CEF 344: Client Server and Web development

Chapter 1: General Introduction

I. Introduction to HTML and CSS

HTML (Hypertext Markup Language) is the tool which will allow you to build webpages and **CSS** (Cascading Style Sheets) is what you use to make them look the way you want them to.

We'll explore both languages and write your first snippets of each. Let's start with something a little more familiar: web pages!

Pages

Web pages are like regular pages, but online. Thanks, Captain Obvious! This is a useful comparison to explore, though. Consider a newspaper page. On it, you have headings, sections, articles, images, and layouts which organize content that's styled to look a certain way.

Back in the old days, it was someone's responsibility to physically arrange elements on a newspaper printing press in order to tell a story. They would set attention-grabbing headlines, for example, that were complemented by content in paragraphs. They would lay out the different sections and articles on a page according to the hierarchy of the content and the story they wanted to tell via these elements.

Then, depending on the newspaper, the exact type-setting would have a certain font and characteristic style. *The aesthetic appearance of the page is different from the content itself.*

This separation is the same for web pages. You have:

- content
- its appearance

HTML is the language that handles the first concern: *creating structured content* to tell a story.

CSS covers the second concern: *customizing the appearance* of that content to visually bring it to life.

Writing the two languages separately means that it's much easier just to focus on content *or* appearance without worrying about both at the same time. For example, let's say OpenClassrooms were to entirely overhaul its brand colors and design. This would only require adapting the CSS, meaning the appearance of the site. The content (like courses) could be left alone, unchanged.

Exercice1:

1. create a file with the following content and save with the name index.html

```
<html>
  <head>
    <link rel="stylesheet" href="css/style.css" type="text/css">
  </head>
  <body>
    <h1> Hello my name is sop deffo</h1>
  </body>
</html>
```

2. in the same project folder, create a folder named css then create in that folder a file with the contents

```
h1 {
    color: blue;
    background-color: yellow;
    font-family: sans-serif;
}
```

save with the name style.css

II. Create content using HTML

Basic HTML syntax

HTML uses **tags** in order to describe each piece of content on a webpage. It's up to you, as a developer, to choose the right tags!

If this seems complex, let's think about how you format elements in a word processing document. Often, you highlight them from start to finish, and then turn them into a heading (if that's what you want).

The same is true in HTML. You must turn your content into a heading. By writing opening and closing <h1> tags around content, you indicate that the content inside should be turned into a big heading.

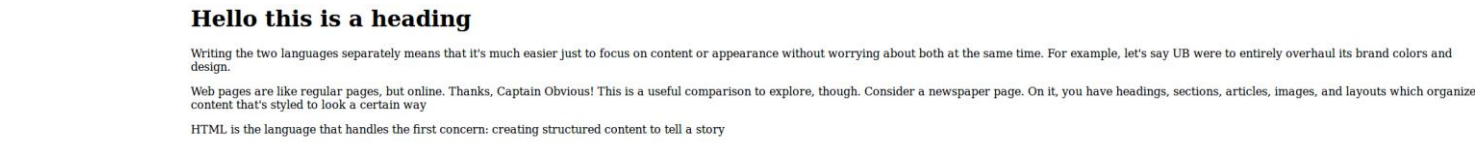
 is the opening tag (therefore, is the closing tag). The content in the middle becomes a paragraph because "p" stands for "paragraph".

Exercise2 :

here is a raw text

Hello this is a heading. Writing the two languages separately means that it's much easier just to focus on content or appearance without worrying about both at the same time. For example, let's say OpenClassrooms were to entirely overhaul its brand colors and design.Web pages are like regular

pages, but online. Thanks, Captain Obvious! This is a useful comparison to explore, though. Consider a newspaper page. On it, you have headings, sections, articles, images, and layouts which organize content that's styled to look a certain way. HTML is the language that handles the first concern: creating structured content to tell a story.



We want it to look like that

do that using h1 and p tags

III. Decorate your content with CSS

Cascading style sheets

CSS stands for Cascading Style Sheets. That's a complicated way of saying that CSS allows you to set **styles**.

Styles are like decoration. Imagine saying "I want my website background color to be yellow, the text up top to be black, the text in the bottom section of the page to be white, the images to have a bit of space around them, and the whole thing to appear differently on mobile." All of those colors and spacing and placement is style. You just write it in code, not in English!

