Homepage

Build a simple homepage using HTML, CSS, and JavaScript.

Background

The internet has enabled incredible things: we can use a search engine to research anything imaginable, communicate with friends and family members around the globe, play games, take courses, and so much more. But it turns out that nearly all pages we may visit are built on three core languages, each of which serves a slightly different purpose:

1. HTML, or *HyperText Markup Language*, which is used to describe the content of websites;
2. CSS, *Cascading Style Sheets*, which is used to describe the aesthetics of websites; and
3. JavaScript, which is used to make websites interactive and dynamic.

Create a simple homepage that introduces yourself, your favorite hobby or extracurricular, or anything else of interest to you.

Getting Started

Log into code.cs50.io, click on your terminal window, and execute `cd` by itself. You should find that your terminal window's prompt resembles the below:

```
$
```

Next execute

```
wget https://cdn.cs50.net/2021/fall/psets/8/homepage.zip
```

Then execute

```
unzip homepage.zip
```

to create a folder called `homepage`. You no longer need the ZIP file, so you can execute

```
rm homepage.zip
```

and respond with "y" followed by Enter at the prompt to remove the ZIP file you downloaded.

Now type

```
cd homepage
```

followed by Enter to move yourself into (i.e., open) that directory. Your prompt should now resemble the below.

```
homepage/ $
```

Execute `ls` by itself, and you should see a few files:

```
index.html  styles.css
```

If you run into any trouble, follow these same steps again and see if you can determine where you went wrong! You can immediately start a server to view your site by running

```
http-server
```

in the terminal window. Then, command-click (if on Mac) or control-click (if on PC) on the first link that appears:

```
http-server running on LINK
```

Where LINK is the address of your server.

Specification

Implement in your `homepage` directory a website that must:

- Contain at least four different `.html` pages, at least one of which is `index.html` (the main page of your website), and it should be possible to get from any page on your website to any other page by following one or more hyperlinks.
- Use at least ten (10) distinct HTML tags besides `<html>`, `<head>`, `<body>`, and `<title>`. Using some tag (e.g., `<p>`) multiple times still counts as just one (1) of those ten!
- Integrate one or more features from Bootstrap into your site. Bootstrap is a popular library (that comes with lots of CSS classes and more) via which you can beautify your site. See Bootstrap's documentation to get started. In particular, you might find some of Bootstrap's components of interest. To add Bootstrap to your site, it suffices to include

  ```
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA/Ms
  ```

  in your pages' `<head>`, below which you can also include

  ```
  <link href="styles.css" rel="stylesheet">
  ```

  to link your own CSS.
- Have at least one stylesheet file of your own creation, `styles.css`, which uses at least five (5) different CSS selectors (e.g. tag (`example`), class (`.example`), or ID (`#example`)), and within which you use a total of at least five (5) different CSS properties, such as `font-size`, or `margin`; and
- Integrate one or more features of JavaScript into your site to make your site more interactive. For example, you can use JavaScript to add alerts, to have an effect at a recurring interval, or to add interactivity to buttons, dropdowns, or forms. Feel free to be creative!
- Ensure that your site looks nice on browsers both on mobile devices as well as laptops and desktops.\

Testing

If you want to see how your site looks while you work on it, you can run `http-server`. Command- or control-click on the first link presented by http-server, which should open your webpage in a new tab. You should then be able to refresh the tab containing your webpage to see your latest changes.

Recall also that by opening Developer Tools in Google Chrome, you can *simulate* visiting your page on a mobile device by clicking the phone-shaped icon to the left of **Elements** in the developer tools window, or, once the Developer Tools tab has already been opened, by typing `Ctrl` + `Shift` + `M` on a PC or `Cmd` + `Shift` + `M` on a Mac, rather than needing to visit your site on a mobile device separately!

Assessment

No `check50` for this assignment! Instead, your site's correctness will be assessed based on whether you meet the requirements of the specification as outlined above, and whether your HTML is well-formed and valid. To ensure that your pages are, you can use this Markup Validation Service, copying and pasting your HTML directly into the provided text box. Take care to eliminate any warnings or errors suggested by the validator before submitting!

Consider also:

- whether the aesthetics of your site are such that it is intuitive and straightforward for a user to navigate;
- whether your CSS has been factored out into a separate CSS file(s); and
- whether you have avoided repetition and redundancy by "cascading" style properties from parent tags.

Afraid `style50` does not support HTML files, and so it is incumbent upon you to indent and align your HTML tags cleanly. Know also that you can create an HTML comment with:

```
<!-- Comment goes here -->
```

but commenting your HTML code is not as imperative as it is when commenting code in, say, C or Python. You can also comment your CSS, in CSS files, with:

```
/* Comment goes here */
```

Hints

For fairly comprehensive guides on the languages introduced in this problem, check out these tutorials:

- HTML
- CSS
- JavaScript

How to Submit

In your terminal, execute the below to submit your work.

```
submit50 cs50/problems/2022/x/homepage
```