Unit 3
1.4 Document Object Model (DOM)

1.4 Document Object Model (DOM)

The HTML DOM **defines a standard for accessing and manipulating HTML documents**. It is **a platform and language-neutral interface** , hierarchically arranged, that will **allow programs and scripts to dynamically access and update the content, structure and style of documents**. The document can be further processed and the results of that processing can be incorporated back into the presented page.

The DOM is separated into 3 different parts / levels:

- Core DOM - standard model for any structured document
- XML DOM - standard model for XML documents
- HTML DOM - standard model for HTML documents

The DOM defines the objects and properties of all document elements, and the methods (interface) to access them.

1.4.1. Core Dome

The Core DOM defines a set of objects and interfaces for accessing and manipulating document objects. The Core functionality is sufficient to allow software developers and web script authors to access and manipulate parsed HTML and XML content inside conforming products. The DOM Core API also allows creation and population of a Document object using only DOM API calls; loading a Document and saving it persistently is left to the product that implements the DOM API.

1.4.2 What is the XML DOM?

The XML DOM defines the objects and properties of all XML elements, and the methods (interface) to access them.

1.4.3 What is the HTML DOM?

The HTML DOM is:

- A standard object model for HTML
- A standard programming interface for HTML
- Platform- and language-independent

The HTML DOM is a w3c standard. It defines the objects and properties of all HTML elements, and the methods (interface) to access them.
In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

1.4.4 DOM Nodes

According to the DOM, everything in an HTML document is a node.
The DOM says:

- The entire document is a document node
- Every HTML element is an element node
- The text in the HTML elements are text nodes
- Every HTML attribute is an attribute node
- Comments are comment nodes

1.4.4.1 DOM Example

Look at the following HTML document:

```
<html>
<head>
<title>DOM Example</title>
</head>
<body>
<h1>DOM Lesson one</h1>
<p>Hello world!</p>
</body>
</html>
```

Example 1.40

The root node in the HTML above is <html>. All other nodes in the document are contained within <html>. The <html> node has two child nodes; <head> and <body>. The <head> node holds a <title> node. The <body> node holds a <h1> and <p> node.

1.4.4.2 Text is Always Stored in Text Nodes

A common error in DOM processing is to expect an element node to contain text. However, the text of an element node is stored in a text node. In this example: **<title>DOM Example </title>**, the element node <title>, holds a text node with the value "DOM Example ".
"DOM Example " is **not** the value of the <title> element!
However, in the HTML DOM the value of the text node can be accessed by the **innerHTML** property.

1.4.5 The HTML DOM Node Tree

The HTML DOM views a HTML document as a tree-structure. The tree structure is called a **node-tree.** All nodes can be accessed through the tree. Their contents can be modified or deleted, and new elements can be created. The node tree below shows the set of nodes, and the connections between them. The tree starts at the root node and branches out to the text nodes at the lowest level of the tree:

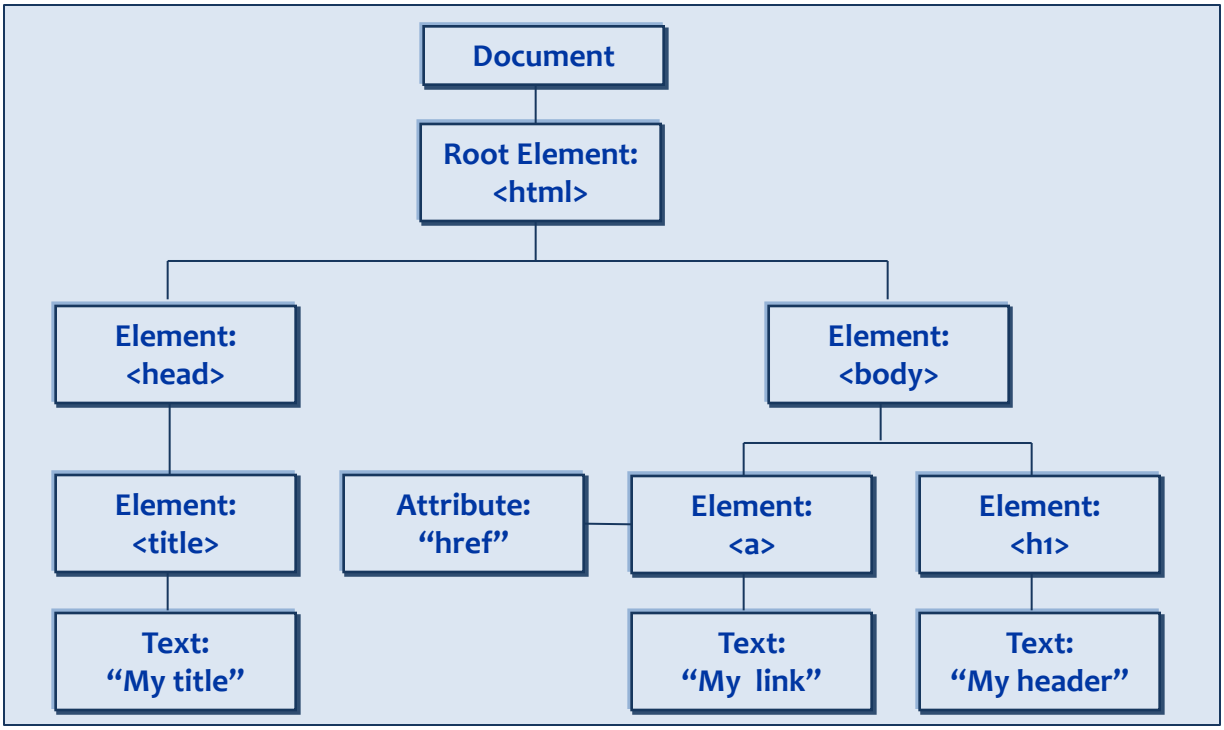

Figure 1.48

1.4.5.1 Node Parents, Children, and Siblings

The nodes in the node tree have a hierarchical relationship to each other.

