HTML & CSS 1

Intro

Overview

- Front End Basics
- Intro To HTML
- HTML Syntax
- Intro To CSS

Objectives

- Understand the unique purposes of HTML and CSS markup languages.
- Be able to create HTML elements including: head, body, p, div, h, nav, img…
- Be able to add styling to an HTML document using CSS selectors for element
- Understand CSS properties including font, color, height, width, padding, and margin.
- Student can use attributes to configure elements.

Front End Basics

HTML, CSS, and JS

- “Front end” refers to the parts of an application that users interact with
- HTML & CSS are the foundation of front end applications
- HTML gives content structure and meaning
- CSS is used to style and create page layouts
- We’ll cover how JavaScript interacts with them later in this unit

Intro To HTML

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is used to create the infrastructure of a webpage
- HTML is NOT a programming language; HTML is considered a “markup” language

Markup languages are used to categorize content

Without markup

```
What is HTML? HTML stands for HyperText Markup Language. Markup
languages are used to organize text into logical sections.
```

With markup

```
<h1>What is HTML?</h1>
<p>
  HTML stands for <cite>HyperText Markup Language</cite>.
</p>
<p>
  Markup languages are used to organize text into logical
  sections.
</p>
```

Note: HTML version history

Over the years, there have been many versions of HTML. As of 2014, the most up-to-date version is HTML5, which introduced a number of new features to the language. For more details, see MDN Web Docs Glossary — HTML 5

Making an HTML File

HTML files are created by ending the name of a file with an `.html` extension. This extension tells browsers and code editors to read the file as HTML.

Note: *index.html*

index.html is the standard name for a root HTML file since the browser looks for files called `index` by default.

HTML Syntax

Elements

A fundamental unit of HTML is called an ***element***

Some elements contain text

A paragraph element

```
<p>HTML is cool.</p>
```

- Start the element with an ***open tag*** (`<p>`)
- End with a ***close tag*** (`</p>`)

Some elements don't contain text content

An image element

```
<img src="/static/cat.png">
```

They comprise of a single, ***self-closing tag***

Attributes

Attributes are used to configure elements

An input of type, "password"

```
<input type="password">
```

- If you don't explicitly add attributes, you'll use the default values
- To override defaults, add `attr="value"` to the element's tag

Boolean attributes default to ***false***

- You can explicitly set their value to ***true***

  A required input

  ```
  <input required="true">
  ```

- But most prefer using this shortcut

  ```
  <input required>
  ```

A Few Common Elements

** **
div tags are used to divide a page into sections, they usually contain other elements

** **
paragraph, or p, tags are used to hold text

<h1></h1>
heading tags (h1 - h6) are used to define headings, with 1 being the largest

<img/>
image tags are used to display images

How many elements are there?

There are over 100 HTML elements. Don't worry about memorizing them, you'll end up remembering them naturally through practice.

Basic HTML Page Structure

HTML pages are made up of many elements. Here's an example of a basic HTML page structure:

```
<!DOCTYPE html>
  <html lang="en">
    <head>
      <!-- Meta tags and other external links go here, and the link and title tags below can
be used to change the icon/title of your webpage -->
      <link rel="shortcut icon" href="some-URL-or-path-goes-here" />
      <title>The Title</title>
    </head>
    <body>
      <!-- Content of page goes inside the body tag -->
    </body>
  </html>
```

Broken Down

`<!DOCTYPE html>`

- Not actually an HTML element, just a declaration
- Lets browsers know to expect HTML in this file
- Should always be included, otherwise the page may not work

`<html>`

- The root element of the file, the parent of everything else
- Everything contained in this tag will be read as HTML
- The `lang` attribute can be used here to specify human languages

`<head>`

- The head tag is a container used to contain metadata (data about data)
- This data is not displayed on the page, but is readable by browsers
- Meta tags are used inside the head to store the data
- Meta tags provide information about a site that can be used by search engines and other software
- We use attributes to specify the data such as a website description, keywords, author, and more

