

Document Object Model Continued - Creating Content Dynamically

How to Create and Append Nodes Using the DOM

1. Create the element

createElement() – creates a new DOM HTML element with the tag name in lower case, adds the ending tag automatically

```
var element = document.createElement(tagName);
```

```
var elDiv = document.createElement("div");
```

2. Create the text for the element

createTextNode() – creates a new text node that can be appended to another element, as a child node

```
var text = document.createTextNode(content);
```

```
var txtDiv = document.createTextNode("This is new content");
```

3 Add the text to the element

appendChild() – append a node to the end of a specified parent node, **as the last child node (append means at the end). Append places the node INSIDE the element, not after it**

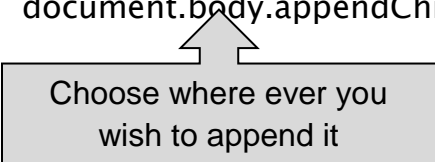
```
var newChild = document.appendChild(node);
```

```
var newDiv = elDiv.appendChild(txtDiv);
```

4 Place the new content into the document using appendChild()

// places it at the end of the document (body) as the last child node

```
document.body.appendChild(newDiv);
```



Choose where ever you
wish to append it

**new node before another node, rather than
appending it to the parent node**

4b Place the new content into the document using insertBefore()

parentNode.insertBefore(newNode, referencedNode) – inserts the new node before the referenced node as a child of the parent node

```
var newChild = document.appendChild(node); // this is the new node (step #3)
```

5 Create an attribute node and value using createAttribute("name")

document.createAttribute("name") – set an attribute, you still need to set a value and attach it to an element as shown below

```
var h1 = document.createElement('h1'); //create or reference the element
```

```
var att = document.createAttribute("class"); // create the attribute name
```

```
att.value = "red"; // set the value of the attributeattribute
```

```
h1.setAttributeNode(att); // set the attribute node to that/ or other the element
```

Other DOM Manipulation Methods

node.removeChild() method – removes a specified child node of the specified element

```
node.removeChild(someChild);
```

Note: white space and text are child nodes also.

Note: There is also a **remove()** method - `element.nextElementSibling.remove()`

node.replaceChild() method – replaces one child node of the specified node with another.

```
replacedNode = parentNode.replaceChild(newChild, oldChild);
```

newChild is the new node to replace oldChild.

oldChild is the existing child to be replaced.

replacedNode is the replaced node. This is the same node as oldChild.

node.cloneNode() method – returns a duplicate of the node

```
var clonedNode = node.cloneNode(true/false);
```

node is the node to be cloned.

clonedNode is the new node that will be a clone of node

true if the children of the node should also be cloned, or false to clone only the specified node.

setAttribute() method – sets an attribute and a value for a specified element

```
document.getElementsByTagName("h1")[0].setAttribute("class", "red");
```

Other Important Concepts

innerHTML property – A JavaScript property, not part of the DOM, allows you to set both text and HTML inside an HTML element. It is widely used in the Industry because it is easy to use. It overwrites the existing content. It will work regardless if the tags are spaced apart or closed - `<div> </div>` or `<div></div>`

firstChild.nodeValue – A way to insert content inside an HTML element. It will overwrite the content. The tags need to be spaced apart or else there is no first child
`<div> </div>`

appendChild() – A method used to append content inside an existing HTML element. It does not overwrite the existing content. It will work regardless if the tags are spaced apart or closed - `<div> </div>` or `<div></div>`