GitHub 101

Gbenga Iyiola

Florida State University – College of Engineering
High Performance Materials Institute
National High Magnetic Field Laboratory

Adviser: Prof. Jose Luis Mendoza-Cortes

June 30, 2016

Table of Contents

1. Table of contents
2. Introduction (Importance)
3. What is Git-Hub ?
4. Getting Started

Why is this important ?

- Due to network effects and the intertwining between projects Github has the highest rate of lock in, and more and more third parties directly integrate into the Git-hub API.
- Its "kinda" like a programmer's CV. Git-hub helps you keep track of all the projects you have been involved in over the years and can act as a public record of your skills and experience.
- Git-hub presents a very good way to learn and strengthen your coding ability.
- Repository - You have your repository, which is usually a library of files which you work from which is good for organization.
- Revisioning - Git keeps an eye on each revision you make with the option of 'reverting' to a previous version, should the need arise.

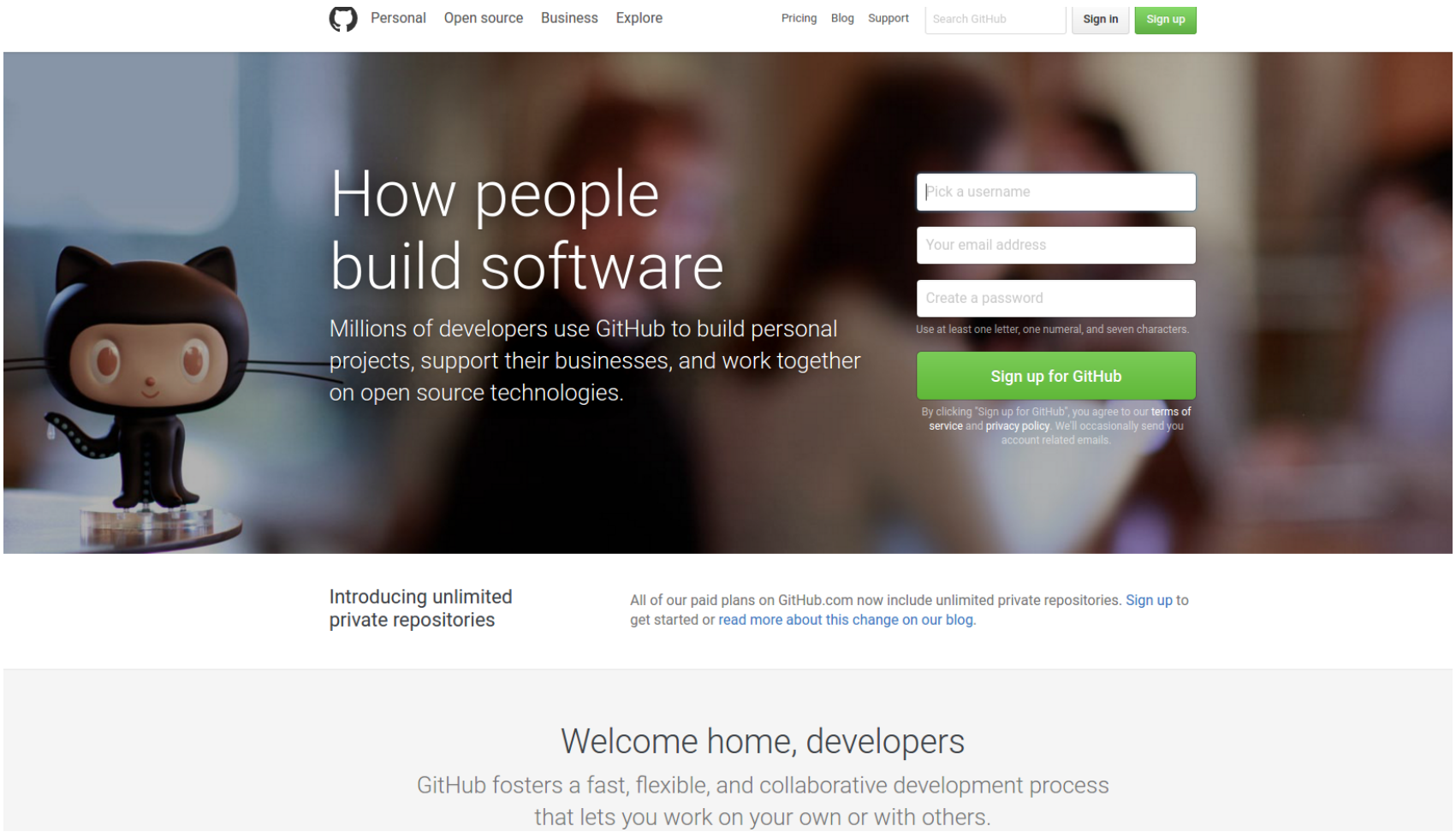
Why is this important ?

