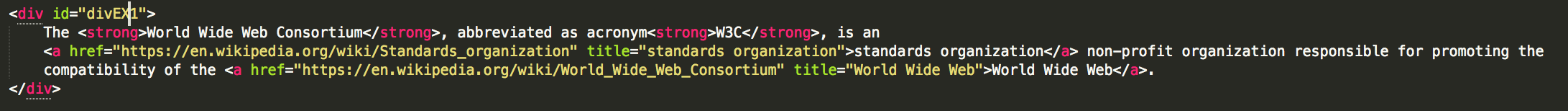
Exercise Javascript Manipulation HTML Code part 2

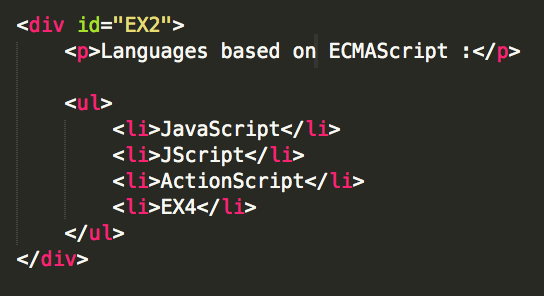
**Evaluated skills :**

*→ Manipulation HTML Code part 2 Javascript lesson*

**Exercise 1 :**For this first exercise, we propose to recreate this HTML structure mixed with various elements such as <a> and <strong>. It's quite simple, with all the text nodes! 

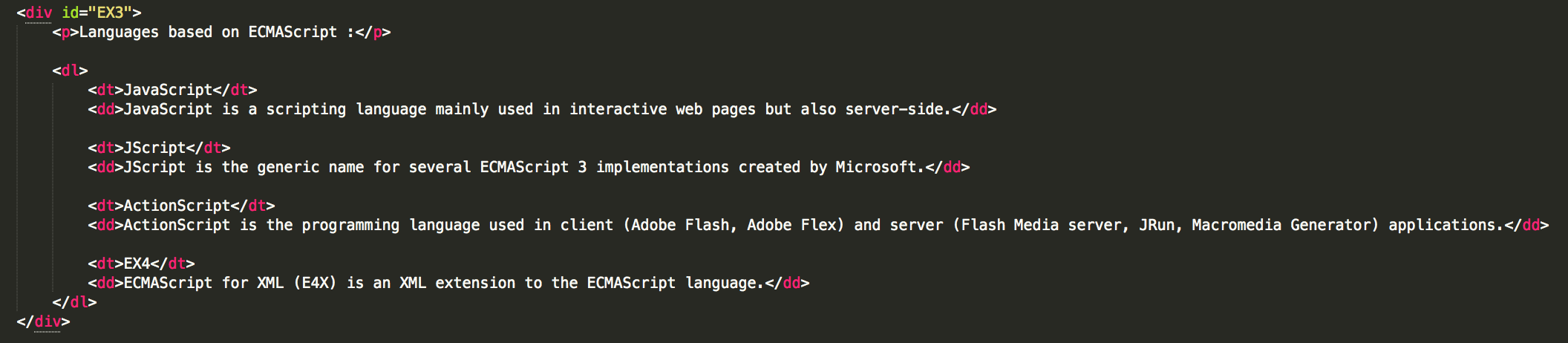
**Exercise 2** :

We are not going to create four <li> "by hand" elements ... Use a loop fo*r! A*nd remember, use an array to define the textual elements.



**Exercise 3 :**

Here is a slightly more complex version of the previous exercise. The schema of operation is the same, but here the “languages array” will contain literal objects, and each of these objects will contain two properties: the <dt> node and the <dd> node.

