

1. The HTML DOM Document Object

HTML DOM Nodes

In the HTML DOM (Document Object Model), everything is a **node**:

- The document itself is a document node
- All HTML elements are element nodes
- All HTML attributes are attribute nodes
- Text inside HTML elements are text nodes
- Comments are comment nodes

1.1 The Document Object

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 and the "owner" of all other nodes: (element nodes, text nodes, attribute nodes, and comment nodes).

The document object provides properties and methods to access all node objects, from within JavaScript.

Tip: The document is a part of the Window object and can be accessed as `window.document`.

Document Object Properties and Methods

The following properties and methods can be used on HTML documents:

Property / Method	Description
<code>document.activeElement</code>	Returns the currently focused element in the document
<code>document.addEventListener()</code>	Attaches an event handler to the document
<code>document.adoptNode()</code>	Adopts a node from another document
<code>document.anchors</code>	Returns a collection of all <code><a></code> elements in the document that have a name attribute
<code>document.applets</code>	Returns a collection of all <code><applet></code> elements in the document
<code>document.baseURI</code>	Returns the absolute base URI of a document
<code>document.body</code>	Sets or returns the document's body (the <code><body></code> element)
<code>document.close()</code>	Closes the output stream previously opened with <code>document.open()</code>
<code>document.cookie</code>	Returns all name/value pairs of cookies in the document
<code>document.createAttribute()</code>	Creates an attribute node
<code>document.createComment()</code>	Creates a Comment node with the specified text

document.createDocumentFragment()	Creates an empty DocumentFragment node
document.createElement()	Creates an Element node
document.createTextNode()	Creates a Text node
document.doctype	Returns the Document Type Declaration associated with the document
document.documentElement	Returns the Document Element of the document (the <html> element)
document.documentMode	Returns the mode used by the browser to render the document
document.documentURI	Sets or returns the location of the document
document.domain	Returns the domain name of the server that loaded the document
document.domConfig	Obsolete. Returns the DOM configuration of the document
document.embeds	Returns a collection of all <embed> elements the document
document.forms	Returns a collection of all <form> elements in the document
document.getElementById()	Returns the element that has the ID attribute with the specified value
document.getElementsByTagName()	Returns a NodeList containing all elements with the specified class name
document.getElementsByName()	Returns a <u>NodeList</u> containing all elements with a specified name
document.getElementsByTagName()	Returns a NodeList containing all elements with the specified tag name
document.hasFocus()	Returns a Boolean value indicating whether the document has focus
document.head	Returns the <head> element of the document
document.images	Returns a collection of all elements in the document
document.implementation	Returns the DOMImplementation object that handles this document
document.importNode()	Imports a node from another document
document.inputEncoding	Returns the encoding, character set, used for the document
document.lastModified	Returns the date and time the document was last modified
document.links	Returns a collection of all <a> and <area> elements in the document that have a href attribute
document.normalize()	Removes empty Text nodes, and joins adjacent nodes
document.normalizeDocument()	Removes empty Text nodes, and joins adjacent nodes
document.open()	Opens an HTML output stream to collect output from document.write()
document.querySelector()	Returns the first element that matches a specified CSS selector(s) in the document
document.querySelectorAll()	Returns a static NodeList containing all elements that matches a specified CSS selector(s) in the document
document.readyState	Returns the (loading) status of the document

document.referrer	Returns the URL of the document that loaded the current document
document.removeEventListener()	Removes an event handler from the document (that has been attached with the <code>addEventListener()</code> method)
document.renameNode()	Renames the specified node
document.scripts	Returns a collection of <code><script></code> elements in the document
document.strictErrorChecking	Sets or returns whether error-checking is enforced or not
document.title	Sets or returns the title of the document
document.URL	Returns the full URL of the HTML document
document.write()	Writes HTML expressions or JavaScript code to a document
document.writeln()	Same as <code>write()</code> , but adds a newline character after each statement

1.2 The Element Object

In the HTML DOM, the **Element object** represents an HTML element.