- Collaboration - Multiple people can pull and push codes to the repository, making it a centralized source.
- Forks - sort of an off-shoot of previous code. Forks are great for taking a basic example and further customizing the code because the initial code is still intact should other avenues be explored.
- Issue system - good way of handling any technical issues with the code with people being able to respond to particular sections.
- Github is a great way to handle and maintain code for a developer in the sense that it keeps everything organized with the ability to undo any unwanted actions. You can also push directly to servers so there is more consistency with that side of the transfer. In general, Github, and Git, are great tools for any developer/designer and relatively easy to learn.

What is Git-Hub ?

- GitHub is an open source code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

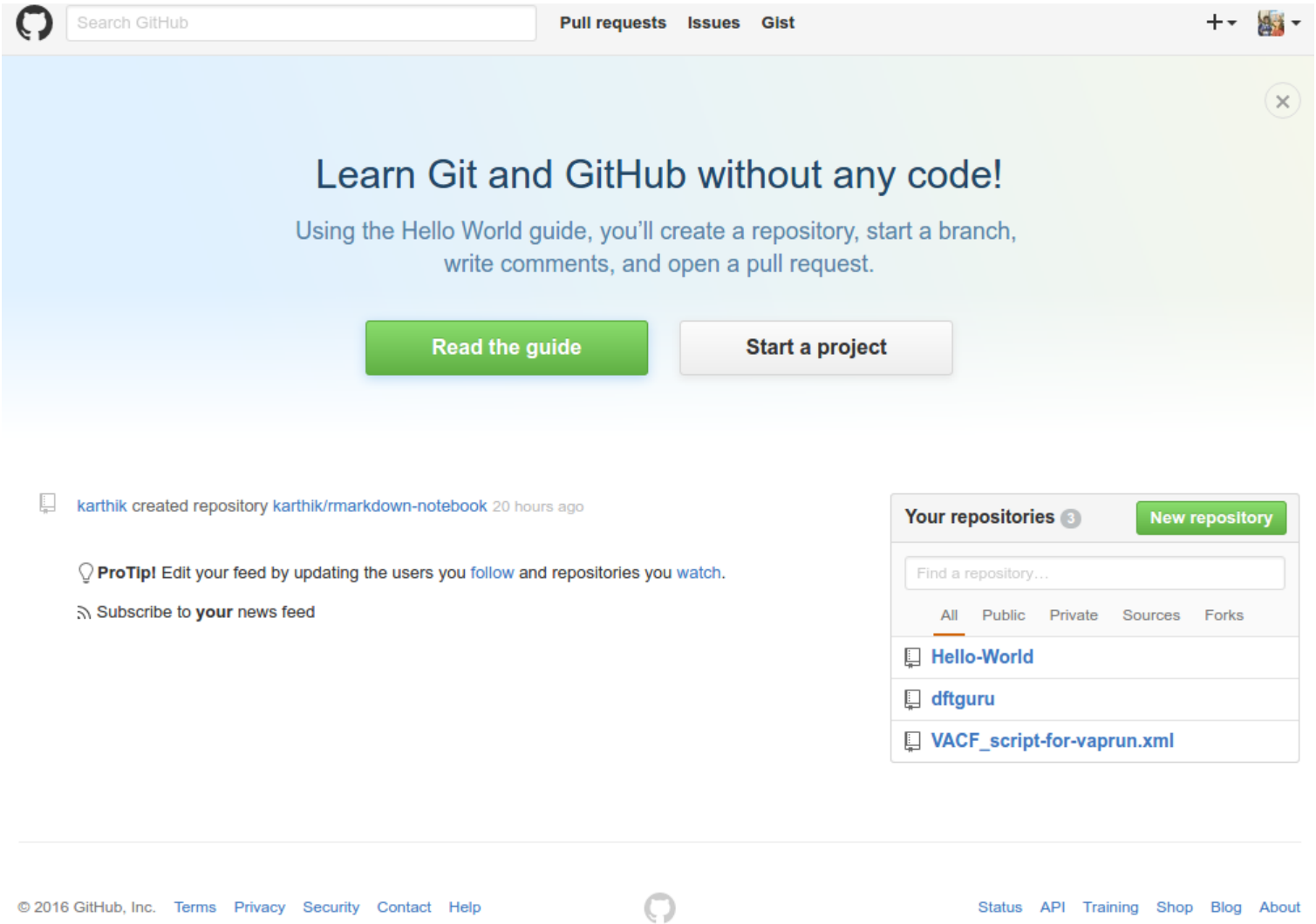
Opening an account



- Lets all open a github account and follow each other.
- Kindly follow me @dftguru and i'll follow back. Also follow Karthik Ram (@karthik) from UC Berkeley.

Welcome to Github

- You are now part of the elite 0.002 percent of the world's population.



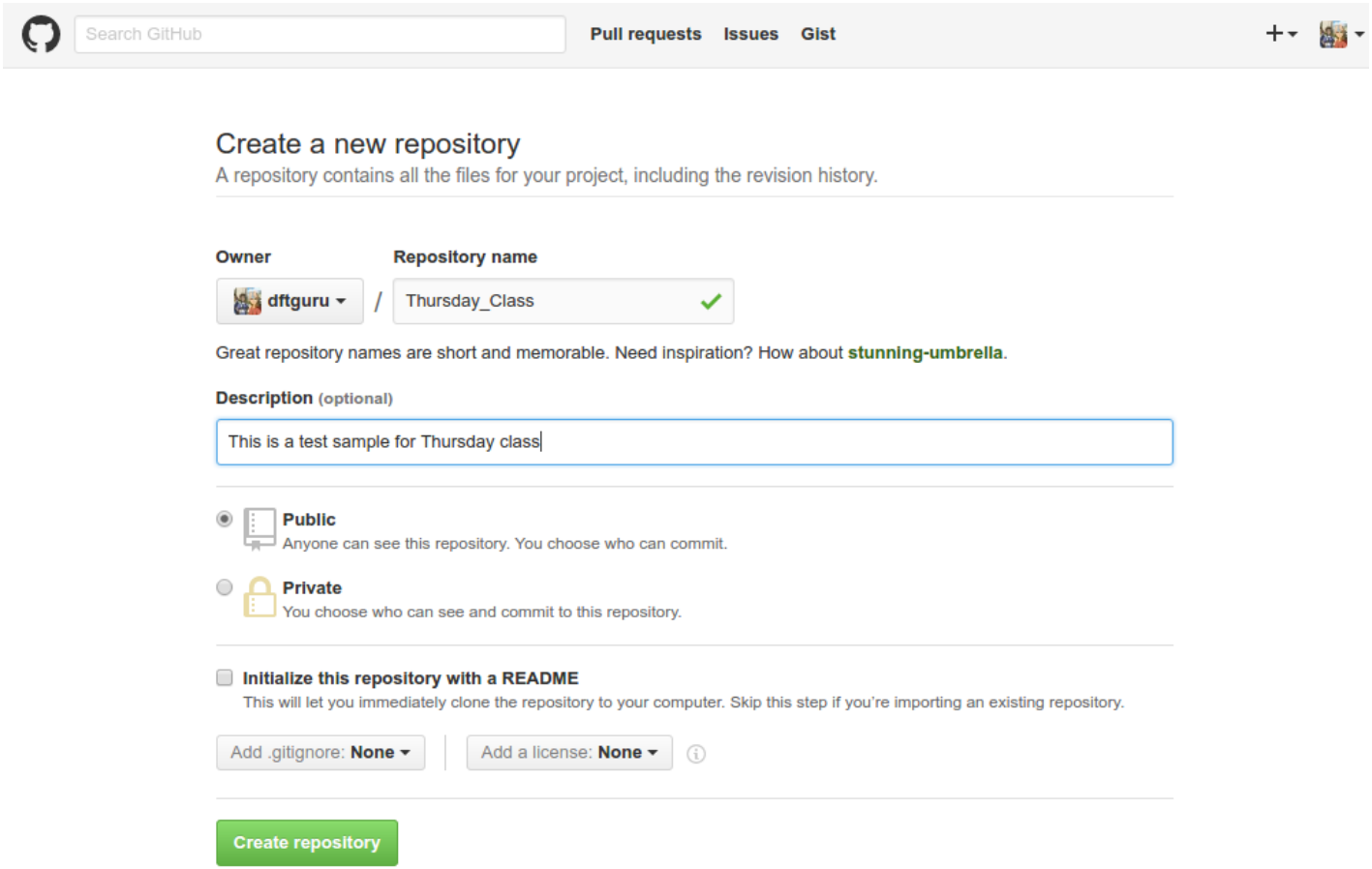
- Remember your avatar and email settings.

