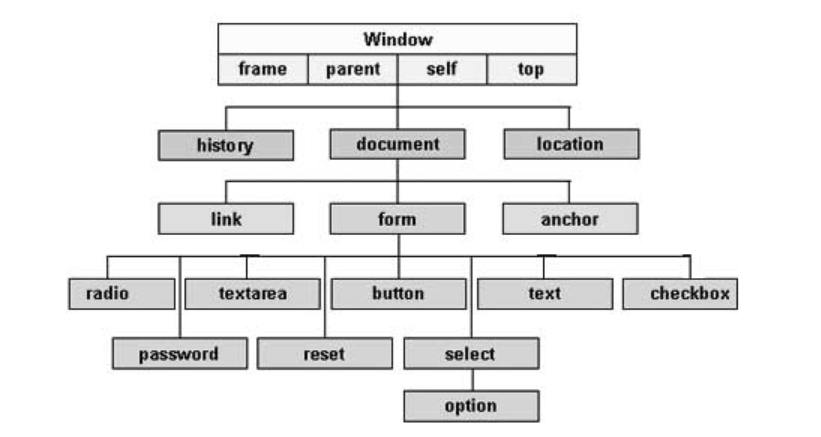
**Advanced JavaScripts**

Every web page resides inside a browser window which can be considered as an object.

A Document object represents the HTML document that is displayed in that window.



|  |  |
| --- | --- |
| window object | |
| Properties | Description |
| window.closed | This property indicates whether the current window is closed or not. |
| window.console | Returns a reference to the console object which provides access to the browser's debugging console. |
| window.document | Returns a reference to the document that the window contains. |
| window.fullScreen | This property indicates whether the window is displayed in full screen or not. |
| window.history | Returns a reference to the history object |
| window.innerHeight | Gets the height of the content area of the browser window including, if rendered, the horizontal scrollbar. |
| window.outerHeight | Gets the height of the outside of the browser window. |
| window.innerWidth | Gets the width of the content area of the browser window including, if rendered, the vertical scrollbar. |
| window.outerWidth | Gets the width of the outside of the browser window. |
| window.location | Gets/sets the location, or current URL, of the window object. |
| window.localStorage | Returns a reference to the local storage object used to store data that may only be accessed by the origin that created it. |
| window.name | Gets/sets the name of the window. |
| window.opener | Returns a reference to the window that opened this current window. |
| window.parent | Returns a reference to the parent of the current window or subframe. |
| window.screen | Returns a reference to the screen object associated with the window. |
| window.screenX | Returns the horizontal distance of the left border of the user's browser from the left side of the screen. |
| window.screenY | Returns the vertical distance of the top border of the user's browser from the top side of the screen. |
| window.scrollX | Returns the number of pixels that the document has already been scrolled horizontally. |
| window.scrollY | Returns the number of pixels that the document has already been scrolled vertically. |

|  |  |
| --- | --- |
| window object | |
| Methods | Description |
| window.alert() | Displays an alert dialog. |
| window.blur() | Sets focus away from the window. |
| window.close() | Closes the current window. |
| window.confirm() | Displays a dialog with a message that the user needs to respond to |
| window.focus() | Sets focus on the current window. |
| window.moveTo() | Moves the window to the specified coordinates. |
| window.open() | Opens a new window. |
| window.print() | Opens the Print Dialog to print the current document. |
| window.prompt() | Returns the text entered by the user in a prompt dialog. |
| window.resizeTo() | Dynamically resizes window. |
| window.scroll() | Scrolls the window to a particular place in the document. |
| window.stop() | This method stops window loading. |

The Screen interface represents a screen, usually the one on which the current window is being rendered, and is obtained using window.screen.

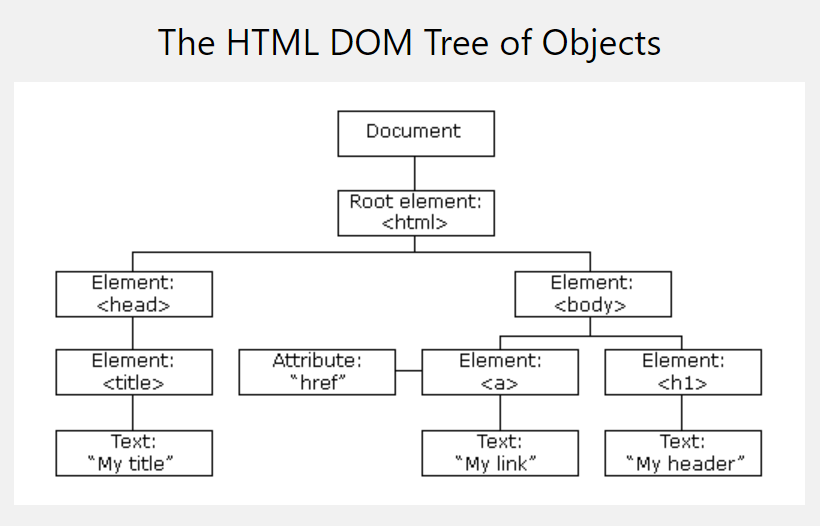
|  |  |
| --- | --- |
| screen object | |
| Properties | Description |
| screen.width | Returns the width of the screen |
| screen.top | Returns the distance in pixels from the top side of the current screen. |
| screen.left | Returns the distance in pixels from the left side of the main screen to the left side of the current screen. |
| screen.height | Returns the height of the screen in pixels |
| screen.availWidth | Returns the amount of horizontal space in pixels available to the window |
| screen.availTop | Specifies the y-coordinate of the first pixel that is not allocated to permanent or semipermanent user interface features. |
| screen.availLeft | Returns the first available pixel available from the left side of the screen. |
| screen.availHeight | Specifies the height of the screen, in pixels, minus permanent or semipermanent user interface features displayed by the operating system, such as the Taskbar on Windows. |