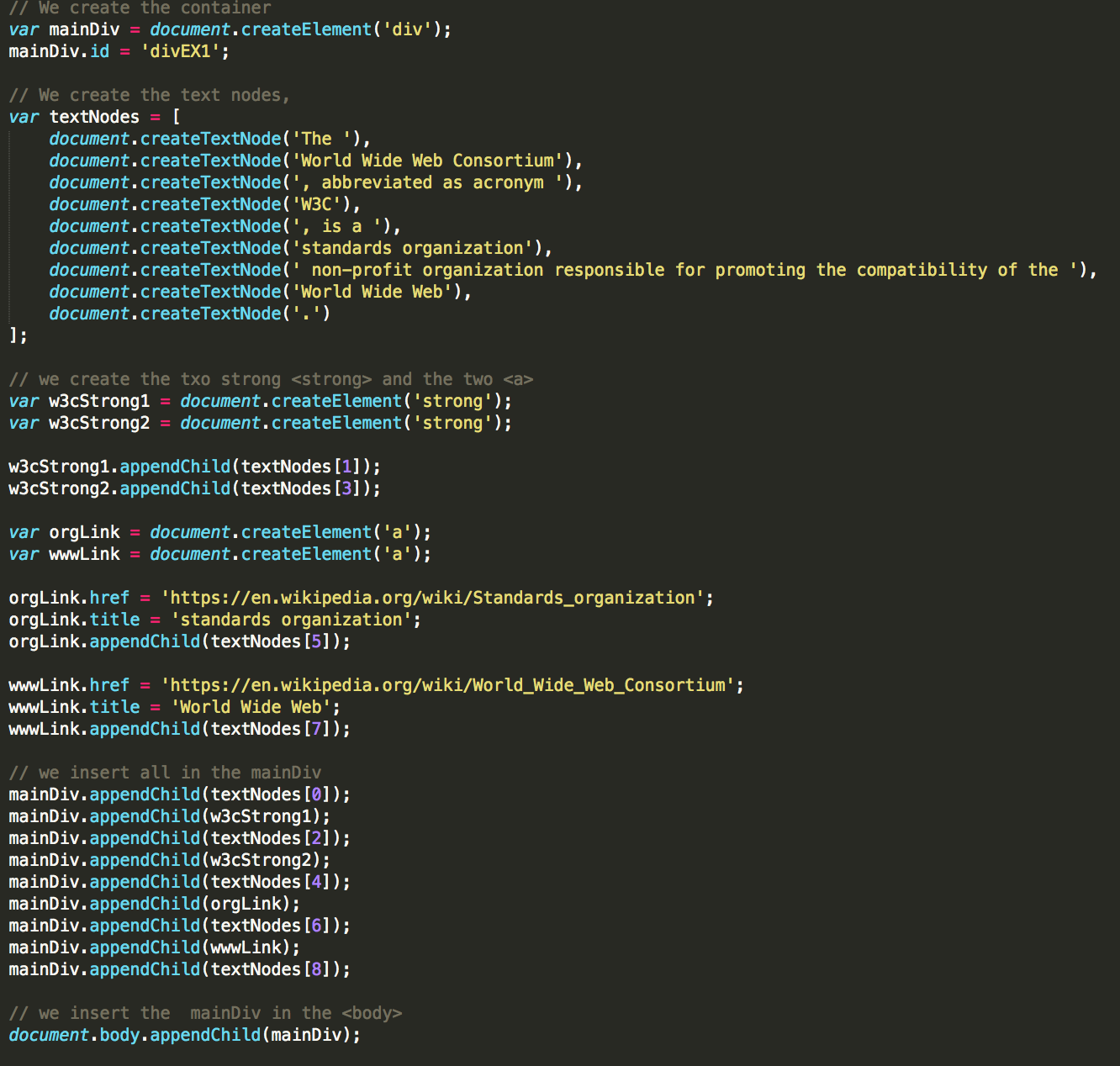