Creating a Repository

- A repository is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets anything your project needs. GitHub recommend including a README, or a file with information about your project. GitHub makes it easy to add one at the same time you create your new repository. It also offers other common options such as a license file.
- There are three ways to create a new repository
  1. Using the "Start a project" tab on your homepage.
  2. Using the "New repository" tab on your homepage.
  3. Using the "+" tab at the top right corner beside your avatar.

Creating a Repository

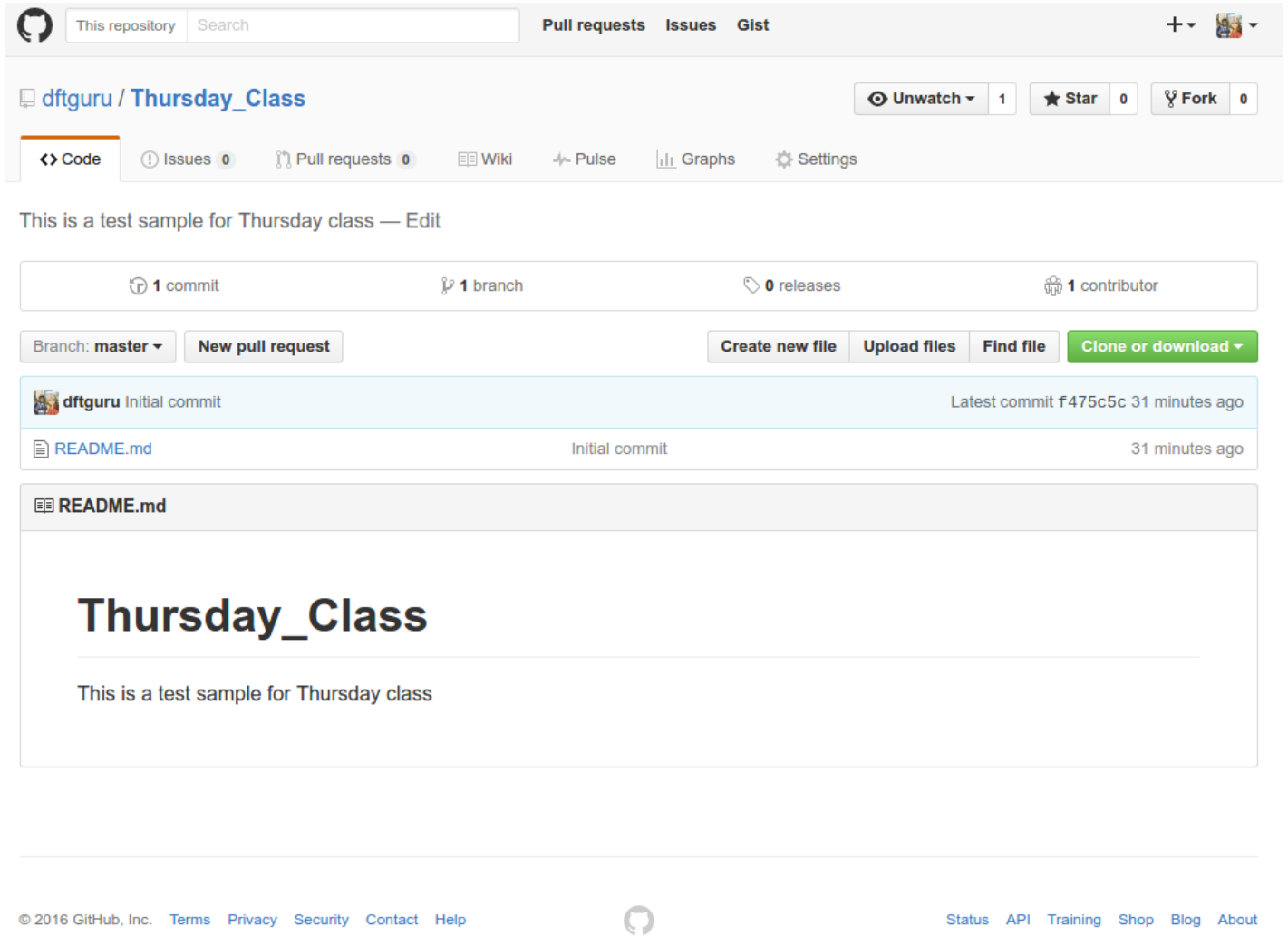
- Go ahead and name the repository, add a description and initialize with a README file.



- You can always edit the README file by clicking on README.md after creating the repository. Create the repository if you haven't already.