You also use CSS to create a layout for your site, such as placing a certain section on the left of the page, determining the size of the navigation bar that appears at the top of your website, or making multiple sections appear side-by-side in one row.

CSS controls the way different pieces of content are arranged on websites. For example, the fact that there is a column with articles on the left of the New York Times and newspaper sections arranged in a horizontal bar (World, U.S., etc.) is not a coincidence! Developers created this layout with CSS:

While CSS is very powerful, **it depends on HTML.** Consider decorating a house. First, you select your element, like the bedroom walls or living room floor. *Then*, you specify how it should look: the dining room walls should be red, or the living room floor should be carpeted. If an interior designer asked how you want to decorate the dining room, and you just said "Red," they wouldn't know what you're talking about. You must say you're talking about the *walls* first.

CSS applied to HTML

There are two steps while using CSS to make your HTML look awesome:

- Identify and select the relevant HTML element (ex. paragraph, header, etc).
- Add some styles (specify how it should look).

Let's say we want the text of our paragraph to be blue:

```
p {

 color: blue;

   font-family: Arial;}
```

This may seem completely foreign, especially the curly brackets, but take another look. Can you recognize some elements?

- p: refers to the paragraph tag in HTML, which is <p>.
- color: blue: color is a CSS property. Here, we've made the text color blue.
- font-family: this is also a CSS property to set the font of your text. In the example above, it is set to Arial. We will get into the intricacies of choosing fonts later in the course.
- { }: curly brackets open and close a set of style rules in CSS

IV. Create HTML text elements

a) Heading elements

Create headings in HTML by using tags that include a lowercase "h" and a number from 1-6. The numbers 1-6 let you control the default size of your heading and therefore the importance it has in your page structure.

What do each of these tags — <h1>, <h2>, <h3>, <h4>, <h5>, <h6> — actually look like on a web page?

```
<h1>I'm an h1 heading!</h1>
<h2>I'm an h2 heading!</h2>

<h3>I'm an h3 heading!</h3>

<h4>I'm an h4 heading!</h4>
```

I'm an h1 heading!

I'm an h2 heading!

I'm an h3 heading!

I'm an h4 heading!

I'm an h5 heading!

I'm an h6 heading!

<h5>I'm an h5 heading!</h5>

<h6>I'm an h6 heading!</h6>

As the number from 1-6 increases, the heading becomes smaller!

b) Add text in paragraphs

when writing HTML, you must manually create the meaning you want your content to have by wrapping it in the proper element tags. Because you want paragraphs, you must add <p> tags. You cannot just add line breaks in your code between text that you want to appear as separate paragraphs.

You would then go ahead and throw some <p> tags around each paragraph

c)Strengthen and emphasize text

In a normal word processing scenario, you can highlight the text you want to make bold or italic and then click buttons that usually look like this:

This will make your text bold or italic. However, to do the same thing in HTML, we must adjust our thinking about text appearance.

Strong text

Bold text is considered to be "strong" in HTML. Surround the text you wish to strengthen with a opening tag and a closing tag.

Whether the text you wish to strengthen appears in the middle of a sentence or is an entire sentence itself, the tag wrapping process is the same.

Emphasized text

Strong ("bold") text is useful, but sometimes you need more nuance. Enter emphasized text, which produces a default result that is visually identical to italicized text.

The "em" (which stands for emphasis) tag in HTML makes text italicized by default.

Why don't these tags have simpler names, like <bold>?

This is a very good question. Remember that HTML controls content, not appearances. CSS controls appearances instead. By adding or tags around text you want to make stronger or

emphasize, *you can use CSS later to change the default behavior of that text.* For example, even though makes text bold by default, you can add custom CSS to make elements red instead of bold. Either way, the text is stronger. Making it "bold" isn't what counts; it's the importance of the text that matters, and how you reflect that in the text's appearance is up to you.

d)Add links and understand attributes

Hyperlinks — "links" for short — are the core of the web. They let you hop from one page to another, connecting all information online.

It's therefore imperative to learn how to create them in HTML. On the way, we'll see how links differ slightly from other elements you've seen so far!

<a> tag

Links are made with an **<a>** tag, where the "a" stands for "anchor." There is one noteworthy difference between <a> tags and the others you've learned so far.

