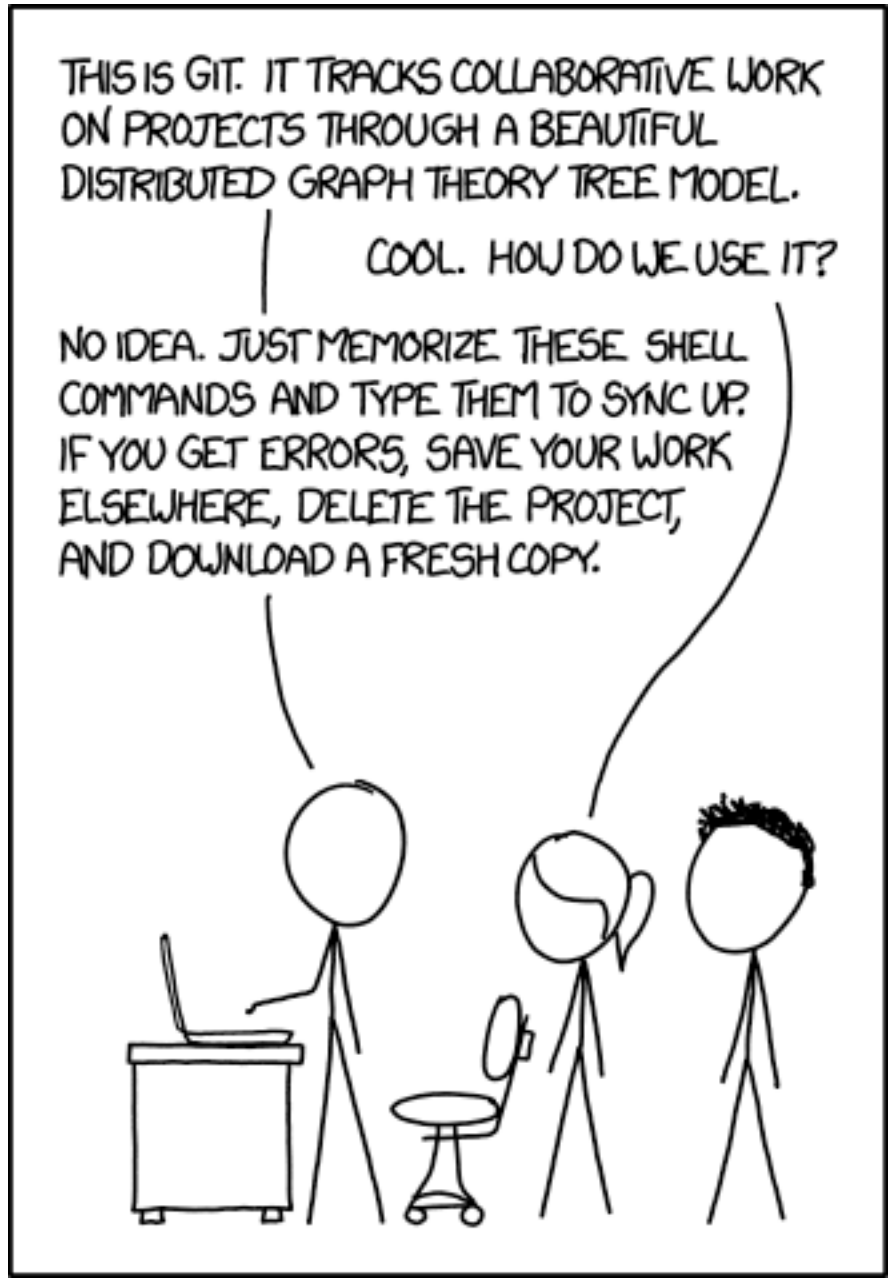
Bioinformatics Lessons Schedule

Date	Subject
10-22	Terminus
10-29	Server Basics
11-5	Server Basics, continued
11-12	Server Basics, continued
11-19	Server Basics, continued
11-26	No lesson, week of Thanksgiving
12-03	Server Basics, repeat of Week 5, continued
12-10	Skipped for symposium
12-17	Basic Git and GitHub
12-24	Christmas break
12-31	Christmas break
01-07	Resume lessons with RNA-seq

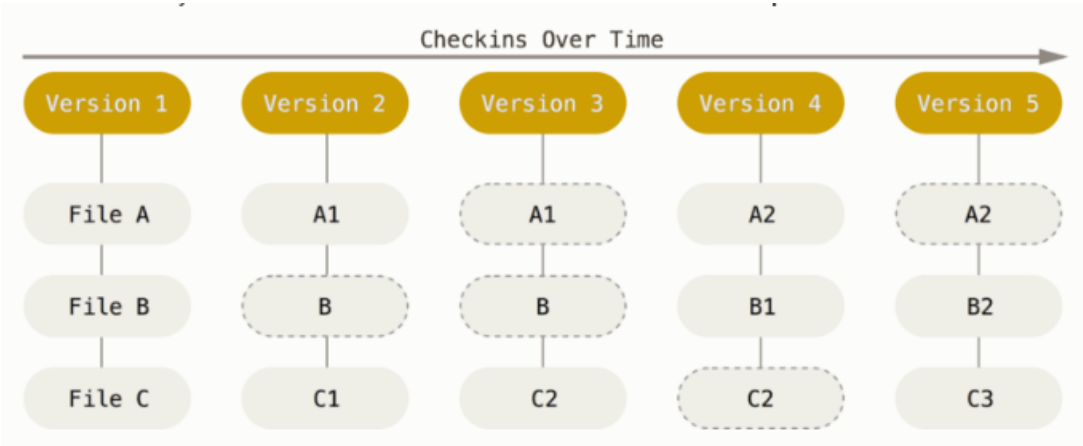
Git and GitHub

What is Git?


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.

