skillza

Future Skills for Kids

Coding with Skillza

Lesson plans for HTML and CSS

Done by: Nour Islam

Table of Contents

Introduction

WHY HTML and CSS?

HTML, stands for Hyper Text Markup Language, is a web-based language that tells your browser how to showcase text and images on a webpage. Tags is used to add the various elements and determine the placement of them.

CSS, stands for Cascading Style Sheet, is used to add styling such as fonts, colors or layouts to the webpage making it more aesthetically pleasing.

Therefore, HTML helps in building the structure of the website and CSS makes that structure comes to life.

Here are the top 4 reasons why anyone should consider up skilling himself with these foundational languages.

1) Applications across industries

Being able to build websites or having an understanding of how these languages are utilized can help build optimized, user-friendly websites. Not only are you more sought after when it comes to web-development but also in industries such as marketing, design and advertising.

2) Stand out from the crowd

Whilst editors such as WordPress and WIX exist providing you with free web-templates to build a website your website looks like everyone else's and doesn't capture your uniqueness. Armed with the knowledge of web-development you can create a website for yourself and you don't need to pay someone else to do it.

3) Easy to learn

As fundamental web development languages HTML and CSS are a great way to get started with development. This is because these languages offer simple sets of rules that define how to code and are easy to learn. In fact, HTML uses a tag-based structure which is easy to understand even without any prior technical experience, CSS uses a simplistic structure as well making the learning curve for mastery relatively easy.

4) Learn other coding languages faster

By understanding HTML and CSS, you will soon be able to get a grasp on other technologies much more quickly and easily, these include JavaScript, PHP and SQL and Python. If you're looking into becoming more serious about web development, knowing the fundamentals is a good place to start and adding additional languages to your skillset arms you with technical advantages and expertise.

Session 1

Front end development

Lesson Overview

Students are introduced to web development by understanding what the components of any website are and what the difference between the backend and the frontend is.

Simply, frontend development focuses on the user-facing side of the website. Frontend developers ensure that visitors can interact with and navigate sites easily by designing user friendly interfaces using HTML and CSS. While backend development focus on the server side of the website by performing behind of the scenes work that creates the website's structure and overall functionality using programming languages like Java, Python and Ruby.

In this course **Sublime** text editor will be used to write HTML codes, the following are the steps to get started:

1. Download sublime text editor using this link:

   https://www.sublimetext.com/download_thanks?target=win-x64

2. Install sublime
3. Create a new folder on the desktop
4. Open sublime then create a new file by pressing on file → new file
5. Save the file by pressing file → save as, then name the file session1.html and select the folder created, **the file must be saved as .html file**.

Learning Objective

By the end of this session students will be able to:

- Have a wider vision about web development.
- Differentiate between frontend and backend.
- Understand the language used in this course and its functionality.
- Understand the final output of this course.
- Download and install sublime text editor on their devices.
- Create the first html file.

Session 2

Elements and Attributes

Lesson Overview

Elements

HTML consists of series of elements, which is defined by a start tag, some content and an end tag.

< start tagname > content < /close tagname >

There are 4 main elements that should be included in any HTML file, those elements are:

- **<!DOCTYPE html>**: defines that this document is an HTML document.
- **<html>**: root element of an HTML page.
- **<head>**: contains meta information about the page like the title.
- **<body>**: container for all the webpage elements.

There some other important elements such as:

- **<p>**: defines a paragraph
- **<h1>**: defines a heading with size 1. There are 6 sizes for the heading where 1 is the biggest and 6 is the smallest.
- **<img>**: defines an image.
- **<a>**: defines a hyper link.

Attributes

Attribute provides additional information about HTML elements, its always specified in the start tag and usually come in name/value pairs like: name= "value".

For example, to define a hyper link <a> is used and within the start tag an attribute named **href** is included where the value is equal to the URL of the website.

**< a href =" https://www.google.com"> Click here to open google **
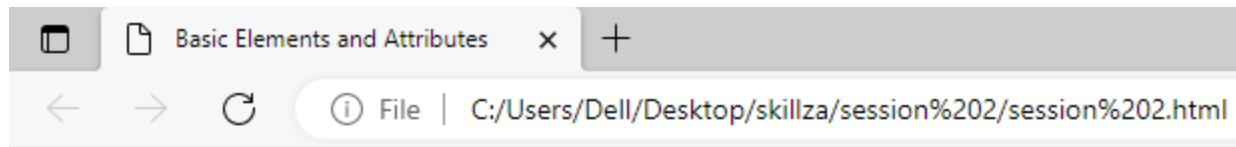
Also, to define an image <img> is used and within the start tag an attribute named **src** is included where the value is equal to the name of the image to be added. **Note that the image must be saved in the same folder where the HTML file is saved.**

