
RU Silicon?
Git
Anthony Sistilli

# Today’s schedule

- What is github
- Why is it useful
- Features
- Basics

🥺🥺 BONUS: CARE PACKAGE !!!! 🥺🥺

INCLUDES:

Presentation + Notes

Helpful resources for Internships, and further learnings

(thank you 6godanthony)

# What is github?

“GitHub is a web-based Git repository hosting service. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features.” - Wikipedia

# Version Control System

- Tool used by developers to simultaneously work on a code source together

- Coders working on the same project can write code simultaneously, and combine it remotely using a VCS

# Why is it useful?

# Example:

- Person A and Person B are working on a game
- Person A wants to create the main menu
- Person B wants to create the login

Person A and Person B can simultaneously create their features on separate computers, and then combine it using git

# Real life application

- Companies use version control systems like Git for all of their projects to easily document, peer review, and revise their code.
- When entering a job as a software developer or software engineer, you will probably be expected to know the basics of VCS

# Features of Github

- Repositories
- Open Source
- Forking

# Repositories

- You would make a new repository for every project you want to work on
- Basically a folder on github for your project
- You can make your repository private or public
  - Public repositories and their files are accessible by everyone
  - Private repositories can only be seen by the owner, and those invited

# Open Source

“denoting software for which the original source code is made freely available and may be redistributed and modified.”

- Mozilla Firefox
- Ubuntu
- Wordpress

# Forking

“A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.”

Basically taking a public repository, and creating a copy of it on your account, so that you can freely change the code around without messing with the original

# Let’s Start!

Let’s start by making an account on Github installing Git!

https://github.com/

Mac Git Install

https://git-scm.com/download/mac

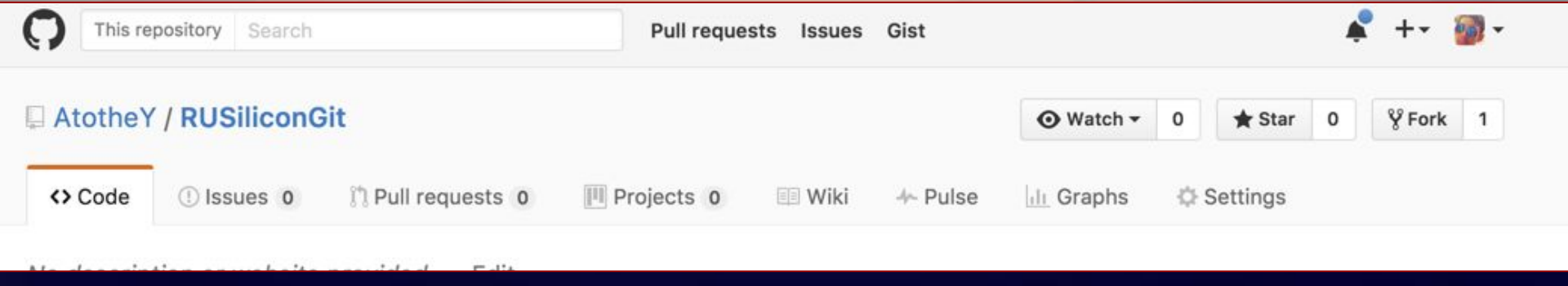
Windows Git-bash Install

https://git-scm.com/download/win

# Let’s Fork!

Let’s start by forking a sample repository I made, that contains the care package

Go to github.com/AtotheY/RUSiliconGit and click the “Fork” button on the top right

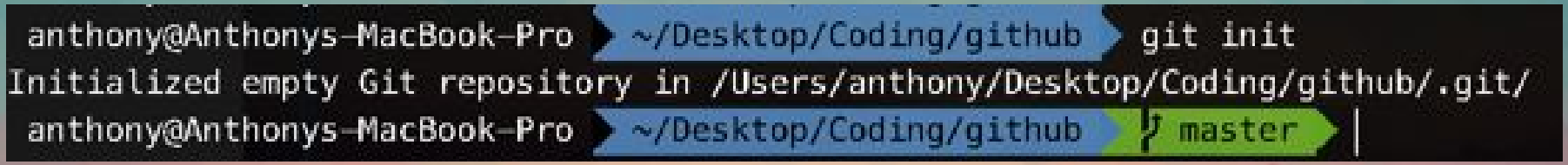


# Initializing a repository locally

In terminal, navigate to the folder you want your project to be in, and type

git init

This will create an empty repository in that directory



# Connecting the repository

Now that you've made an empty repository on your computer, you have to link the github repository to it. We can do this by typing

git remote add origin project_url.git

Origin is the pseudo name we assign to the specific remote url

Where project_url.git is the link to your forked url, with .git at the end. Make sure you replace "AtotheY" with your username