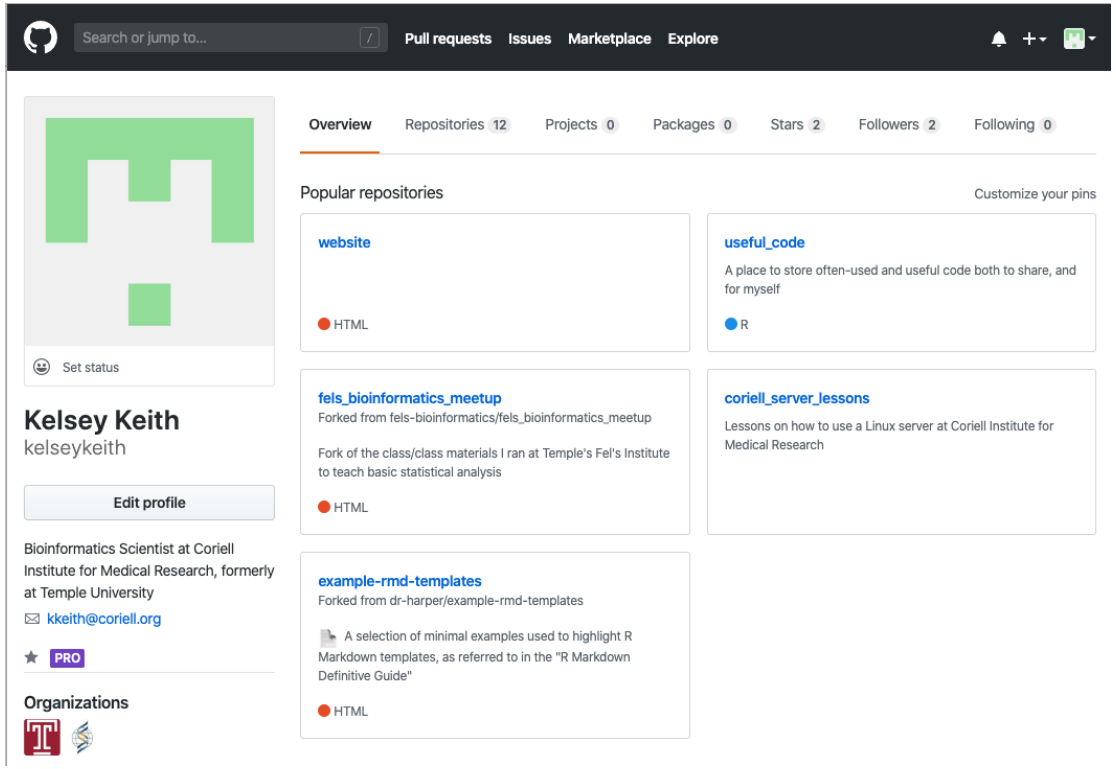
Git is a Version Control System

- Git works as a series of snapshots of a file system
- Originally made for collaborative development of the Linux operating system
- Every time you save your project through Git, Git takes a picture of it, with links between the current and past versions of files
- If you need to undo a change you can go back to previous versions of the file



What do we use Git for?

- Documenting a project's progress
- Working with other people
- Reproducible research
- Skills advertisement (for the future)



Getting Started

Set Up Your `.gitconfig` Files

Set Up Your `.gitconfig` Files

1. Log onto the server

Set Up Your `.gitconfig` Files

1. Log onto the server
2. Set your username and email address

```
git config --global user.name “Your Name”
git config --global user.email youremail@coriell.org
```

Set Up Your `.gitconfig` Files

1. Log onto the server
2. Set your username and email address

```
git config --global user.name "Your Name"
git config --global user.email youremail@coriell.org
```

3. Set your default text editor

```
git config --global core.editor nano
```

Set Up Your `.gitconfig` Files

1. Log onto the server
2. Set your username and email address

```
git config --global user.name "Your Name"
git config --global user.email youremail@coriell.org
```

3. Set your default text editor

```
git config --global core.editor nano
```

4. Check your settings

```
git config --list
```

Set Up Your SSH Key

Set Up Your SSH Key

1. Generate an SSH key for yourself
   - Type: `ssh-keygen -t rsa -b 4096 -C "youremail@coriell.org"`
   - When you're prompted to "Enter a file in which to save the key", press Enter to accept the default file location.
   - At the prompt, "Enter passphrase" hit Enter to save the SSH key without a password (although you can enter a password) if you really want to.

Set Up Your SSH Key

1. Generate an SSH key for yourself
   - Type:

   ```
   ssh-keygen -t rsa -b 4096 -C "youremail@coriell.org"
   ```
   - When you're prompted to "Enter a file in which to save the key", press Enter to accept the default file location.
   - At the prompt, "Enter passphrase" hit Enter to save the SSH key without a password (although you can enter a password) if you really want to.
2. Add your SSH key to the ssh-agent
   1. Start the ssh-agent in the background by entering:

   ```
   eval "$(ssh-agent -s)"
   ```
   2. Add your SSH key to the ssh-agent by entering:

   ```
   ssh-add ~/.ssh/id_rsa
   ```

Add Your SSH Key to Your GitHub Account

1. Copy your ssh key to your clipboard

   ```
   pbcopy < ~/.ssh/id_rsa.pub
   ```

2. Follow along with me as I show you how to add the key to your GitHub account.

Make Your First Repository

Initializing a Git Repository

1. Log onto the server

Initializing a Git Repository

1. Log onto the server
2. Go to your data directory

```
cd data
```

Initializing a Git Repository

1. Log onto the server
2. Go to your data directory

   ```
   cd data
   ```

3. Make a new directory to practice with:

   ```
   mkdir practice_repository
   ```

Initializing a Git Repository

1. Log onto the server
2. Go to your data directory

   ```
   cd data
   ```

3. Make a new directory to practice with:

   ```
   mkdir practice_repository
   ```

4. Make a git repository

   ```
   git init
   ```

Making Your First Commit

Making Your First Commit

1. Make a file to commit
    1. `nano README.md`
    2. Add something to the file and save it

Making Your First Commit

1. Make a file to commit
    1. `nano README.md`
    2. Add something to the file and save it
2. Check your repository's status

