

Tutorial 5

You may use repl.it (Node.JS template), Visual Studio Code or simply use your browser to work on your exercises.

Download and refer to the files provided in tutorial5.zip . Once completed, submit the files to Brightspace as a .zip

Reference:

- [HTML Basics](#)
- [Semantic Web](#)

Exercise 1: Create a simple recipe HTML page

In this activity, you will start to make your own recipe. You will be introduced to five common HTML tags; headings, paragraphs, image, ordered lists and unordered lists.

Exercise 2: Create a media recipe HTML page

Based on your earlier code, add in embedded video resources, tables to showcase more info
Upgrade it using semantic HTML

Exercise 3: Semantic

Instructions

With reference to the semantic tags taught, revamp the html using HTML5 semantic tags instead.

Refer to the html files provided in tutorial5.zip

Steps

1. Place the ul, h1 and p elements at the top of the page inside an element that represents a group of introductory content.
2. Next, place the paragraphs at the bottom of the page inside an element that typically contains information about the site, copyright data or related links.
3. Place the content between the header and footer inside an element that represents standalone sections of content.
4. Finally, place the top `` inside an element that represents a major section of navigation.

Exercise 4: Applying CSS

Instructions

This is a simple exercise to get you going with CSS. Try it out and get some creative juice flowing.

Steps

1. Use the existing external CSS file "style.css"
2. Add the css file using the `<link>` method
3. Add a few CSS properties (color, background-color, font-family, text-align) to the content created in the last activity
4. Try messing around a little more :)

Exercise 5: Class & ID

1. Apply the #heading ID to one appropriate element to affect the result A unique style is applied to the heading area.
2. Apply the class styles to the appropriate elements to affect the result class="?" to your tag elements

Note

1. Notice that block-level child elements inherit the applied style

Refer to the html files provided in tutorial5.zip

Exercise 6: Simple Calculator

There are two textboxes. After the user types in two numbers in the two textboxes, they will click on the Calculate button.

Using the alert function, display the sum of the two.

Steps

- Extract out the value from the first textbox
- Extract out the value from the second textbox
- Use alert to display the sum

Exercise 7: TaskMaster

Solve the following questions and place them in the javascript file

Learning Objective:

- Cycle through nodelists and manipulate nodes accordingly
- Write code using a loop to select only the elements with the class li.hot and change all the class attributes of the elements to cool.
- Write code using the getElementsByName() function and check if there are 3 or more elements found with the class attribute of hot. Run through the nodelist retrieved using a loop and use a if statement to check. Then change only the 3rd element class attribute to cool.
- Write code using the getElementsByTagName() function and check if there are at least one list item li found with the class attribute of hot. Then change only the 1st element class attribute to "cool".

Exercise 8: Character Counter

Instructions

- Select the `text` class and assign to variable `text`
- Select the `words` class and assign to variable `words`
- Select the `characters` class and assign to variable `characters`
- Assign an event listener to the textarea. The event listener listens to the user's input
- Retrieve the user's input value and display the number of characters and words
- Write suitable statements that updates the character count