Element objects can have **child nodes** of type element nodes, text nodes, or comment nodes.

A **NodeList object** represents a list of nodes, like an HTML element's collection of child nodes.

Elements can also have attributes. Attributes are attribute nodes (See next chapter).

Properties and Methods

The following properties and methods can be used on all HTML elements:

Property / Method	Description
<i>element.accessKey</i>	Sets or returns the accesskey attribute of an element
<i>element.addEventListener()</i>	Attaches an event handler to the specified element
<i>element.appendChild()</i>	Adds a new child node, to an element, as the last child node
<i>element.attributes</i>	Returns a NamedNodeMap of an element's attributes
<i>element.blur()</i>	Removes focus from an element
<i>element.childElementCount</i>	Returns the number of child elements an element has
<i>element.childNodes</i>	Returns a collection of an element's child nodes (including text and comment nodes)
<i>element.children</i>	Returns a collection of an element's child element (excluding text and comment nodes)
<i>element.classList</i>	Returns the class name(s) of an element
<i>element.className</i>	Sets or returns the value of the class attribute of an element
<i>element.click()</i>	Simulates a mouse-click on an element
<i>element.clientHeight</i>	Returns the height of an element, including padding
<i>element.clientLeft</i>	Returns the width of the left border of an element
<i>element.clientTop</i>	Returns the width of the top border of an element
<i>element.clientWidth</i>	Returns the width of an element, including padding
<i>element.cloneNode()</i>	Clones an element
<i>element.compareDocumentPosition()</i>	Compares the document position of two elements
<i>element.contains()</i>	Returns true if a node is a descendant of a node, otherwise false
<i>element.contentEditable</i>	Sets or returns whether the content of an element is editable or not
<i>element.dir</i>	Sets or returns the value of the dir attribute of an element
<i>element.firstChild</i>	Returns the first child node of an element
<i>element.firstElementChild</i>	Returns the first child element of an element
<i>element.focus()</i>	Gives focus to an element
<i>element.getAttribute()</i>	Returns the specified attribute value of an element node

<i>element.getAttributeNode()</i>	Returns the specified attribute node
<i>element.getElementsByClassName()</i>	Returns a collection of all child elements with the specified class name
<i>element.getElementsByTagName()</i>	Returns a collection of all child elements with the specified tag name
<i>element.getFeature()</i>	Returns an object which implements the APIs of a specified feature
<i>element.hasAttribute()</i>	Returns true if an element has the specified attribute, otherwise false
<i>element.hasAttributes()</i>	Returns true if an element has any attributes, otherwise false
<i>element.hasChildNodes()</i>	Returns true if an element has any child nodes, otherwise false
<i>element.id</i>	Sets or returns the value of the id attribute of an element
<i>element.innerHTML</i>	Sets or returns the content of an element
<i>element.insertBefore()</i>	Inserts a new child node before a specified, existing, child node
<i>element.isContentEditable</i>	Returns true if the content of an element is editable, otherwise false
<i>element.isDefaultNamespace()</i>	Returns true if a specified namespaceURI is the default, otherwise false
<i>element.isEqualNode()</i>	Checks if two elements are equal
<i>element.isSameNode()</i>	Checks if two elements are the same node
<i>element.isSupported()</i>	Returns true if a specified feature is supported on the element
<i>element.lang</i>	Sets or returns the value of the lang attribute of an element
<i>element.lastChild</i>	Returns the last child node of an element
<i>element.lastElementChild</i>	Returns the last child element of an element
<i>element.namespaceURI</i>	Returns the namespace URI of an element
<i>element.nextSibling</i>	Returns the next node at the same node tree level
<i>element.nextElementSibling</i>	Returns the next element at the same node tree level
<i>element.nodeName</i>	Returns the name of a node
<i>element.nodeType</i>	Returns the node type of a node
<i>element.nodeValue</i>	Sets or returns the value of a node
<i>element.normalize()</i>	Joins adjacent text nodes and removes empty text nodes in an element
<i>element.offsetHeight</i>	Returns the height of an element, including padding, border and scrollbar
<i>element.offsetWidth</i>	Returns the width of an element, including padding, border and scrollbar
<i>element.offsetLeft</i>	Returns the horizontal offset position of an element
<i>element.offsetParent</i>	Returns the offset container of an element
<i>element.offsetTop</i>	Returns the vertical offset position of an element
<i>element.ownerDocument</i>	Returns the root element (document object) for an element

