HTML

Learning Objectives

After completing this section you should understand:

- How to structure a basic HTML website
  - all important HTML elements such as `<h1>`, `<p>`, `<ul>` and `<li>`, `<blockquote>` and `<div>` to create websites.

Basic structure of every HTML page

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title></title>
  </head>
  <body>
    <!-- we put the content here -->
  </body>
</html>
```

The structure you see above is the most basic structure of every modern HTML webpage. The `<html>` tags tell the browser what language the file is written in. The `<head>` tag contains relevant information for the browser to help it display the webpage correctly. While the `<body>` tag contains the majority of our code, this is where we put all the content we want to display. Since the `<body>` looks a bit empty, let's fill it with some HTML code.

Basic HTML syntax

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title></title>
  </head>
  <body>
    <div>
      <div>
        <h1>Welcome to my page</h1>
      </div>
      <div>
        <div>
          <h3>Navigate my page</h3>
          <ul>
            <li><a href="/index.html">Home</a></li>
            <li><a href="/about.html">About</a></li>
          </ul>
        </div>
        <div>
          <h2>Text about me</h2>
          <p>
            Roof party fap synth meh bushwick. Stumptown cray farm-to-table
            celiac marfa, ugh chillwave DIY everyday carry. Seitan wolf
            taxidermy fanny pack, gastropub activated charcoal la croix bitters.
            Four dollar toast umami echo park post-ironic pop-up man braid.
            Try-hard normcore tofu, chartreuse next level mlkshk gluten-free
            chicharrones prism vegan glossier seitan health goth banjo.
          </p>
        </div>
      </div>
    </div>
  </body>
</html>
```

That code above turns into:

Welcome to my page

Navigate my page

- Home
- About

Text about me

Roof party fap synth meh bushwick. Stumptown cray farm-to-table celiac marfa, ugh chillwave DIY everyday carry. Seitan wolf taxidermy fanny pack, gastropub activated charcoal la croix bitters. Four dollar toast umami echo park post-ironic pop-up man braid. Try-hard normcore tofu, chartreuse next level mlkshk gluten-free chicharrones prism vegan glossier seitan health goth banjo.

Can you figure out which tag achieved what purpose?

`<h>` tags are heading tags, they help build a necessary structure to capture the visitors attention. In total, we have six heading levels to choose from—h1 to h6—to add structure to the web page. h1 is the highest heading level (and, by default, the largest in terms of font size) and h6 the lowest (and smallest).

`<p>` or paragraph tags on the other hand are used to give all the other text parts some structure. Paragraphs are usually represented in visual media as blocks of text that are separated from adjacent blocks by vertical blank space and/or first-line indentation. Paragraphs are block-level elements, which occupies the entire space of its parent element (container), thereby creating a “block.”

`<ul>` unordered list, help create structure through listing `<li>` list items. There is also the `<ol>` ordered list, which relies on numbers instead of bullet points.

`<a>` tags or anchor tags link words, pictures or any html to different a page. The `<a>` tag defines a hyperlink, which is used to link from one page to another. The most important attribute of the `<a>` element is the `href` attribute, which indicates the link’s destination.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue.
- A visited link is underlined and purple.
- An active link is underlined and red.

These can be internal, on the same website or external, a different webpage.

`<div>` tag or division tag helps us structure the page itself. The user usually doesn't see the `<div>` tag but it helps us move different parts of the page around and give it styling in general. It is generally good practice to put every building block of the page into a `<div>` tag. So in the above example we wrapped the navigation in a `<div>` since those elements make up one building block of our page.

One other very useful tag is the `<img>` tag, it allows us to add images to a webpage. It works a bit similar to the `<a>` tag, but instead of a `href` part we have a `src="address of image"` part in the tag. The `<img>` tag has two required attributes: src and alt.

Note: Images are not technically inserted into an HTML page, images are linked to HTML pages. The `<img>` tag creates a holding space for the referenced image.

Tip: To link an image to another document, simply nest the `<img>` tag inside `<a>` tags.

Exercise

A

Can you add an image below the paragraph? Use the code below, remember the `<img>` tag:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title></title>
  </head>
  <body>
    <div>
      <div>
        <h1>Welcome to my page</h1>
      </div>
      <div>
        <div>
          <h3>Navigate my page</h3>
          <ul>
            <li><a href="/index.html">Home</a></li>
            <li><a href="/about.html">About</a></li>
          </ul>
        </div>
        <div>
          <h2>Text about me</h2>
          <p>
            Roof party fap synth meh bushwick. Stumptown cray farm-to-table
            celiac marfa, ugh chillwave DIY everyday carry. Seitan wolf
            taxidermy fanny pack, gastropub activated charcoal la croix bitters.
            Four dollar toast umami echo park post-ironic pop-up man braid.
            Try-hard normcore tofu, chartreuse next level mlkshk gluten-free
            chicharrones prism vegan glossier seitan health goth banjo.
          </p>
        </div>
      </div>
    </div>
  </body>
</html>
```

B

Create a html page that lists every day of the week, remember the `<ol>` tag, it should look like this when displayed in your browser:

1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday
6. Saturday
7. Sunday

C

Create two html pages (files). Call one index.html and the other about.html.
On the index.html page put an image that links to the about.html website.
If the user clicks on the image he/she should end up on the about.html page.
On the about.html site put an email address, link that up as well. If the user clicks on the email, your mail program on your laptop should open. Remember to use the `<a>` tag and the "href" needed to link the pages.

For C, the folder structure should be:

C/
|-- index.html
|-- about.html

For all exercises, make a document on your local machine, add in the extra code you need. Ensure this file is in a local directory which is linked to your GitHub exercise repository (git init or cloned from GitHub) so that you can add and commit your work at the end of the day.

You can check the output of your code by running the file in your browser (you can drag and drop from your gui).