

# WEB222 Assignment 3

## Assessment Weight:

5% of your final course Grade

## Objective:

Practice writing HTML Containers / Media Elements, Updating the DOM with data

## Specification:

Update HTML & JavaScript code for each of the following pages according to the instructions outlined below. To begin, download the assignment3.zip file containing all of the files required for Assignment 3 from blackboard.

Unzip the zip files somewhere on your local machine. When you're ready to begin editing the files, open the uncompressed folder (assignment3) in Visual Studio Code (<https://code.visualstudio.com>) using "Open Folder".

You may test your html files in any modern browser (Chrome, Firefox, Safari, Internet Explorer, etc). Modify the pages only as stated below.

NOTE: many of the sections below ask you to use JavaScript to create content in the page. You should use document.createElement() (see <https://developer.mozilla.org/en-US/docs/Web/API/Document/createElement>) to create such content vs. writing HTML using innerHTML whenever possible.

### "Assignment 3 Home" - index.html:

Update the "Assignment 3 Home" index.html page with the following:

1. Add a **professional greeting** to the visitor, i.e.: "Welcome to my website, I will be demonstrating HTML5 principles and techniques, DOM manipulation"... and so on.
2. Add a relevant **header** as a title for the next section (step 3)
3. Add a **short paragraph** that introduces this website, i.e.: "This site contains 8 pages, including: Home, HTML Text, HTML Lists"... and so on. **NOTE:** This paragraph **must include all relevant links** to the **8 pages**.
4. Add a relevant **header** as a title for the next section (step 5)
5. Pick an HTML5 Element to research from <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>. Write a short summary about what it is used for, and give some example HTML code that shows how to use it. Make sure you use HTML entities and <code> to display your example.

## "HTML Lists" - list.html:

Update the "HTML Lists" page with the following:

1. Create two <div> elements with unique id attributes, one for fruit, the other for directories. Add headings to each of the <div>s to indicate what they are.
2. Use the **fruits** array inside the **src/js/lists.js** file to create an Ordered List containing all the fruit inside the array using JavaScript,( ie: "Apples", "Oranges", "Pears", "Grapes", "Pineapples", "Mangos") to one of your fruit container in the list.html page. The ordered list should look like the image below when complete:

1. Apples
2. Oranges
3. Pears
4. Grapes
5. Pineapples
6. Mangos

5. Add a relevant **header** as a title for the next section (step 6)

6. Use the **directory** array inside the **list.js** file (js/list.js) to write (using JavaScript) a nested unordered list containing all the files and directories (with their files) inside the array to one of your containers in the list.html page.

You will notice that files have the type "**file**" and consist of a "**name**" property, whereas directories have the type "**directory**" and consist of a "**name**" property in addition to an **array of files**. Use these properties to correctly construct your nested unordered list (using JavaScript).

The nested unordered list should look like the image below when complete:

- file1.txt
- file2.txt
- HTML Files
  - file1.html
  - file2.html
- file3.txt
- JavaScript Files
  - file1.js
  - file2.js
  - file3.js

"HTML Tables" - table.html:

Update the "HTML Tables" page with the following:

3. Inside table.html, create the following two tables, each with a relevant **header** as a title:
  - The **1st table** should be composed of the elements: `<table>`, `<tr>`, `<th>`, `<td>` and `<caption>` with meaningful table contents, i.e.: a table **of your program, courses enrolled, hours of each course etc.**
  - The **2nd table** should be composed of the elements: `<table>`, `<tr>`, `<th>`, `<td>`, `<thead>`, `<tfoot>` and `<tbody>` with meaningful table contents, i.e.: a table of your **favourite songs, the singer, and a link to the song on YouTube.**
2. Update the table.html page to include an additional container (i.e.: `<div>`) with a unique id value.
3. Add a relevant **header** as a title for the next section (step 4)
4. Use the **users** array inside the **table.js** file (js/table.js) to write a complete **3rd table** containing all the users inside the array (and an appropriate header row), to your container in the table.html page.

You will notice that users consist of the properties: **first\_name**, **last\_name**, **age**, and **email**. Use these properties to correctly construct your table (using JavaScript) with the following headers: **First Name**, **Last Name**, **Age** and **Email**. You must also ensure that all email addresses are rendered as a valid "mailto" link and will open as a new message in the user's default mail client when clicked.