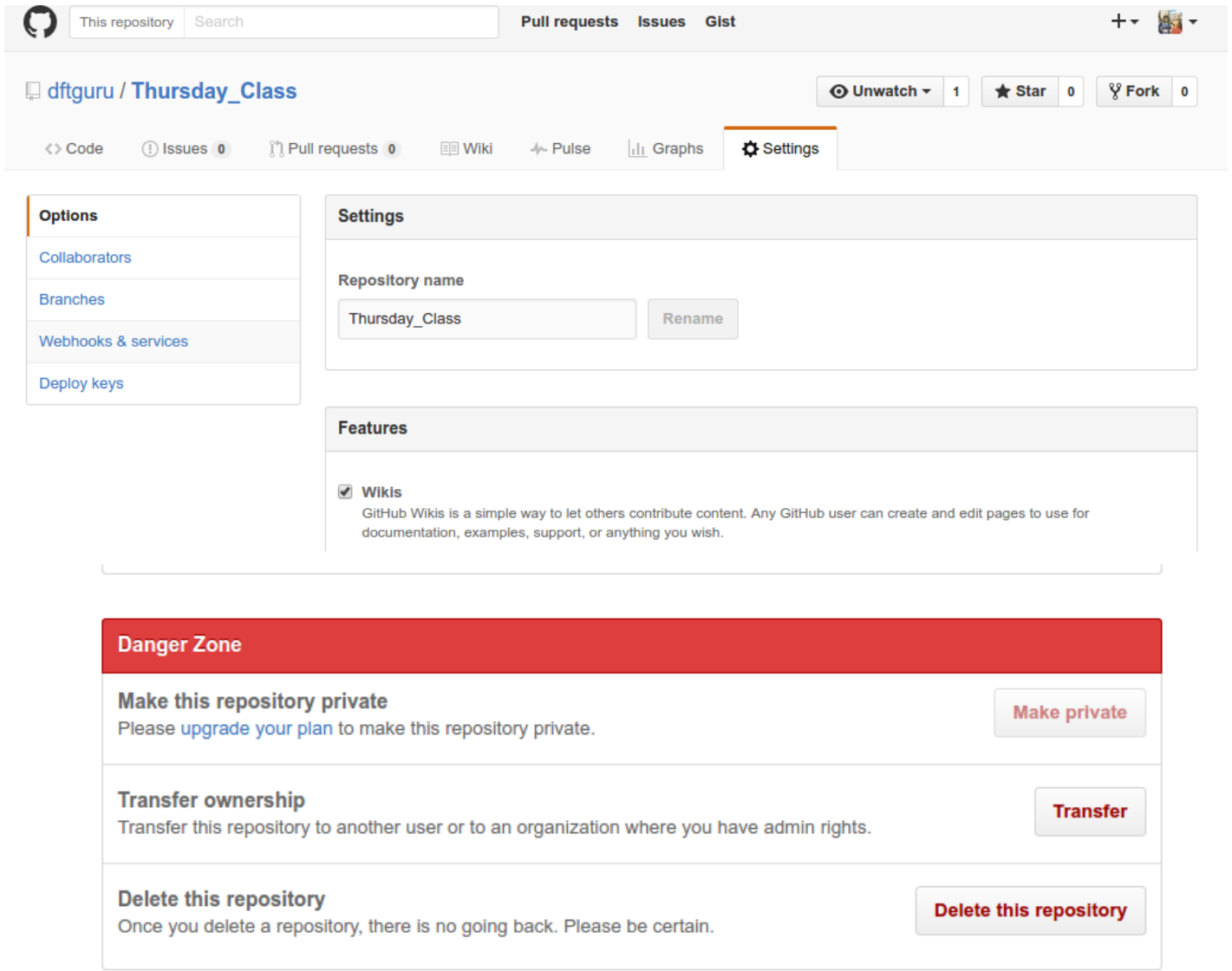
You have now created a repository

- Go ahead and explore the various tabs and what they do...



Deleting or Transfering ownership of a Repository

- The new repository you created can be deleted by going to "Settings"- located besides "Graph"



Creating a Branch

- Branching is the way to work on different versions of a repository at one time.

