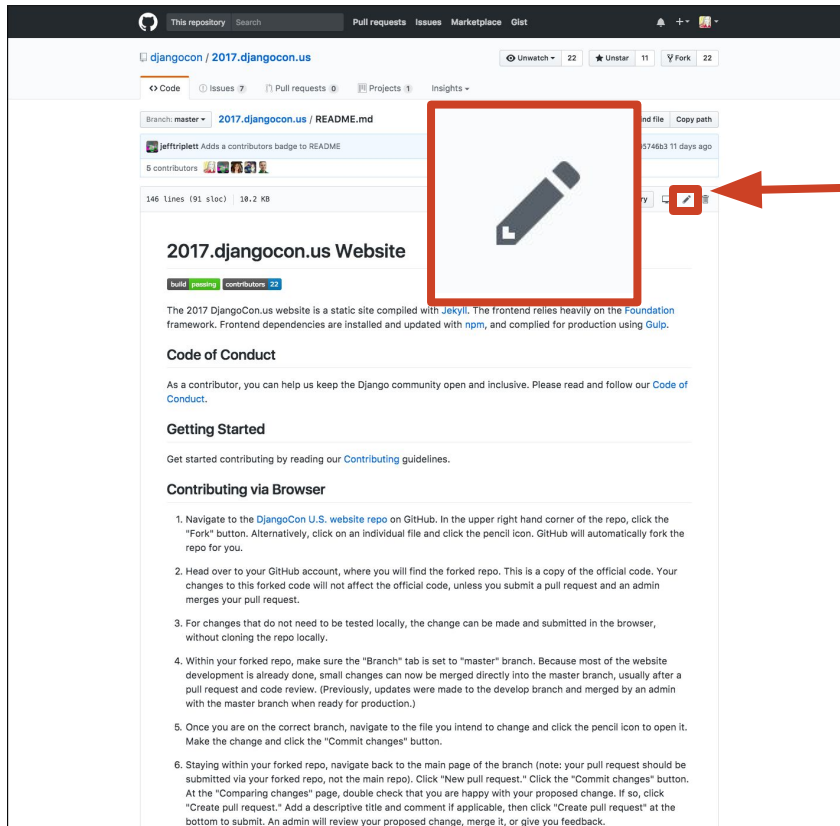
Collaboration and Code Review Best Practice Workflow

You need to switch between multiple tasks:
- Keep codebase up-to-date
- Create one or more features
- Do code review

Working on a File in GitHub
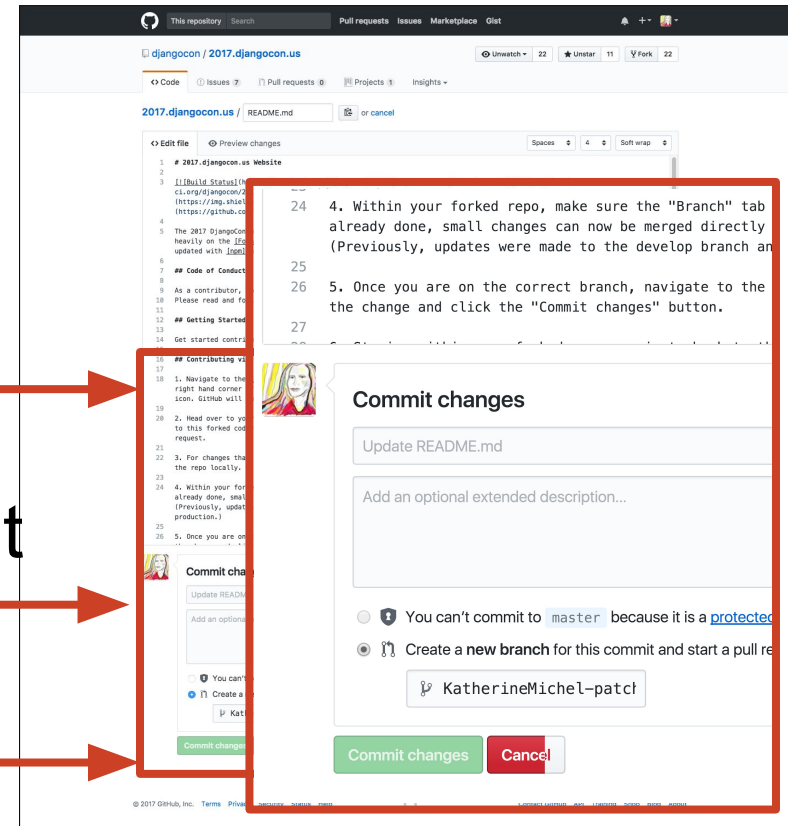
File on GitHub (README)
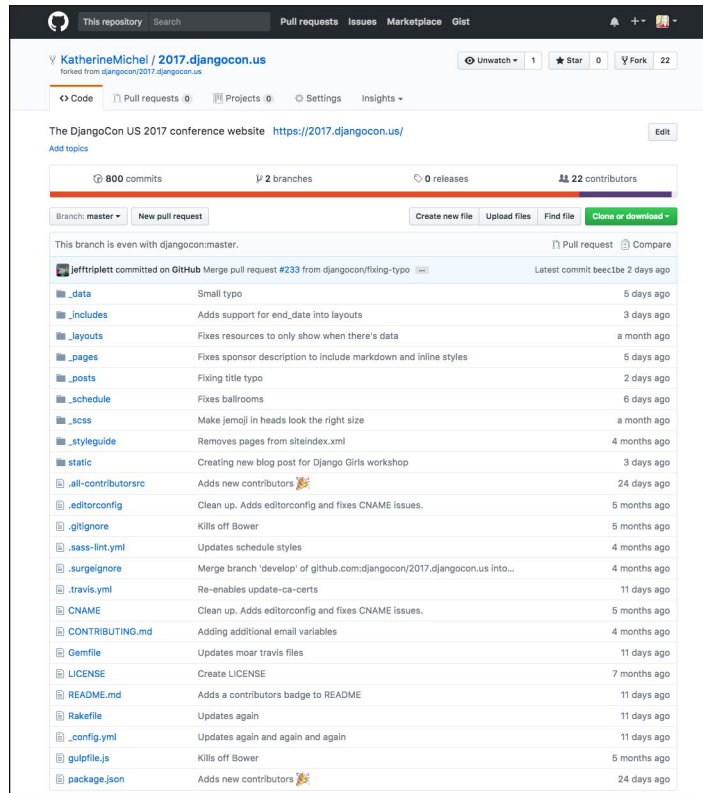
Editing File

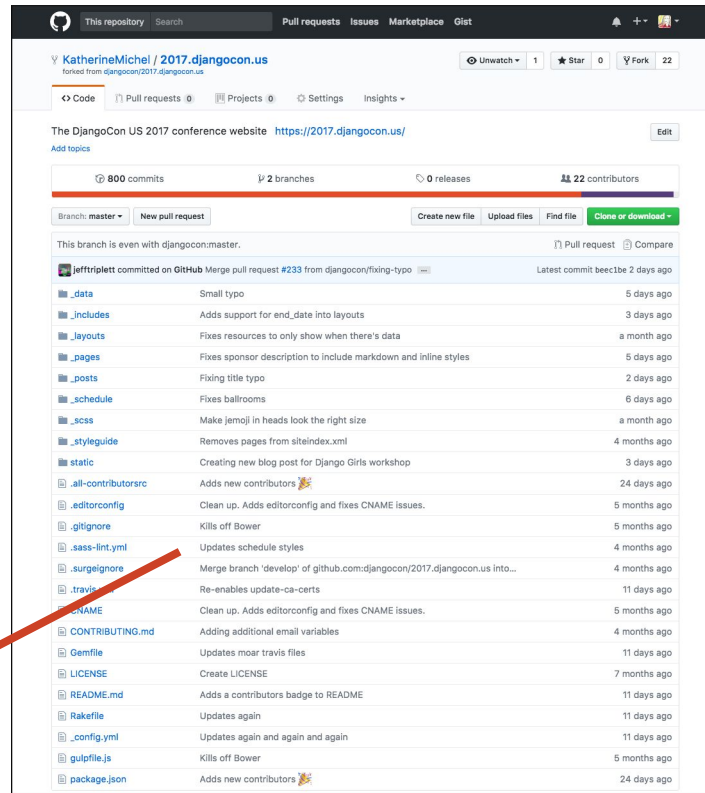Click pencil icon

Edit

Commit Info

Save

Branches

Master Branch

Branches

Master Branch

Feature Branch

Branches

Feature Branch

Master Branch

Feature Branch

Branches

Feature Branch

Feature Branch

Master Branch

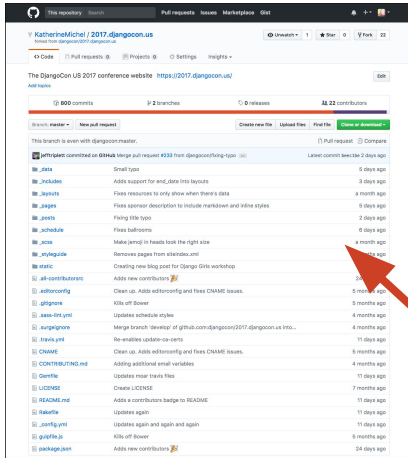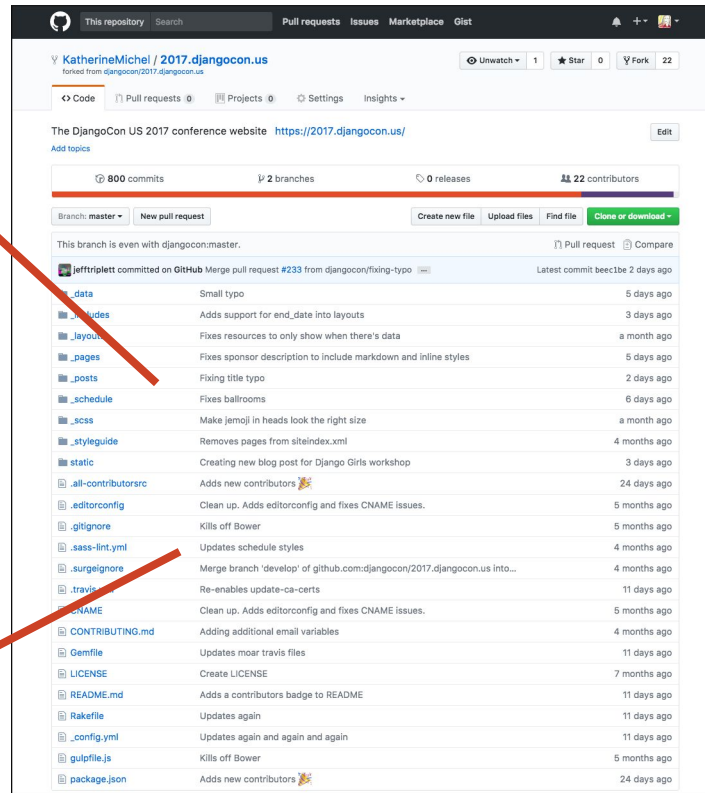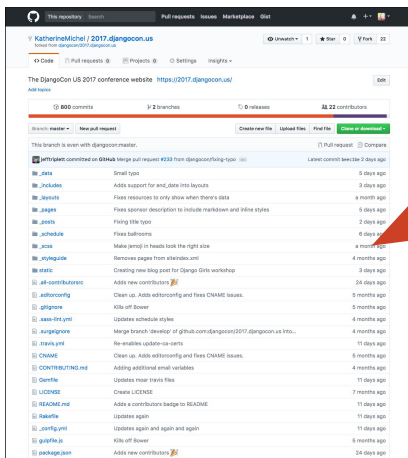## Feature Branch

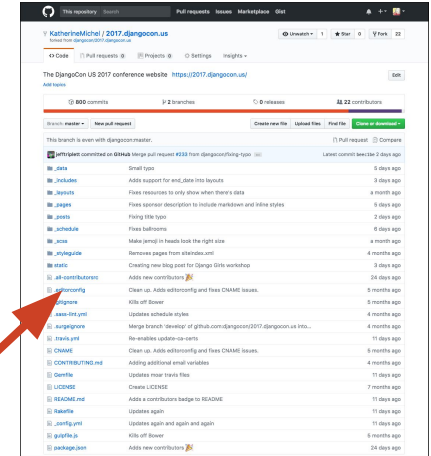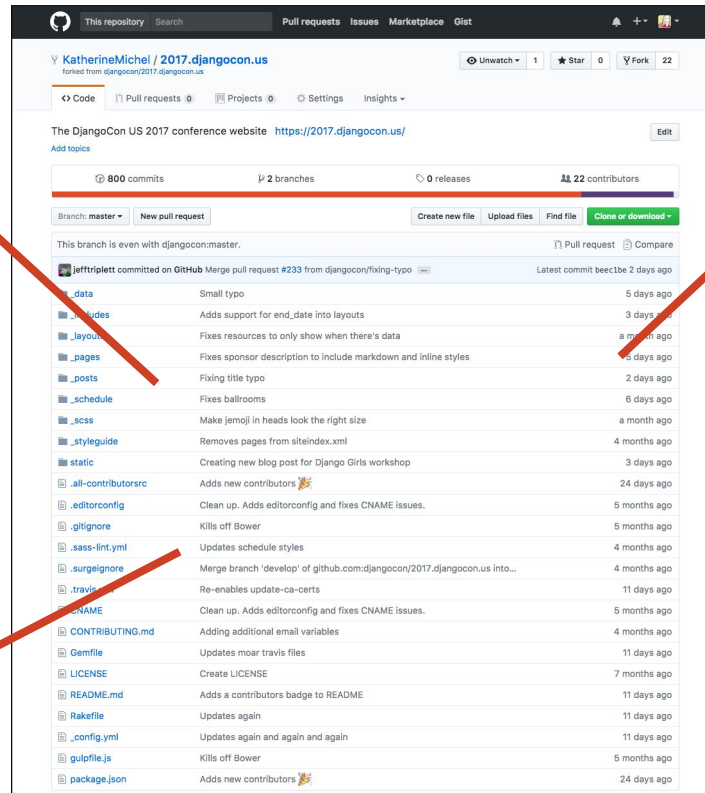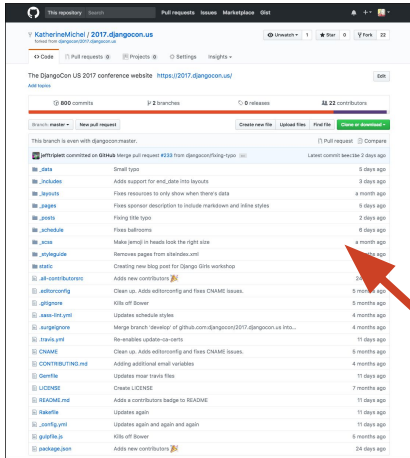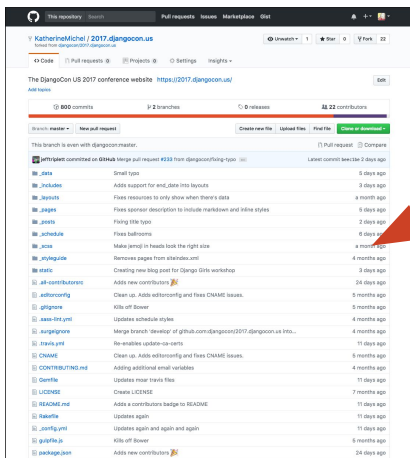# Branches