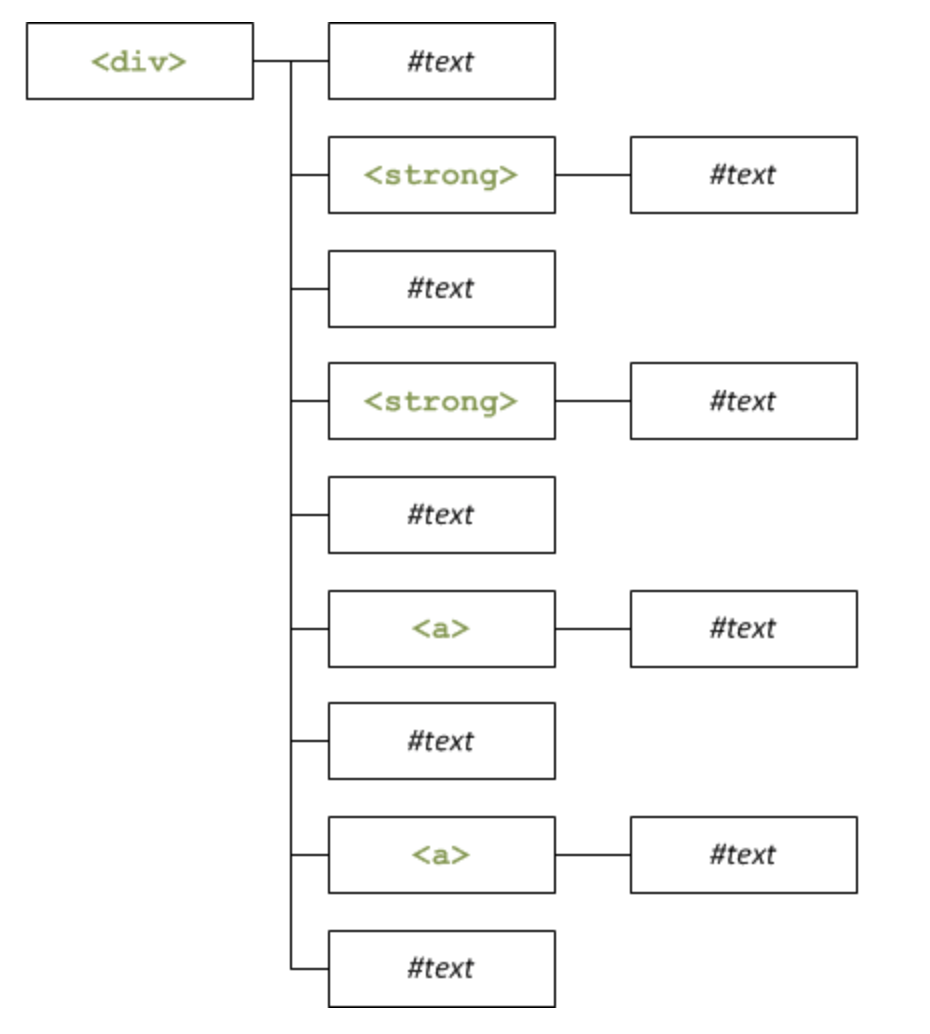