If you follow the same steps we've used for other tags, you'd just put "your link" between an <a> opening tag and closing tag:

```
<a>my link</a>
```

However, there's one crucial piece of information missing: *which page should the link go to when clicked?*

Somewhere in this tag, we need to tell the link where it should go when someone clicks it.

Attributes

HTML elements sometimes require additional information in order to do their jobs. This additional information lives in **attributes**.

There are many, many kinds of attributes. The attributes you use will depend on the elements you're creating. Sometimes, you don't need any attributes at all. For example, when you created h1 tags, you weren't obligated to give the element additional information.

When you do need to give an element additional information, you add it within the opening tag. In the case of links, we use an attribute called **href**:

```
<a href="my_other_page.html"> my link</a>
```

href (pronounced "h" — like the letter — "ref) is short for Hypertext Reference. This is a goofy name that really only means a URL (otherwise known as a web site address, like my_other_page.html). This is the first attribute we'll learn about in this course, but it's not the last!

In the illustration above, you see that in order to add an attribute to an element, you must:

- Give name of the attribute (in our case, "href")
- Add an equal sign
- Add the contents of the attribute in quotes (in this example, the URL of the website we want to link)

e) Organize elements in a list

Lists allow you to better structure your text and order information on your web page in a way that is helpful to the reader as well as to browsers and search engines. In this chapter, we'll see two types of lists:

- unordered lists (you may be familiar with them as bulleted lists)
- ordered lists (you may be familiar with them as numbered lists)

HTML allows you to create both, of course, and the list you choose will depend on your needs. *List* get started! ?

Unordered lists

Unordered lists are perfect for when you need to show several different items, and the numerical order of them doesn't matter. You may already think of them as "bulleted lists" or other list forms where a tiny shape appears in front of each item (a dash, an empty circle, a star, etc).

For example, a general list of fruits could be shown in text form as:

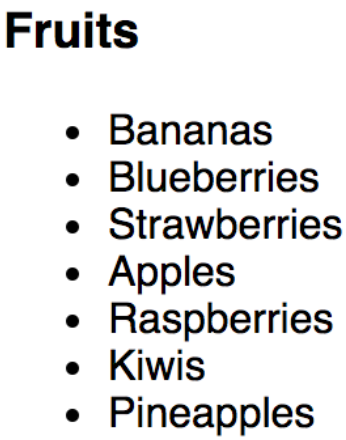
- Bananas
- Blueberries
- Strawberries
- Apples
- Raspberries
- Kiwis
- Pineapples

To create lists like this, the HTML tag <ul> is appropriate. **<ul>** stands for **unordered list.**

The <ul> tag is only used to define the type of list you'll be working with. It is not used to individually define the items in the list. In order to do so, you need another HTML tag: **<li>**, which stands for **list item.**

Therefore, you'll first create your opening and closing <ul> and tags to indicate you want to create an unordered list. Then, add your list items using
 tags inside as follows:

```
<h1>Fruits</h1>

<ul>

   <li>Bananas</li>

   <li>Blueberries</li>

   <li>Strawberries</li>

   <li>Apples</li>

   <li>Raspberries</li>

   <li>Kiwis</li>

   <li>Pineapples</li>

</ul>
```



Ordered list

Imagine you'd rather create a list of your favorite fruits. The order of that list would matter (the closer a fruit is to the top of the list, the more you like it)! Alternatively, maybe you want to create a step-by-step list of how to do something or the top 10 reasons to visit a city. In either example, the order of the list items matters.

Numbers are useful in these cases. Good news: all you need to do to end up with a numbered list instead of a bulleted list is to use a *slightly* different tag.

Create numbered lists using the tag <ol>. **
** stands for **ordered list.** As usual, wrap your list items in an <ol> opening tag and closing tag.

```
<h1>My favorite fruits</h1>
<ol>

   <li>Blueberries</li>

   <li>Strawberries</li>

   <li>Blackberries</li>
```

```
    <li>Apples</li>
    <li>Honeydew</li>
    <li>Mangoes</li>
    <li>Bananas</li>
</ol>
```

My favorite fruits

1. Blueberries
2. Strawberries
3. Blackberries
4. Apples
5. Honeydew
6. Mangoes
7. Bananas

f) Add images to your web page

A picture is worth a thousand words! That's why you'll likely include images in your web pages at some point, whether it's to display a user's profile picture, show a vacation photo, add some visual enhancement to an article, or something else!