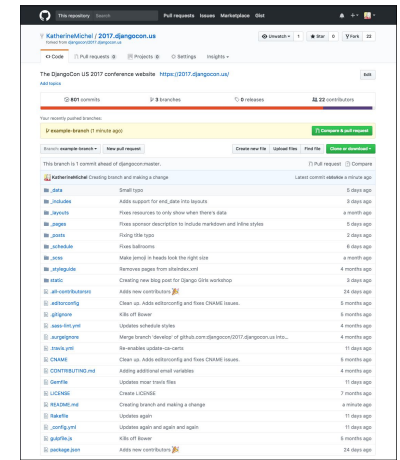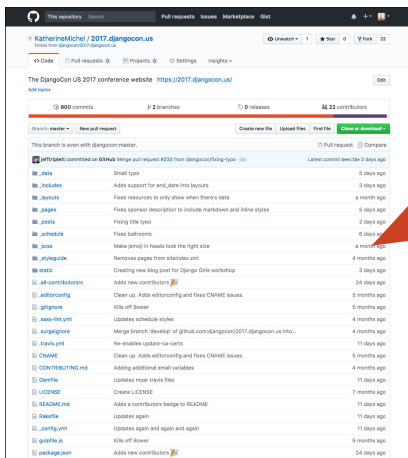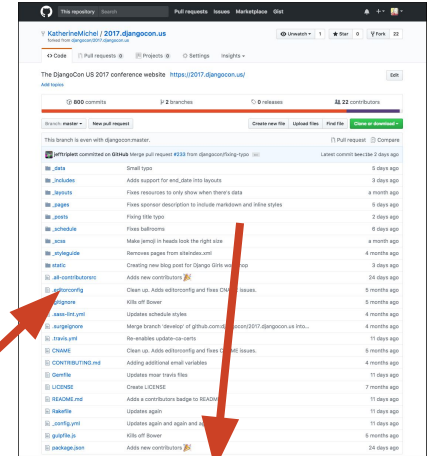Feature Branch

Feature Branch

Master Branch

Feature Branch

Pull Request
Branch

How to Create a Branch

File on GitHub (README)
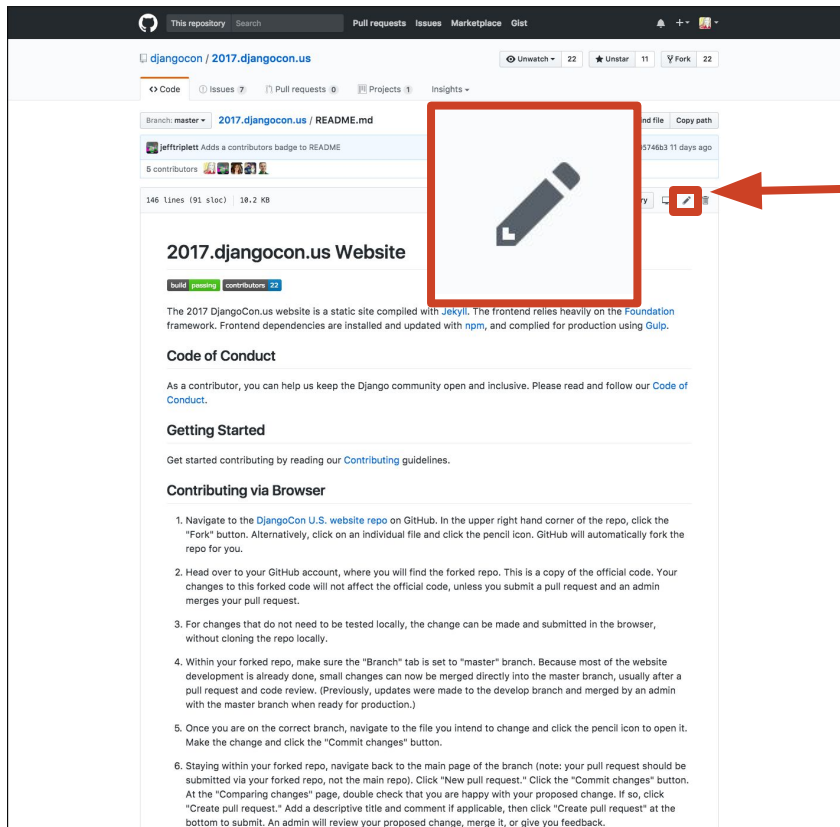
Editing File

Click pencil icon

Edit

Commit Info

Save
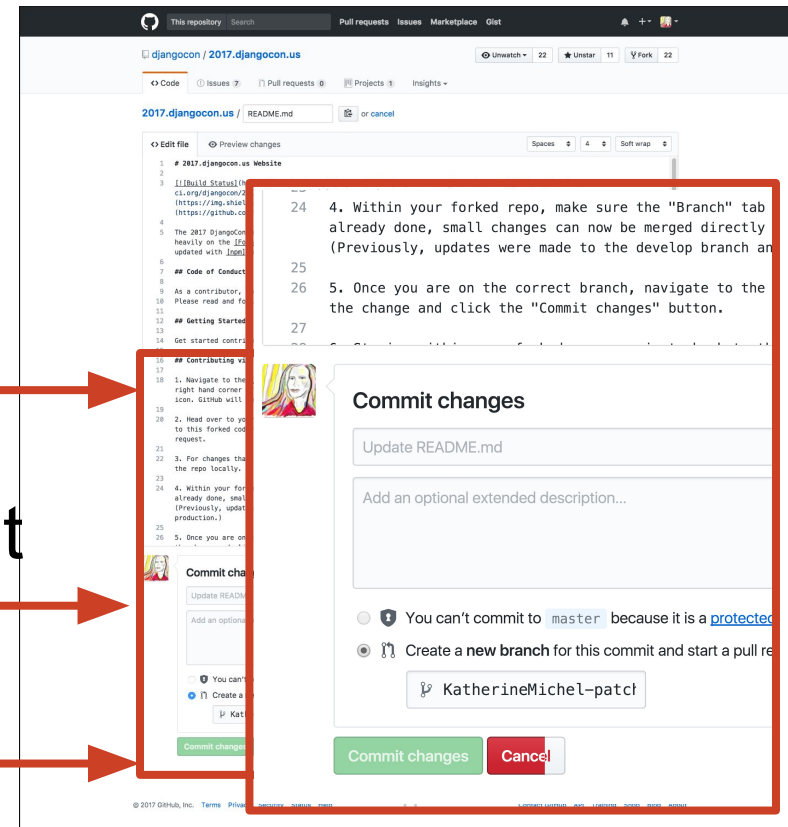
How to Create a Branch

File on GitHub (README)

Editing File
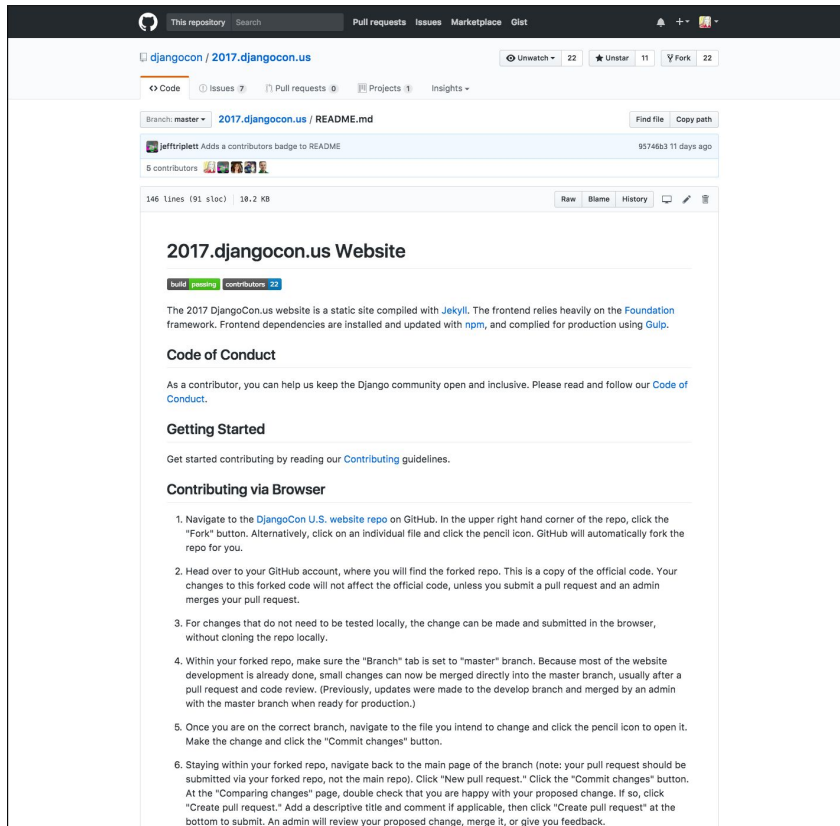
New branch and name
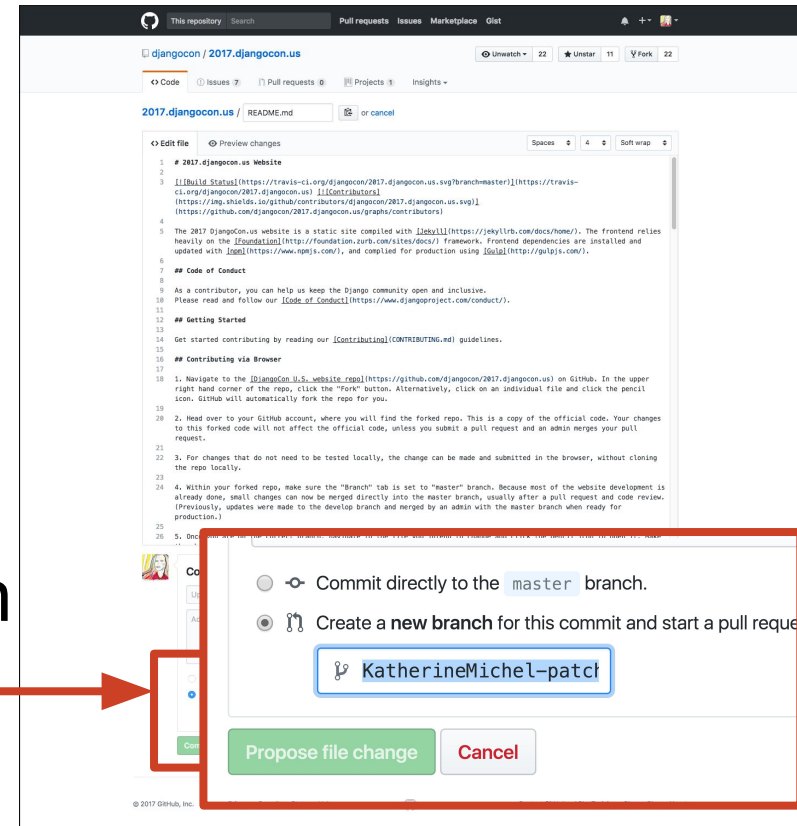
Overview

- Determine which collaboration approach to use
  - Clone a repo into our local development environment, create a feature branch, make a change, push the branch to the GitHub repo, and submit a pull request
- Review the two different types of pull request branches as a maintainer
  - Recommendations

Section 1

- Determine which collaboration approach to use

The Two Collaborative Development Models

- "Shared Repository" Model
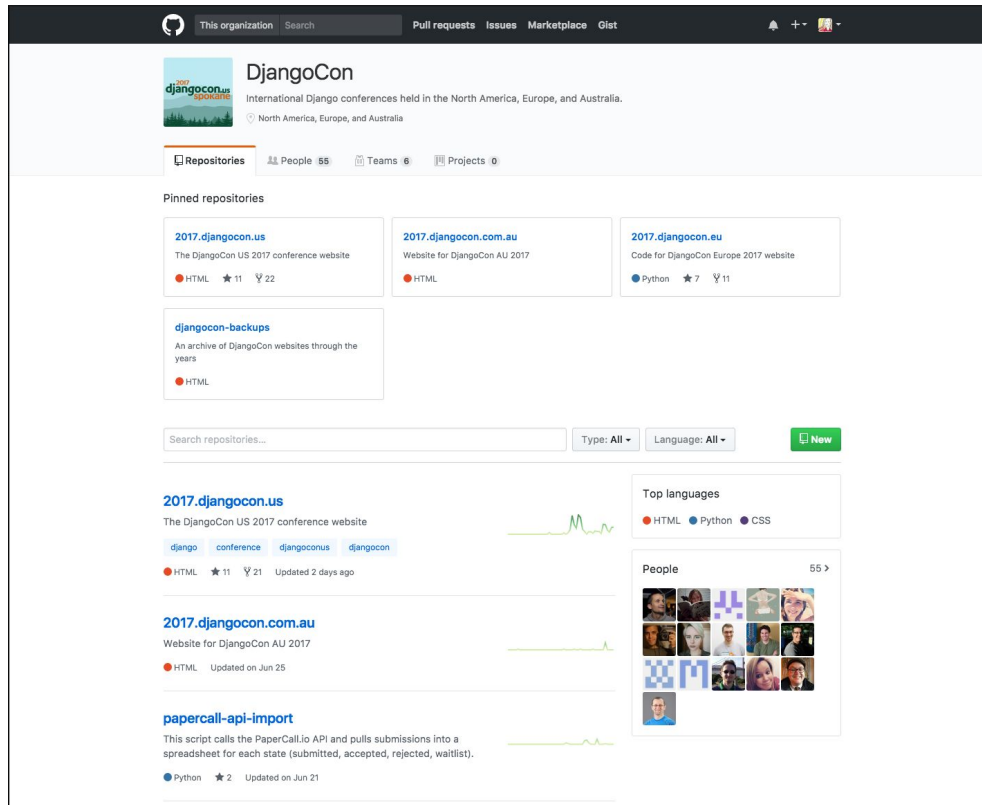- "Fork and Pull" Model