    `git status`

Making Your First Commit

1. Make a file to commit
   1. `nano README.md`
   2. Add something to the file and save it
2. Check your repository's status

   `git status`
3. Add the file(s)

   `git add README.md`

Making Your First Commit

1. Make a file to commit
   1. `nano README.md`
   2. Add something to the file and save it
2. Check your repository's status

   `git status`
3. Add the file(s)

   `git add README.md`
4. Commit

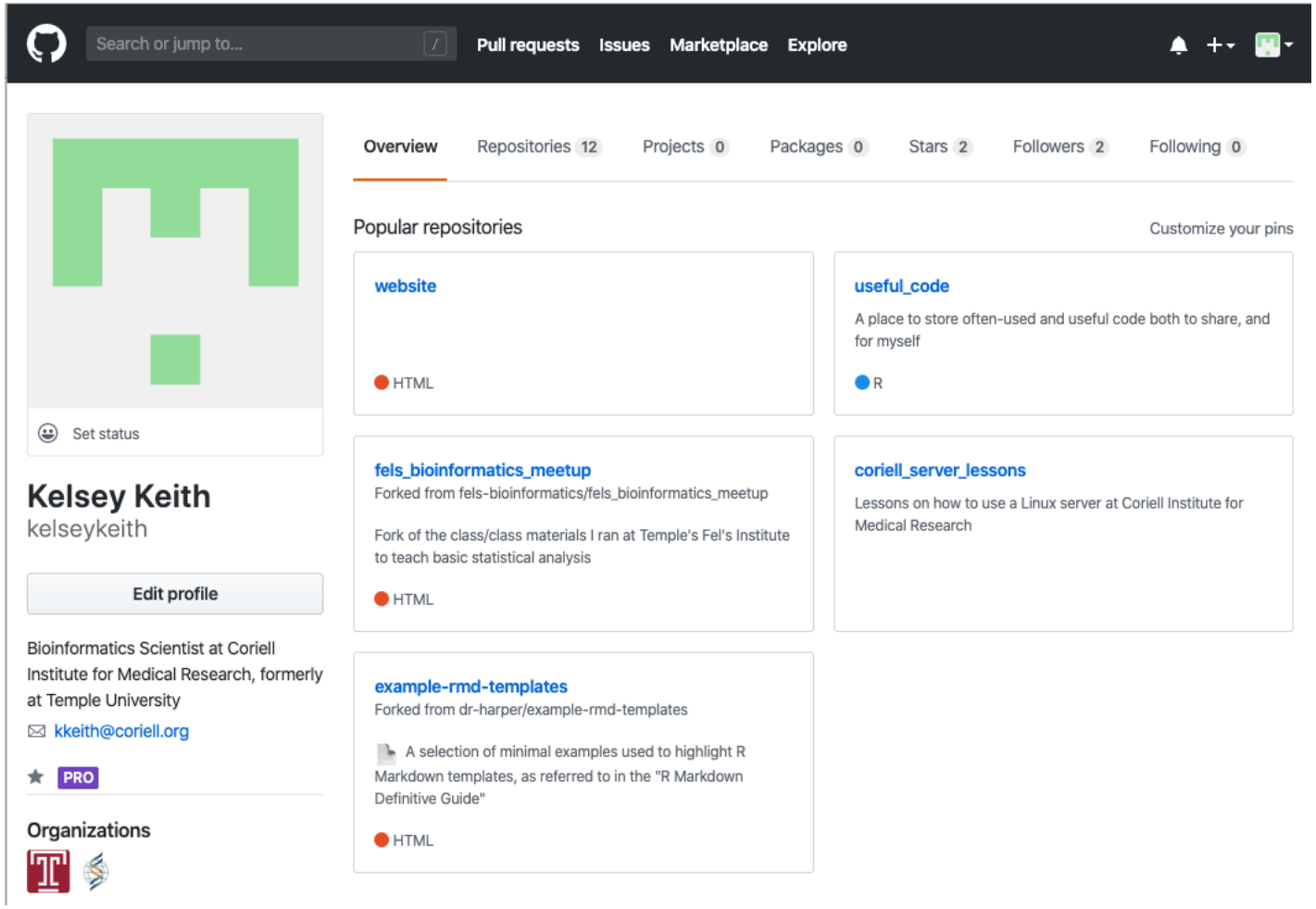
   `git commit –m 'first commit'`

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote

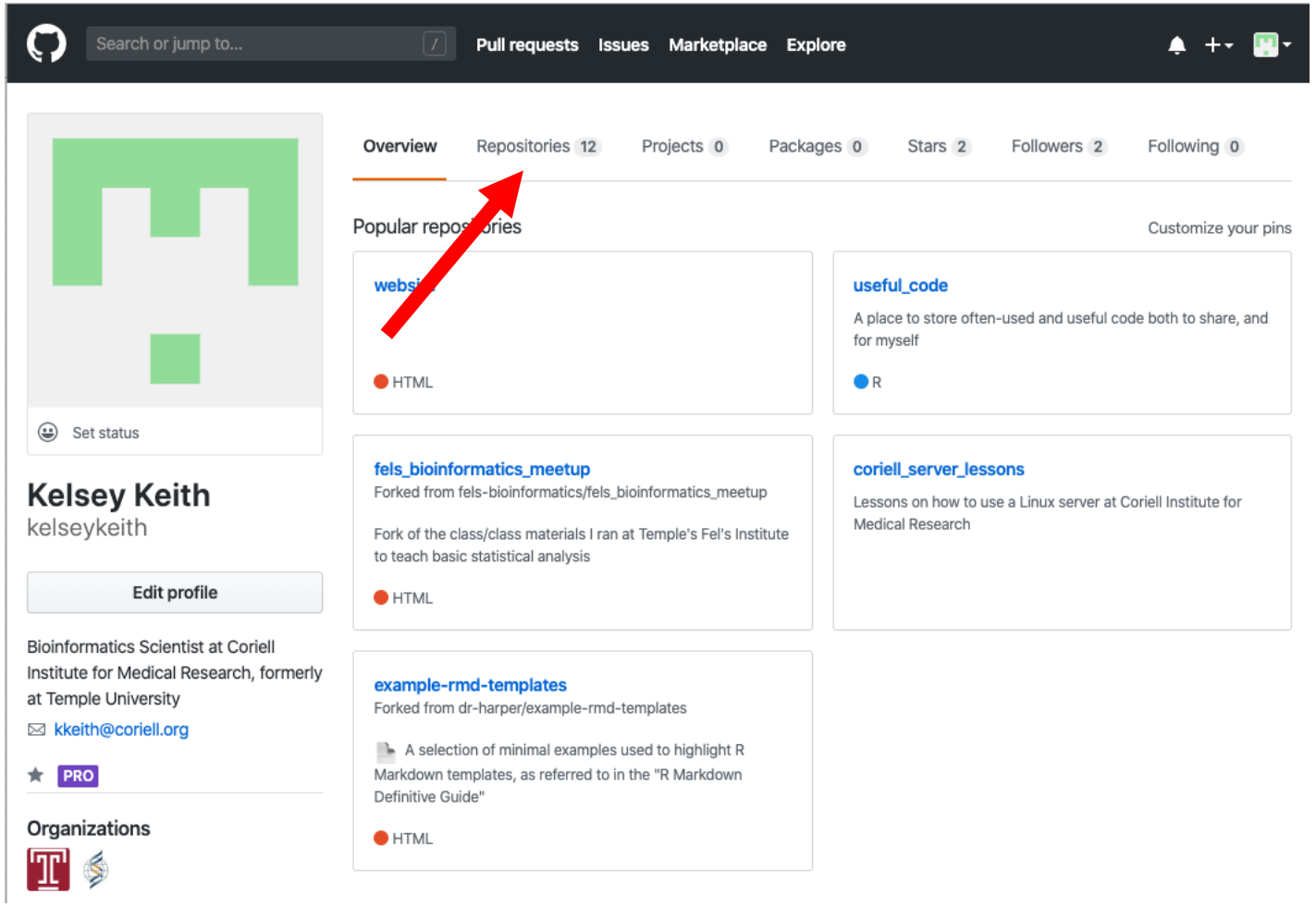
Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



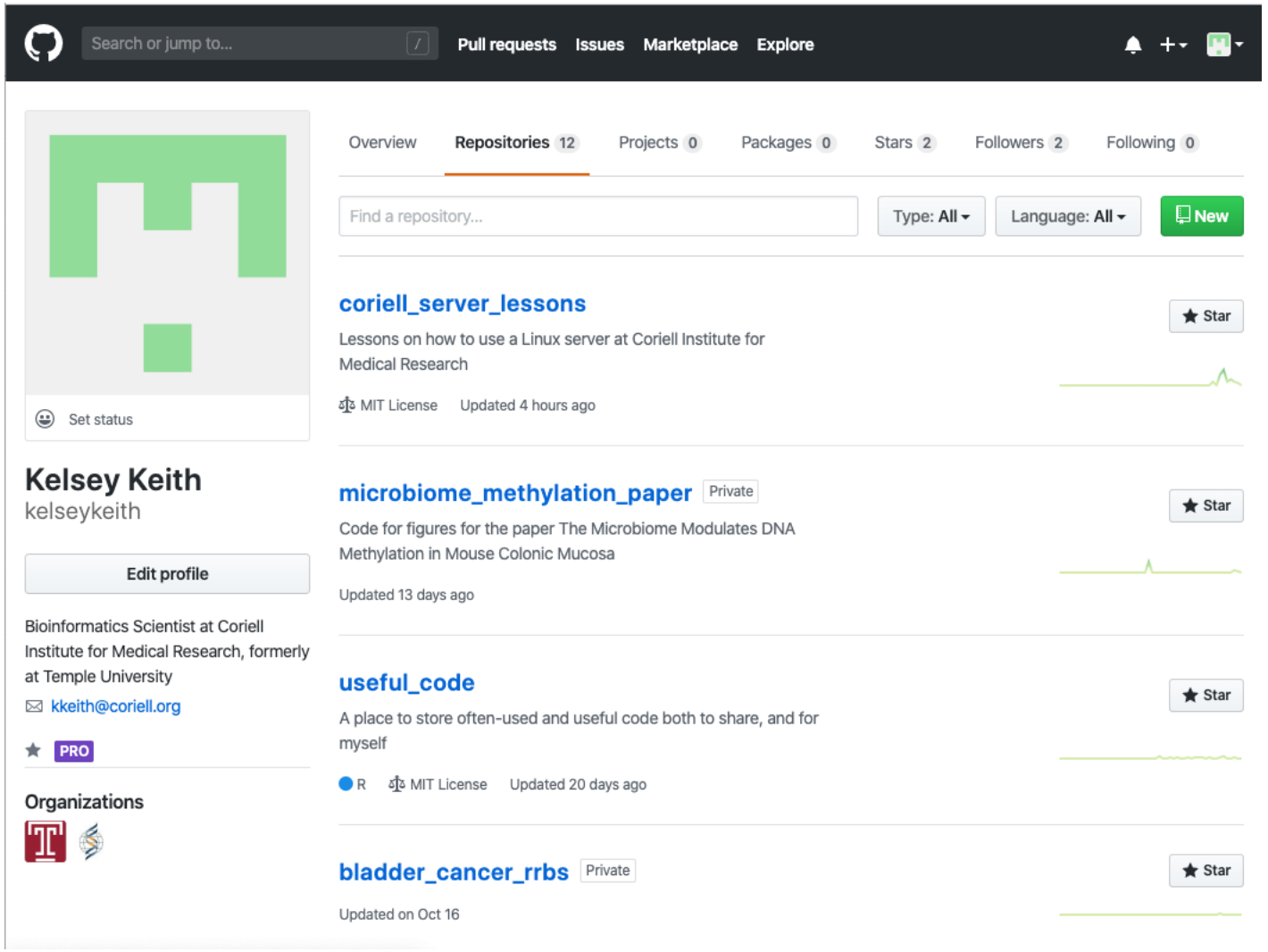
Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