The terms parent, child, and sibling are used to describe the relationships. Parent nodes have children. Children on the same level are called siblings (brothers or sisters).

- In a node tree, the top node is called the root
- Every node, except the root, has exactly one parent node
- A node can have any number of children
- A leaf is a node with no children
- Siblings are nodes with the same parent

The following image illustrates a part of the node tree and the relationship between the nodes:

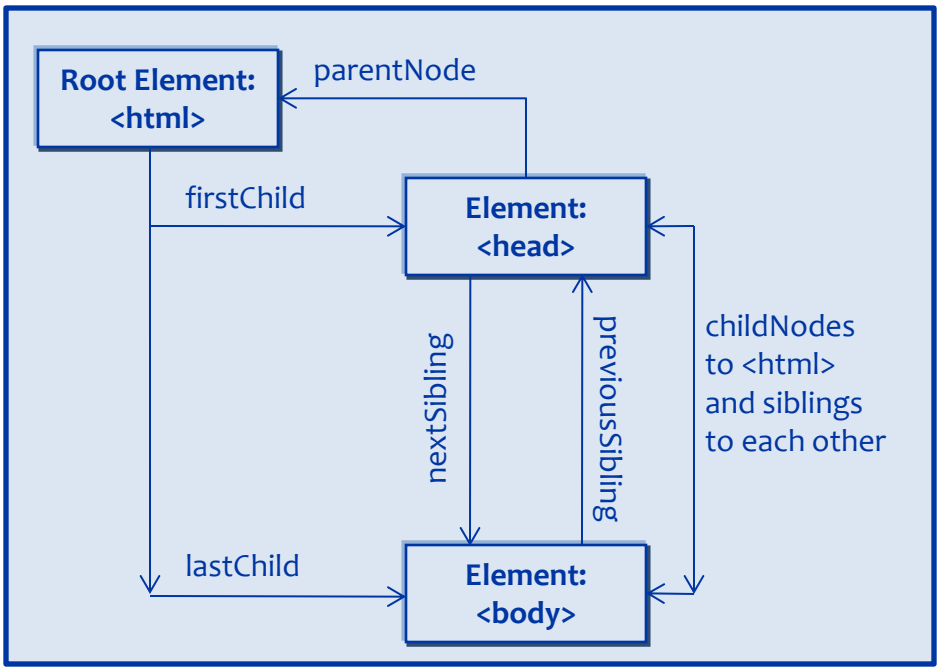

Figure 1.49

Look at the following HTML fragment:

```
<html>
<head>
<title>DOM Example</title>
</head>
<body>
<h1>DOM Lesson one</h1>
<p>Hello world!</p>
</body>
</html>
```

Example 1.41

From the HTML above:

- The **<html>** node has no parent node; it is the root node
- The parent node of the **<head>** and **<body>** nodes is the **<html>** node
- The parent node of the "Hello world!" text node is the **<p>** node

and:

- The **<html>** node has two child nodes; <**head**> and <**body**>
- The <**head**> node has one child node; the **<title>** node
- The **<title>** node also has one child node; the text node "DOM Tutorial"
- The **<h1>** and **<p>** nodes are siblings, and both child nodes of **<body>**

1.4.5.2 First Child - Last Child

From the HTML above:
the **<head>** element is the first child of the **<html>** element, and the **<body>** element is the last child of the **<html>** element
the **<h1>** element is the first child of the **<body>** element, and the **<p>** element is the last child of the **<body>** element

1.4.6 HTML DOM Properties and Methods

Properties and methods define the programming interface of the HTML DOM.

1.4.6.1 Programming Interface

As mentioned previously, in the DOM, HTML documents consist of a set of node objects. These nodes can be accessed with **JavaScript** or other programming languages. We will discuss javascripts next section**. The programming interface of the DOM is defined by standard properties and methods.**

Properties are often referred to as something that is an attribute of a object. (i.e. the name of a node).
Methods are often referred to as something that is done (i.e. remove a node).

1.4.6.2 HTML DOM Properties

Some **object specific** DOM properties are listed below.

Property Name	Description
obj.innerHTML	The text value of x component
obj.nodeName	The name of x component
obj.nodeValue	The value of x component
objx.parentNode	The parent node of x component
obj.childNodes	The child nodes of x component
obj.Attributes	The attributes nodes of x compoent
Note: In the list above, x is a node object (HTML element)	

Figure 1.50

1.4.6.3 HTML DOM Methods

Some **document specific** DOM methods are listed below.

Method Name	Description
document.getElementById(id)	get the element with a specified id
document.getElementsByTagName(name)	get all elements with a specified tag name
document.appendChild(node)	insert a child node to x
document.removeChild(node)	remove a child node from x
Note: In the list above, x is a node object (HTML element)	

Figure 1.51

Using these properties and methods will be revisited after learning javascripts in the next section.