

# JavaScript Workshop

---

Creating markup with DOM

# innerHTML

- The innerHTML method is not part of the W3C DOM specification, but is supported by current browsers (IE 6, FF, Opera 7). It is used to manipulate HTML in an element:

```
<div id = "testdiv">
```

```
    <p>Old <em>content</em>.</p>
```

```
</div>
```

```
var testdiv = document.getElementById("testdiv");
```

```
testdiv.innerHTML = "<p><strong>New</strong>content.</p>"
```

# Pros and Cons of innerHTML

- Pros
  - 1) Easier to view or change element values
  - 2) More efficient coding
  - 3) Faster in most browsers
- Cons
  - 1) Does not return any references to the content
  - 2) HTML-specific (cannot use in XML)
  - 3) Not web standard (support could stop in future browsers)

# Creating element nodes

- Steps using DOM methods to add element nodes to the DOM tree:
  - 1) Create the element – createElement
  - 2) Insert the element into node tree - appendChild
  - 3) Add text to element node - createTextNode

```
var para = document.createElement("p");
```

```
var testdiv = document.getElementById("testdiv");
```

```
testdiv.appendChild(para);
```

```
var txt = document.createTextNode("Greetings");
```

```
para.appendChild(txt);
```

# innerHTML versus DOM methods

```
testdiv.innerHTML = "<p><strong>New </strong>content.</p>"
```

---

```
var para = document.createElement("p");  
var txt1 = document.createTextNode("New ");
```

```
para.appendChild(txt1);
```

```
var strong = document.createElement("strong");  
var txt2 = document.createTextNode("content");
```

```
strong.appendChild(txt2);  
para.appendChild(strong);
```

# insertBefore

- The DOM method `insertBefore` can be used to insert a new element before an existing element.

```
parentElement.insertBefore(newElement, targetElement);
```

```
var testdiv = document.getElementById("testdiv");
```

```
var para = document.createElement("p");
```

```
var txt = document.createTextNode("Greetings");
```

```
para.appendChild(txt);
```

```
testdiv.insertBefore(para, testdiv);
```

# insertAfter

- There is no insertAfter DOM method, but there are many custom functions have been written to extend the DOM functionality.

```
function insertAfter(newElement, targetElement) {  
  
    var parent = targetElement.parentNode;  
  
    if (parent.lastChild === targetElement) {  
  
        parent.appendChild(newChild);  
    } else {  
        parent.insertBefore(newElement, targetElement.nextSibling);  
    }  
}
```

- JavaScript Workshop website:  
<http://www.javascriptworkshop.com>
- Any questions?  
[frank@javascriptworkshop.com](mailto:frank@javascriptworkshop.com) - Frank Stepanski
- Recommended reading:  
[DOM Scripting: Web Design with JavaScript](#)