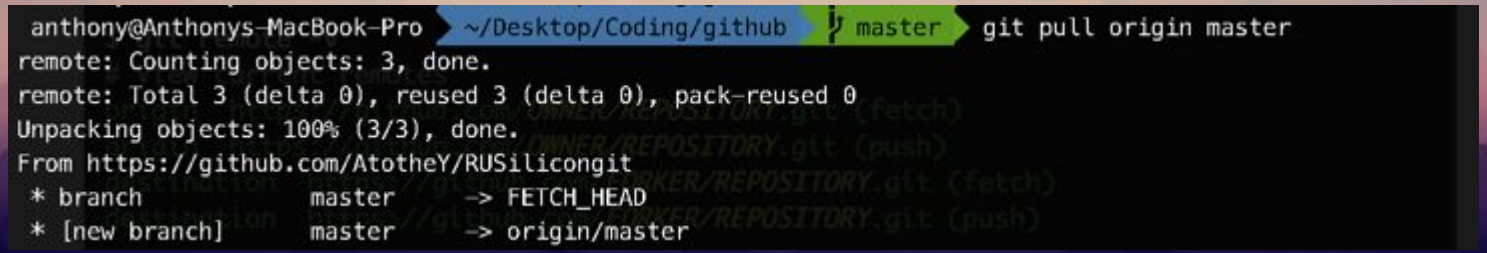
# Pulling

We now want to take the files on the github online repository, and transfer it to our local repository. This is called **pulling**. To pull, we type the following

git pull origin master

Master is the main branch of the program

*Note that it may require you to type in your login credentials



# Adding stuff to the repo

Our folder should now have the files from the github repository added in. Let's try adding something to it.

1. Open "Readme.md" with a text editor, and add your name to the top
2. Create a new file in the folder called "hello.txt"

# Pushing our changes to the online repo

If we edit our local repo, and want to push it onto our online github repo, we have to go through 3 steps

1. Adding
2. Committing
3. Pushing

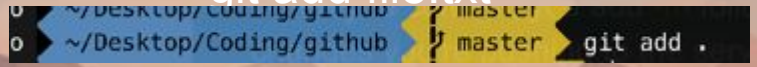
# Adding

The first step is to locally “Stage” the files and changes you’ve added. This step stages your changes to be committed in the next step.

git add .

Or alternatively,

git add file.txt




With “git add .”, it stages all the changes you’ve made since the last pull

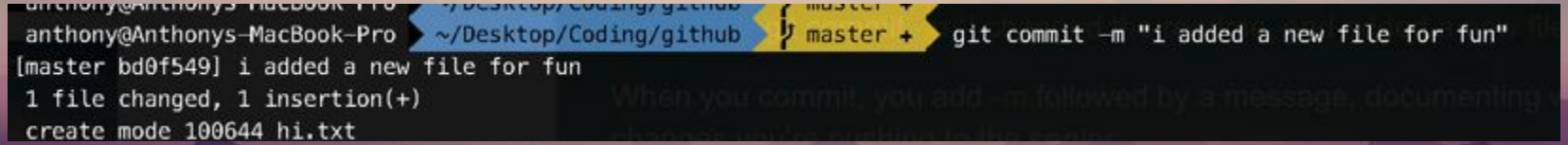
With “git add file.txt”, it only stages the specific file you specify

# Committing

Committing is the act of taking your staged changes, and putting them together to prepare to be sent to the online repository

git commit -m “I changed the readme, and added a new file”

When you commit, you add -m followed by a message, documenting what changes you’re pushing to the server

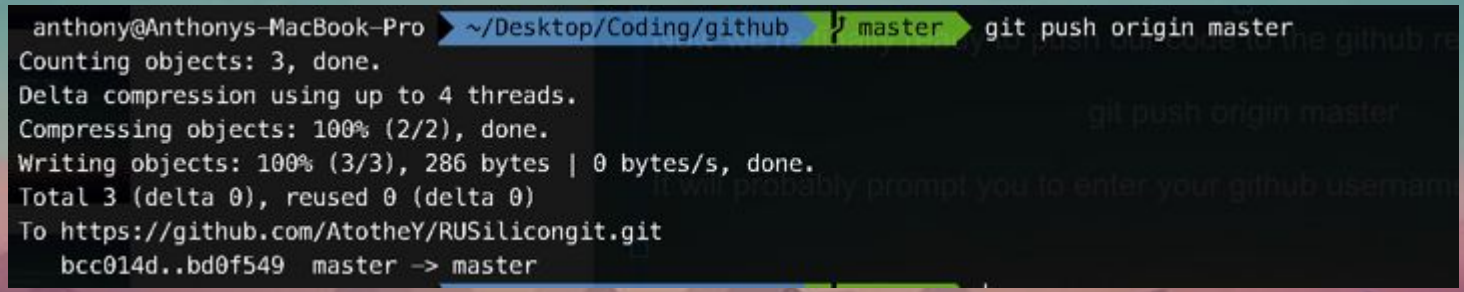


# Pushing

Now we're finally ready to push our code to the github repo!

git push origin master

It will probably prompt you to enter your github username and password




Check your online repository by refreshing the page, and your new files+changes should be there!