Be sure to write to your container **after** the window has completely loaded. The table should look like the image below when complete:

First Name	Last Name	Age	Email
Kaitlin	Burns	23	<a href="mailto:kburns99753@usermail.com">kburns99753@usermail.com</a>
Joshua	Feir	31	<a href="mailto:josh319726@usermail.com">josh319726@usermail.com</a>
Stephen	Shaw	28	<a href="mailto:steve.shaw47628@usermail.com">steve.shaw47628@usermail.com</a>
Timothy	McAlpine	37	<a href="mailto:Timbo72469@usermail.com">Timbo72469@usermail.com</a>
Sarah	Connor	19	<a href="mailto:SarahC6320@usermail.com">SarahC6320@usermail.com</a>

"HTML Images" - image.html:

Update the "HTML Images" page with the following:

1. Find the given image with the file named **ict.png** under the "**img**" sub-folder. Show the image in your web page using a relative path. Ensure that when the image is not available, the text "**ICT School, Seneca College**" is visible on the page and make the image a **hyperlink to the ICT School website**.
2. Update the **image.html** page to include an additional container (i.e.: **<div>**) with a unique id value.
3. Add a relevant **header** as a title for the next section (step 4)
4. Use the **images** array inside the **image.js** file (js/image.js) to write 5 new **<figure>** elements to your container in the **image.html** page (using JavaScript). You will notice that each image in the array has the following properties: **caption**, **alt** and **url**. Use these properties to correctly construct (using JavaScript) your **<figure>** elements such that each element contains a valid **image** (using **url** and **alt**) as well as a **caption** underneath the image with the message contained within the **caption** property

Be sure to write to your container **after** the window has completely loaded (using JavaScript). The **<figure>** elements should look like the image below when complete (only the first 2 shown):



Red Slate Mountain



Indonesian Jungle

## "HTML5 Audio" - audio.html:

Update the "HTML5 Audio" page with the following:

1. Update the audio.html page to include an additional container (i.e.: <div>) with a unique id value.
2. Use the **audio** object inside the **audio.js** file (js/audio.js) to render an audio player in your container within the audio.html page.

You will notice that the audio object consists of the properties: **controls** and **source**. Use these properties to correctly construct (using JavaScript) your audio player with (or without) controls, and the correct source options.

## "HTML5 Video" - video.html:

Update the "HTML5 Video" page with the following:

1. **Update the video.html page to include an additional container (i.e.: <div>) with a unique id value.**
2. Use the **video** object inside the **video.js** file (js/video.js) to render a video player in your container within the video.html page. You will notice that the video object consists of the properties: **controls**, **width**, **height** and **source**. Use these properties to correctly construct (using JavaScript) your video player with (or without) controls, in the correct dimensions and using the correct source options.
3. **In the YouTube Iframe Video section, create an additional container (i.e., <div>) with a unique id value.**
4. Use one of the songs you listed in your Table of favourite songs. Find the video on YouTube
5. Follow the instructions on <https://support.google.com/youtube/answer/171780?hl=en> to embed an <iframe> to the YouTube video on your page.

## "Honesty Statement" - honesty.html:

Update the "Honesty Statement" page with the following:

1. Add date and your name in the provided places to complete the academic honesty declaration.

## Other Requirements

- All tags/attributes must be in **lower case**.
- Make sure to **update the title** with relevant text on **each of the pages**. **Each page footer must have your full name as Student name (zero will be assigned if you do not write your name)**

- All of your html files **MUST not contain any errors** when tested using the W3C Markup Validation Service: <https://validator.w3.org/>

#### Assignment Submission:

- Zip **all of your files** (i.e., your **assignment3** folder) as **assignment3.zip**
- Upload the zip file to **My.Seneca** under **Assignments -> Assignment 3** (same submission procedure as Assignments 1 & 2)
- **NOTE:** Your **HTML must not contain any errors** when validated ( <https://validator.w3.org/> )

#### Important Note:

- **NO LATE SUBMISSIONS** for assignments. Late assignment submissions will not be accepted and will receive a **grade of zero (0)**.
- After the end (11:00PM) of the due date, the assignment submission link on My.Seneca will no longer be available.