COMP1073 – Client-Side JavaScript

ASSIGNMENT 3
To-Do List

DESCRIPTION

Assemble a simple to-do list application that uses a few simple HTML elements, some CSS, and JavaScript to handle the functionality. **DO NOT COPY IN-CLASS CODE OR CODE FROM ANY THIRD-PARTY SOURCE – THIS REQUIRES YOUR OWN ORIGINAL WORK (HTML, CSS, AND JS).**

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 to-do 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. Upload your application to a webserver.
10. Post the URL to **Assignment 2** on Blackboard.

TAKE IT FURTHER

1. Figure out how to generate a satisfying 'ding' sound when a to-do item is checked.
2. Add some color 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?

Image thanks to j4p4n - https://openclipart.org/detail/281418/epa-checklist

SUBMITTING YOUR WORK

Upload the HTML, CSS and any media files to a web server and then post a link to the document in the assignment on Blackboard. *Your work will not be graded unless it is posted on a web server*.

EVALUATION

Please refer to the chart (assessment rubric) below. You will be graded on *how well* you followed the *assignment instructions* from both a *technical* and a *creative* perspective as spelled out in the rubric. This assignment is weighted in terms of your final mark as indicated on the course syllabus.

Criteria	Mark
TECHNICAL EVALUATION	
HTML, CSS, and JavaScript is valid, properly structured, formatted and commented.	**/5**
JavaScript is well organized, with appropriate variable names and helpful descriptions for each section of the script.	**/5**
The basic functionality of the application is complete.	**/5**
The JavaScript includes properly-built *variables, arrays*, *functions*, *loops*, and *conditional* structures as appropriate to the functional requirements.	**/5**
There are additional features and/or functionality that go beyond the basic application requirements.	**/5**
CREATIVE EVALUATION	
The application is fun, intuitive, and easy-to-use – a pleasurable user experience.	**/5**
The JavaScript is thoughtfully organized, and it offers a creative solution to the application requirements.	**/5**
There are other innovative or imaginative elements of the script that go beyond the basic application requirements.	**/5**
TOTAL	**/40**