The two different models typically correspond to the two different account types and which model you use depends on whether you have write permission to the repo
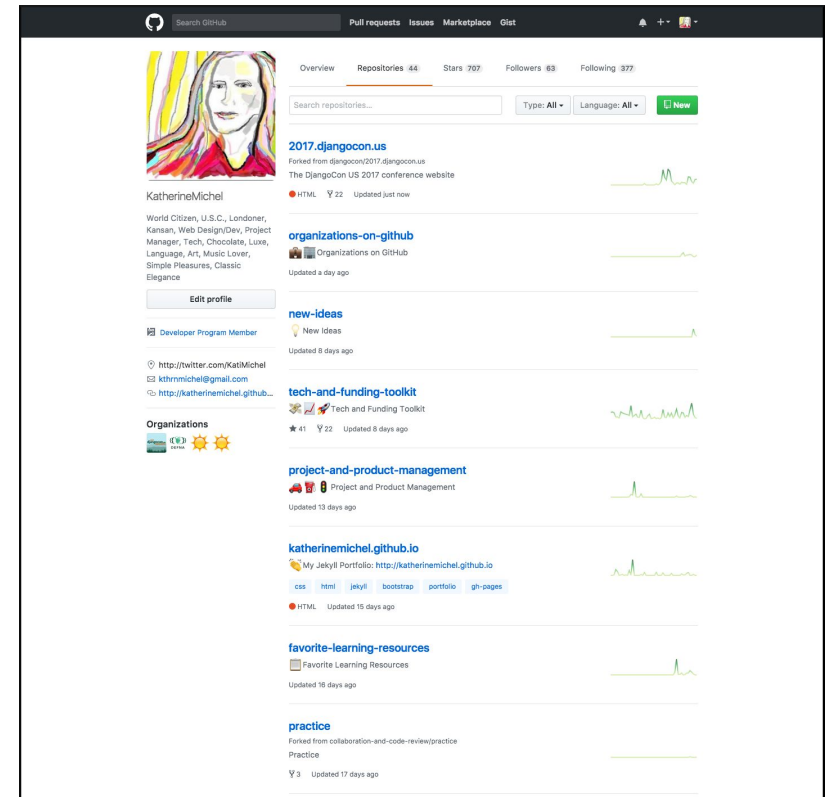
Two Types of Accounts

Organization Account

User Account

Write Permission

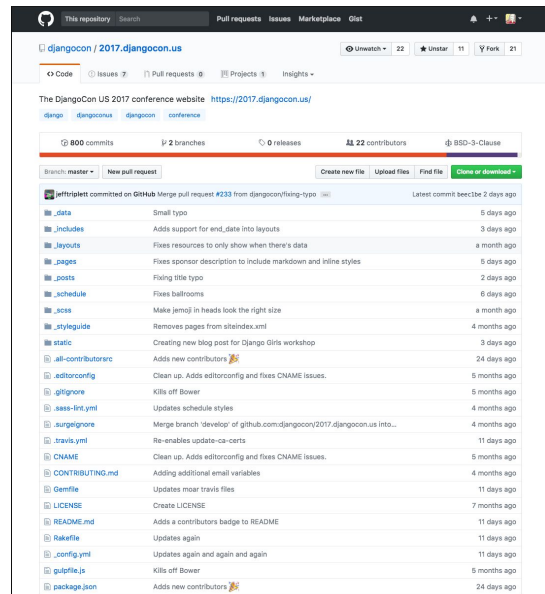When a user has write permission to a repo, it means they can make changes directly inside of the repo

Write Permission and Collaboration Examples

Shared Repository

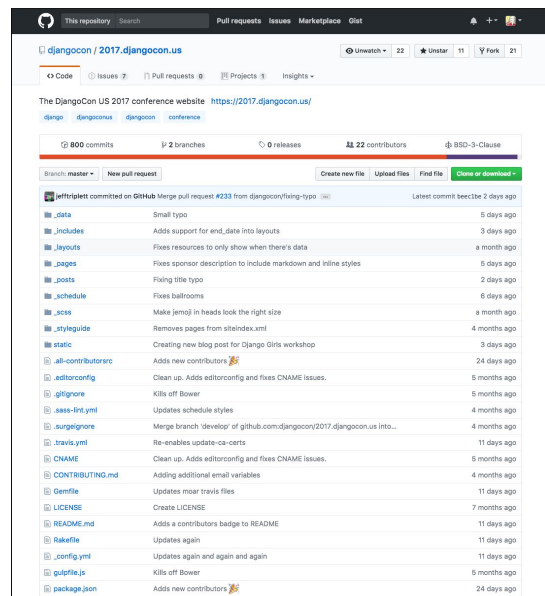DjangoCon US Website Repo
Write Permission: Maintainers

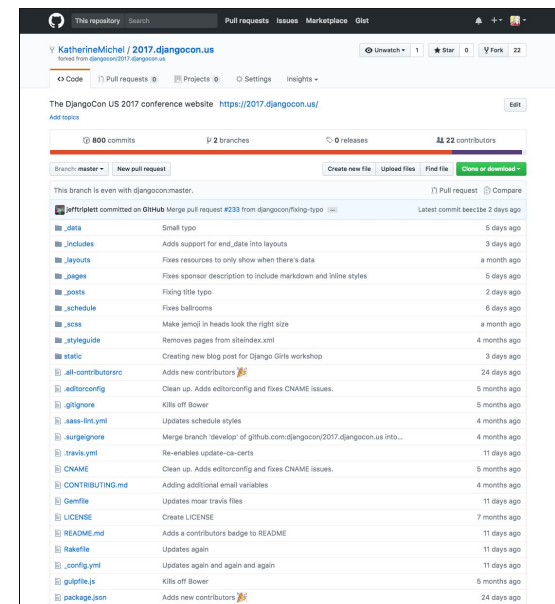# Write Permission and Collaboration Examples

Shared Repository

DjangoCon US Website Repo

Write Permission: Maintainers

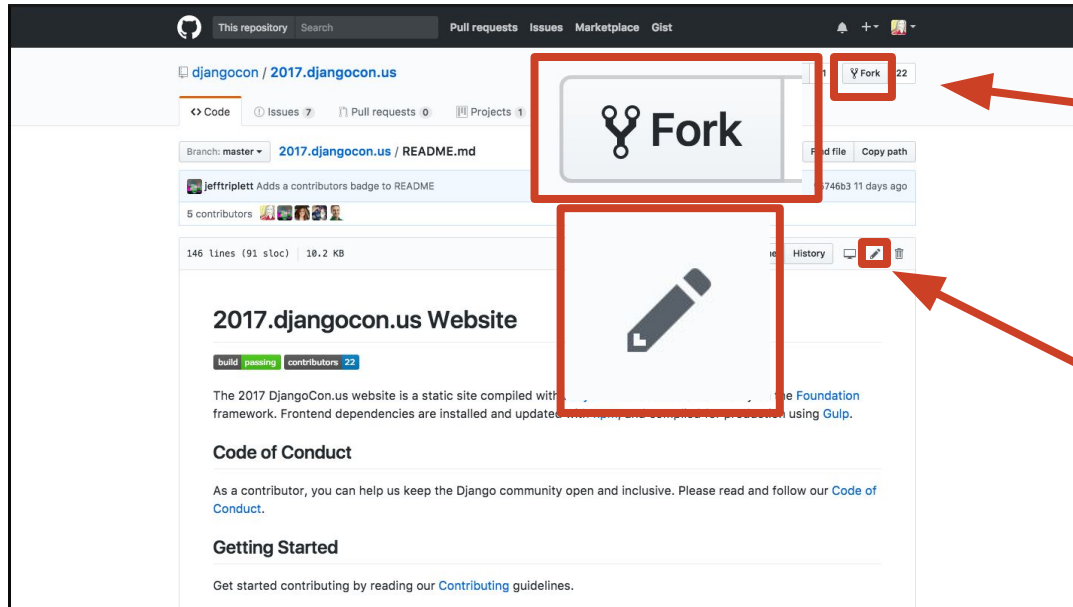## Fork and Pull

Fork

Write Permission: User

Fork

Submit
Pull Request

How to Fork a Repo

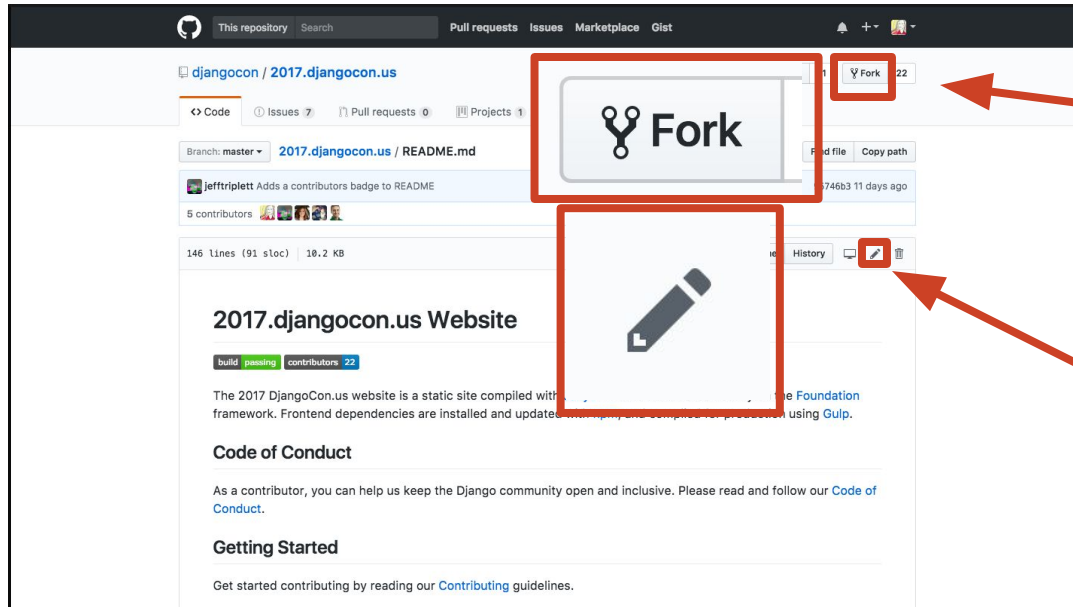DjangoCon US Website Repo

Click the "Fork" button

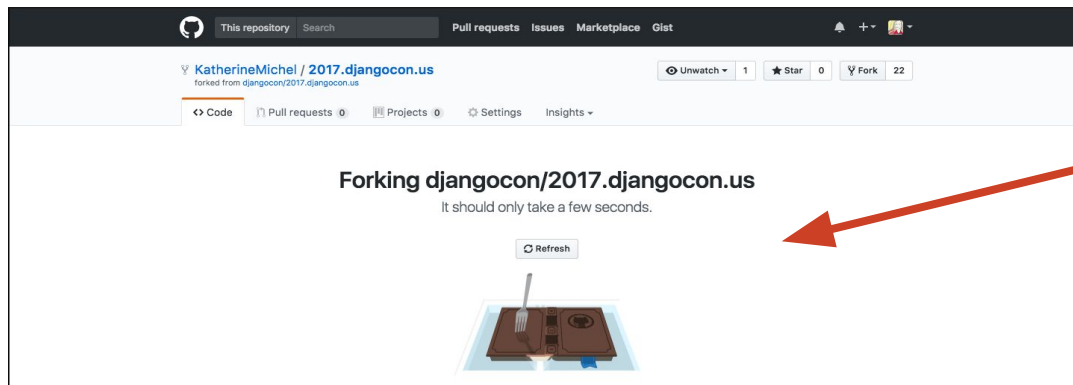Click pencil icon to try to edit a file you don't have write permission to

How to Fork a Repo

DjangoCon US Website Repo

Click the "Fork" button

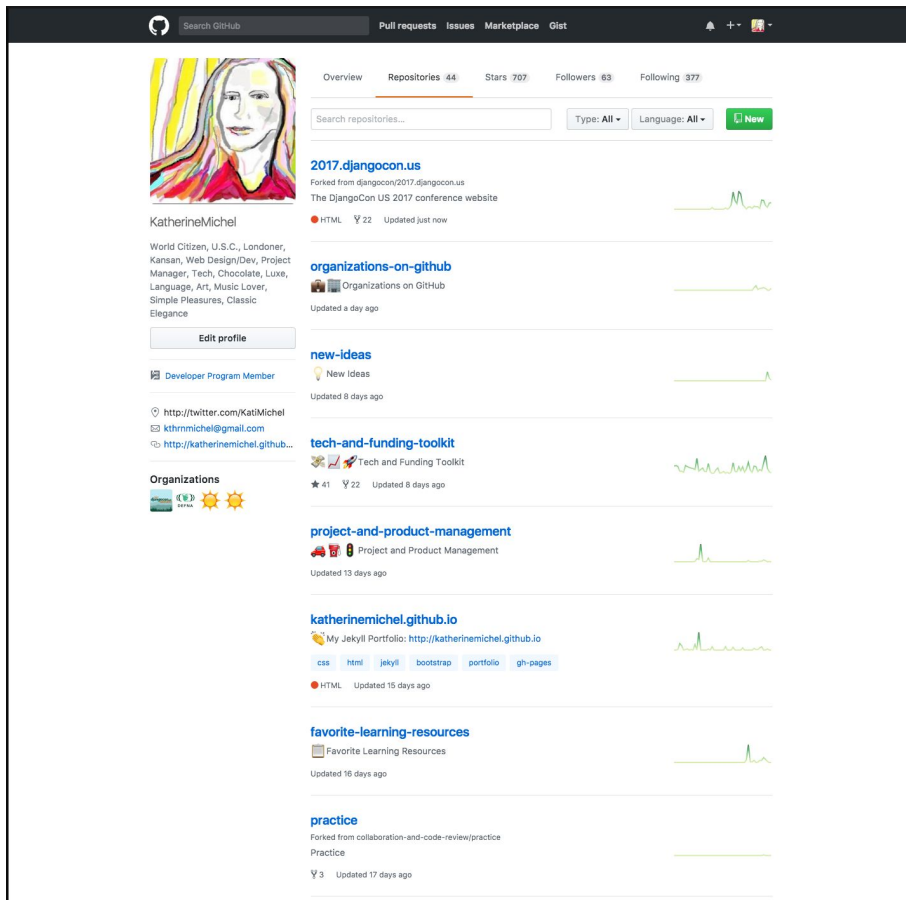Click pencil icon to try to edit a file you don't have write permission to

