
ASSIGNMENT 2- JAVASCRIPT

This assignment has to be tested in the Automated Test Environment introduced in class. Check the ATE-settings.pdf document for more information. You will need to install the jsdom library:

- Open the Command window (cmd)
- Change directory to the **folder of your project**
- Install jsdom with npm:

`npm install jsdom`

```
C:\Users\Lucia\Desktop\Lenguajes de marcas 21-22\Unit4.Javascript\Assignment2-DOM>c:\Tools\nodejs\npm install jsdom
added 59 packages, and audited 60 packages in 9s
found 0 vulnerabilities
C:\Users\Lucia\Desktop\Lenguajes de marcas 21-22\Unit4.Javascript\Assignment2-DOM>c:\Tools\nodejs\mocha .
```

Submit the *practica2.js* document filled with the code of the functions corresponding to each exercise (*exercise1* in *misterio1*, and so on)

```
function misterio1(html, id){
return true;
}
```

The results have to be correct (check spaces or capital letters) as they will be tested in an automatic process. It is highly recommended to study the tests to better understand the exercises.

Compulsory exercises: Number 3, 7, 8, 10, 12, 14

Exercises:

1. The function receives two parameters: a html document and an id. It should return the content of the HTML element that has that id. If there is no element with that id it will return: "No existe ese id";

Example:

```
<p id="test" class="clase">ask</p>
```

misterio1(document,"test") will return "ask"

2. The function receives two parameters: a html document and a string with the name of a class. It will return the color of the first HTML element of

that class. If there is no element with that class, it will return "No existen con esa clase";

Example:

```
<h1 class="rojo" style="color:red">Este es un h1 rojo</h1>
```

misterio2(document,"rojo") will return "red"

3. The function receives two parameters: a html document and a string with the name of a html tag (ex: "h1")

If the document does not contain elements with that tag, it will return "No existen con esa etiqueta"

Otherwise, the function will change the id to each of the elements of that tag, to *el<label-name><number>*.

Example with h1:

the first h1 would be elh11, the second elh12, the third elh13, ...

4. The function receives three parameters: a html document, a string with the name of a html tag (ex: "h1"), and a string with the name of a class. Use *querySelector* to return *true* if there is an element with that tag and class or *false* otherwise.

5. The function receives two parameters: a html document and a string with the name of a class. It will return the number of elements in the document that have that class.

6. The function receives two parameters: a html document and a string (id). It will add a h1 tag with that id to the body of the document

7. The function receives four parameters: a html document, a string (*id*), a string (*text*) and a string (*parentId*). It will add a h2 tag with the *id* and the *text* as a child of the *parentId* element, if exists. If it does not exist, it will add the new element to the body.

8. The function receives one parameter: a html document. It will add to the body section an unordered list with three list items (li) belonging to the class "linea" and with value "HOLA".

9. The function receives one parameter: a html document. It will modify a list (similar to the one created for exercise 8):

- Add the id "primero" and the text "primero" to the first li of the list

- Add the id "segundo" and the text "segundo" to the second li
- Add the id "tercero" and the text "tercero" to the third li

10. The function receives six parameters: a html document, a string(tag), a string(id), a string(color), a string (background color), and a string (text). It will create an element of type tag, with id, color, background color and text, and it will add it to the body.

11. The function receives two parameters: a html document and a string (id). It will change the list of classes of the element with that id so it will belong only to the class "clase". If there is no element with that id it will return "No existe ese id";

12. The function receives two parameters: a html document and a string (id). It will return the number of child nodes of the element with the id. It will return 0 if the element does not exist.

13. The function receives two parameters: a html document and a string. It will set the string as the title of the document

14. The function receives two parameters: a html document and a string with the name of a class. It will remove all the elements of that class, and it will return a string with the number of elements deleted: "Eliminados:x"