

JavaScript Workshop

The Document Object Model

What is the DOM?

- When you create a web page and load it in a browser the DOM is created which converts that document into an object into an object.
- The M in DOM stands for Model, which like a map represents the web page that's currently loaded in the browser. The browser provides a map (or model) of the page and all its elements. JavaScript can then be used to read and modify this map.
- The Document Object Model represents the document as a family tree. A family tree describes the relationships between family members, and uses conventions such as parent, child and sibling.

Nodes

- A document is a collection of nodes, with nodes as the branches and leaves on the document tree.
- Element nodes represent any type of tag element (`<p>`, ``, ``). The tag provides the name of the element. Paragraph elements have the name `p`, unordered lists have the name `ul`, list items have the name `li`.
- Elements can contain other elements. All the list items (``) are contained within an unordered list (``) element. The only element that isn't contained within another element is the `<html>` element, it is the root of the document tree.

Nodes cont.

- Text nodes - Are enclosed within element nodes. Though not all elements contain text nodes. The `` element doesn't contain any text directly, it contains other element nodes (``) which contain text nodes.

```
<ul>          ---> element node  
<li>One</li> ---> text node (One) enclosed by element node  
<li>Two</li> ---> text node (Two) enclosed by element node  
</ul>       ---> element node
```

- Attribute nodes - Used to give specific info about an element node.

```
<p title = "greeting">Hello</p> ---> attribute node (title = "one")
```

Layers of the web

- Web content in a web page can be wrapped in three layers:

Structure - HTML or XHTML tags describe content

Presentation - CSS to describe how content is displayed

Behavior – JavaScript describe how content reacts to events

- Though there is a certain amount of crossover (:hover CSS) between the layers in which the DOM can alter the structure of a web page.
- We'll also review how to use the DOM to keep each layer separate to create more accessible and maintainable web pages.

DOM Core methods

- Objects we have used previously to reference web page elements, (form and image for example) have been provided by the web browser. These objects (and its methods and properties) are specific to a web browser only.
- DOM objects and methods are not specific to a browser and can be used on any document that uses a markup language (XML) with any programming language that has DOM support. The advantage is its portability and supporting the latest W3C standards.

getElementById

getElementsByTagName

getAttribute

setAttribute

getElementById

- The DOM has a method called `getElementById`, which allows you to get straight to the element node with the specific id. Remember that JavaScript is case-sensitive so it must be written with case preserved.

document.getElementById(id)

```
<ul id = "costs">  
<li>$100</li>  
<li>$500</li>  
</ul>
```

alert(typeof document.getElementById("costs")); ---> returns object

getElementsByTagName

- The `getElementsByTagName` method gives you instant access to an array populated with every occurrence of a specified tag.

```
document.getElementsByTagName(tag)
```

```
<ul id = "costs">  
<li>$100</li>  
<li>$500</li>  
</ul>
```

```
alert(document.getElementsByTagName("li").length); ---> returns 2
```

getAttribute

- After you get to an element node by either the `getElementById` or the `getElementsByTagName` method you can find out the values of any of its attributes by using the `getAttribute` method.

object.getAttribute(attribute); ---> used on element node object

ESPN

var link = document.getElementsByTagName("a");

alert(link[0].getAttribute("href")); ---> displays "http://www.espn.com"

setAttribute

- The setAttribute method allows you to change the value of an attribute node. It takes two arguments, the attribute and the new value.

object.setAttribute(attribute, value); ---> used on element node object

**

var img = document.getElementsByTagName("img");

img[0].setAttribute("src", "batman.jpg"); ---> changes src of image

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