Assignment 7: JavaScript continued

Are you having trouble remembering what you have to do? Great news everyone! In this assignment you will make a todo list.

Your submission shall consist of two files, an HTML page named `todo.html` and a JavaScript file named `todo.js`. Go ahead and create them, and we will get along with this exciting exercise.

Deliver the two files in a zip file on Blackboard.

Part 1: Prepare the HTML file (10%)

The HTML page should consist of a title, one button for adding tasks, a correlating input field, an output element, and a list where the tasks are presented. Make sure to add IDs on the appropriate places.

When the user enters the page, the input file shall automatically get the focus.

Connect your HTML page with the JavaScript file. The rest of the task, except part 4, should be done in the JavaScript file.

Hint: The button and input field could both be input elements. An input element, with type set to submit, could act as the button.

Part 2: Implementing Functionalities (20%)

Now you will start to implement some functionality.

When the user writes a task in the input field and clicks the add button, the task shall be added to the empty list with a checkbox in front of it. Make a JavaScript function named "addTask()" that does this. Place the newest task on the top of the list.

Hint: Use eventlisteners to listen on button clicks.
Hinter: If you want to remove the list bullet, you can do so using CSS.
Hintest: Use `preventDefault()` so that the page does not reload itself if you use a form.

Part 3: Storing Users' Activities (20%)

Instead of simply showing the tasks in the DOM, we also want to store them as an object.

Create an empty list in `todo.js` called `tasks`. Then expand your function "addTask()" so that it adds a task object to the list. The object should consist of the attributes: * timestamp * task

The timestamp could be the number of milliseconds between 1 January 1970 00:00:00 UTC and now (i.e. use the JavaScript Date object), or any other unique identifier using the Date object. The task should be a string.

You can check that this works by using console.log on the list 'tasks'.

Part 4: Adding more Functionalities (25%)

Part 4 should be completed using CSS.

If the user checks the checkbox, the following should happen: * the task should have a line through it

Should the user uncheck the checkbox, the task should be reverted to it's default state (no line through).

Part 5: Making it Interactive (25%)

We have not forgotten about the output element. Every time you update a task (check it, uncheck it or add it) the output element should be updated with the number of completed tasks and the number of tasks. E.g. if you have 7 tasks and 3 are completed, it should read '3/7 completed'.

The encouraged last part

Style your todo list to look smashing using CSS. This part will not be graded, but is a nice task when you do not want to do your assignments in other courses.

Deliverables

For this assignment deliver `todo.html`, `todo.js` and optionally `todo.css` as a zip file into Blackboard before the deadline. Submissions are ONLY accepted via Blackboard. We DON'T accept late assignments. Emails or any other messages with late assignments are automatically discarded without further communication