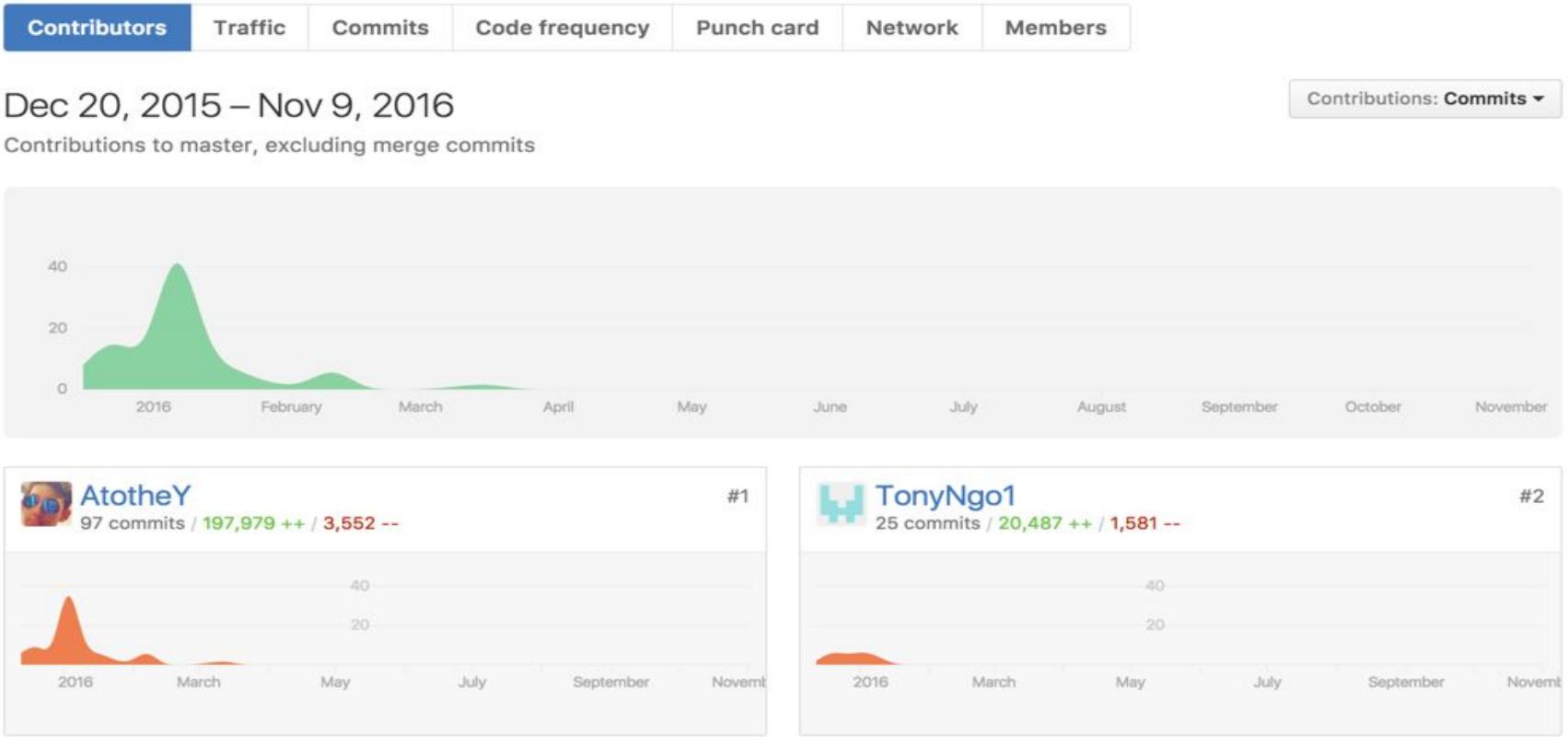
# Final Notes

It’s important to understand, that if someone else pushes to the repository while you’re working on it locally, you **WON’T** be able to push your changes until you pull theirs.

Future things to read up on:

- Branches
- Cloning
- Pull requests

# Online features



# Portfolio Opportunities

- Employers will often ask for your github username so they can see if you've contributed to any major projects, and how often you actually do projects