**Exercise 4:**

A little more difficult ... though. Here, the difficulty lies in the large number of elements to nest into each other. If you proceed methodically, you are unlikely to go wrong.

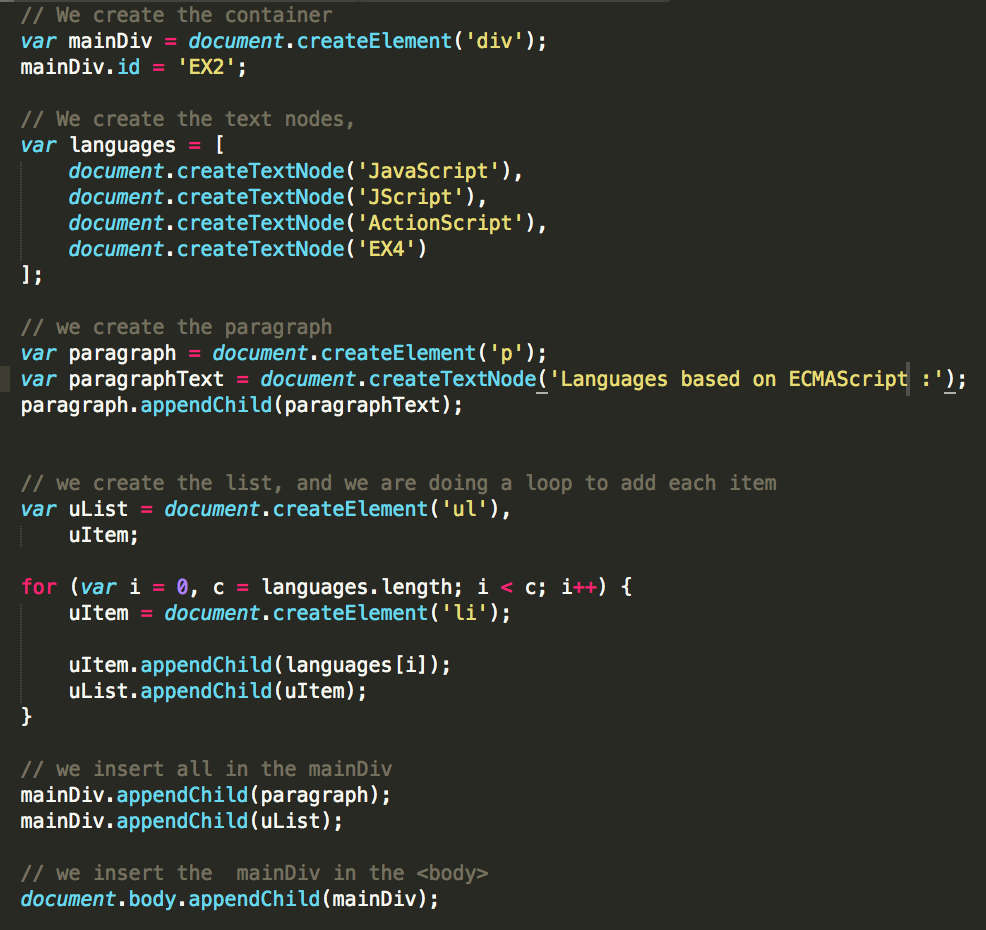
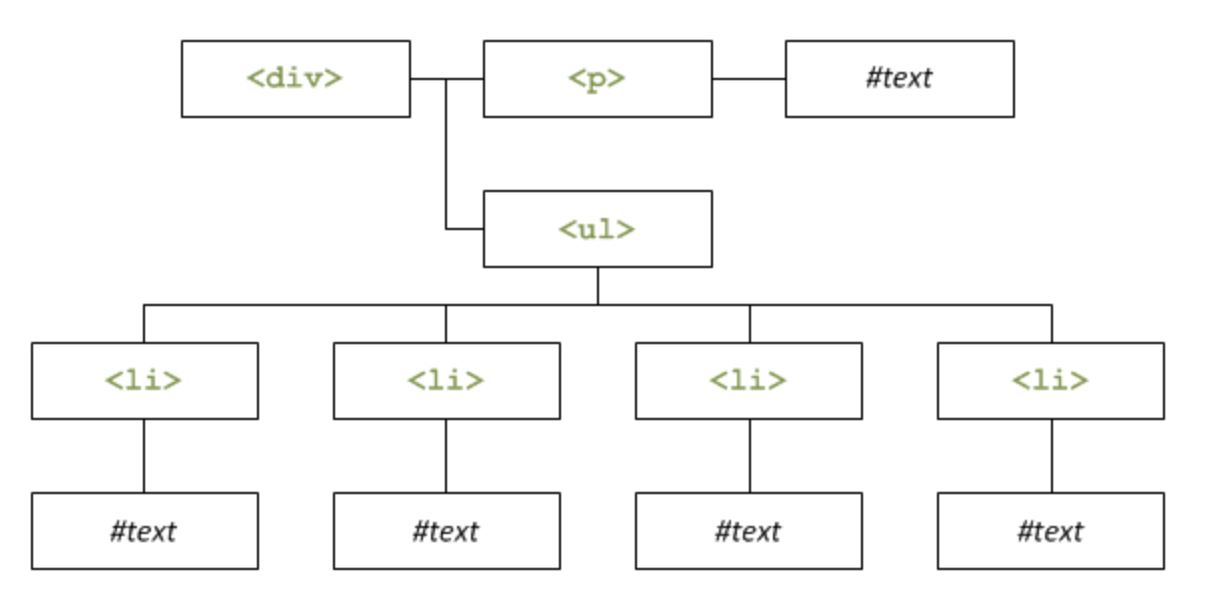
****

