Assignment 5: JavaScript introduction

Write all the JavaScript within the same provided file `script.js`. Write your code on the places indicated by the comments.

Comment your code so that the teaching assistant understands what is happening, e.g:

```
/* Declares the constant hey */
const hey = 'This is a constant named hey'
```

Part 1: Intro (10%)

Download the provided files `index.html` and `script.js`. Connect the JavaScript file with the HTML document using a relative path. Where in the HTML document would you include it, and why?

Write your answers as a comment below your inclusion.

Part 2: Hello World (10%)

Below `/* Part 2 */` in `script.js`, write a `for loop` which prints the integers from 1 to 20 in the developer console. It shall not print the numbers 0 and 21.

You can check the result by opening the developer console in your web browser.

Part 3: Combine the `for loop` with the `if statement` (20%)

Now you will implement a simplified version of a common interview test, combining the `for loop` with an `if statement`. Write your code below `/* Part 3 */`.

You will see the array 'numbers'. Use a `for loop` to go through that array. For each element in the array, print the number to the console using console.log. If the number is divisible by 3, instead of the number print the string 'eple' to the console. Should the number be divisible by 5, print the string 'kake' to the console instead of the number.

Hint: to check if a number is divisible by another number, use `modulo (%)`. E.g. To check if the number 12 is divisible by 6, you can write `12%6`, which will return 0.

Comment from the author: If a number is divisible by both 3 and 5, either 'eple' or 'kake' can be printed, this is dependent on your implementation.

Optional: If you want an extra challenge you can add an extra conditional that checks if the number is divisible by both 3 and 5. If it is the word 'eplekake' should be printed out. This part will not be graded, but is part of the normal test.

Part 4: DOM Manipulation (15%)

What is a page without a title? Well, what is a snake without its head? Pretty darn boring. Instead of simply adding the title in the HTML document, we will manipulate the DOM to add it. Sounds scary? It's not. In your HTML document `index.html`, you will see a `h1` element with the id `title`. Use JavaScript to access this element and add the string 'Hello, JavaScript' to it.

As you might have guessed, write your code below `/* Part 4 */`

Hint: you can use `.innerText` to add text content to an element.

Part 5: Use Functions (25%)

We have manipulated the DOM by adding content to an element. Now we will use JavaScript to change a style attribute of two `divs`.

The provided HTML file contains three buttons and two divs. In `script.js` you will find three functions, one for each button. When clicked each button should activate its corresponding function, as listed below.

- Display: none -> changeDisplay()
- Visibility: hidden -> changeVisibility()
- Reset -> reset()

Below is a description of what each function should do. You will need to implement them correctly.

changeDisplay() should change `#magic`'s `display` attribute to `none`.

changeVisibility() should change `#magic`'s `visibility` attribute to `hidden`, and the `display` attribute to `block`.

reset() should set `#magic`'s `display` and `visibility` attributes to their default values, that is `block` and `visible`, respectively.

You can make changes to the HTML document if you need to do so.

Now open your webpage and look at the magic. If you have done it correctly, the div with number 1 should be removed from the document flow when you press the button 'display: none' and be invisible (but take up space in the document) when you press the 'visibility: hidden' button.

You have not only learned to trigger a function with a button, and update the CSS attribute of an element; but you have also learned the difference between `visibility: hidden` and `display: none`. I'm glad you got out of bed today to experience this!

Part 6: Create Dynamic Webpages (20%)

Remember when you made an HTML list? Well, now you will populate that list using JavaScript. This way you can easily create dynamic webpages with your hands behind your back (this is not true).

In the HTML document, below `<h2>Part 6</h2>`, you will find an empty unordered list with the `id='tech'`. In the JavaScript file, below `/* Part 6 */`, you will find the array technologies.

Loop through the array, and for each element, add it to that list in `index.html`.

Hint: One way to solve this task is to combine each list element with the HTML tags `<li></li>`. You then have to concatinate string. In JavaScript you use the operator `+` to concatinate strings. E.g.:

```
const hey = 'Your Name'
console.log('Hello, ' + hey)
```

will give the string `Hello, Your Name`.

To solve this task, you can also use the JavaScript function `appendChild()`, which you can read about here. The important part is to create the list dynamically.

Deliverables

Submission should be uploaded 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.