Variations on Adding elements

Continuing the discussion on adding elements, this lesson will cover several methods you can use to add the elements in an existing HTML page.

We'll cover the following

- Adding a Textual Node
- Adding a Node Before another one
- Determining the Exact Position of the New Node

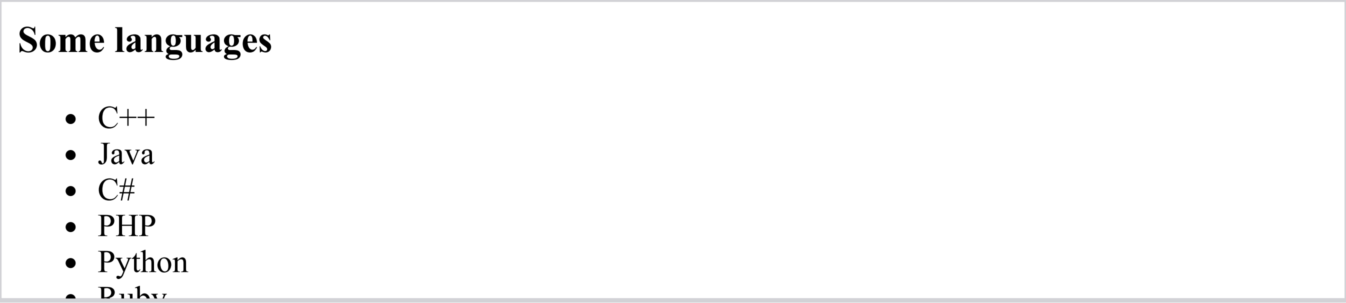
Adding a Textual Node

Instead of using the `textContent` property to define the new element's textual content, you can create a textual node with the `createTextNode()` method. This node can then be added to the new element with `appendChild()`. The following code demonstrates this possibility by inserting the Ruby language at the end of the list.

Output

JavaScript

HTML



Some languages

- C++
- Java
- C#
- PHP
- Python
- Ruby

Adding a Node Before another one

Sometimes, inserting a new node at the end of its parent's children list is not ideal. In that case, you can use the `insertBefore()` method. Called on the ideal. In that case, you can use the `insertBefore()` method. Called on the future parent, this method takes as parameters the new node and the node

before which the new one will be inserted. As an example, here’s how the Perl language could be inserted before PHP in the list.

Output

JavaScript

HTML

Some languages

- C++
- Java
- C#
- Perl
- PHP
- Python

Determining the Exact Position of the New Node

There is a method to more precisely define the position of inserted elements: `insertAdjacentHTML()`. Call it on an existing element and pass it the position and a string of HTML characters that represent the new content to be added. The new content’s position should be either:

- `beforebegin`: before the existing element
- `afterbegin`: inside the existing element, before its first child
- `beforeend`: inside the existing element, after its last child
- `afterend`: after the existing element

Here’s how these positions translate relative to an existing `<p>` tag.

HTML

```
<!-- beforebegin -->
<p>
  <!-- afterbegin -->
  foo
  <!-- beforeend -->
</p>
<!-- afterend -->
```

The following example uses `insertAdjacentHTML()` to add JavaScript at the top of the language list.

Output

JavaScript

HTML

```
// Add an element to the beginning of a list
document.getElementById('languages').insertAdjacentHTML("afterBegin", '<li id="javascript">Ja
```