GitHub will notify you that the repo is being forked

Forked Repo

User Account

www.github.com/KatherineMichel

Forked Repo

User Account

www.github.com/KatherineMichel

Forked Repo

User Account   Forked Repo

www.github.com/KatherineMichel/2017.djangocon.us

Section 2

- Determine which collaboration approach to use
  - Clone a repo into our local development environment, create a feature branch, make a change, push the branch to the GitHub repo, and submit a pull request

Fork and Pull Model

DjangoCon US Website Repo

Fork and Pull Model
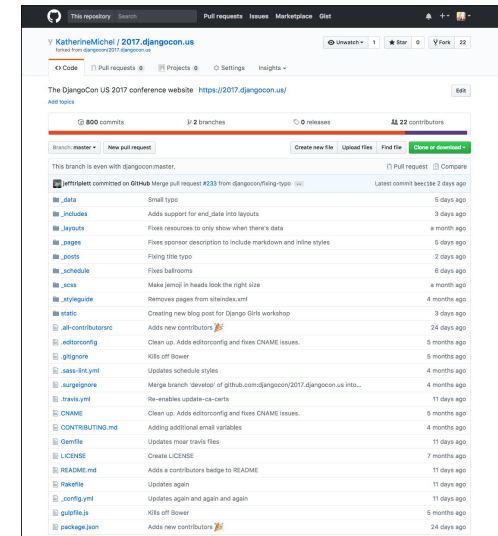
DjangoCon US Website Repo

My Fork

Fork

Fork and Pull Model

DjangoCon US Website Repo

My Fork

Remote origin

Fork

Clone

Local Development Environment

Fork and Pull Model

DjangoCon US Website Repo

My Fork

Remote origin

Fork

Clone

Local Development Environment

Push changes, including branches

Fork and Pull Model

DjangoCon US Website Repo

My Fork

Remote origin

Fork

Submit
Pull Request

Clone

Local
Development
Environment

Push
changes,
including
branches

Shared Repository

DjangoCon US
 Website Repo

Shared Repository

DjangoCon US
Website Repo

Remote origin
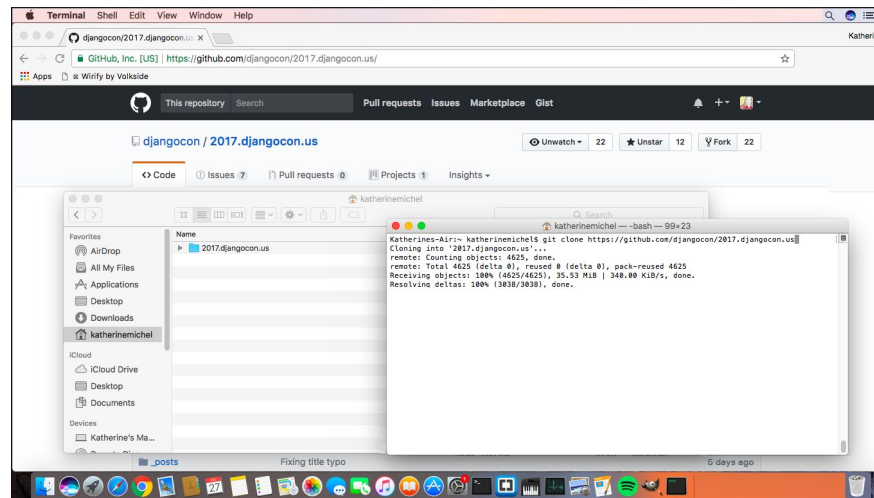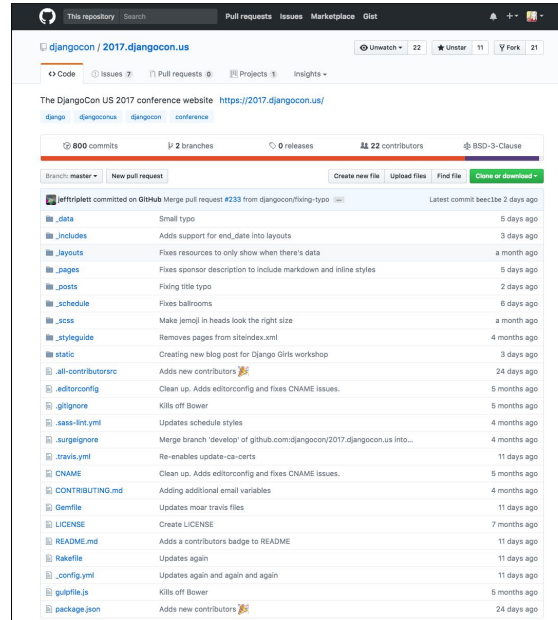
Clone

Local
Development
Environment

Shared Repository

DjangoCon US
Website Repo

Remote origin

Clone

Push
branches,
submit
pull requests

Can push
or pull
because of
write
permission,
No fork
needed

Local
Development
Environment

Local Dev Environment

Working within home directory: katherinemichel
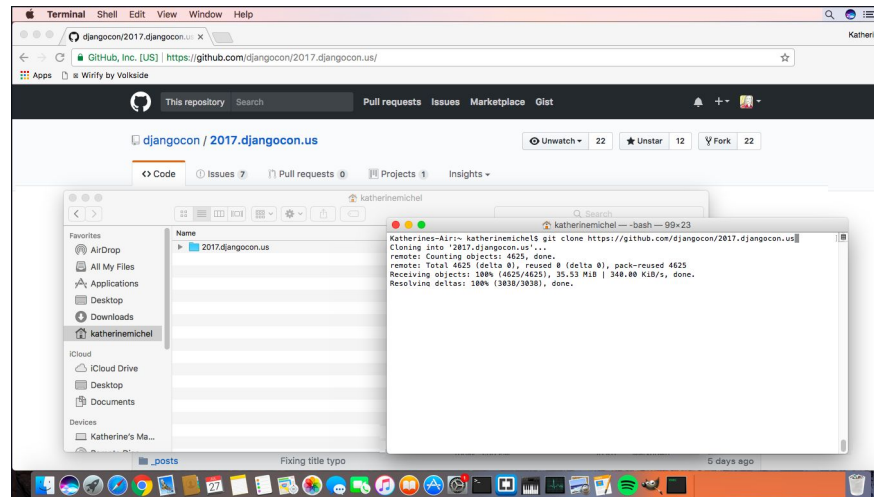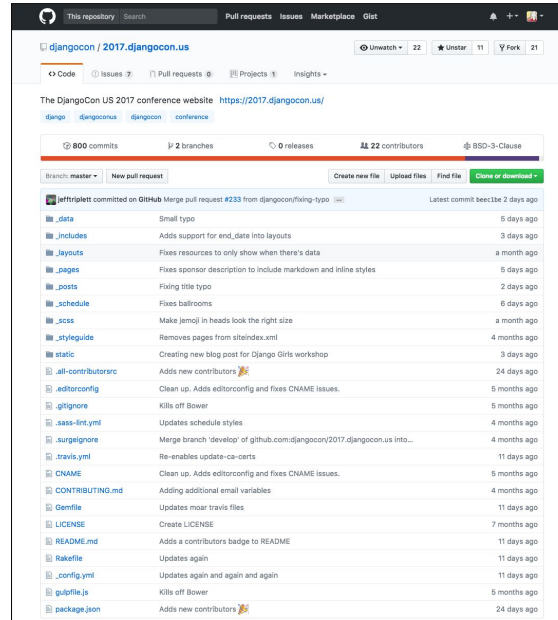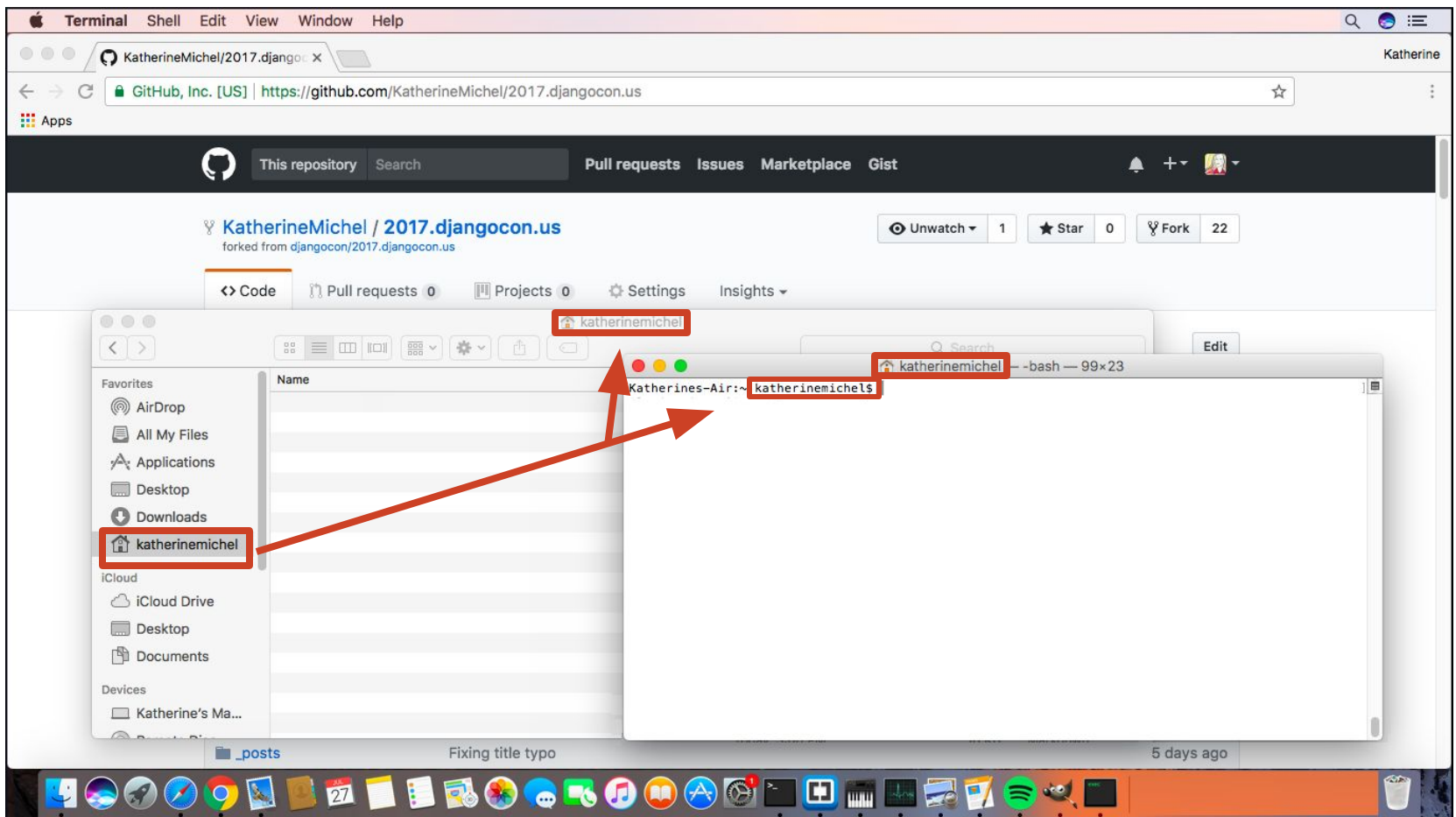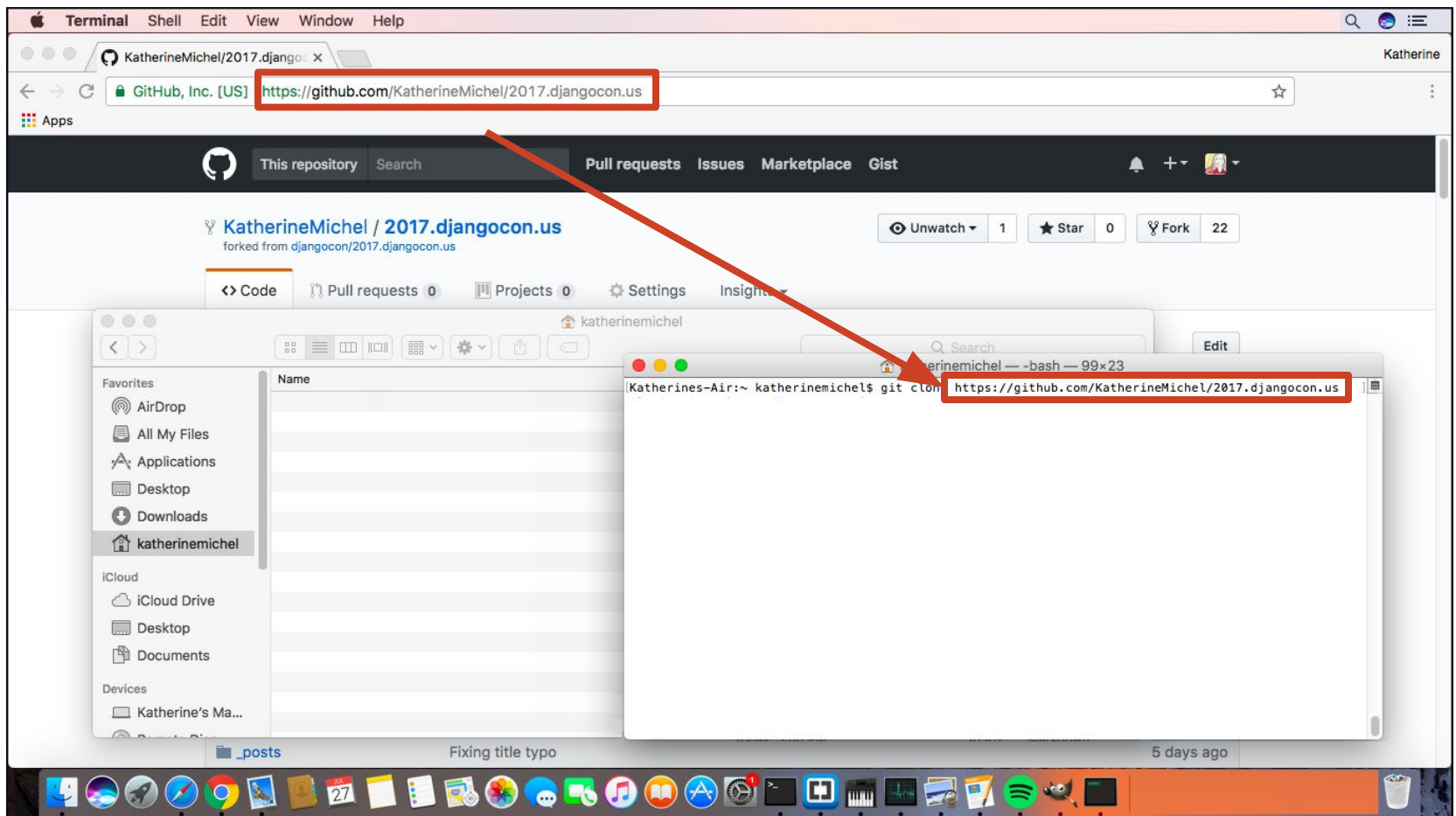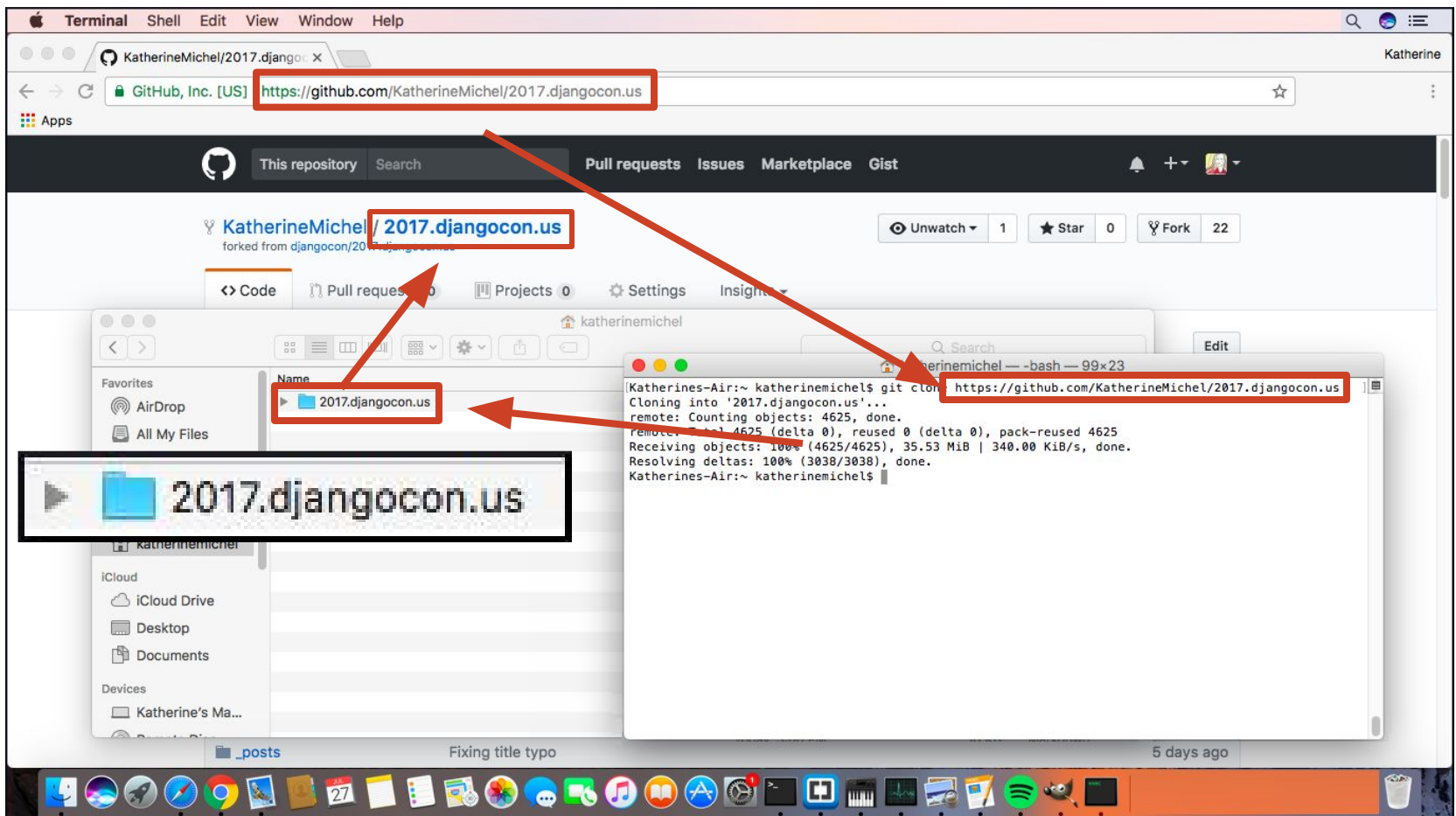
Cloning