<i>element.parentNode</i>	Returns the parent node of an element
<i>element.parentElement</i>	Returns the parent element node of an element
<i>element.previousSibling</i>	Returns the previous node at the same node tree level
<i>element.previousElementSibling</i>	Returns the previous element at the same node tree level
<i>element.querySelector()</i>	Returns the first child element that matches a specified CSS selector(s) of an element
<i>element.querySelectorAll()</i>	Returns all child elements that matches a specified CSS selector(s) of an element
<i>element.removeAttribute()</i>	Removes a specified attribute from an element
<i>element.removeAttributeNode()</i>	Removes a specified attribute node, and returns the removed node
<i>element.removeChild()</i>	Removes a child node from an element
<i>element.replaceChild()</i>	Replaces a child node in an element
<i>element.removeEventListener()</i>	Removes an event handler that has been attached with the <code>addEventListener()</code> method
<i>element.scrollHeight</i>	Returns the entire height of an element, including padding
<i>element.scrollLeft</i>	Sets or returns the number of pixels an element's content is scrolled horizontally
<i>element.scrollTop</i>	Sets or returns the number of pixels an element's content is scrolled vertically
<i>element.scrollWidth</i>	Returns the entire width of an element, including padding
<i>element.setAttribute()</i>	Sets or changes the specified attribute, to the specified value
<i>element.setAttributeNode()</i>	Sets or changes the specified attribute node
<i>element.style</i>	Sets or returns the value of the style attribute of an element
<i>element.tabIndex</i>	Sets or returns the value of the <code>tabindex</code> attribute of an element
<i>element.tagName</i>	Returns the tag name of an element
<i>element.textContent</i>	Sets or returns the textual content of a node and its descendants
<i>element.title</i>	Sets or returns the value of the title attribute of an element
<i>element.toString()</i>	Converts an element to a string
<i>nodelist.item()</i>	Returns the node at the specified index in a <code>NodeList</code>
<i>nodelist.length</i>	Returns the number of nodes in a <code>NodeList</code>

2. The Window Object

Window Object

The window object represents an open window in a browser.

If a document contain frames (<iframe> tags), the browser creates one window object for the HTML document, and one additional window object for each frame.

Note: There is no public standard that applies to the Window object, but all major browsers support it.

Window Object Properties

Property	Description
closed	Returns a Boolean value indicating whether a window has been closed or not
defaultStatus	Sets or returns the default text in the statusbar of a window
document	Returns the Document object for the window (See Document object)
frameElement	Returns the <iframe> element in which the current window is inserted
frames	Returns all <iframe> elements in the current window
history	Returns the History object for the window (See History object)
innerHeight	Returns the inner height of a window's content area
innerWidth	Returns the inner width of a window's content area
length	Returns the number of <iframe> elements in the current window
localStorage	Returns a reference to the local storage object used to store data. Stores data with no expiration date
location	Returns the Location object for the window (See Location object)
name	Sets or returns the name of a window
navigator	Returns the Navigator object for the window (See Navigator object)
opener	Returns a reference to the window that created the window
outerHeight	Returns the outer height of a window, including toolbars/scrollbars
outerWidth	Returns the outer width of a window, including toolbars/scrollbars
pageXOffset	Returns the pixels the current document has been scrolled (horizontally) from the upper left corner of the window
pageYOffset	Returns the pixels the current document has been scrolled (vertically) from the upper left corner of the window
parent	Returns the parent window of the current window
screen	Returns the Screen object for the window (See Screen object)
screenLeft	Returns the horizontal coordinate of the window relative to the screen
screenTop	Returns the vertical coordinate of the window relative to the screen
screenX	Returns the horizontal coordinate of the window relative to the screen
screenY	Returns the vertical coordinate of the window relative to the screen