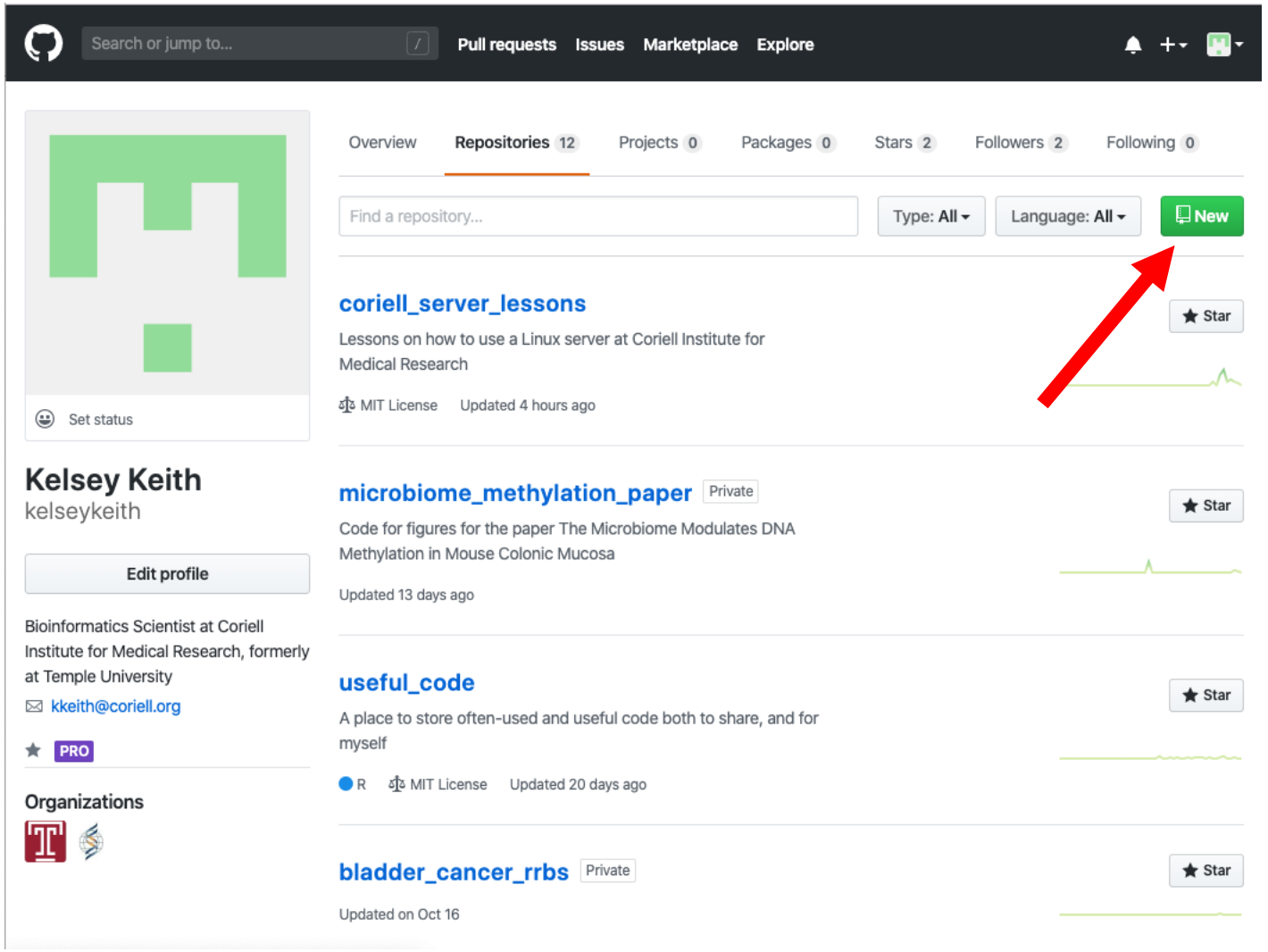
Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



