Generate repo for this assignment?

# GitHub Pages Project Webpage

## Instructions

\* Now that you know how to create webpages for specific projects, navigate into the repository of one of your previous activities or homework assignments and create a basic webpage for that project.

\* Make a simple website that explains what the purpose of the activity was and shows off some of the work that you have done.

\* Create a header that will act as a title for the page

\* Create a few short paragraphs describing the project's purpose

\* Try to include at least one table or a picture of a graph that shows off some of the data you collected

\* Have some fun! Test your HTML skills by going above and beyond with your page!

\* Once you have created your HTML, add, commit, and push your files up to GitHub Pages.

\* Make sure to navigate to your page in the web browser to ensure your website is live

~~## Deploy Guide~~

~~1. Create a new repository on your GitHub account. You can name this repository whatever you would like.~~

2. Once inside of the repository, create a new file and name it index.html

3. Add your HTML into this file, save it, and then navigate into your repository's Settings tab.

4. Scroll down to the GitHub Pages section and then, in the section labeled Source, select that you would like to use the master branch as your source.

5. Navigate to \_username\_.github.io/\_repositoryname\_ and you will find that your new web page has gone live!

* General
  + You must use bootstrap. This includes using the bootstrap navbar component for the header on every page, the bootstrap table component for the data page, and the bootstrap grid for responsiveness on the comparison page.
  + You must deploy your website to GitHub pages, with the website working on a live, publicly accessible URL as a result.
    - Figure out how to make subpages, if that’s a thing?
  + Be sure your website works at all window widths/sizes.
  + Feel free to take some liberty in the visual aspects, but keep the core functionality the same.
* Generate navbar anatomy, header styles and image colors stylesheet to import to all pages
  + Has the name of the site on the left of the nav which allows users to return to the landing page from any page.
  + Contains a dropdown on the right of the navbar named "Plots" which provides links to each individual visualization page.
  + Provides two more links on the right: "Comparisons" which links to the comparisons page, and "Data" which links to the data page.
  + Is responsive (using media queries). The nav must have similar behavior as the screenshots "Navigation Menu" section (notice the background color change).
  + Be sure to use a CSS media query for the navigation menu.
  + Bonus
    - Use a bootstrap theme to customize your website. You may use a tool like Bootswatch. Make it look snazzy, give it some attitude. If using this, be sure you also meet all of the requirements listed above.
    - Add extra visualizations! The more comparisons the better, right?
    - Use meaningful glyphicons next to links in the header.
    - Have visualization navigation on every visualizations page with an active state. See the screenshots below.
* Landing page (index)
  + Description:

“The purpose of this project was to analyze how weather changes as you get closer to the equator. To accomplish this analysis, we first pulled data from the OpenWeatherMap to assemble a dataset on over 500 cities.

After assembling the dataset, we used Matplotlib to plot various aspects of the weather vs. latitude. Factors we looked at included: temperature, cloudiness, wind speed and humidity. This site provides the source data and visualizations created as part of the analysis as well as explanations and descriptions of any trends and correlations witnessed.”

* Visualizations pages
* Comparison page
* Data page
  + You may use the weather data or choose another dataset. Alternatively, you may use the included cities dataset and pull the images from the assets folder.