**Correction :**

**Exercise 1 :** 

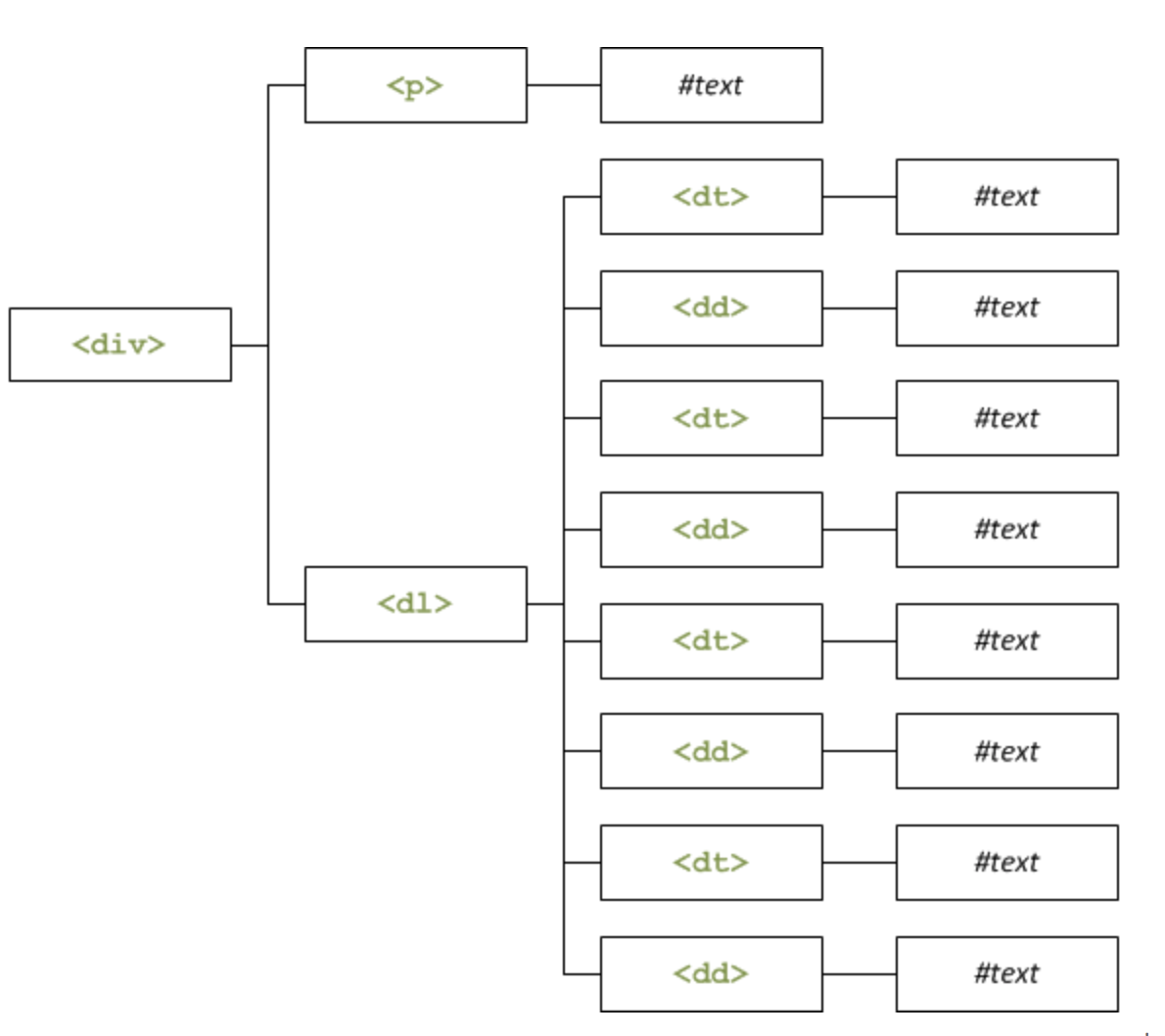
For more convenience, all text nodes are contained in the text Nodes array, so you do not have to make 250 different variables. Once the text nodes are created, we create the <a> and <strong> elements. Once all this is done, we insert everything, one element after another, into the container div.

**Exercise 2 :**

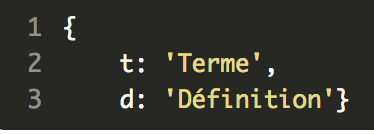


The textual nodes of the list are created through the array languages, and to create each <li> element, just loop over the number of items in the array.