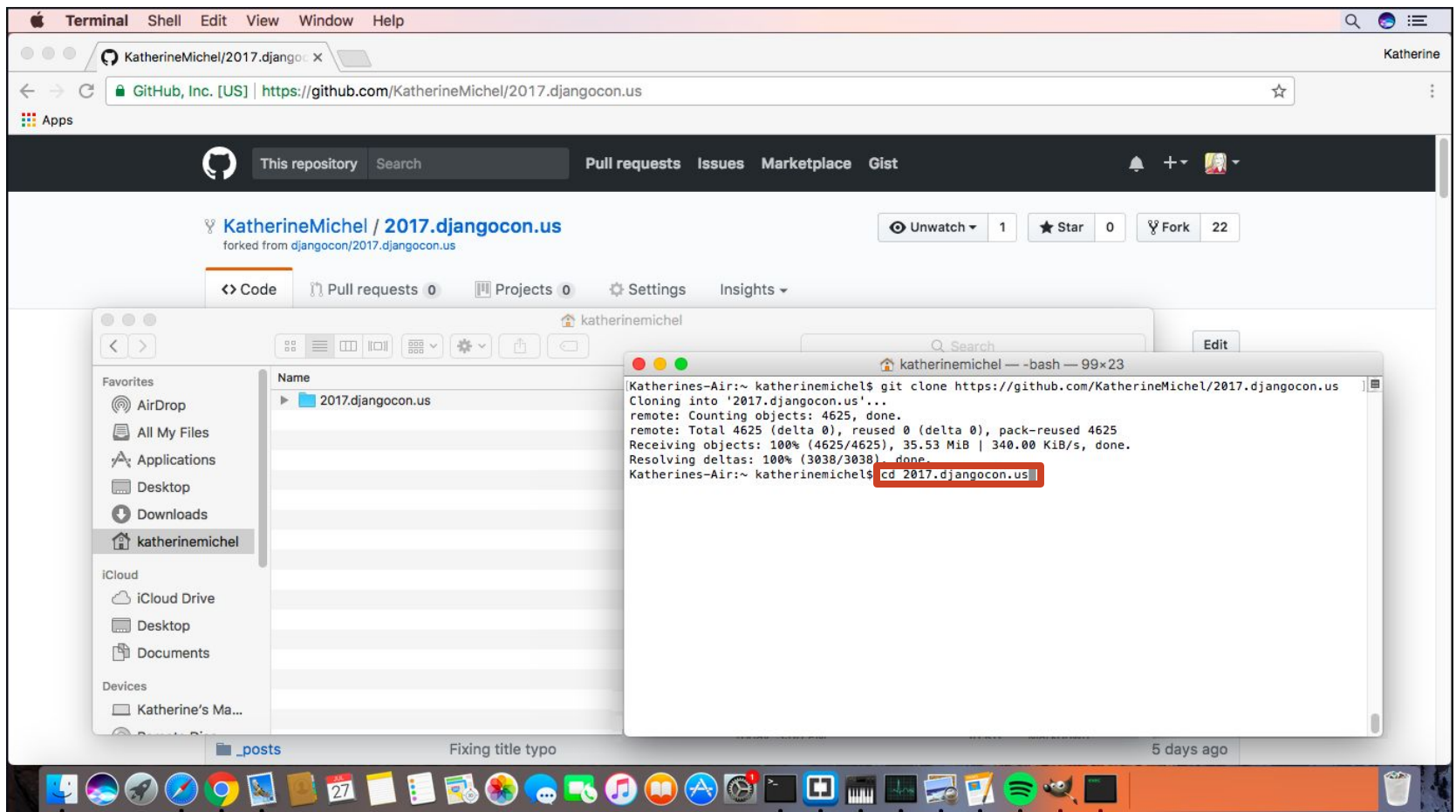$ git clone https://github.com/KatherineMichel/2017.djangocon.us

Cloning

$ git clone https://github.com/KatherineMichel/2017.djangocon.us

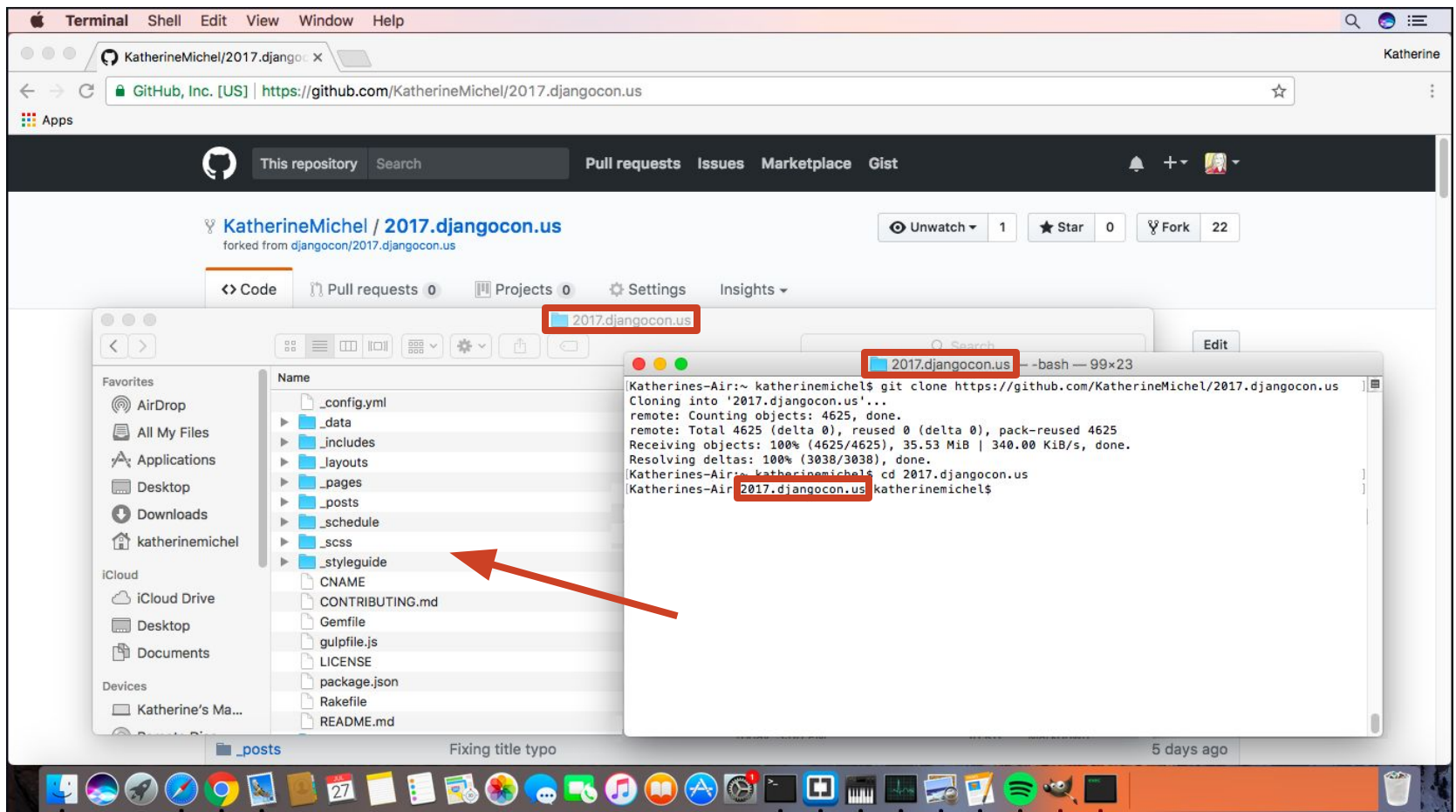# Changing Directory

$ cd 2017.djangocon.us

Changing Directory

Working within new directory: 2017.djangocon.us

GitHub Repo Versus Local Directory

DjangoCon US GitHub Repo

Verifying Branch

$ git branch
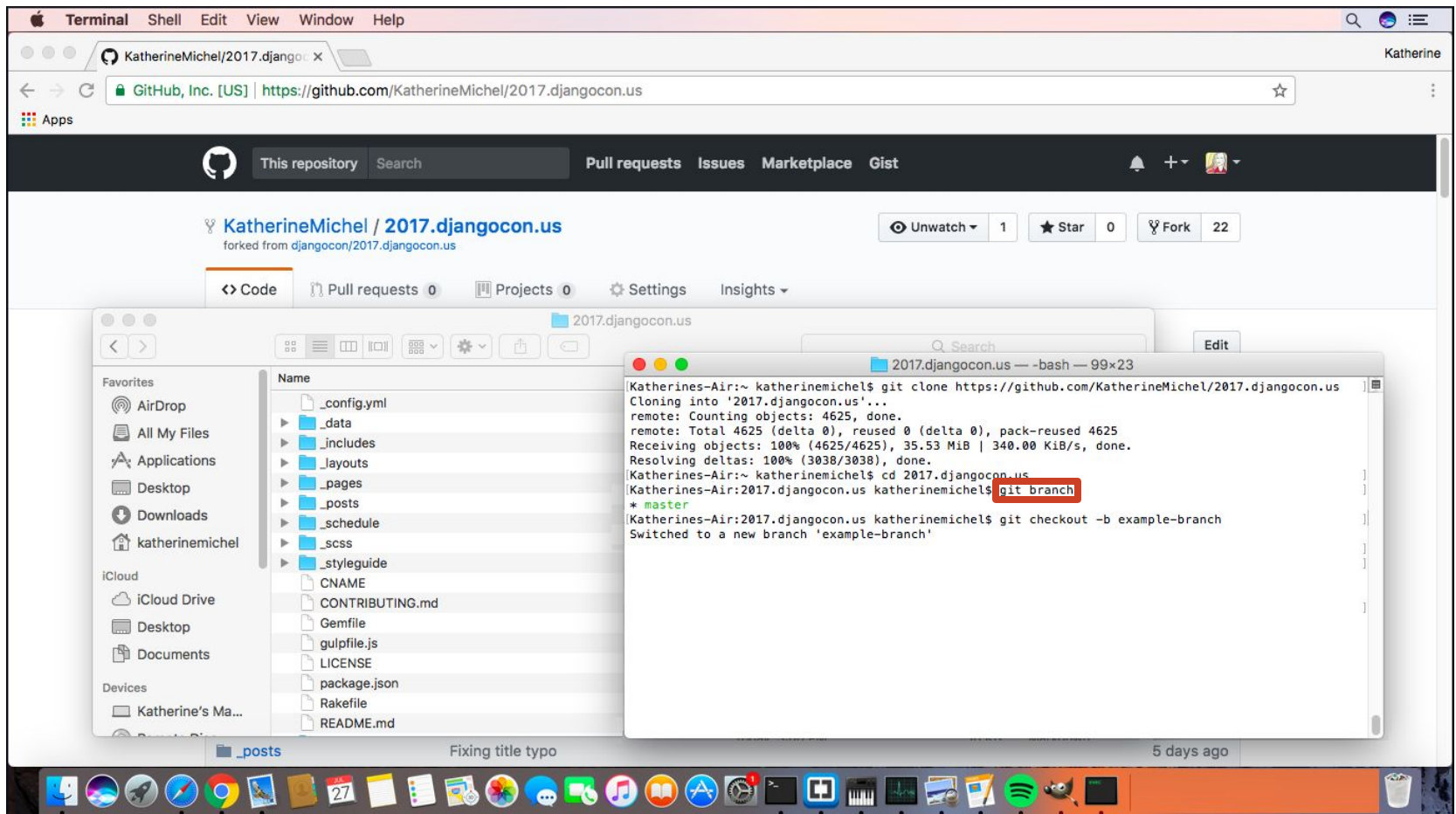
Creating a New Branch

$ git checkout -b example-branch

Working on a File Locally

Adding and Committing

$ git add .
$ git commit -m "Creating branch and updating"