*Examples of **meta** tags*

```
<head>
  <meta charset="UTF-8" />
  <!-- charset stands for character set, and this information is used so the web browser knows which characters, or alphabets, are being used. UTF-8 (Unicode) covers almost all of the characters and symbols used in the world. -->
  <meta name="description" content="best website ever made" />
  <!-- the description tag has a content attribute that dictates the primary description of your site as it appears in search engines -->
  <meta
      name="keywords"
      content="greatest, best, ultimate, GOAT, website, of all time"
  />
  <!-- keywords are used to help search engines recognize what search words can be used to point users toward a site-->
  <meta name="author" content="matias perez-ferrero" />
  <!-- the author tag serves to credit the creator of a website -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <!-- a browser viewport is the area of web page in which the content is visible to the user. The width attribute can be used to set a specific width in pixels of the intended display. Here it is set to a special value (“width= device-width”) which is the width of the device in terms of CSS pixels at a scale of 100%. The initial-scale property governs the zoom level when the page is loaded for the first time. -->
</head>
```

`<title>`

- This is a metadata tag that will contain the title of the webpage
- Webpage titles appear in browser tabs

`<body>`

- The body tag is the container for all the elements that will be rendered on the page
- The head and body pattern is common throughout programming
- Heads/headers contain information about something while the body contains the actual information

Comments in HTML

- Comments help clarify what's going on, for ourselves and other devs
- Comments will not appear in the browser
- Comments in HTML begin with `<!--` and close with `-->`.

```
<body>
  <!-- This is a comment that will not be displayed on the web page -->
</body>
```

Intro To CSS

What is CSS?

- Stands for Cascading Style Sheets
- CSS is used to style HTML and create layouts
- CSS can also respond to some user interactions and create animations

What does CSS look like?

To apply styles, we have to select HTML elements and then declare what values we want the properties to have.

The general syntax for a declaration block and style declarations looks like this:

```
selector {
  property1: value;
  property2: value;
}
```

Selecting Elements

Selectors are how we specify the HTML elements that we want to style.

Tagname

The most basic way to select HTML elements is by their tagname, such as `div`, `h3`, or `button`. Selecting this way will affect all of the elements with that tagname in our HTML.

```
button {
  background-color: green;
}
```

Commas

You can use more than one selector separated by a comma to apply styles to multiple elements.

```
h1, h2, h3 {
  color: teal;
  margin: 20px;
}
```

Combinators

To get even more specific, we can use combinators.

- `>` child selector: selects direct children only
- `' '` descendant selector: selects children, grandchildren, etc.

```
p > span {
  padding: 10px;
}

div span {
  color: blue;
}
```

Where do you write CSS?

We have a few options.

Inline Styling

- You can write CSS in the body of HTML documents
- Not used for the majority of styling, only when needed

```
<button style="color:blue;font-size:12px">Button Text</button>
```

Internal CSS

- There is also the option to include a `<style></style>` tag in the `head` of HTML documents
- Again, not very common and not considered best practice

```
<!DOCTYPE html>
  <html lang="en">
    <head>
      <style>
        h1 {
          background-color: blue;
        }
      </style>
      <title>Document</title>
    </head>
    <body>
      <h1>I'm a header</h1>
    </body>
  </html>
```

External CSS

- Write CSS in an external file and link it the HTML file
- These files are called stylesheets
- This is what's considered best practice

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>My Page</title>
    <link rel="stylesheet"
href="index.css"/>
</head>
<body>
    <h1>Welcome!</h1>
</body>
</html>
```