Creating images in HTML will allow us to re-explore attributes because, as you saw with links, they are necessary for the image element to actually show anything.

The <img> tag

As you've come to expect, everything in HTML is added via a tag. Additionally, every tag you've seen up until now has had a closing tag. For example, to show a paragraph, you have both a opening tag and closing tag around the content.

The tag for images, which is <img>, is different in this regard. Image tags can just stand by themselves: <img>. No closing tag necessary!

All the information necessary for the image to be displayed is placed in the attributes of the first tag. You will often see three attributes on images:

- src
- alt
- title

The src attribute

The first attribute to include in an <img> element is src. The contents of the src attribute indicate **where** the image is stored. An image can be loaded from:

- A URL that's already online (like an image on another website)
- A file that's loaded from elsewhere in your codebase

Either way, **you're loading a file**. The only difference is how you describe its location. Take the above photo as an example.

The alt attribute

Once you have told the <img> tag where the image is located (via the src attribute), you must add a description of the image within another attribute: **alt**.

Think of "alt" like "alternative text." If someone is using a screen reader and can't see your image with their eyes, they will see the descriptive text of the image instead. So will search engines, which is important for SEO.

SEO stands for Search Engine Optimization, which is concerned with, among other things, improving the quality and the quantity of user traffic to websites

The title attribute

The title of your image appears when a user hovers over the image itself, as in the Millie Hughes-Fulford image above. See how hovering with the cursor brings up the contents of the title attribute?

Nonetheless, the "alt" attribute is more important than the title attribute, so if you only have a limited amount of time, just short-hand your title and work on a great alt description instead.

Figures

Now that you know how to insert images plain and simple, HTML also offers a *specific* kind of image tag for figures. *A figure is an image that could be moved around in your web page without dramatically altering the "flow" of the page.* This is similar to how we think of figures in an encyclopedia, for example. Because they're labelled, they can be next to one paragraph or another, and it doesn't really matter.

Figures also allow for captions. Fun fun! To insert a figure, surround your <img> tag with two other tags: <figure> and <figcaption>.

```
<figure>
   <img src="https://upload.wikimedia.org/wikipedia/commons/7/77/NASA-MHughes-Fulford.JPG"
alt="Astronaut Millie Hughes-Fulford displays the modernist Blackburn & Danne NASA logotype."
title="Millie Hughes-Fulford">

   <figcaption>Millie Hughes-Fulford</figcaption>

</figure>
```

You can even include multiple images within one <figure> tag, and they'll appear next to one another.

Using <figure> is a great way to write semantically wonderful HTML. "Figure" is more descriptive than "image", so if figure suits your content, it's a good choice to use it.

g)Use the best images possible

Image formats

We covered in the last chapter that images should often be saved in their own folder in their codebase. There are several different image formats (otherwise known as file types) available, and you should be sure to save or use image with the one most suited to your page's needs:

JPEG

Most digital cameras save photos in JPEG or JPG (they're the same thing) format because it can blend red, green, and blue to produce many colors. JPEG is a good format to use when adding a photo with many colors or with photography in general. After all, the JPEG stands for Joint Photographic Experts Group. Sample image files could be winter_landscape.jpg or family_portrait.jpeg .

GIF

Everyone loves animated GIFs. That's because this file type is perfect for animated images. It's less suited to photography though because it can't handle as many colors. You can still use it for still photos with very few colors, like simple logos. It also allows for transparency, which JPEGs do not. A sample image file could be brand_logo.gif or dancing_bear.gif .

PNG

There are two different types of PNG formats, PNG-8 and PNG-24. The latter allows for more colors, but both formats allow transparency and do not result in quality loss when saved repeatedly (unlike JPEGs). PNG stands for Portable Network Graphics. A sample image file could be simple_illustration.png or transparent_background_cutout.png .

SVG

If the name "Scalable Vector Graphics" doesn't give it away, SVG images are vector images. They can be scaled up or down and not lose any quality. However, they're only suited to simple images like logos or flags. Sample image files could be logo.svg or buea_campus.svg .

One of the most important things you can do when building great web pages is to keep your image sizes reasonable and to measure your images in pixels. If you want to show an image at a size of 200px by 200px, it should be saved at that size. This will help your pages load faster!