Pushing

$ git push origin example-branch

New Branch
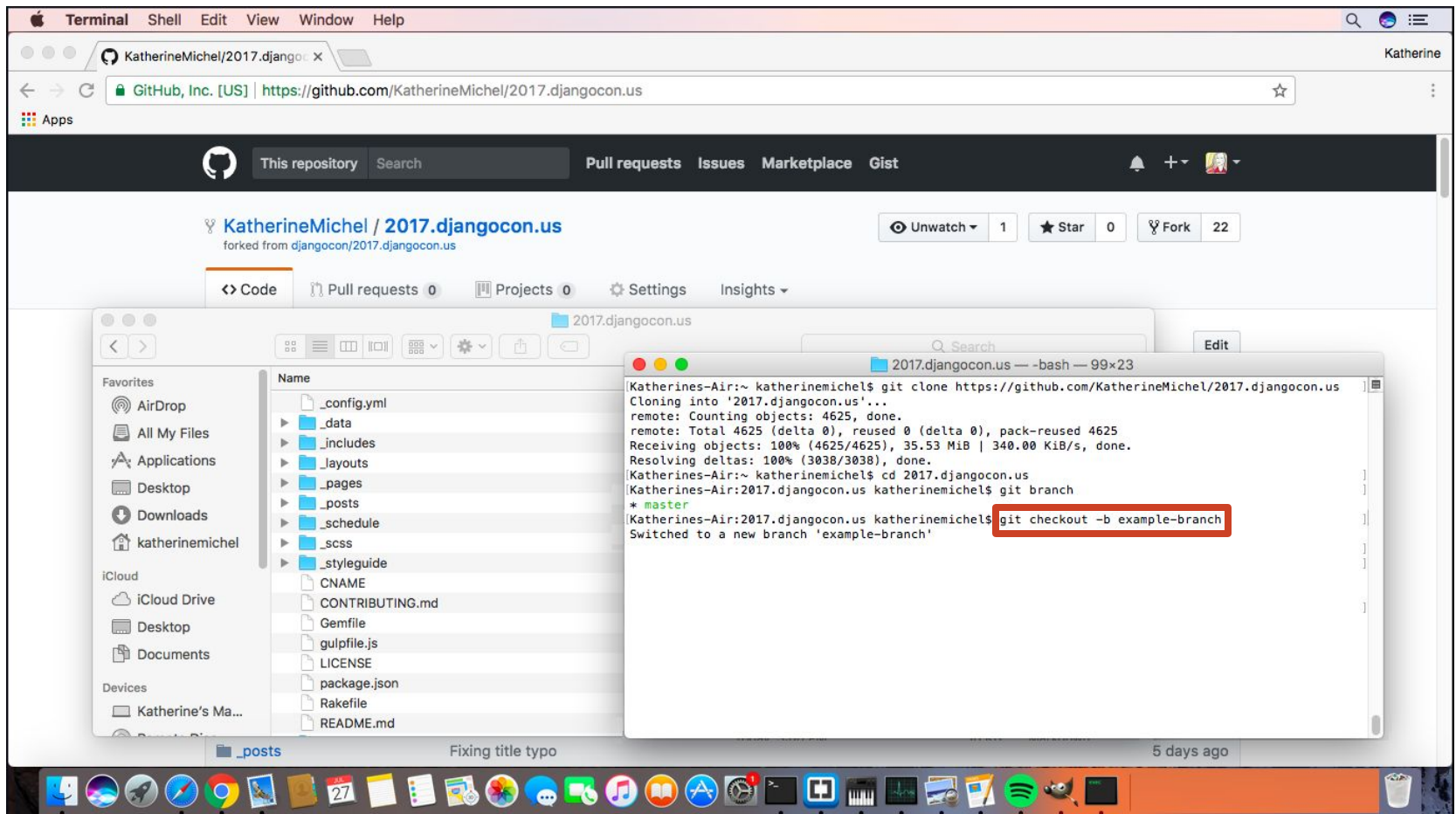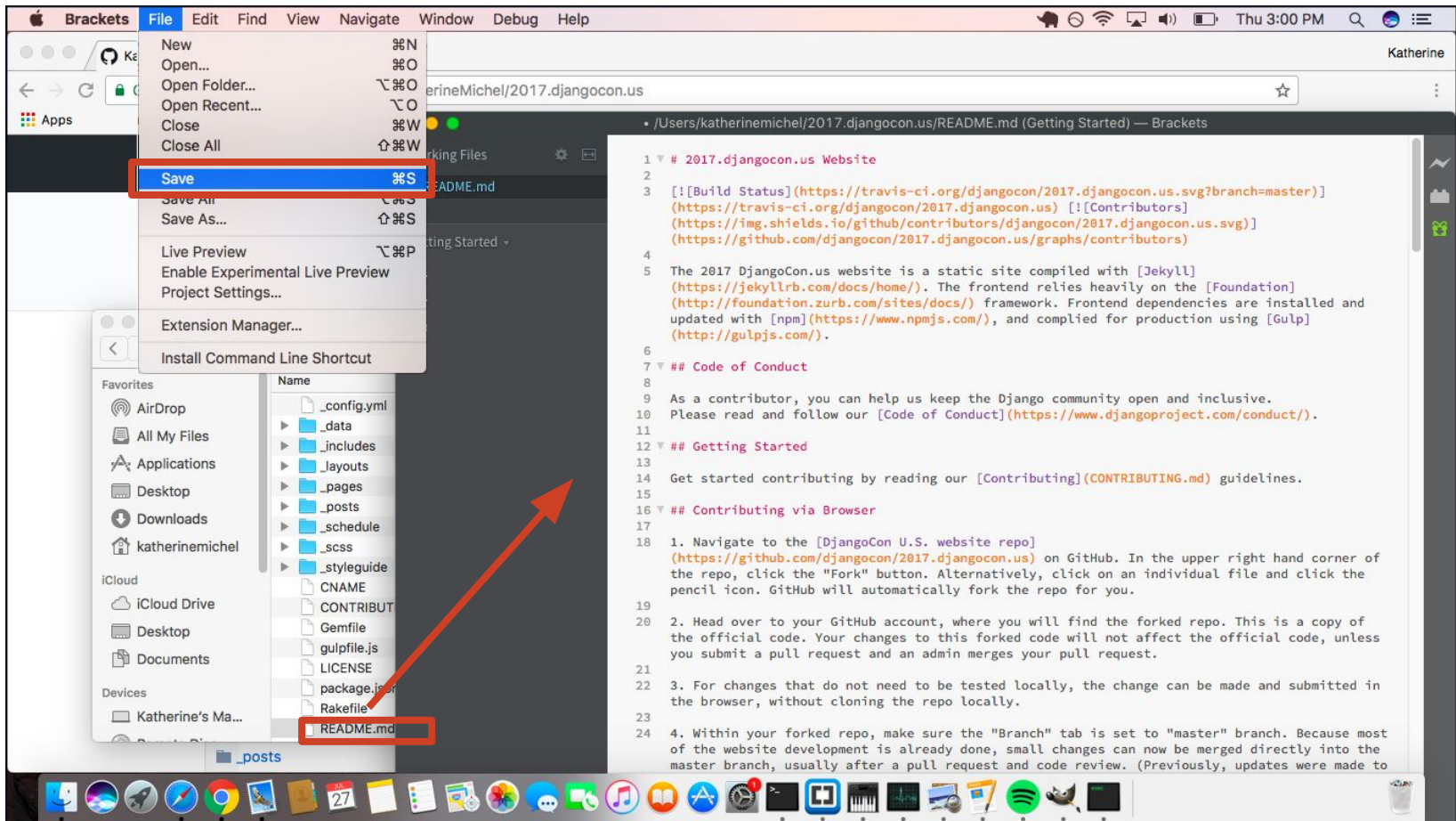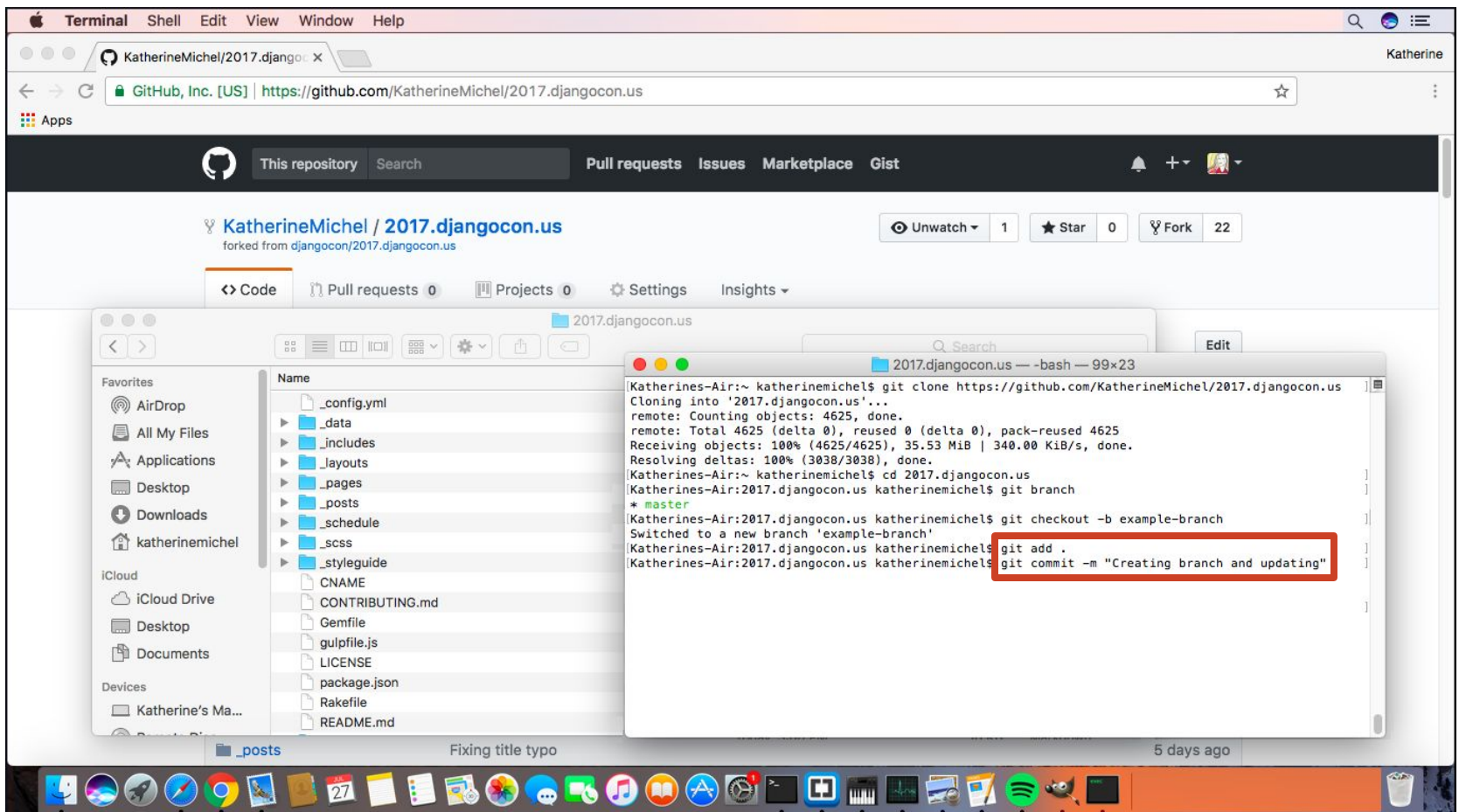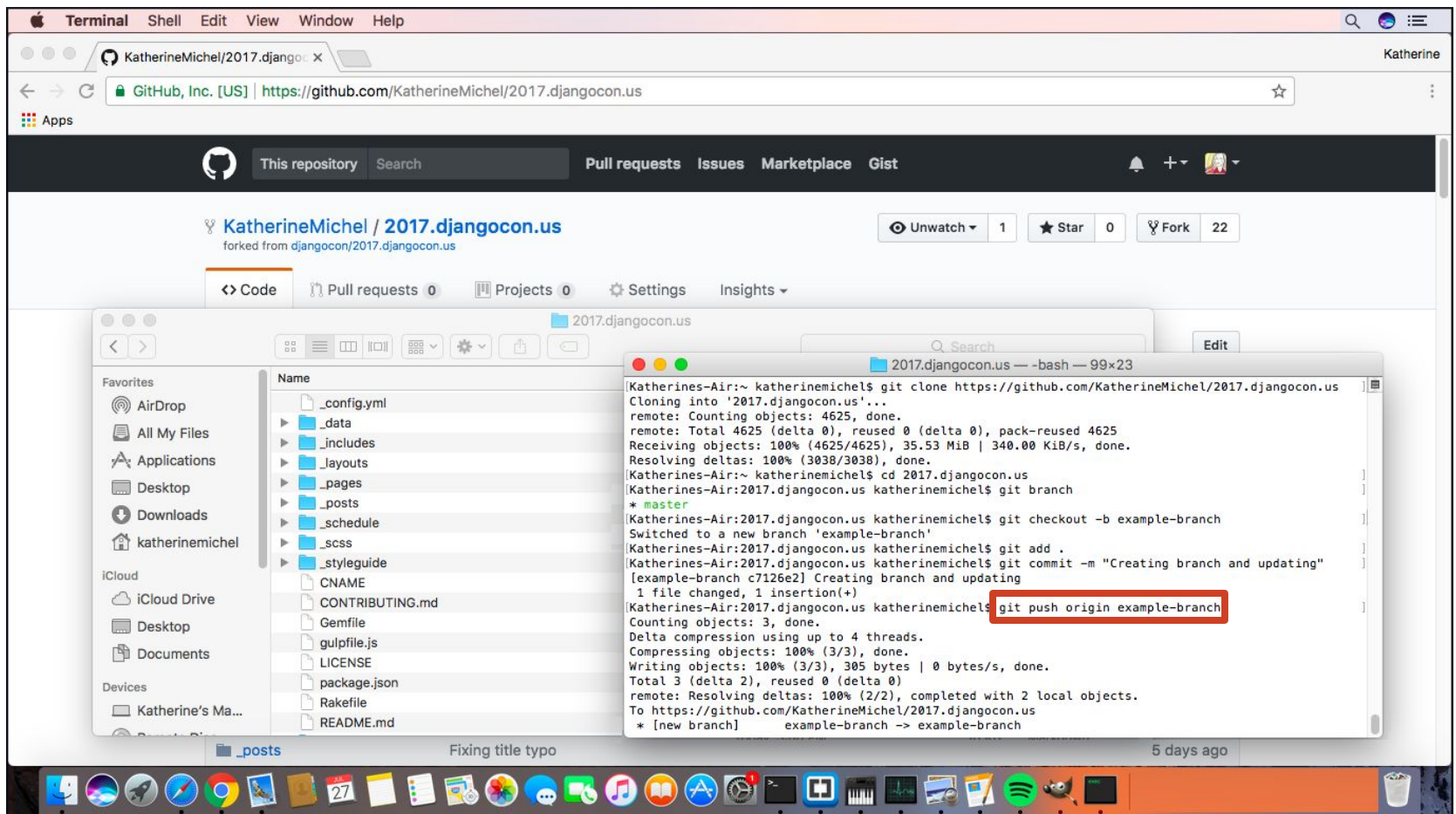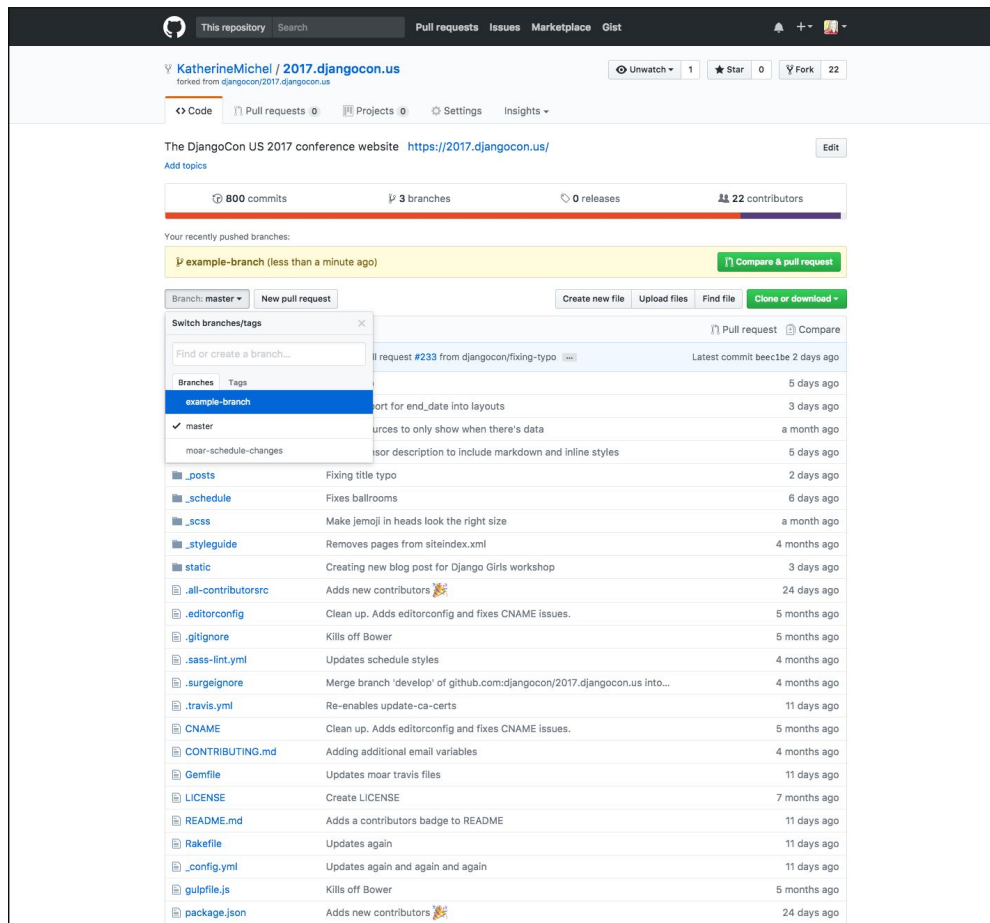
Branch Tab

