WEB APPLICATION ARCHITECTURES

Accessing DOM elements

Objectives

In this exercise you will learn how to access DOM elements using code

Reference material

This exercise is based on material from the “Accessing DOM elements” chapter.

Overview

- In this lab you’ll exercise accessing DOM elements using code.

Estimated duration

The estimated duration for this lab is 30 minutes.

Completed solution

There is a completed solution for this lab.

Step by step instructions

1. Open the completed solution for Lab 4. (If you didn’t manage to complete Lab 4, then you can open a copy of the model solution instead.)
2. Place a on your page in the <body> section.
3. Give this div an ID like divNames
4. Modify the getNames function as follows:
    a. Instead of using document.write for each iteration of the loop, ensure that getNames has a text variable called result, and add each name to the result variable.
    b. After the loop completes, modify the innerHTML of the that you created in steps 2 and 3, and set it equal to the result variable
    c. Question – what would happen if you modified the innerText property of the instead of the innerHTML property? Try it yourself and see if you were right!
5. Run and test your code

Changing CSS

6. Add a class called bigNumber (or any name you like) to the <style> element of the HTML page. Change the style of big numbers, perhaps by changing their colour or their font weight.
7. Add a function called highlightBigNumbers to the <script> section of your HTML page.
8. In this function, you should
    a. Use the document.getElementsByTagName function to get an array of all <td> elements on the page
        - Shortly, we will write some code which runs this function, but we will ensure it runs *after* the multiplicationTable function has been run. Refresh your memory on what the multiplicationTable function does. It uses <td> elements to contain each of the numbers in the table. The hightlightBigNumbers function will go through each of these <td> elements to access each number
    b. Create a loop that goes through each of the <td> elements
    c. Inside the loop, get the element's innerText property, and see if it's bigger than 10.
        - Note that JavaScript allows you to compare an integer to a string, but it's not recommended. A better solution is to do :
          ```
          if (parseInt(allNumbers[i].innerText) > 10) {…
          ```
    d. If it is bigger than 10, then set the element's class to bigNumber
9. Now, in the body of the HTML, after the call to multiplicationTable, add a call to highlightBigNumbers.

When you have tome

10. The steps we followed above, using getElementsByTagName, are not ideal, because they will have unexpected effects if there are other <td> elements on the page. Modify your code so that the <td> elements in the table are given a class when the table is generated (in the multiplicationTable function). Then modify the highlightBigNumbers function to modify only the elements with this class, using getElementsByClassName.

    Hint: you can't use the className property of an element to set its class if it already has another class – this will result in the other class being removed, which will have unwanted side effects. Instead, you can add a class name as follows:
    ```
    element.className += element.className ? " bigNumber" : "bigNumber";
    ```

    If the element already has a class name, this will append the new class (with a space before it); otherwise, it will set the new class.
11. Modify the highlightBigNumbers function. When it finds a big number, after setting the element's class, also use the elements "style" property to change another aspect of its style, e.g.:
    ```
    allNumbers[i].style.backgroundColor =…
    ```

Does it seem a bit strange to set some aspects of the element's style with a class, and other aspects with the style property? We wouldn't normally mix these two methods like this – we're only mixing them in this lab to show you different ways of achieving the same thing.