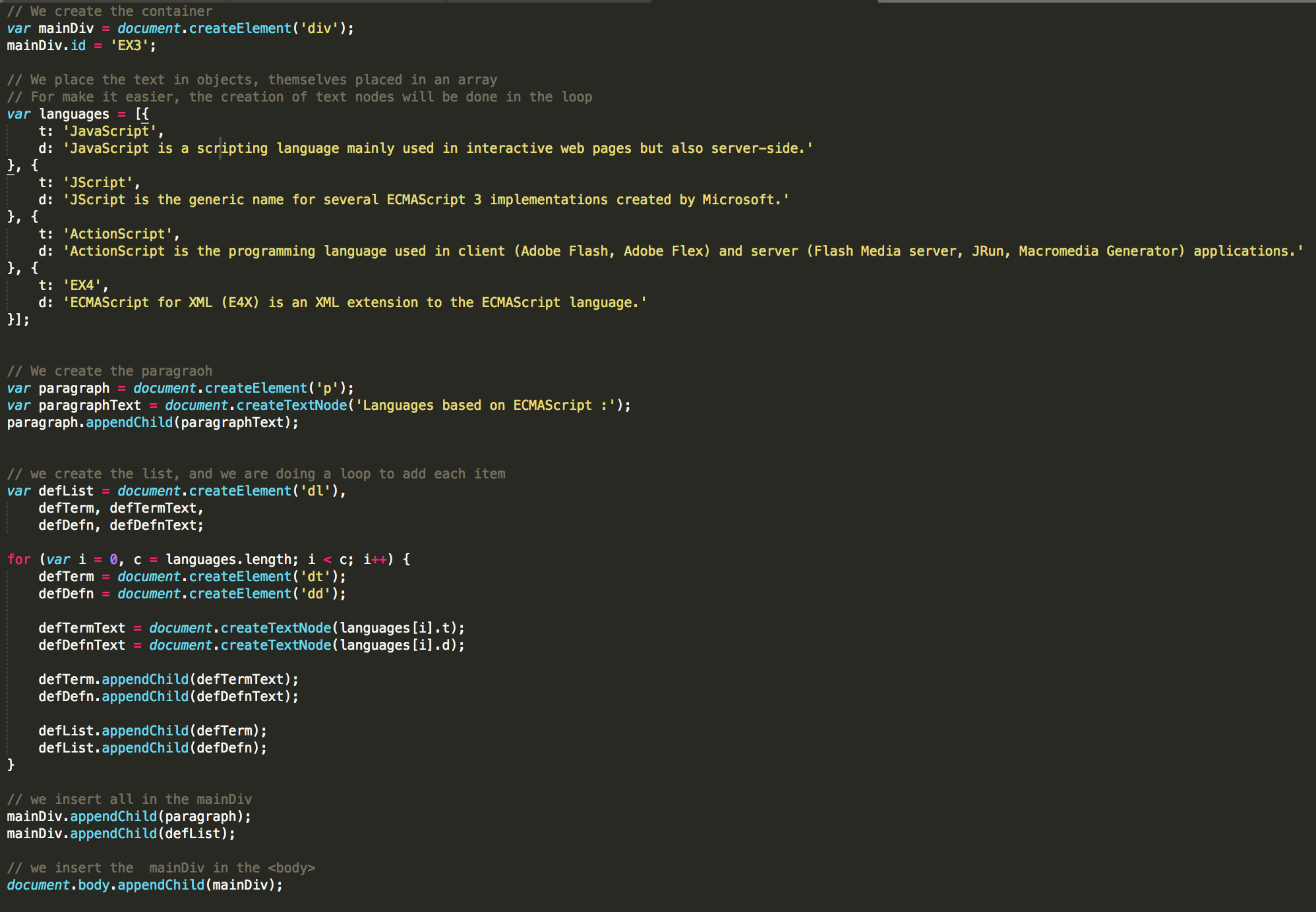
**Exercise 3** :



The table contains objects like this:



Creating a list of definitions (<dl>) is no more complicated than a normal bulleted list, the only thing that differs is that <dt> and <dd> are added together within the loop.



**Exercise 4 :**