New Branch

Branch Tab

New Branch

Submit a Pull Request

DjangoCon US Website Repo

Creating a New Pull Request

Submit a Pull Request

DjangoCon US Website Repo

Creating a New Pull Request

Choose branches

Title and Description

Create

Look over changes

Section 3

- Determine which collaboration approach to use
  - Clone a repo into our local development environment, create a feature branch, make a change, push the branch to the GitHub repo, and submit a pull request
- Review the two different types of pull request branches as a maintainer

Pull Request Review Process

Pull Request Review Process

Look over changes

Title and Description

Command line instructions

Merge button

Pull Request Command Line

Merge pull request ▼

command line instructions.

Merge pull request ▼ You can also open this in GitHub Desktop or view command line instructions.

Merging via command line

If you do not want to use the merge button or an automatic merge cannot be performed, you can perform a manual merge on the command line.

HTTPS | Git | Patch | https://github.com/KatherineMichel/2017.djangocon.us.git

Step 1: From your project repository, bring in the changes and test.

```
git fetch origin
git checkout -b example-branch origin/example-branch
git merge master
```

Step 2: Merge the changes and update on GitHub.

```
git checkout master
git merge --no-ff example-branch
git push origin master
```

Pull Request Review Options

Do no run code locally

Run code locally

Click merge

Make change in browser then click merge

Pull Request Review Options

Do no run code locally

Run code locally

You update and merge locally, push to GitHub

Click merge

Make change in browser then click merge

No change needed, go back to browser and click merge

Ask pull request author to make change

You push change to pull request, then click merge
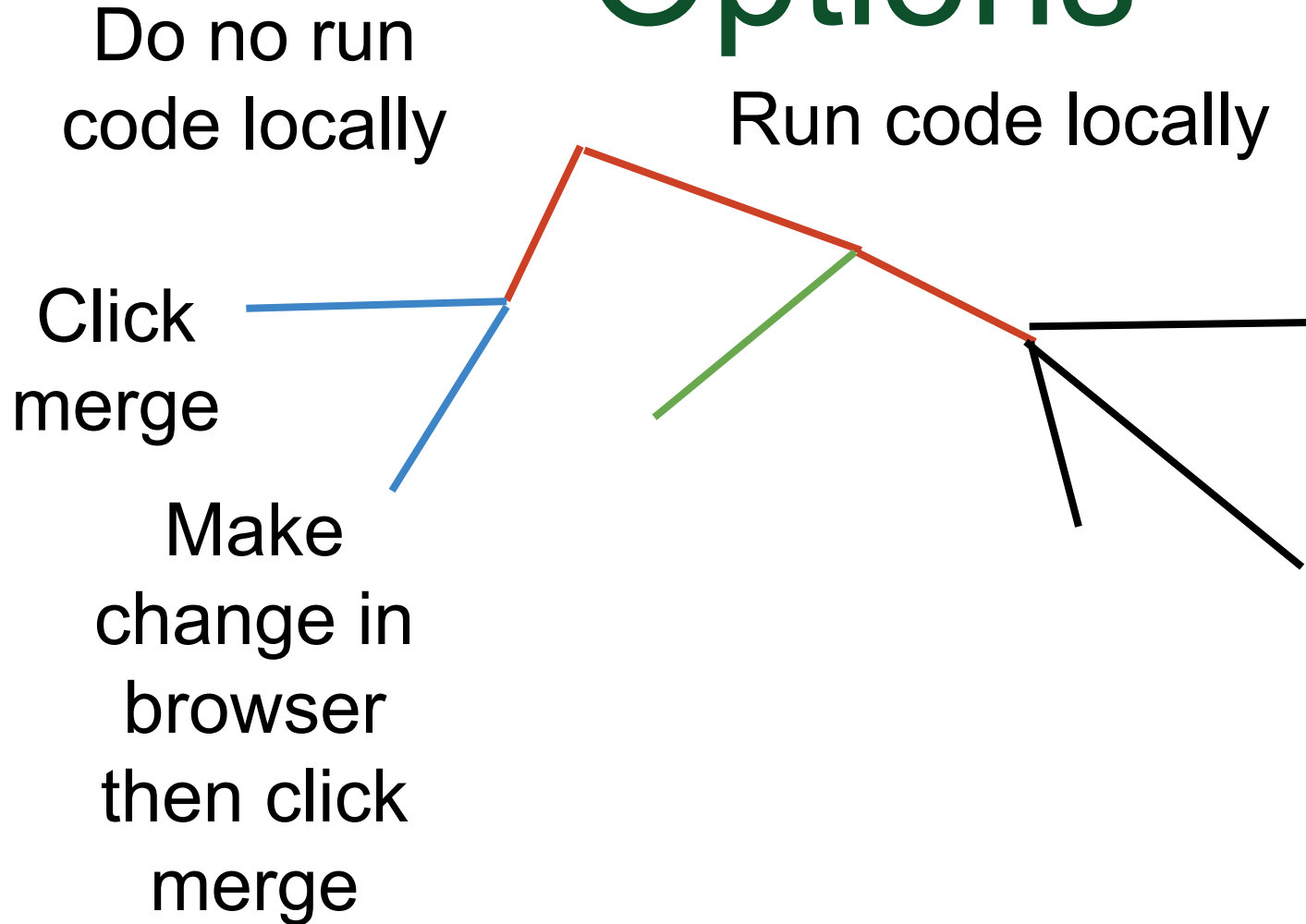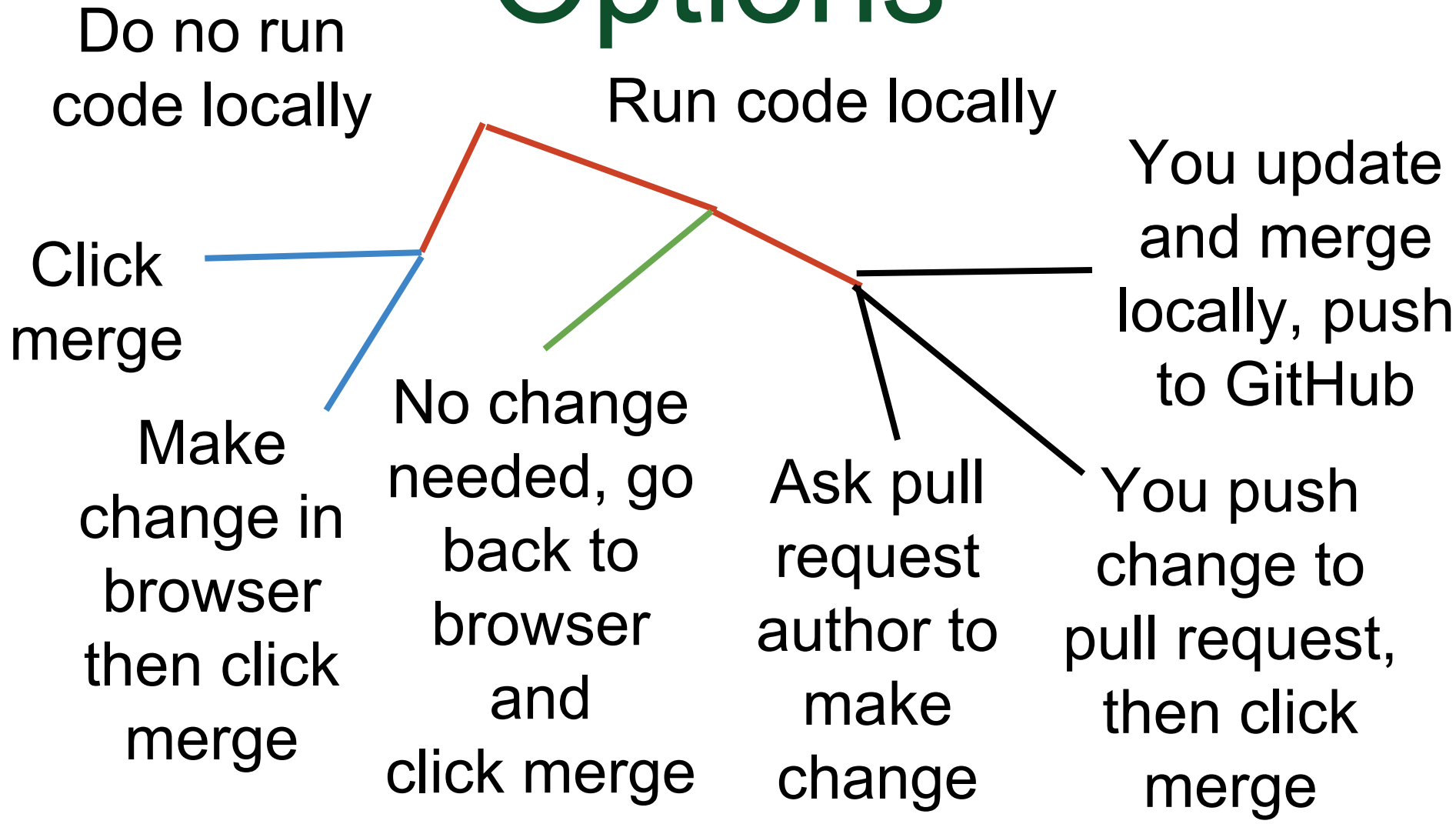
Pull Request Branches

