

# Adding a New Element

In this lesson, you will learn how to add an element on an existing web page via JavaScript.

## WE'LL COVER THE FOLLOWING



- Creating the Element
- Setting Element Properties
- Inserting the Element into the DOM

Adding a new element to a web page can be broken into three steps:

- Create the new element
- Set element properties
- Insert the new element in the DOM

For example, suppose you want to add the language “Python” to the list of languages on our page. Here’s the JavaScript code you’d use to do so.

Let’s study each of these steps.

## Creating the Element #

You’d create an element using the `createElement()` method (surprising, isn’t it?). This method is used on the document object and takes the tag of the new element as a parameter. This method also returns the element created as an object (here stored in a variable called `pythonElement`).

JavaScript

```
const pythonElement = document.createElement("li"); // Create an li element
```



## Setting Element Properties #

Once the element is created and stored in a variable, you can add some detail to it (ID, class, text content, etc.) by using the aforementioned DOM properties. In the example, the element ID becomes `"python"` and its text content becomes `"Python"`.

JavaScript

```
// ...  
pythonElement.id = "python";           // Define element ID  
pythonElement.textContent = "Python"; // Define its text content
```



## Inserting the Element into the DOM #

There are several techniques to insert a new node in the DOM. The most common is to call the `appendChild()` method on the element that will be the future parent of the new node. The new node is added to the end of the list of child nodes of that parent. In our example, the new item is added as a new child of the `<ul>` tag identified by “languages”, after all the other children of this tag.

JavaScript

```
// ...  
document.getElementById("languages").appendChild(pythonElement); // Insert the new element in
```