< img src =" image.jpg" >

Session example:

```
session 2.html                    ×
 1  <!DOCTYPE html>
 2  <html>
 3
 4 ▼ <head>
 5      <meta charset="utf-8">
 6      <meta name="viewport" content="width=device-width, initial-scale=1">
 7      <title>Basic Elements and Attributes</title>
 8  </head>
 9
10 ▼ <body>
11      <h1>My first heading</h1>
12      <p>My first paragraph</p>
13      <img src="download.png">
14      <br>
15      <a href="https://www.google.com">Click here to open google</a>
16  </body>
17
18  </html>
```

Learning Objective

By the end of this session students will be able to:

- Identify multiple HTML elements.
- Differentiate between elements and attributes.
- Create html file and include some elements like: header, paragraph, image and hyperlink with different attributes

Session 3

Styles and formatting

Lesson Overview

<u>Styles</u>

CSS is used to add styles to the HTML elements; there is many ways to achieve that. In this session one of these ways is used which is the *Inline CSS*.

Inline CSS is achieved by adding the *style attribute* to any element which is used to define styles to that element such as color, font, size and more.

< start tagname style = "property: value;" >

The property is CSS property, the value is CSS value. There are tons of CSS properties in HTML, the following are some of the most important ones:

- **Color**: defines the text color
- **Background-color**: defines the background color
- **Font-family**: defines the font
- **Font-size**: defines the text size
- **Text-align**: defines the horizontal text alignment

Formatting

HTML contains several formatting elements for adding special effects to text.

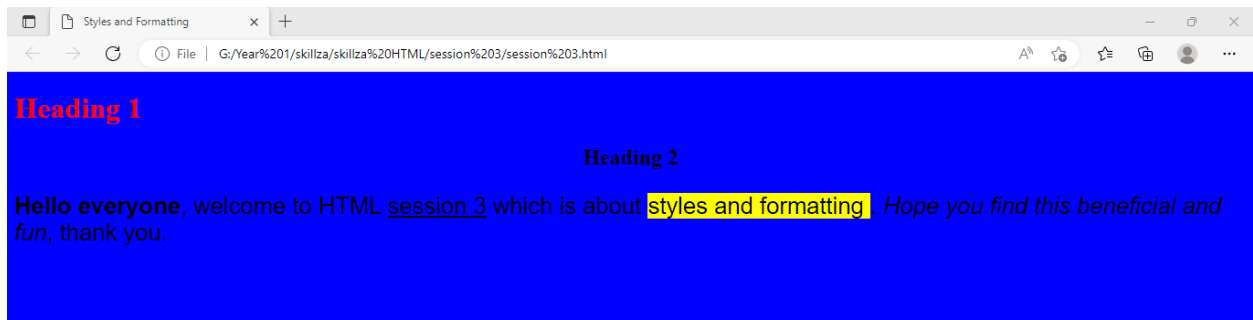Formatting elements:

- **<b>**: Bold text
- **<i>**: Italic text
- **<mark>**: Marked text
- **<ins>**: Underlined text

Session example:

```
session 3.html                    ×
1    <!DOCTYPE html>
2    <html>
3
4    <head>
5        <meta charset="utf-8">
6        <meta name="viewport" content="width=device-width, initial-scale=1">
7        <title>Styles and Formatting</title>
8    </head>
9
10   <body style=" background-color: blue;">
11       <h1 style="color: red;">Heading 1</h1>
12       <h2 style="text-align: center;"> Heading 2</h2>
13       <p style="font-family: arial; font-size: 160%;">
14           <b>Hello everyone</b>, welcome to HTML <ins>session 3</ins> which is about
15           <mark>styles and formatting </mark>.
16           <i>Hope you find this beneficial and fun</i>, thank you.
17       </p>
18   </body>
19
20   </html>
```

Output:

Learning Objective

By the end of this session students will be able to:

- Add styles to elements using inline CSS.
- Identify some of the style's properties.
- Add special effects to text using the formatting elements.

Session 4

Lists

Lesson Overview

Learning Objective

Lesson 5

Tables

Lesson Overview

Learning Objective

Lesson 6

Forms

Lesson Overview

Learning Objective

Lesson 7

Emojis

Lesson Overview

Learning Objective

Lesson 8

Divisions

Lesson Overview

Learning Objective

Lesson 9

CSS

Lesson Overview

Learning Objective

Lesson 10

Internal CSS

Lesson Overview

Learning Objective

Lesson 11

External CSS

Lesson Overview

Learning Objective

Lesson 12

Classes and IDs

Lesson Overview

Learning Objective