Day 6:
Collaborative Causes

Goal

To explore collaborative development by building a web page with a team.

Learning Objectives

By the end of this exercise the student will be able to:

- take a webpage from design to deploy.
- collaborate with a team of developers using github.

Introduction

The world needs coders like you! The internet is one of the best places to raise awareness about anything. People spend so much time on social media and general web browsing that having content online is a must for getting the word out about anything.

For this assignment you'll be creating a webpage for a cause or nonprofit of your choice. This will be a group effort since collaboration is important in software engineering. We're going to be using lots of words that you may not be used to yet - you should go through the github glossary here to familiarize yourself: https://help.github.com/articles/github-glossary/.

Setup

- Form a group with 1-2 other students.
- Use Git + Github to collaborate on this project. Feel free to lean on your instructor for help with this. We're going to follow some simple rules to do so:
    - Assign one group member as the **git czar**. This person will take ownership of reviewing code before it is merged into the master branch.
    - If the **czar** is opening a pull request for themself another team member must review it.
    - Everyone will name their branches after the feature they're working on. Eg: **about-section**
    - Each group member should contribute at least two things to the project. That means they should have at least 2 closed pull requests reviewed by a team mate.
- Choose a cause that your group really cares about. This is totally open ended. If you can't think of any look at the following for inspiration: http://www.goodnet.org/articles/512

Instructions

- Before you even begin coding, sketch out what you'd like the website to look like with your team. The website should have the following **specifications:**
    - consist of a single page.
    - have at least 3 sections.
    - have a header and footer.
    - include relevant images.
- Divvy up your design to each member for implementation. If you're having trouble deciding how to equally split the work, check in with your instructor. Sticky notes could be helpful here.

- Set up collaboration:
    - The **git czar** should create a repository on github.
    - Then invite everyone as a collaborator.
    - Everyone should clone this repository.
- Everyone would by default be on the master branch, the czar should create a webpage as **index.html** using the basic html boilerplate on their computer and push it up. This way everyone starts at the same place Make sure to pull to get that change. .
- Now everyone is ready to work on their own features. That should follow this workflow:
    - Create <feature name> branch locally.
    - Checkout your new <feature name> branch.
    - Code your feature making regular commits.
    - Push changes to github.
    - Switch to the **master** branch and pull to make sure you have the latest changes your teammates made.
    - Switch to your branch and run **git merge master.**
    - Deal with any merge conflicts with your teammates if possible (to avoid deleting their work).
    - Go to github.com and open a pull request.
    - Wait for your changes to be reviewed by your team's czar.
    - Hit the merge pull request button when approved.
    - Switch to your master branch locally and pull to get your changes there.
    - Rinse and repeat.

Bonus: Follow the steps here to deploy your project to github pages:
https://github.com/blog/2289-publishing-with-github-pages-now-as-easy-as-1-2-3

Completion Requirements

- Workshop completed before end of allotted time.
- Group built website has the following specifications:
    - at least a basic sketch/wireframe.
    - 2 pull requests each to the repository.
    - A github organization set up.
- Group review with your instructor.