Replacing or Removing nodes

Just like you can add elements from an existing web page, you can also remove or replace those elements via JavaScript.

WE'LL COVER THE FOLLOWING

- Replacing a Node
- Removing a Node

Replacing a Node

A DOM element can be replaced with the `replaceChild()` method. This replaces a child node of the current element with another node. The new node and node-to-be-replaced are passed as parameters (in that order). The example shows replacing the Perl language with Lisp.

Output

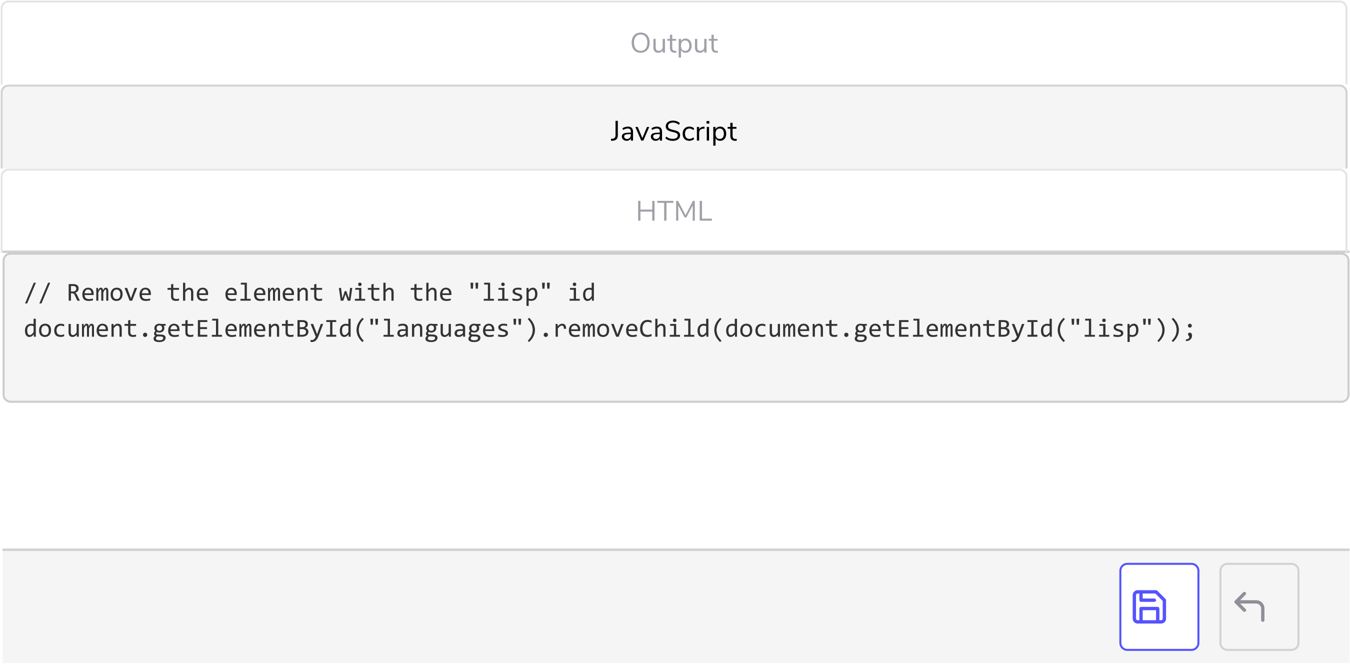
JavaScript

HTML

```
const lispElement = document.createElement("li"); // Create an li element
lispElement.id = "lisp";          // Define its ID
lispElement.textContent = "Lisp"; // Define its text content
// Replace the element identified by "perl" with the new element
document.getElementById("languages").replaceChild(lispElement, document.getElementById("perl'
```

Removing a Node

Lastly, you can delete a node thanks to a method called `removeChild()`, to which you'll pass the node-to-be-removed as a parameter.



```javascript
// Remove the element with the "lisp" id
document.getElementById("languages").removeChild(document.getElementById("lisp"));
```