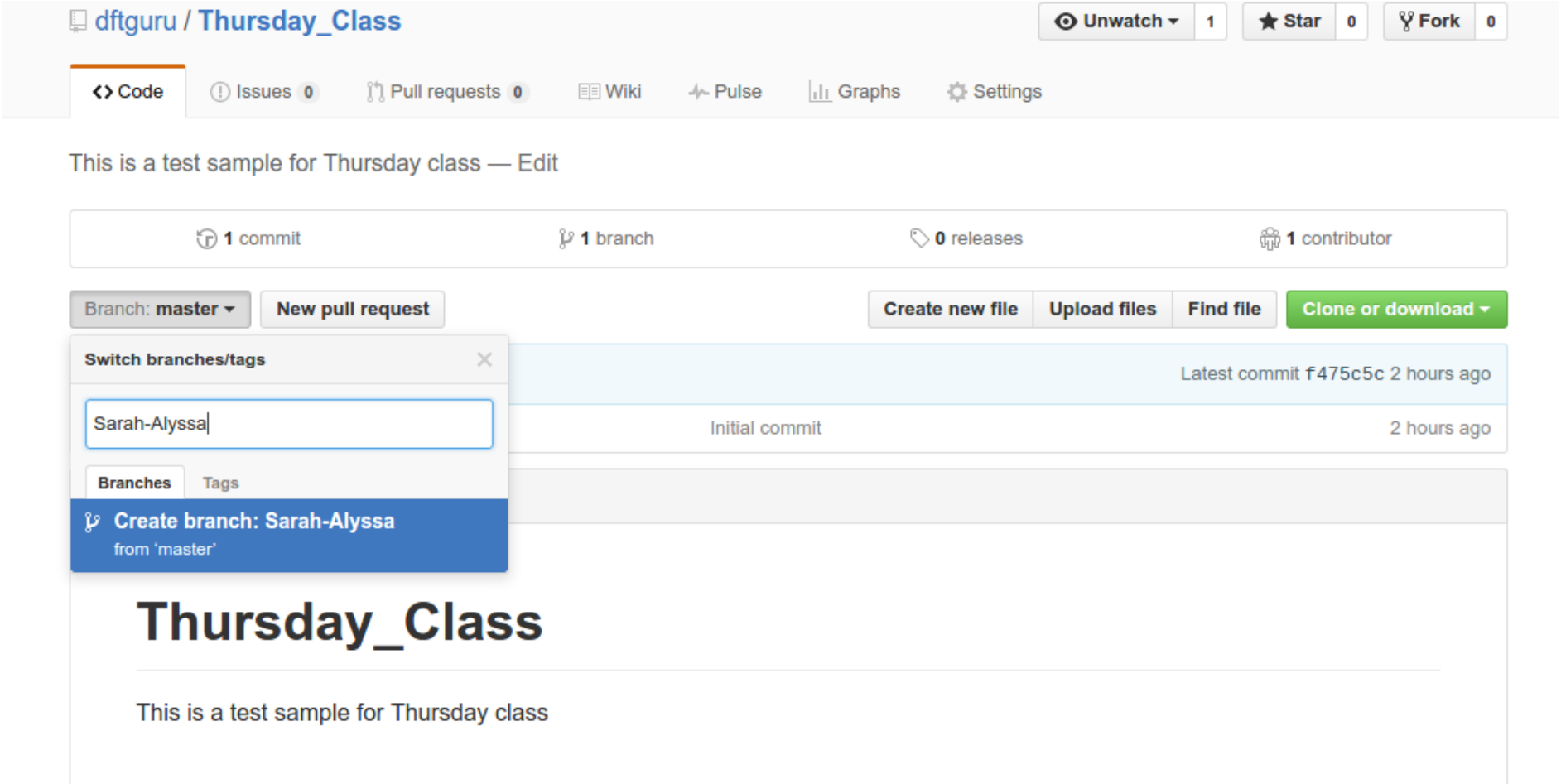
  By default your repository has one branch named "master" which is considered to be the definitive branch. We use branches to experiment and make edits before committing them to "master".

  When you create a branch off the "master" branch, youre making a copy, or snapshot, of "master" as it was at that point in time. If someone else made changes to the "master" branch while you were working on your branch, you could pull in those updates.

- To create a new branch

  1. Go to your new repository Thursday-Class.
  2. Click the drop down at the top of the file list that says Branch: master.
  3. Type a branch name, "Sarah-Alyssa-Srimanta", into the new branch text box.
  4. Select the blue Create branch box or hit Enter on your keyboard.

You have created a branch



- Now you have two branches, "master" and "Sarah-Alyssa-Srimanta". They look exactly the same, but not for long! Next well add our changes to the new branch.