|  |  |
| --- | --- |
| Location Object | |
| Properties | Description |
| location.href | Is s DOMString containing the entire URL. If changed, the associated document navigates to the new page. |
| location.protocol | Is a DOMString containing the protocol scheme of the URL, including the final ':'. |
| location.host | Is a DOMString containing the host, that is the hostname, a ':', and the port of the URL. |
| location.pathname | Is a DOMString containing an initial '/' followed by the path of the URL |
| location.search | Is a DOMString containing a '?' followed by the parameters or "querystring" of the URL. |

|  |  |
| --- | --- |
| Location Object | |
| Methods | Description |
| location.assign() | Loads the resource at the URL provided in parameter |
| location.reload() | It Reloads the resource from the current URL. |

The **History** interface allows manipulation of the browser session history, that is the pages visited in the tab or frame that the current page is loaded in.

|  |  |
| --- | --- |
| History Object | |
| Properties and Methods | Description |
| history.length | Returns an Integer representing the number of elements in the session history, including the currently loaded page. |
| history.back() | Goes to the previous page in session history, the same action as when the user clicks the browser's Back button. Equivalent to history.go(-1). |
| history.forward() | Goes to the next page in session history, the same action as when the user clicks the browser |
| history.go() | Loads a page from the session history, identified by its relative location to the current page, for example -1 for the previous page or 1  for the next page. If you specify an out-of-bounds value (for instance, specifying -1 when there are no previously-visited pages in the session history), this method silently has no effect. Calling go() without parameters or a value of 0 reloads the current page. |



The **Document** interface represents any web page loaded in the browser and serves as an entry point into the web page's content, which is the DOM tree.

|  |  |
| --- | --- |
| document object | |
| Properties | Description |
| document.URL | Returns the document location as a string. |
| document.title | Sets or gets the title of the current document. |
| document.lastModified | Returns the date on which the document was last modified. |
| document.cookie | Returns a semicolon-separated list of the cookies for that document or sets a single cookie. |
| document.body | Returns the <body> node of the current document. |
| document.forms | Returns a list of the <form> elements within the current document. |
| document.head | Returns the <head> element of the current document. |
| document.images | Returns a list of the images in the current document. |
| document.links | Returns a list of all the hyperlinks in the document. |
| document.scripts | Returns all the <script> elements on the document. |
| document.styleSheetSets | Returns a list of the style sheet sets available on the document. |
| parentnode.childElementCount | Returns the number of children of this ParentNode which are elements. |
| parentnode.children | Returns a live HTML Collection containing all of the Element objects that are children of this ParentNode, omitting all of its non-element nodes. |