```
h1 {
    color: red;
}
```

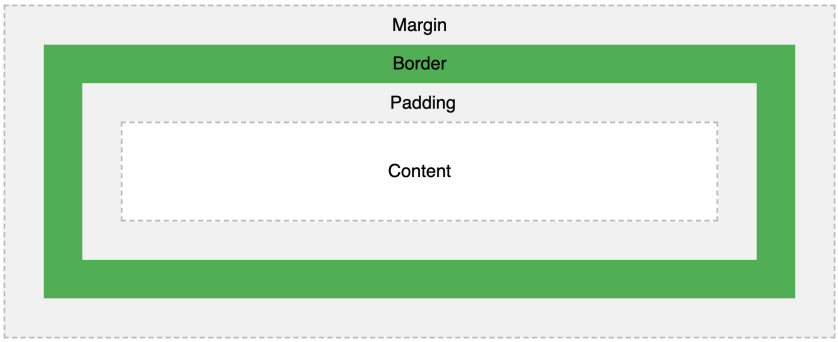
Cascading and Specificity

- CSS is read from top to bottom
- Declarations that are lower in will override previous declarations about the same property

Box Model

- All HTML elements can be thought of as boxes
- In CSS, the "box model" refers to a 4 part box that wraps around every element

Here's a visual representation:



Box Model Described

margin
The white space that separates one element from another

border
The border of the box that separates the padding from the margin

padding
The padding or buffer space between the content of an element and its border

content
The space allocated to the actual content of an element, which may be text, images, etc.

Block Elements

- Elements whose "box" automatically takes up the entire width of its parent
- Starts on a new line (they stack like blocks)
- The height is determined by the contents of the element
- The width and height can then be changed using CSS
- Can hold other block elements or inline elements
- `div`, `h1`, and `p` are all block elements

Inline Elements

- Elements whose width only occupies the minimum space necessary
- They do not start a new line
- Styling can be a little trickier
- Generally they do not contain other elements
- `button`, `input`, and `span` are all inline elements

Text Properties and Fonts

Text on a web page can be manipulated through various properties in CSS. Here are a few:

- `font-size` - changes the size of a font
- `font-weight` - controls how bold the font appears
- `color` - changes the color of the text inside of an element, **not the background**
- `line-height` - determines the space between two inline elements
- `text-align` - defines the horizontal alignment of text within the content box of an element
- `font-family` - determines the font family of the text on the page
- `letter-spacing` - determines how much white space should be between each character

External Fonts

We can also bring in fonts that are not native to a browser through `<link>` tags in our HTML. Here's an example of what that looks like:

```
<!DOCTYPE html>
  <html lang="en">
    <head>
      <link
          href="https://fonts.googleapis.com/css?family=Liu+Jian+Mao+Cao&display=swap"
          rel="stylesheet"
      />
    </head>
    <body>
      <!-- Content of page-->
    </body>
  </html>
```

Background Properties

These can be used to modify the background appearance of an element. Colors as well as images can be used as backgrounds for elements.

Examples of background properties include:

- `background-color` - can be used to assign a background color to an element
- `background-image` - can be used to assign an image as a background for an element
- `background-size` - determines the size of a background image, the image can be left to its natural size or adjusted
- `background-position` - can be used to adjust the alignment of the background image
- `background-repeat` - can be used to set if/how a background image will be repeated.

Using background properties

```
div {
  background-image: url('https://cdn.pixabay.com/photo/2015/04/23/22/00/tree-736885__340.jpg');
  background-size: cover;
  background-position: center;
  background-repeat: none;
}
```

Units

CSS has a number of units for expressing length, and they fall into two main categories, “absolute” and “relative” units.

Absolute Units

- `cm` - centimeters
- `mm` - millimeters
- `in` - inches (1in = 96px = 2.54cm)
- `px` - pixels (1px = 1/96th of 1in)
- `pt` - points (1pt = 1/72 of 1in)
- `pc` - picas (1pc = 12 pt)

Relative Units

- `em` - Relative to the font-size of the element (2em means 2 times the size of the current font)
- `rem` - Relative to font-size of the root element
- `vw` - Relative to 1% of the width of the viewport
- `vh` - Relative to 1% of the height of the viewport
- `%` - Relative to the parent element

Summary

- HTML is the markup language we use to dictate structure and content
- It is comprised of elements and their content
- CSS is the styling language we use to define styles and layouts
- It is made of selectors and declarations
- HTML and CSS work together to create web pages

The End