

A-Very-Hacktober Introduction to Open Source

Michael DeMarco

October 18th, 2022

A-Very-Hacktober Introduction to Open Source

- Welcome to this workshop on contributing to OSS!
- We'll get started shortly. . .

Introduction

- I'm Michael!
- 4th year Honours Computer Science and Data Science minor
- Currently an SDE Intern @ Tesla and Academic Officer with the CSS
- @michaelfromyeg on the Internet

Agenda

- Introduction to open-source (5min.)
- Basic commands; a first contribution (15min.)
- Hacktoberfest (10min.)
- Break-out and exploration (20min.)
- Q&A, wrap-up (10min.)

Goals

- Get an understanding of open source (what it is, why it exists)
- Know enough `git` to confidently contribute to open source
- Learn about Hacktoberfest and look at making some significant contributions this month
- Meet other folks to work on open source projects with!

Expectations

- Some familiarity with writing programs beyond CPSC 110 (i.e., with something other than Racket)
- Some experience with `git` (though I can cover it, if need be!)

Getting started

- So... what the heck is this open source thing?!

Your turn: what do you *think* open source is?

- Go ahead!

What is open source? (1/)

- Software that is **publicly accessible** so that the code can be seen, modified and use
 - This is restricted by a “license” (a bit like copyright)
 - More accurately, this is “free and open source” or FOSS; open source is focused on the visibility of the source code
 - “Free as in freedom, not as in beer”
- Projects often hosted on GitHub, GitLab, or similar in public repositories

What is open source? (2/)

- Many common examples: Linux, Visual Studio Code, React, Flutter, Kubernetes, . . . countless others
- Companies even being built around open source (e.g., Mattermost vs. Slack, Supabase vs. Firebase)

Surprise Linus!



Figure 1: Linus Torvalds... founder of Linux!

Why contribute?

- Give back to software you use
- Improve your skills on a significant project (. . . though personal projects are cool too!)
- Find mentors
- Grow a reputation
- . . . get a job!

Maybe... even a job at Facebook

- Take a look at this article



Figure 2: Jumping in GIF

What does contributing look like?

- While it may be code, there are lots of other ways to give back
- Reporting issues, improving docs, suggesting features, reviewing PRs, etc., are all valuable

Joining a project (1/)

- Open source is based around the idea of community, and to be successful, you should know the language of the community!
- Community of **maintainers** and **contributors**
- Created by an **author**

Joining a project (2/)

- You should be aware of the LICENSE and README.md
- CONTRIBUTING.md is a document outlining how to get involved
- CODE_OF_CONDUCT.md outlines appropriate behavior
- Look for custom issue trackers (outside, say, GitHub), online communities (Slack, Discord)

Finding a project

- Your best bet... projects you already use!
 - PrairieLearn! We'll take a look at their issues today
- Look for small things, too
 - ~28% of casual contributions are documentation (like typo fixes)
- Variety of resources online: Good First Issue, GitHub Explore, many more

Before you contribute...

- Check the following things
 - What's the license?
 - Does it accept contributions?
 - Is it active?
 - Are there open issues? How fast are folks getting responses?
 - Same for pull requests!
 - Are the maintainers friendly?

Our first contribution

- Let's make a quick detour... it's git time



Figure 3: Detour

Why use git?

- Open-source depends on community, so we need a sane way of managing versions of software
- Enter: version control!
- `git` is a command-line tool for managing versions of software projects, efficiently
- (Also authored by Linus Torvalds!)

Git for contributing to open source

- (Note... this is not a `git` workshop; we'll just do enough to make you functional)
- First, make sure `git` is installed
- (If on Windows, please use Chocolatey)

Configure git

- Run commands to inform who the contribution will be from

```
git config --global user.name "Your Name"
```

```
git config --global user.email "youremail@domain.com"
```

Fork the repository

- To make changes **to a repository we can't edit directly**, we make a fork
- This is do-able right within GitHub

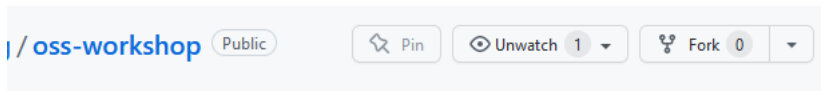


Figure 4: Fork repository on GitHub

After the fork

- Now you have a copy of the project to work in

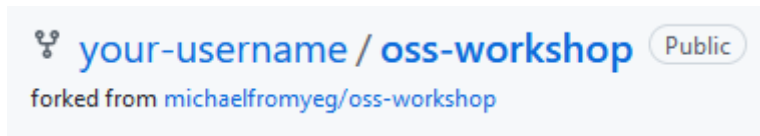


Figure 5: After the fork