Node object

The Node object represents a node in the HTML document.A node in an HTML document is:

* The Document
* An element
* An attribute
* Text
* A comment

Document Object

Each HTML document loaded into a browser window becomes a Document object. The Document object provides access to all HTML elements in a page, from within a script , The Document object is also part of the Window object, and can be accessed through the window.document property.The Document object can also use the properties and methods of the Node object.

# Document Object Properties

**Anchors** : Returns a collection of all the anchors in the document

**body** : Returns the body element of the document

**cookie** : Returns all name/value pairs of cookies in the document

**domain** : Returns the domain name of the server that loaded the document

**forms** : Returns a collection of all the forms in the document

**images** : Returns a collection of all the images in the document

**lastModified** : Returns the date and time the document was last modified

**links** : Returns a collection of all the links in the document

**readyState** : Returns the (loading) status of the document

**referrer** : Returns the URL of the document that loaded the current document

**title** : Sets or returns the title of the document

**URL** : Returns the full URL of the document

**NOTE : These are Object Properties so you can access them directly without the need to create new instance object of document , see below examples :**

* ***document.write(document.URL);***
* ***document.write(document.title);***

# Document Object Methods

# getElementsByName()

The getElementsByName() method accesses all elements with the specified name.

**Syntax**

***document.getElementsByName(name)***

* **name** [Required]. The name of the element you want to access/manipulate

**Example** : Alert the number of elements with a specific name:

***var x=document.getElementsByName("x");***

***alert(x.length);***

# open()

The open() method opens an output stream to collect the output from any document.write() or document.writeln() methods.Once all the writes are performed, the document.close() method causes any output written to the output stream to be displayed.Note If a document already exists in the target it will be cleared. If this method has no arguments, a new window (about:blank) is displayed.

**Syntax**

***document.open(MIMEtype,replace)***

* **MIMEtype** [Optional]. The type of document you are writing to. Default value is "text/html"
* **replace** [Optional]. If set, the history entry for the new document inherits the history entry from the document which opened this document

**Example** Open an output stream, add some text, then close the output stream:

***var doc=document.open("text/html","replace");***

***var txt="<html><body>Learning about the HTML DOM is fun!</body></html>";***

***doc.write(txt);***

***doc.close();***

# close()

Closes the output stream previously opened with document.open()

# write()

Writes HTML expressions or JavaScript code to a document

The write() method writes HTML expressions or JavaScript code to a document.

**Syntax**

***document.write(exp1,exp2,exp3,...)***

# writeln()

The writeln() method is identical to the write() method, with the addition of writing a newline character after each statement.

**Syntax**

***document.writeln(exp1,exp2,exp3,...)***

Element Object

The Element object represents an element in the HTML document. The Element object can have child nodes of type Element, Text, Comment, CDATASection, ProcessingInstruction, and EntityReference. The Element object can have attributes, which have the node type Attr. The Element object can also use the properties and methods of the Node object.

Element Object Properties

* **schemaTypeInfo** Returns the type information of the element.
* **tagName** Returns the tag name of the element

**Example**

Get the tagName of an element:

***document.getElementById("demo").tagName;***

The result will be: **P**

History Object

The history object contains the URLs visited by the user (within a browser window). The history object is part of the window object and is accessed through the window.history property. Notethat There is no public standard that applies to the history object, but all major browsers support it.

# History Object Properties

* **Length** Returns the number of URLs in the history list

# History Object Methods

# back()

Loads the previous URL in the history list ,This is the same as clicking the Back button or history.go(-1).

**Example**

***<script type="text/javascript">  
function goBack()  {  
  window.history.back()  }  
</script>  
<input type="button" value="Back" onclick="goBack()" />***

# forward()

Loads the next URL in the history list , The length property returns the number of URLs in the history list. Notethat Internet Explorer and Opera start at 0, while Firefox, Chrome, and Safari start at 1.

**Syntax**

***history.length***

**Example**

Return the number of URLs in the history list:

***<script type="text/javascript">***

***document.write("Number of URLs in history list: " + history.length);***

***</script>***

The output of the code above will be:

**Number of URLs in history list: 2**

# go()

Loads a specific URL from the history list ,The parameter can either be a number which goes to the URL within the specific position (-1 goes back one page, 1 goes forward one page), or a string. The string must be a partial or full URL, and the function will go to the first URL that matches the string.

**Syntax**

***history.go(number|URL)***

**Example**

ExampleClick on the button to go back two pages:

***function goBack() {***

***window.history.go(-2) }***