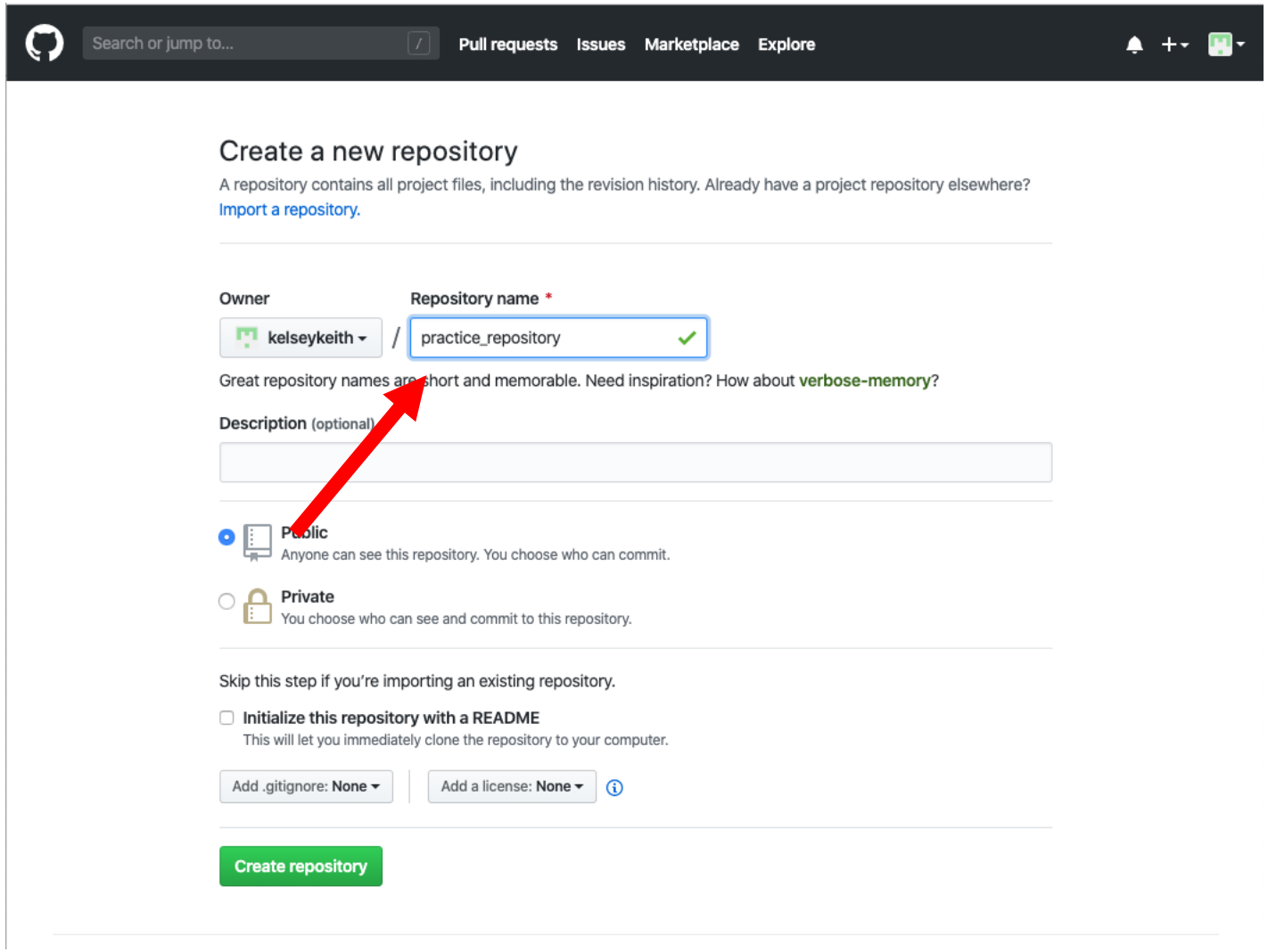
Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



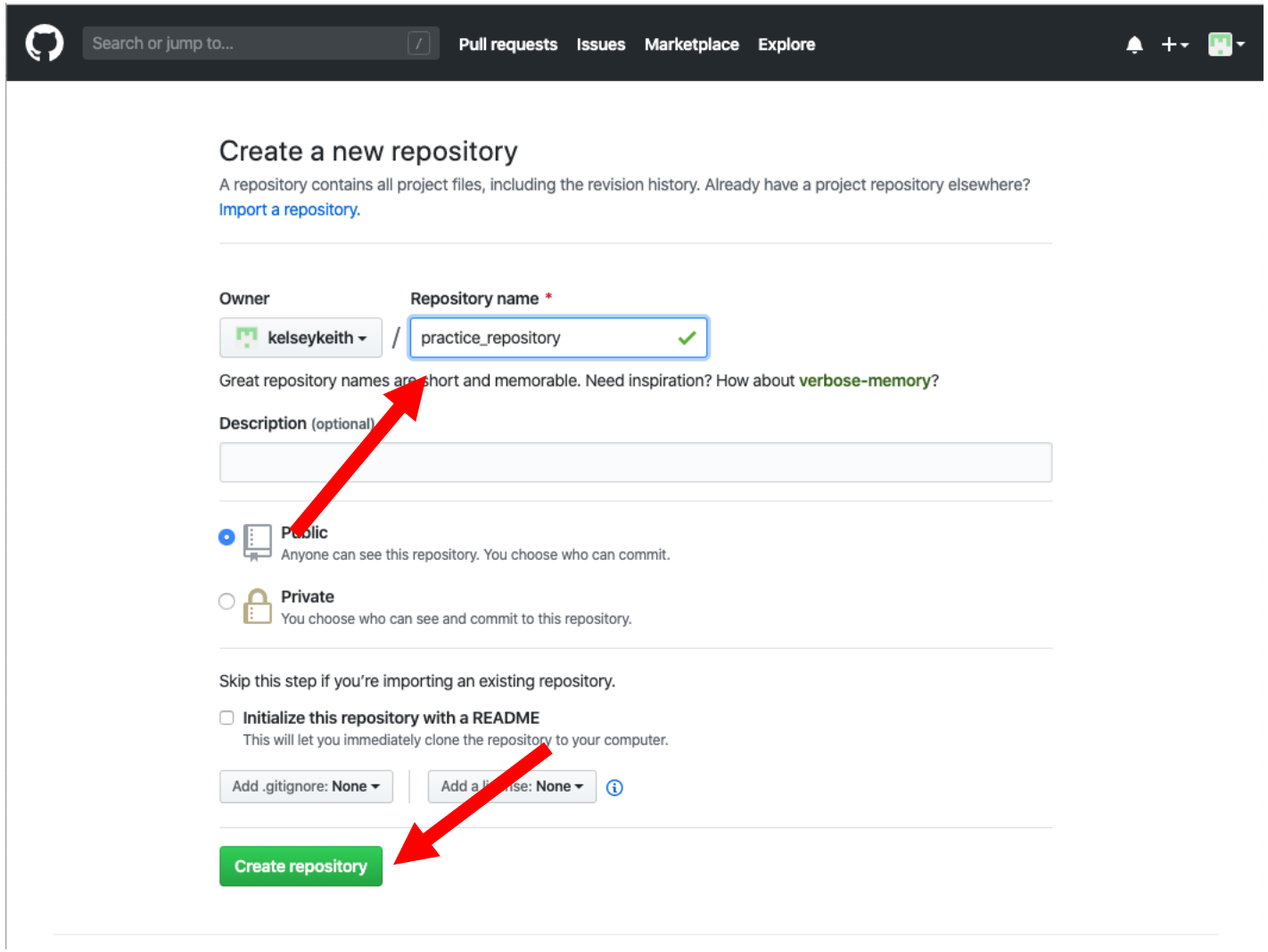
Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