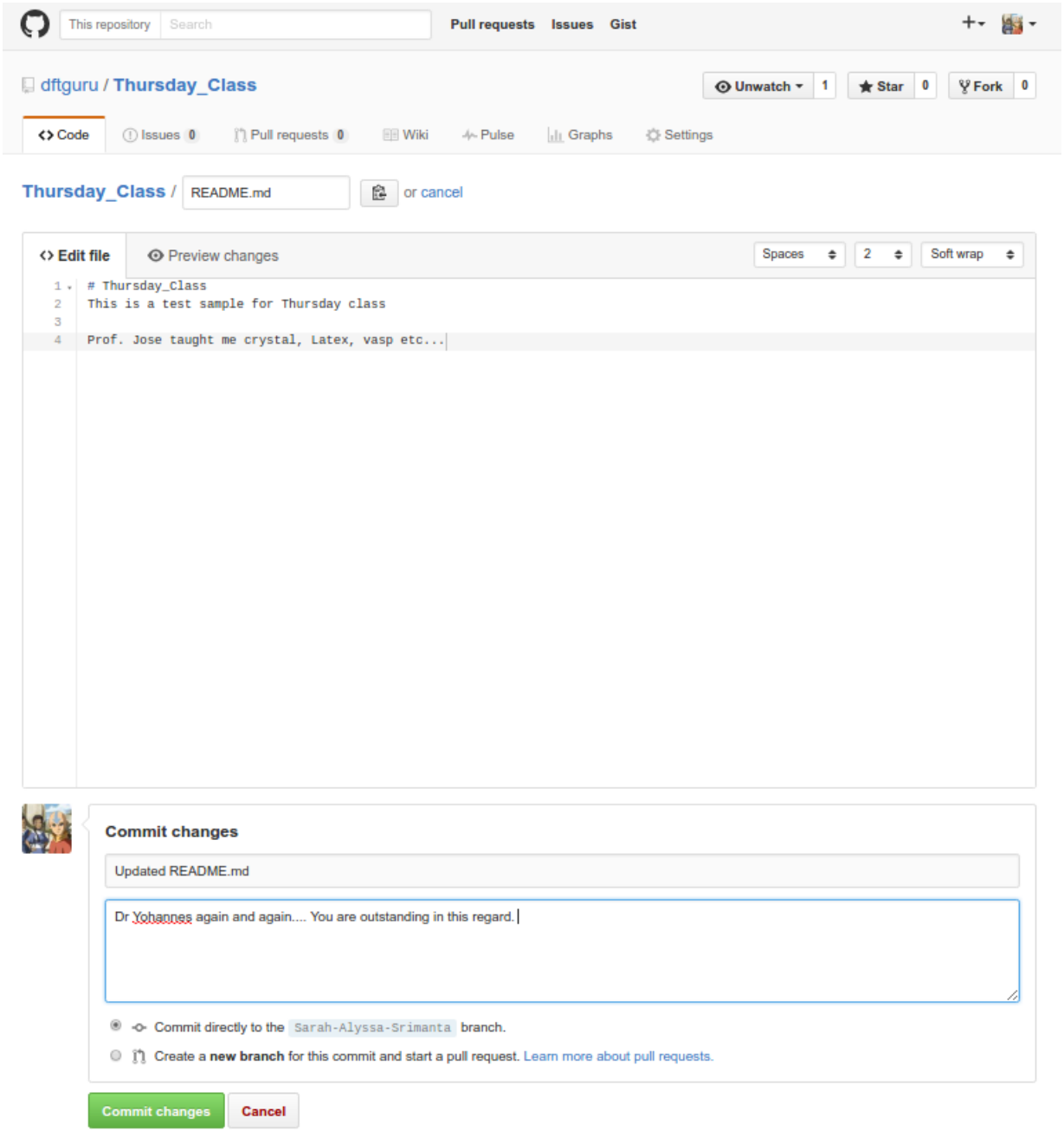
Make and commit changes

- Now we can make changes to the branch "Sarah-Alyssa-Srimanta" which is a branch of our "master.

To make and commit changes:

1. Click the README.md file.
2. Click the pencil icon in the upper right corner of the file view to edit.
3. In the editor, write a bit about yourself e.g Prof. Jose taught me crystal, latex, vasp etc...
4. Write a commit message that describes your changes.
5. Click Commit changes button.

Make and commit changes



Open a Pull Request

Nice edits! Now that you have changes in a branch off of master, you can open a pull request.
Pull Requests are the heart of collaboration on GitHub. When you open a pull request, youre proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch.
Pull requests show diffs, or differences, of the content from both branches. The changes, additions, and subtractions are shown in green and red.
As soon as you make a commit, you can open a pull request and start a discussion, even before the code is finished.
By using GitHubs @mention system in your pull request message, you can ask for feedback from specific people or teams, whether theyre down the hall or 10 time zones away.
You can even open pull requests in your own repository and merge them yourself. Its a great way to learn the GitHub Flow before working on larger projects.