**DOM, CSS, XPath, HTML**

**What is the Document Object Model?**

The Document Object Model is a platform- and language-neutral interface 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 Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated. In the DOM specification, the term "document" is used in the broad sense - increasingly, XML is being used as a way of representing many different kinds of information that may be stored in diverse systems, and much of this would traditionally be seen as data rather than as documents. Nevertheless, XML presents this data as documents, and the DOM may be used to manage this data.

With the Document Object Model, programmers can create and build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model,

## What the Document Object Model is

The Document Object Model is a programming API for documents. The object model itself closely resembles the structure of the documents it models. For instance, consider this table, taken from an HTML document:

<TABLE>

<ROWS>

<TR>

<TD>Shady Grove</TD>

<TD>Aeolian</TD>

</TR>

<TR>

<TD>Over the River, Charlie</TD>

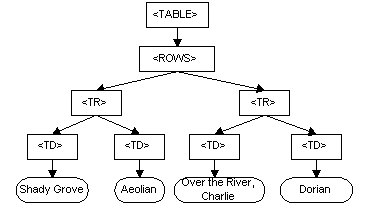
<TD>Dorian</TD>

</TR>

</ROWS>

</TABLE>

The Document Object Model represents this table like this:



**DOM representation of the example table**

In the Document Object Model, documents have a logical structure which is very much like a tree;

**HTML**

When an HTML document is loaded into a web browser, it becomes a **document object**.

The document object is the root node of the HTML document

HTML Element:

An HTML element usually consists of a **start** tag and **end** tag, with the content inserted in between:

<tagname>Content goes here...</tagname>

The HTML **element** is everything from the start tag to the end tag:

<p>My first paragraph</p>

<userName>Admin</username>

**Nested HTML Elements**

The HTML element  <body>, <h1>,<p> are the nested elements

<html>  
 <body>  
 <h1>My First Heading</h1>  
 <p>My first paragraph.</p>  
 </body>  
</html>

HTML Attributes

* All HTML elements can have **attributes**
* Attributes provide **additional information** about an element
* Attributes are always specified in **the start tag**
* Attributes usually come in name/value pairs like: **name="value"**

<img src="img\_girl.jpg" width="500" height="600">

Attributes: src, width and height

Styling HTML with CSS

**CSS** stands for **C**ascading **S**tyle **S**heets.

CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**.

CSS **saves a lot of work**. It can control the layout of multiple web pages all at once.

CSS can be added to HTML elements in 3 ways:

* **Inline** - by using the style attribute in HTML elements
* **Internal** - by using a <style> element in the <head> section
* **External** - by using an external CSS file

The most common way to add CSS, is to keep the styles in separate CSS files. However, here we will use inline and internal styling, because this is easier to demonstrate, and easier for you to try it yourself.

Ex: <h1 style="color:blue;">This is a Blue Heading</h1>

# XML and XPath

* XPath is a syntax for defining parts of an XML document
* XPath uses path expressions to navigakte in XML documents
* XPath contains a library of standard functions
* XPath is a major element in XSLT and in XQuery
* XPath is a W3C recommendation

## XPath Path Expressions

XPath uses path expressions to select nodes or node-sets in an XML document. These path expressions look very much like the expressions you see when you work with a traditional computer file system.

XPath expressions can be used in JavaScript, Java, XML Schema, PHP, Python, C and C++, and lots of other languages.

## XPath Example

We will use the following XML document:

<?xml version="1.0" encoding="UTF-8"?>  
  
<bookstore>  
  
<book category="cooking">  
  <title lang="en">Everyday Italian</title>  
  <author>Giada De Laurentiis</author>  
  <year>2005</year>  
  <price>30.00</price>  
</book>  
  
<book category="children">  
  <title lang="en">Harry Potter</title>  
  <author>J K. Rowling</author>  
  <year>2005</year>  
  <price>29.99</price>  
</book>  
  
<book category="web">  
  <title lang="en">XQuery Kick Start</title>  
  <author>James McGovern</author>  
  <author>Per Bothner</author>  
  <author>Kurt Cagle</author>  
  <author>James Linn</author>  
  <author>Vaidyanathan Nagarajan</author>  
  <year>2003</year>  
  <price>49.99</price>  
</book>  
  
<book category="web">  
  <title lang="en">Learning XML</title>  
  <author>Erik T. Ray</author>  
  <year>2003</year>  
  <price>39.95</price>  
</book>  
  
</bookstore>

In the table below we have listed some XPath expressions and the result of the expressions:

|  |  |
| --- | --- |
| **XPath Expression** | **Result** |
| /bookstore/book[1] | Selects the first book element that is the child of the bookstore element |
| /bookstore/book[last()] | Selects the last book element that is the child of the bookstore element |
| /bookstore/book[last()-1] | Selects the last but one book element that is the child of the bookstore element |
| /bookstore/book[position()<3] | Selects the first two book elements that are children of the bookstore element |
| //title[@lang] | Selects all the title elements that have an attribute named lang |
| //title[@lang='en'] | Selects all the title elements that have a "lang" attribute with a value of "en" |
| /bookstore/book[price>35.00] | Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00 |
| /bookstore/book[price>35.00]/title | Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00 |