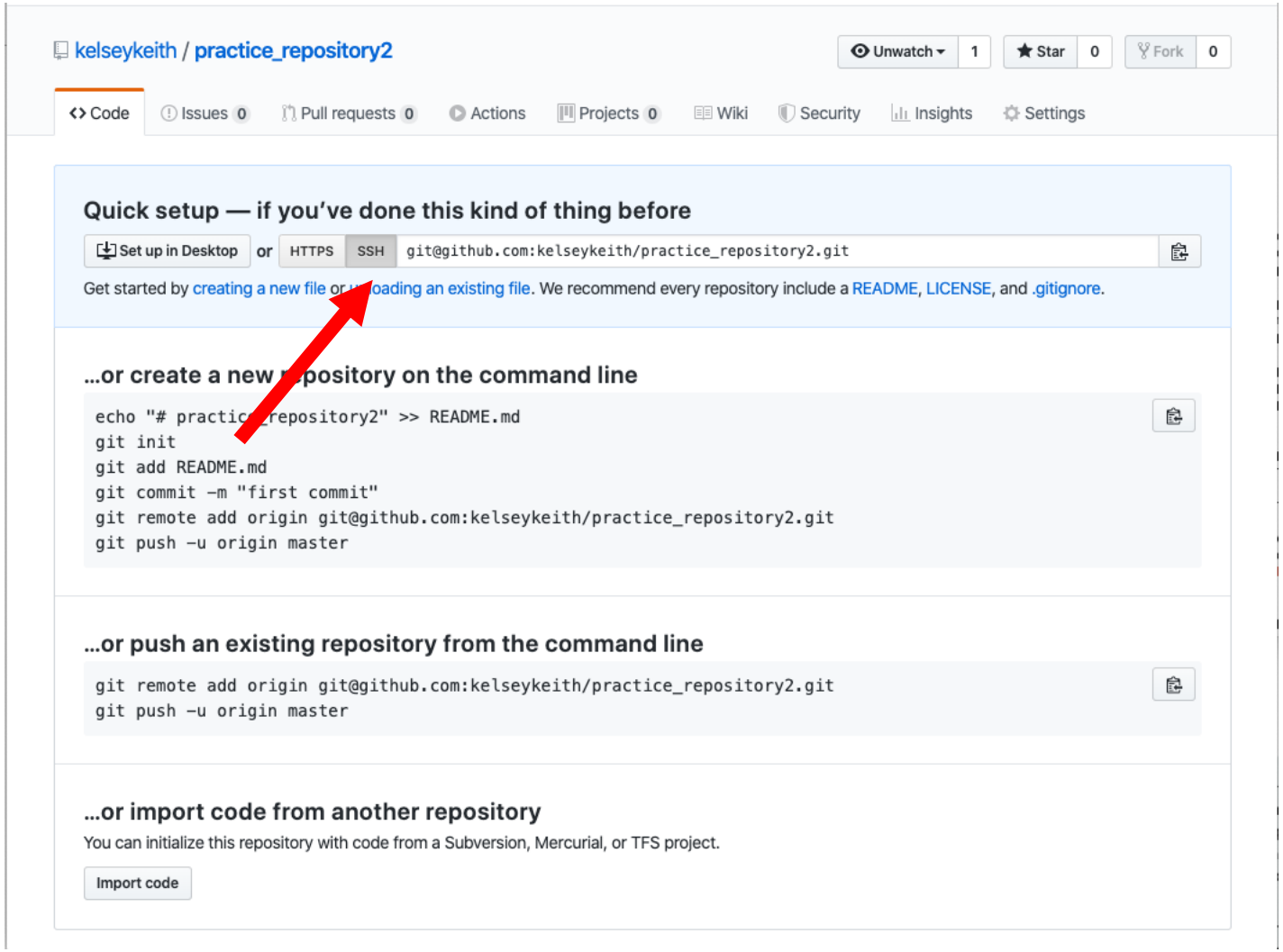
Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