|  |  |
| --- | --- |
| document object | |
| methods | Description |
| document.getElementById(“node id name”) | <!DOCTYPE html>  <html>  <body>  <p id=**"intro"**>**Hello World!**</p>  <p>**This example demonstrates the** <b>**getElementById**</b> **method!**</p>  <p id=**"demo"**></p>  <script>  ***var*** myElement **=** document.getElementById**(**"intro"**);**  document.getElementById**(**"demo"**).**innerHTML **=**  "The text from the intro paragraph is " **+** myElement.innerHTML**;**  </script>  </body>  </html> |
| document.getElementsByTagname(“tag name”) | <!DOCTYPE html>  <html>  <body>  <p>**Hello World!**</p>  <p>**The DOM is very useful.**</p>  <p>**This example demonstrates the** <b>**getElementsByTagName**</b> **method**</p>  <p id=**"demo"**></p>  <script>  ***var*** x **=** document.getElementsByTagName**(**"p"**);**  document.getElementById**(**"demo"**).**innerHTML **=**  'The first paragraph (index 0) is: ' **+** x**[**0**].**innerHTML**;**  </script>  </body>  </html> |
| Document.getElementsByName(“name”) |  |
| document.getElementsByClassName | <!DOCTYPE html>  <html>  <body>  <p>**Hello World!**</p>  <p class=**"intro"**>**The DOM is very useful.**</p>  <p class=**"intro"**>**This example demonstrates the** <b>**getElementsByClassName**</b> **method.**</p>  <p id=**"demo"**></p>  <script>  ***var*** x **=** document.getElementsByClassName**(**"intro"**);**  document.getElementById**(**"demo"**).**innerHTML **=**  'The first paragraph (index 0) with class="intro": ' **+** x**[**0**].**innerHTML**;**  </script>  </body>  </html> |
| document.querySelectorAll(“css selector”)  or  document.querySelector(“css selector”); ///returns the first matched item | <!DOCTYPE html>  <html>  <body>  <p>**Hello World!**</p>  <p class=**"intro"**>**The DOM is very useful.**</p>  <p class=**"intro"**>**This example demonstrates the** <b>**querySelectorAll**</b> **method.**</p>  <p id=**"demo"**></p>  <script>  ***var*** x **=** document.querySelectorAll**(**"p.intro"**);**  document.getElementById**(**"demo"**).**innerHTML **=**  'The first paragraph (index 0) with class="intro": ' **+** x**[**0**].**innerHTML**;**  </script>  </body>  </html> |
| document.forms | <!DOCTYPE html>  <html>  <body>  <form id=**"frm1"** action=**"/action\_page.php"**>  **First name:** <input type=**"text"** name=**"fname"** value=**"Donald"**><br>  **Last name:** <input type=**"text"** name=**"lname"** value=**"Duck"**><br><br>  <input type=**"submit"** value=**"Submit"**>  </form>  <p>**Click "Try it" to display the value of each element in the form.**</p>  <button onclick=**"myFunction()"**>**Try it**</button>  <p id=**"demo"**></p>  <script>  ***function*** myFunction**()** **{**  ***var*** x **=** document.forms**[**"frm1"**];**  ***var*** text **=** ""**;**  ***var*** i**;**  ***for*** **(**i **=** 0**;** i **<** x.length **;**i**++)** **{**  text **+=** x.elements**[**i**].**value **+** "<br>"**;**  **}**  document.getElementById**(**"demo"**).**innerHTML **=** text**;**  **}**  </script>  </body>  </html> |
| **Changing HTML Content:**  document.getElementById(id).innerHTML= “new value” | <html>  <body>  <p id=**"p1"**>**Hello World!**</p>  <script>  document.getElementById**(**"p1"**).**innerHTML **=** "New text!"**;**  </script>  </body>  </html> |
| **Changing the value of an Attribute:**  document.getElementById(*id*).*attribute = new value* | <!DOCTYPE html>  <html>  <body>  <img id=**"myImage"** src=**"smiley.gif"**>  <script>  document.getElementById**(**"myImage"**).**src **=** "landscape.jpg"**;**  </script>  </body>  </html> |
| **Changing CSS property values:** document.getElementById(id).style.property = new style | <html>  <body>  <p id=**"p2"**>**Hello World!**</p>  <script>  document.getElementById**(**"p2"**).**style.color **=** "blue"**;**  </script>  <p>**The paragraph above was changed by a script.**</p>  </body>  </html> |
| 1. document.createElement(element) 🡪 to create an HTML element 2. node.removeChild(node) 🡪 to remove an HTML element 3. node.appendChild(node) 🡪 to add an HTML element 4. node.replaceChild(node) 🡪 to replace an HTML element 5. document.write(text) 🡪 to write into the HTML output stream | 1. **to create a new node,** <!DOCTYPE html>   <html>  <body>  <div id=**"div1"**>  <p id=**"p1"**>**This is a paragraph.**</p>  <p id=**"p2"**>**This is another paragraph.**</p>  </div>  <script>  ***var*** para **=** document.createElement**(**"p"**);**  ***var*** node **=** document.createTextNode**(**"This is new."**);**  para.appendChild**(**node**);**  ***var*** element **=** document.getElementById**(**"div1"**);**  ***var*** child **=** document.getElementById**(**"p1"**);**  element.insertBefore**(**para**,**child**);**  </script>  </body>  </html>   1. to replace  <!DOCTYPE html>   <html>  <body>  <div id=**"div1"**>  <p id=**"p1"**>**This is a paragraph.**</p>  <p id=**"p2"**>**This is another paragraph.**</p>  </div>  <script>  ***var*** parent **=** document.getElementById**(**"div1"**);**  ***var*** child **=** document.getElementById**(**"p1"**);**  ***var*** para **=** document.createElement**(**"p"**);**  ***var*** node **=** document.createTextNode**(**"This is new."**);**  para.appendChild**(**node**);**  parent.replaceChild**(**para**,**child**);**  </script>  </body>  </html>   1. to remove, <!DOCTYPE html>   <html>  <body>  <div id=**"div1"**>  <p id=**"p1"**>**This is a paragraph.**</p>  <p id=**"p2"**>**This is another paragraph.**</p>  </div>  <script>  ***var*** parent **=** document.getElementById**(**"div1"**);**  ***var*** child **=** document.getElementById**(**"p1"**);**  parent.removeChild**(**child**);**  </script>  </body>  </html> |