Assignment 8: JavaScript

Create a directory assignment8 under your personal course website to store your solution for this assignment.

1. Writing to the console. (20%)

Create an empty HTML document and add a `<script>` element in the `<head>`. Write a loop which prints the integers from 1 to 20 in the developer console using `console.log`.

To view the output, open your file in a browser and open the developer console.

- Firefox: Ctrl+Shift+K
- Chrome: Ctrl+Shift+J
- Safari: Command+Option+I (the developer menu must be enabled in “Preferences > Advanced”)
- Opera: Ctrl+Shift+I, click “Console”
- Internet Explorer: F12, click “Console”

2. Arrays and built-in functions (30%)

Create an empty HTML document and add a `<script>` element in the `<head>`. Add an event listener to the `load` event using `addEventListener` to run some code when the page is loaded.

Within this code, initialize an array with at least 10 strings. They can be titles of movies, inspirational quotes or anything you like.

Write a script that, when the page is loaded selects a random string from the array and displays it on the page. You can use an `<output>` element to mark the spot where the string should appear.

- Setting the `id` attribute of the `<output>` element will allow you to access it using `document.getElementById`. You can then set the `value` property to alter the content of the element.
- To do random selection, you can use the built-in function `Math.random()` to get a random number in the interval [0, 1).
- Since JavaScript only has floating-point numbers, you’ll probably need to do some rounding before indexing the array. The built-in functions `Math.round`, `Math.floor` or `Math.ceil` may be useful here.

3. Event handling and setting timers (50%)

For this exercise, you will implement a simple stopwatch. A simple user interface is provided in `stopwatch.html`. You will create a new file `stopwatch.js` containing the JavaScript necessary to turn this into a functional stopwatch.

The start/stop button should start the stopwatch if it's not running, or stop it when it's running. The reset button should reset the elapsed time to zero.

- Use `window.setInterval` to set a callback to be called every `x` number of milliseconds to update the elapsed time.
- Keep the timer object returned by `window.setInterval` in a variable and use `window.clearInterval` when you want to stop the callbacks.
- There is no need to change the provided html file.
- One way to go about this problem is to use the `Date()` object's `getTime()` method and check your current time against the time when the stopwatch started. Meaning: `elapsedTime = currentTime - startTime`.

Submission

Submission can be posted in your course page under an appropriate sub-page (e.g., assignment-08) and also has to be uploaded as a zip file into It's Learning before the deadline.