Build a working, live website that's available to view at http://*your Github username*.github.io, and **submit this URL via Canvas**. You should:

1. Create at least 3 HTML pages, connected by a common navigation element (like a menu or navbar).
2. Make this website your own. It could be your personal profile page, site for a startup idea or small business, blog, list of recipes, or any other idea you might have. Use the HTML guidelines as listed below.
3. The browser tab should display a meaningful title describing what the page is about (e.g. "About Me", "Brian's Recipes", etc.)
4. Link to a separate stylesheet file that's shared across all pages on your site (see CSS guidelines below).
5. Sync your code into your GitHub.com account. The repository should be named *(your Github username)*.github.io. For example, my Github username is *eng*, so my repository name for this assignment should be *eng.github.io*. This will automatically make your page available on the public Internet at that address (e.g. [http://eng.github.io/ (Links to an external site.)Links to an external site.](http://eng.github.io/)).

Linking the pages

If all goes well, your directory structure should resemble something like this:

Your code folder

- username.github.io

-- index.html

-- page2.html

-- page3.html

The "home page" (i.e. what visitors will see when they go to username.github.io) is the index.html file, and is the only filename that really matters. You can name the other files whatever you want.

Creating links between the pages is as simple as <a href="page2.html">My second page</a>. Don't use "absolute" paths, for instance <a href="C:\\code\username.github.io\page2.html">My second page</a> or <a href="http://username.github.io/page2.html">My second page</a> -- just use the filename by itself. This is known as the "relative" path and works great, as long as the files are all in the same directory.

HTML Guidelines

You should use these elements at least once, and use proper indentation and spacing:

* Headers (h1, h2, etc)
* p
* ul/li or table
* a
* body
* html
* title
* head

CSS Guidelines

Add visual styles that apply good design principles and that are responsive across a variety of devices and platforms.

You should:

* Link to the Bootstrap CSS framework
* Utilize the Bootstrap grid system to create a logical separation in your design elements (e.g. header, content, sidebar, footer) on medium-sized (992px wide and up) devices and up
* Ensure that on smaller sized devices, like a tablet or phone, your site is still usable. Content should be stacked.
* Use at least one of Bootstrap's built-in CSS classes to style HTML elements. Examples:
  + Make a link look like a button
  + Style a table
  + Manipulate an image (make it square, circle, etc. or responsive in size)
* Use at least one other Bootstrap design component
  + Nav
  + Navbar
  + Label
  + Alert
  + Etc.
* **Use your own CSS in addition to Bootstrap, in order to customize the look of your site. Override at least 3 of Bootstrap's style rules.**
* Use Font Awesome and/or Google Fonts to add custom typography/icongraphy to your site.

Commit and sync your changes to your Github account no later than 11:59pm on Monday.

Grading Rubric

* 5 points: 3 distinct pages, linked by navbar, and the links work on any computer, not just yours.
* 5 points: All required HTML elements used.
* 5 points: Override 3 bootstrap rules.
* 5 points. Use at least one Bootstrap component.

Again, be sure to **submit your username.github.io website URL via Canvas**- this is the ONLY way we will know where your finished assignment lives!

Happy coding!