COMP 1073 - Summer 2020

COURSE PROJECT - PHASE ONE
- CREATE YOUR OWN INTERACTIVE TO DO LIST

In Module One, we started to really get to know the basics of JavaScript - including how to declare and initialize variables, the difference between var and let, different data types and how to use built-in browser functions, create custom functions (both named and anonymous) and how to use conditionals and loops.

In Module Two, we began to explore the DOM and our ability to use the DOM to interface with and manipulate our webpages. We soon realized that everything is an object and these objects provide us with access to methods, properties and events.

Your Task:

Using your knowledge of JavaScript basics, the DOM, selecting elements, DOM traversal and manipulation and working with browser events, create a simple interactive to-do list application that uses a few simple HTML elements, some CSS, and JavaScript.

Instructions :

1. Build a simple HTML template for the application (valid HTML, CSS and an external JavaScript file).

2. Create an <h1> element that features the name of your to-do list.

3. Build an <input> element that allows someone to type out a to-do item, and a <button> element to add it to the list.

4. Add an event handler that captures the value of the above <input> element and creates a new HTML element that features a checkbox (<input type="checkbox">), the text of the new to-do item, and a <button> element to delete the to-do.

5. For each to-do item, build an event handler that listens for the change event for the checkbox (https://developer.mozilla.org/en-US/docs/Web/Events/change) – when a todo is checked, the item is styled with a CSS text-decoration: line-through property, and it is moved to the bottom of the list of to-dos.

6. Also, include an event handler that removes the respective to-do item when a user clicks on the delete <button> element.

7. Once you've completed the functionality of the interface, add some CSS to make the page visually attractive (don't spend too much time on this – the focus is the JS).

8. Ensure that all your HTML, CSS, and JS is well-commented, formatted, and organized.

9. Publish your page on a web server (AWS, Github pages or your own web server).

TAKE IT FURTHER:

1. Figure out how to generate a satisfying 'ding' sound when a to-do item is checked.

2. Add some colour to the interaction of the user interface – maybe checked off to-do items fade to green, and deleted ones fade to red before vanishing.

3. Be original – what might make this application a bit more fun?

Project Objectives:

- demonstrate understanding of basic JavaScript syntax, integration, functionality & proper formatting
- apply and demonstrate understanding of the Document Object Model (DOM), selecting DOM elements, DOM traversal and DOM manipulation
- apply and demonstrate an understanding of browser events, event handlers and event listeners and registering event handlers.

Project Assessment:

You will be assessed on the following:

	Missing Something	Getting There	Great Work	Awesomesauce
JavaScript **(5 marks)**	Developer used JS that is not valid, properly structured, formatted and commented. Variables, arrays, functions, loops, and conditional structures that are not valid or appropriate to the functional requirements. (0 - 0.5 marks)	Developer used JS that is somewhat valid, properly structured, formatted and commented. Variables, arrays, functions, loops, and conditional structures are somewhat valid and appropriate to the functional requirements. (1 - 2 marks)	Developer used JS that is mostly valid, properly structured, formatted and commented. Variables, arrays, functions, loops, and conditional structures are mostly valid and appropriate to the functional requirements. (3 - 4 marks)	Developer used valid, properly structured, formatted and commented JS. The JavaScript includes properly-built variables, arrays, functions, loops, and conditional structures as appropriate to the functional requirements. (4.5 - 5 marks)

Functionality **(5 marks)**	The basic functionality of the application is not complete. Most functionality requirements are not working/not present. (0 - 0.5 marks)	The basic functionality of the application is almost complete. Some functionality requirements are not working/not present. (1 - 2 marks)	The basic functionality of the application is complete. (3 - 4 marks)	The basic functionality of the application is complete. There are additional features and/or functionality that go beyond the basic application requirements. (4.5 - 5 marks)
Working with the DOM **(5 marks)**	Developer does not demonstrate understanding of working with the DOM and is not able to select elements, and traverse and manipulate the DOM. (0 - 0.5 marks)	Developer demonstrates a fair understanding of working with the DOM and is able to select elements, and traverse and manipulate the DOM. (1 - 2 marks)	Developer demonstrates good understanding of working with the DOM and is able to select elements, and traverse and manipulate the DOM. (3 - 4 marks)	Developer demonstrates strong understanding of working with the DOM and is able to select elements, and traverse and manipulate the DOM. (4.5 - 5 marks)
Events **(5 marks)**	Developer does not demonstrate understanding of browser events, including registering event handlers (0 - 0.5 marks)	Developer demonstrates a fair understanding of browser events, including registering event handlers (1 - 2 marks)	Developer demonstrates good understanding of browser events, including registering event handlers (3 - 4 marks)	Developer demonstrates strong understanding of browser events, including registering event handlers (4.5 - 5 marks)

Project Due Date:

Monday, June 29th at 11:59pm

Project Weight:

10% of final grade

Submission Details:

Please submit all code files as a zipped folder on Blackboard. Please also submit a valid link to your published page with your submission. Your work will not be graded unless it is posted on a web server.

!important

Please ensure that any work you submit is your own unique and independent work. You are not permitted to collaborate with other learners or use code found online (tutorials, resources etc.)

Work submitted that is found to be not your own unique, and independent work will be subjected to a grade of 0 and considered to be academic misconduct.