DjangoCon US Website Repo

Fork

remote origin

git fetch origin

pull request branch

pull request branch

<remote>/<branch>

git pull <url> <branch>

.git folder

local branches

Write permission granted to pull request as maintainers

updates
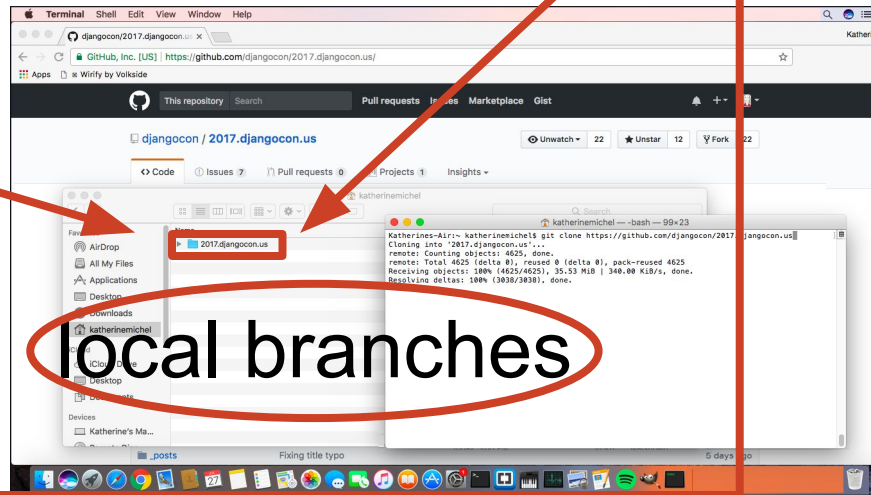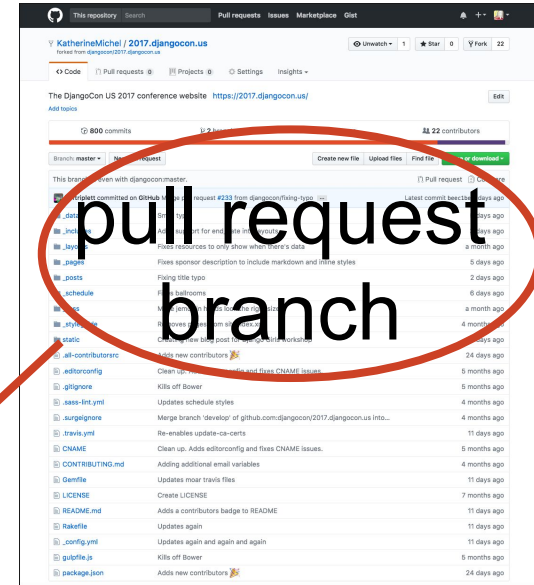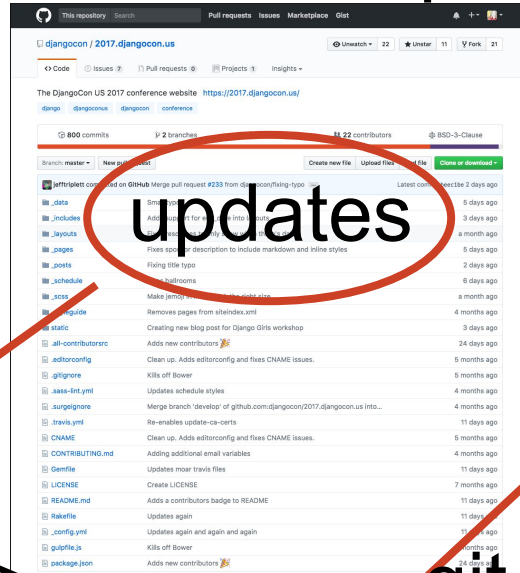
Pull Request Review

Shared Repo Branch

```
$ git fetch origin
$ git checkout -b <branch-name> origin/<branch-name>
$ git merge master


$ git push origin <branch-name>
```

Forked Repo Branch

```
$ git checkout -b <branch-name> master
$ git pull https://github.com/<user-name>/<repo-name> <branch-name>


$ git push https://github.com/<user-name>/<repo-name> <branch-name>
```

Merge Pull Request Locally and Push to Master Branch

```
$ git checkout master
$ git merge --no-ff <branch-name>
$ git push origin master
```

Section 4

- Determine which collaboration approach to use
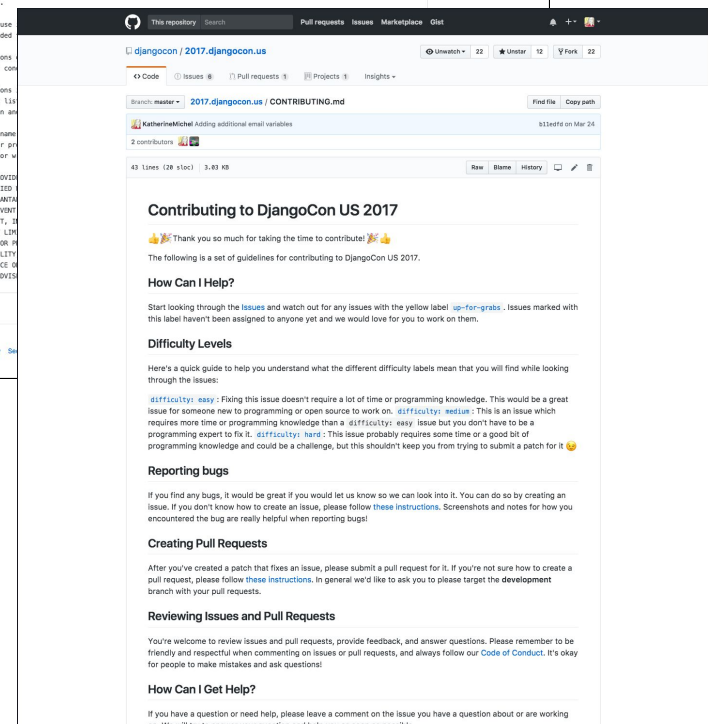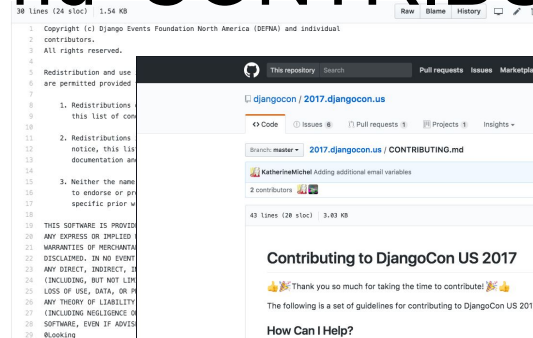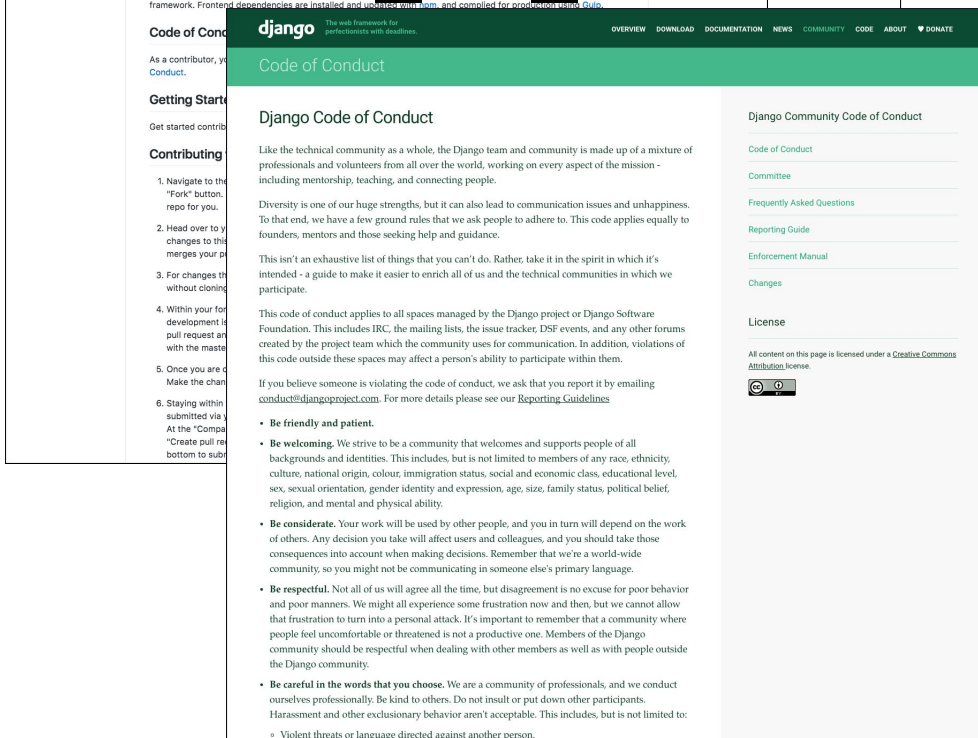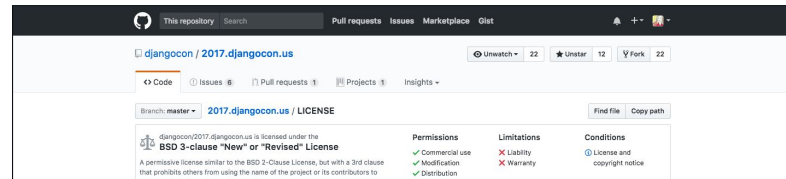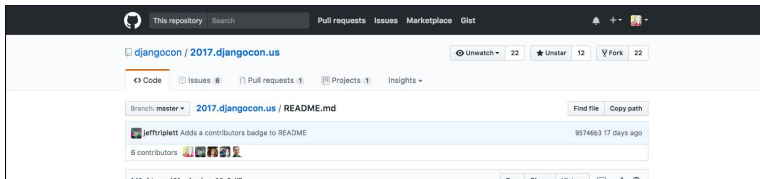 - Clone a repo into our local development environment, create a feature branch, make a change, push the branch to the GitHub repo, and submit a pull request
- Review the two different types of pull request branches as a maintainer
  - Recommendations

Documentation

README.md

LICENSE

CODE_OF_CONDUCT.md CONTRIBUTING.md

Code Safety

Tips for Getting Started

Triaging Labels
- help wanted
  - first-timer
  - good first contribution
- mentor available
- up-for-grabs

Sandboxing
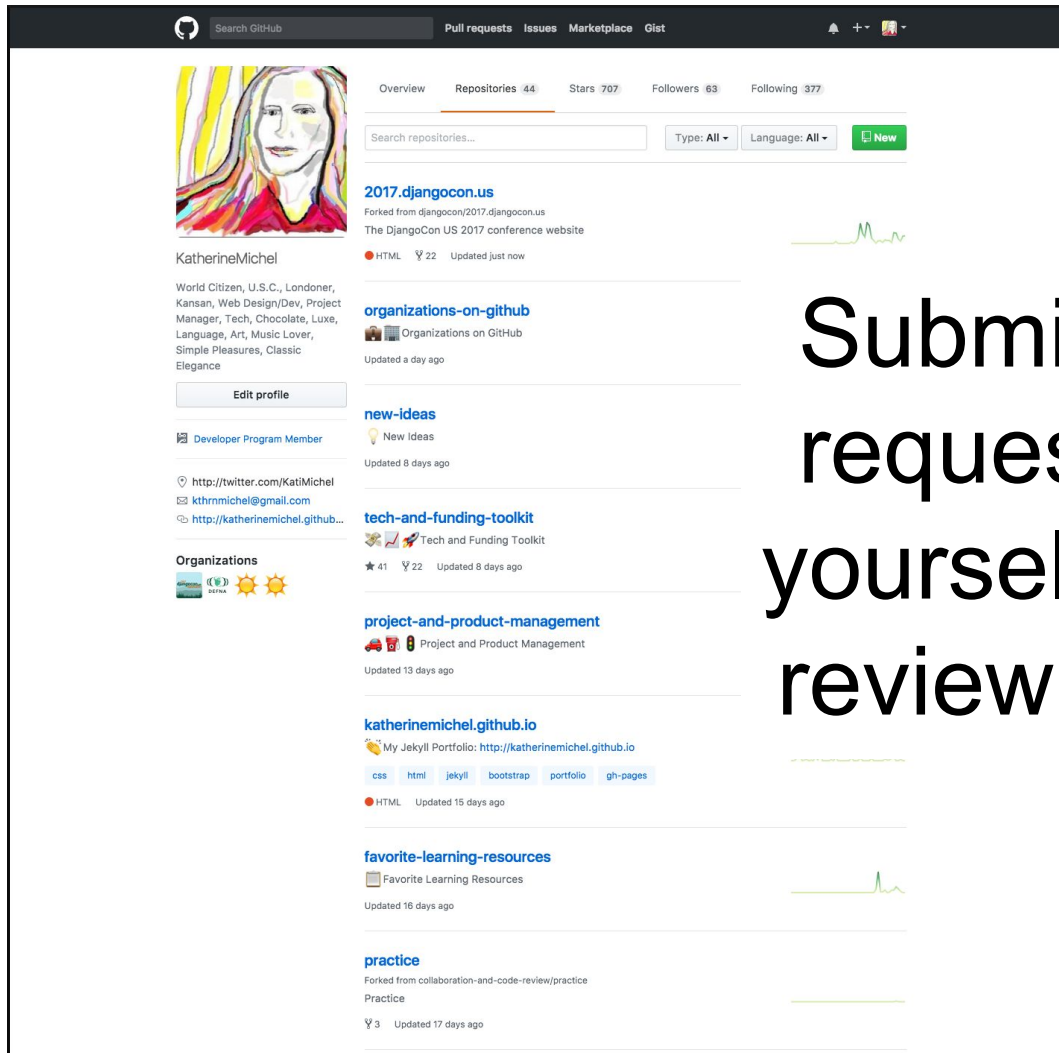
Submit pull requests to yourself then review them

Go For It!

Contribute to the DjangoCon US website next year! We have a diverse group of contributors of all skill levels and we are always looking for more contributors. Let us know if you are interested.

Thank You!

Useful Resources: https://git.io/v5fRh

Katherine "Kati" Michel
Twitter handle: @KatiMichel
GitHub username: KatherineMichel