

DOM Methods & Properties

Method Name	Description
<code>getElementById(id)</code> (<code>document</code>)	Retrieves the element in the document that has the specified unique ID attribute value
<code>getElementsByName(name)</code>	Returns an array of the current element's children that have the specified tag name
<code>hasChildNodes()</code>	Returns a Boolean indicating whether the element has any child elements
<code>getAttribute(name)</code>	Returns the value of the element's attribute specified by name
Property Identifier	Description
<code>attributes[]</code>	Contains a read-only collection of the attributes associated with the HTML element. An individual attribute can be retrieved by using the method <code>getAttribute</code> . To assign or overwrite an attribute, the method <code>setAttribute</code> is used. To remove an attribute, the method <code>removeAttribute</code> is used.
<code>childNodes[]</code>	Is an instance of <code>NodeList</code> that most likely is referenced by using an array notation, but the array is read-only. To add a child node to the current element, the method <code>appendChild</code> is used. To remove a child node, the method <code>removeChild</code> is used; and to replace a child node, <code>replaceChild</code> is used.
<code>className</code>	Assigns a stylesheet class identifier to an element. A class type is very important in Dynamic HTML in that the look and feel of the element can be dynamically assigned.
<code>dir</code>	Indicates the direction of the text, either left to right (<code>ltr</code>) or right to left (<code>rtl</code>).
<code>disabled</code>	Enables (<code>false</code>) or disables (<code>true</code>) an element. Useful when the script does not want a user to click a certain button or other GUI element before completing a required step.
<code>firstChild</code> , <code>lastChild</code>	Retrieves either the first child node or the last child node.
<code>id</code>	Is the identifier of the element used to find a particular element. For example, this property is referenced when a script calls the method <code>getElementById</code> .
<code>nextSibling</code> , <code>previousSibling</code>	Retrieves either the next or previous sibling. When used in combination with <code>firstChild</code> and <code>lastChild</code> , can be used to iterate a set of elements. This approach would be used to iterate a list in which the element is responsible for indicating what the next element should be—for example, when implementing a Decorator pattern or similar structure.
<code>nodeName</code>	Contains the name of the element, which in HTML means the tag (for example, <code>td</code> , <code>table</code> , and so on).
<code>nodeType</code>	Contains the type of element but is geared for use when processing XML documents. With respect to HTML, this property has very little use.
<code>nodeValue</code>	Contains the value of the data in the node. Again, this property has more use when processing XML documents. With respect to HTML, this property cannot be used as a replacement for <code>innerHTML</code> .
<code>parentElement</code>	Retrieves the parent element for the current element. For example, can be used to navigate to the table that contains a row cell.
<code>style</code>	Identifies the current style properties associated with the element and is an instance of <code>CSSStyleDeclaration</code> type.
<code>tabIndex</code>	Defines the tab stop of the element with respect to the entire HTML document.
<code>tagName</code>	Identifies the tag of the current element. Use this property when attempting to figure out the element type after the element has been retrieved via the method <code>getElementById</code> .

XMLHttpRequest Object

Method	Description
<code>abort()</code>	The current request.
<code>getAllResponseHeaders()</code>	Returns all the response headers for the HTTP request as key/value pairs.
<code>getResponseHeader("header")</code>	Returns the string value of the specified header.
<code>open("method", "url")</code>	Sets the stage for a call to the server. The method argument can be either GET, POST, or PUT. The <code>url</code> argument can be relative or absolute. This method includes three optional arguments.
<code>send(content)</code>	Sends the request to the server.
<code>setRequestHeader("header", "value")</code>	Sets the specified header to the supplied value. <code>open()</code> must be called before attempting to set any headers.
Property	Description
<code>onreadystatechange</code>	The event handler that fires at every state change, typically a call to a JavaScript function.
<code>readyState</code>	The state of the request. The five possible values are 0 = uninitialized, 1 = loading, 2 = loaded, 3 = interactive, and 4 = complete.
<code>responseText</code>	The response from the server as a string.
<code>responseXML</code>	The response from the server as XML. This object can be parsed and examined as a DOM object.
<code>status</code>	The HTTP status code from the server (that is, 200 for OK, 404 for Not Found, and so on).
<code>statusText</code>	The text version of the HTTP status code (that is, OK or Not Found, and so on).