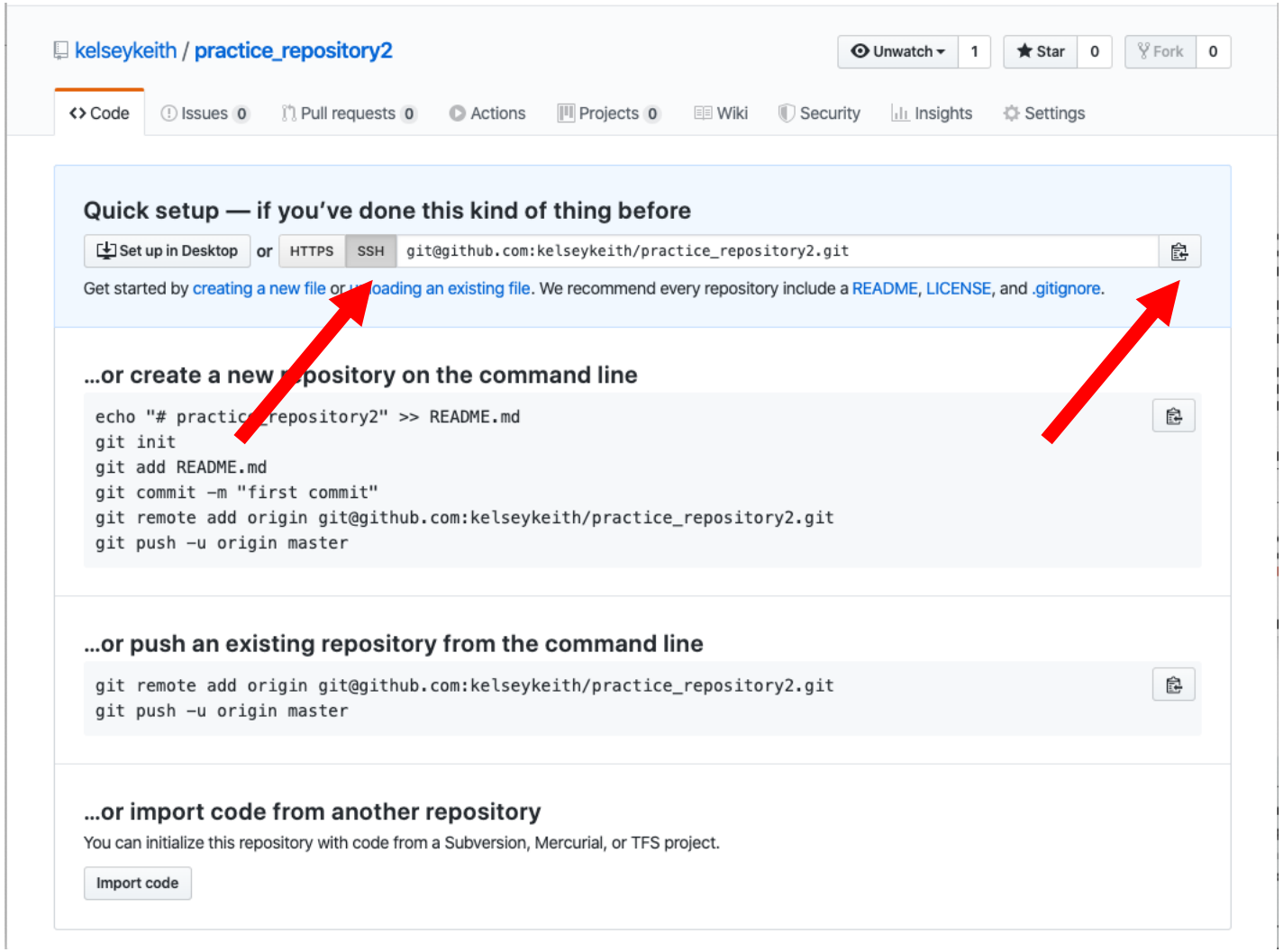
Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote



Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote
2. Add the GitHub repository as a remote

```
git remote add origin
git@github.com:kelseykeith/practice_repository.git
```

Connecting Your Local Repository to GitHub

1. Create a repository on GitHub to server as the remote
2. Add the GitHub repository as a remote

```
git remote add origin
git@github.com:kelseykeith/practice_repository.git
```

3. Push everything to the remote

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

Working with GitHub

Working with the GitHub

Working with the GitHub

1. Make an edit to your README on GitHub

Working with the GitHub

1. Make an edit to your README on GitHub
2. Use the command `git pull` on the server to grab the changes.

Working with the GitHub

1. Make an edit to your README on GitHub
2. Use the command `git pull` on the server to grab the changes.
3. Make a change on the server

Working with the GitHub

1. Make an edit to your README on GitHub
2. Use the command `git pull` on the server to grab the changes.
3. Make a change on the server
4. Add, commit, and push the change to GitHub.
    1. `git status`
    2. `git add README.md`
    3. `git commit –m ‘made a practice change’`
    4. `git push`

Adding a `.gitignore`

Adding a `.gitignore`

1. What is a .gitignore?
   - A .gitignore is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.

Adding a `.gitignore`

1. What is a .gitignore?
   - A .gitignore is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.
   - WARNING: The file upload limit for GitHub is **100 Mb**. Do NOT upload data to GitHub. We are using GitHub for documentation purposes only.

Adding a `.gitignore`

1. What is a .gitignore?
   - A .gitignore is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.
   - WARNING: The file upload limit for GitHub is **100 Mb**. Do NOT upload data to GitHub. We are using GitHub for documentation purposes only.
2. Make a .gitignore

Adding a `.gitignore`

1. What is a .gitignore?
   - A .gitignore is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.
   - WARNING: The file upload limit for GitHub is **100 Mb**. Do NOT upload data to GitHub. We are using GitHub for documentation purposes only.
2. Make a .gitignore
   1. `nano .gitignore`

Adding a `.gitignore`

1. What is a .gitignore?
   - A .gitignore is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.
   - WARNING: The file upload limit for GitHub is **100 Mb**. Do NOT upload data to GitHub. We are using GitHub for documentation purposes only.
2. Make a .gitignore
   1. `nano .gitignore`
   2. Add a file type, for example, `.fastq.gz` for it to ignore

Adding a `.gitignore`

1. What is a .gitignore?
   - A .gitignore is a file that tells git to ignore stuff; it can be certain files types or entire folder or one specific file.
   - WARNING: The file upload limit for GitHub is **100 Mb**. Do NOT upload data to GitHub. We are using GitHub for documentation purposes only.
2. Make a .gitignore
   1. `nano .gitignore`
   2. Add a file type, for example, `.fastq.gz` for it to ignore
3. For more information, check out the great documentation for .gitignore https://git-scm.com/docs/gitignore

PRACTICE

Practice pulling and pushing to and from GitHub

Resources

- Git Book https://git-scm.com/book/en/v2
  - Used this a reference to write this presentation
  - Much, much more here on advanced topics I didn't get into
- Adding SSH keys to your GitHub account https://help.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh
- .gitignore documentation https://git-scm.com/docs/gitignore