This is CS50x

OpenCourseWare

David J. Malan (https://cs.harvard.edu/malan/)
malan@harvard.edu
(https://www.facebook.com/dmalan) (https://github.com/dmalan) (https://www.instagram.com/davidjmalan/)
(https://www.linkedin.com/in/malan/) (https://www.quora.com/profile/David-J-Malan) (https://twitter.com/davidjmalan)

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

Here's how to download this problem's "distribution code" (i.e., starter code) into your own CS50 IDE. Log into CS50 IDE (https://ide.cs50.io/) and then, in a terminal window, execute each of the below.

1. Execute `cd` to ensure that you're in `~/` (i.e., your home directory).
2. Execute `mkdir pset8` to make (i.e., create) a directory called `pset8` in your home directory.
3. Execute `cd pset8` to change into (i.e., open) that directory.
4. Execute `wget https://cdn.cs50.net/2019/fall/tracks/web/homepage/homepage.zip` to download a (compressed) ZIP file with this problem's distribution.
5. Execute `unzip homepage.zip` to uncompress that file.
6. Execute `rm homepage.zip` followed by `yes` or `y` to delete that ZIP file.
7. Execute `ls`. You should see a directory called `homepage`, which was inside of that ZIP file.
8. Execute `cd homepage` to change into that directory.
9. Execute `ls`. You should see this problem's distribution, including `index.html` and `styles.css`.
10. You can immediately start a server to view the site by running

```
$ http-server
```

in the terminal window and clicking on the link that appears.

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 (https://getbootstrap.com/docs/4.1/getting-started/introduction/) to get started. To add Bootstrap to your site, it suffices to include

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
```

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 view how your site looks while you work on it, there are two options:

1. Within CS50 IDE, navigate to your `homepage` directory (remember how?) and then execute

```
$ http-server
```

1. Within CS50 IDE, right-click (or Ctrl+click, on a Mac) on the `homepage` directory in the file tree at left. From the options that appear, select **Serve**, which should open a new tab in your browser (it may take a second or two) with your site therein.

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 the W3Schools HTML Validator (https://validator.w3.org/#validate_by_input) 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 the documentation for each on W3Schools.

- HTML (https://www.w3schools.com/html)
- CSS (https://www.w3schools.com/css)
- JavaScript (https://www.w3schools.com/js)

How to Submit

Execute the below, logging in with your GitHub username and password when prompted. For security, you'll see asterisks (*) instead of the actual characters in your password.

```
submit50 cs50/problems/2020/x/tracks/web/homepage
```