BTI-225 Assignment 3

Submission Deadline:

Sunday, March 26th, 2017 @ 11:59 PM

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:

assignment3.zip

Uncompress the assignment3.zip file 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).

"Assignment 3 Home" - index.html:

Update the "Assignment 3 Home" page with the following:

1. Add a **professional greeting** to the visitor, ie: "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, ie: "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. Add a **short paragraph** introducing HTML 5, ie: "This site utilizes HTML5: a markup language used for structuring and presenting content on the World Wide Web"… and so on.

"HTML Lists" - list.html:

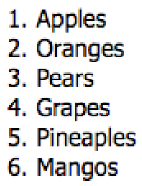
Update the "HTML Lists" page with the following:

1. Inside list.html, create any **nested list** with meaningful items, ie: a nested list of fruits, vegetables, sports, video games, etc.
   - The nested list should contain at least **one ordered list** and at least **one unordered list.**
2. Update the list.html page to include two containers (ie: <div>) with unique id values.

3. Add a relevant **header** as a title for the next section (step 4)

4. Use the **fruits** array inside the **lists.js** file (js/lists.js) to write an ordered list containing all the fruit inside the array, (ie:"Apples","Oranges","Pears","Grapes","Pineapples","Mangos") to one of your containers in the list.html page.

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



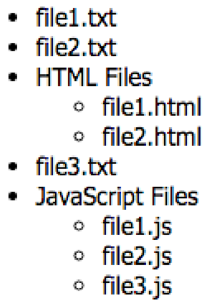
1. Apples
2. Oranges
3. Pears
4. Grapes
5. Pineaples
6. Mangos

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 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.

   Be sure to write to your container **after** the window has completely loaded. The nested unordered list should look like the image to the right 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:

1. 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 border="1">**, **<tr>**, **<th>**, **<td>** and **<caption>** with meaningful table contents, ie: a table of grades, hockey scores, movies, etc.

   - The **2nd table** should be composed of the elements: **<table border="1">**, **<tr>**, **<th>**, **<td>**, **<thead>**, **<tfoot>** and **<tbody>** with meaningful table contents, ie a table of cell phone plans, basketball scores, types of animals, etc.

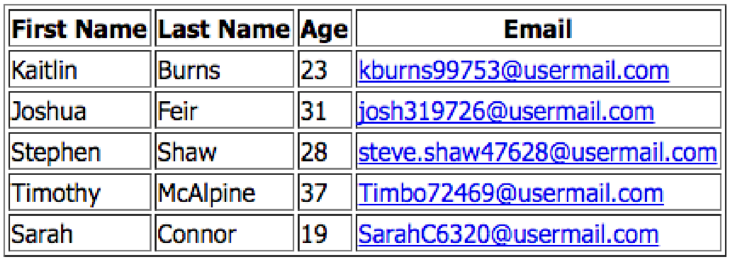
2. Update the table.html page to include an additional container (ie: <div>) with a unique id value.

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 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 to the right when complete:



First Name	Last Name	Age	Email
Kaitlin	Burns	23	kburns99753@usermail.com
Joshua	Feir	31	josh319726@usermail.com
Stephen	Shaw	28	steve.shaw47628@usermail.com
Timothy	McAlpine	37	Timbo72469@usermail.com
Sarah	Connor	19	SarahC6320@usermail.com

"HTML Images" - image.html:

Update the "HTML Images" page with the following:

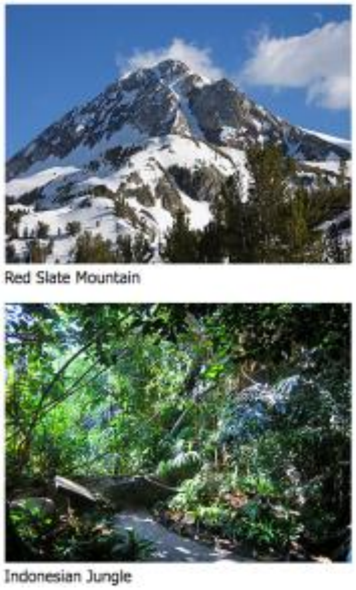
1. Find the given image with the file named **ict.png** under the "**img**" sub-folder included in the zip file. Show the image in your web page using a relative path. Ensure that when the image is not available, the text “**ICT School, Seneca College**” is visible on the page and make the image a **hyperlink to the ICT School website**.

2. Update the image.html page to include an additional container (ie: <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. You will notice that each image in the array has the following properties: **caption**, **alt** and **url**. Use these properties to correctly construct 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. The **<figure>** elements should look like the image to the right 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 (ie: <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 your audio player with (or without) controls, and the correct source options.

   Be sure to write to your container **after** the window has completely loaded.

"HTML5 Video" - video.html:

Update the "HTML5 Video" page with the following:

1. Update the video.html page to include an additional container (ie: <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 your video player with (or without) controls, in the correct dimensions and using the correct source options.

   Be sure to write to your container **after** the window has completely loaded.

"Seneca College" - seneca.html:

Update the "Seneca College" page with the following:

1. Find the image Seneca and its URL from the **http://www.senecacollege.ca** website. Without downloading this image file, show the image in your web page (ie: use an absolute link to the file) – **HINT:** Right-click on the image and choose "**inspect**" – this will take you directly to the html responsible for rendering this image and you can see the relative path to its location on the server. Generate your **absolute path** using this as a starting point.

   Lastly, ensure that when the image is not available, the text "**Seneca College**" is visible on the page and make the image is a **hyperlink to the Seneca website** (this link must open in a new tab/window).

2. Copy several paragraphs (with appropriate headings) about **Seneca College** from **Seneca College** or **Wikipedia** websites and include them in the seneca.html page.

3. Add a relevant **header** as a title for the next section (step 4)

4. Create a list containing 4 important links to pages within Seneca's website, such as important dates, information about Scholarships/Bursaries, Corporate partners, etc (make sure they open in a new window/tab)

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

6. Create a table of all courses **available this semester** for your program.

"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.

This should be the only place in the assignment where your name or other identifying information appears.

Other Requirements

- All tags/attributes must be in **lower case.**
- Make sure to **update the title** with relevant text on **each of the pages.**
- 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** (ie, your **assignment3** folder) as **assignment3.zip**
- Upload the zip file to **My.Seneca** under **Assignment 3** (same submission procedure as Assignment 2)
- **NOTE**: Your **HTML must not contain any errors** when validated (https://validator.w3.org/)