sessionStorage	Returns a reference to the local storage object used to store data. Stores data for one session (lost when the browser tab is closed)
scrollX	An alias of pageXOffset
scrollY	An alias of pageYOffset
self	Returns the current window
status	Sets or returns the text in the statusbar of a window
top	Returns the topmost browser window

Window Object Methods

Method	Description
alert()	Displays an alert box with a message and an OK button
atob()	Decodes a base-64 encoded string
blur()	Removes focus from the current window
btoa()	Encodes a string in base-64
clearInterval()	Clears a timer set with setInterval()
clearTimeout()	Clears a timer set with setTimeout()
close()	Closes the current window
confirm()	Displays a dialog box with a message and an OK and a Cancel button
createPopup()	Creates a pop-up window
focus()	Sets focus to the current window
getComputedStyle()	Gets the current computed CSS styles applied to an element
getSelection()	Returns a Selection object representing the range of text selected by the user
matchMedia()	Returns a MediaQueryList object representing the specified CSS media query string
moveBy()	Moves a window relative to its current position
moveTo()	Moves a window to the specified position
open()	Opens a new browser window
print()	Prints the content of the current window
prompt()	Displays a dialog box that prompts the visitor for input
resizeBy()	Resizes the window by the specified pixels
resizeTo()	Resizes the window to the specified width and height
scroll()	Deprecated. This method has been replaced by the scrollTo() method.
scrollBy()	Scrolls the document by the specified number of pixels
scrollTo()	Scrolls the document to the specified coordinates
setInterval()	Calls a function or evaluates an expression at specified intervals (in milliseconds)
setTimeout()	Calls a function or evaluates an expression after a specified number of milliseconds
stop()	Stops the window from loading

3. The Navigator Object

Navigator Object

The navigator object contains information about the browser.

Note: There is no public standard that applies to the navigator object, but all major browsers support it.

Navigator Object Properties

Property	Description
appName	Returns the code name of the browser
appName	Returns the name of the browser
appVersion	Returns the version information of the browser
cookieEnabled	Determines whether cookies are enabled in the browser
geolocation	Returns a Geolocation object that can be used to locate the user's position
language	Returns the language of the browser
onLine	Determines whether the browser is online
platform	Returns for which platform the browser is compiled
product	Returns the engine name of the browser
userAgent	Returns the user-agent header sent by the browser to the server

Navigator Object Methods

Method	Description
javaEnabled()	Specifies whether or not the browser has Java enabled
taintEnabled()	Removed in JavaScript version 1.2. Specifies whether the browser has data tainting enabled

4. The Screen Object

Screen Object

The screen object contains information about the visitor's screen.

Note: There is no public standard that applies to the screen object, but all major browsers support it.

Screen Object Properties

Property	Description
availHeight	Returns the height of the screen (excluding the Windows Taskbar)
availWidth	Returns the width of the screen (excluding the Windows Taskbar)
colorDepth	Returns the bit depth of the color palette for displaying images
height	Returns the total height of the screen
pixelDepth	Returns the color resolution (in bits per pixel) of the screen
width	Returns the total width of the screen

5. 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.

Note: There is no public standard that applies to the history object, but all major browsers support it.

History Object Properties

Property	Description
length	Returns the number of URLs in the history list

History Object Methods

Method	Description
back()	Loads the previous URL in the history list
forward()	Loads the next URL in the history list
go()	Loads a specific URL from the history list