

Window Object Methods

Method	Description
alert()	<p>Displays an alert box with a message and an OK button</p> <pre><script> window.alert("hello"); </script></pre>
atob()	<p>Decodes a base-64 encoded string</p> <pre><script> varstr = "Hello World!"; varenc = window.btoa(str); vardec = window.atob(enc); var res = "Encoded String: " + enc + "
" + "Decoded String: " + dec; alert(res); </script></pre>
blur()	<p>Removes focus from the current window</p> <pre><script> varlitwindow= window.open("", "", "width=200, height=100"); litwindow.document.write("<p>A new window!</p>"); litwindow.blur(); </script></pre>
btoa()	<p>Encodes a string in base-64</p> <pre><script> varstr = "Hello World!"; varenc = window.btoa(str); var res = "Encoded String: " + enc; alert(res); </script></pre>
clearInterval()	<p>Clears a timer set with setInterval()</p> <pre><script> varmyVar = setInterval(function(){ myTimer() }, 1000); function myTimer() { var d = new Date(); var t = d.toLocaleTimeString(); document.getElementById("demo").innerHTML = t; } function myStopFunction() { clearInterval(myVar); } </script> <p id="demo"></p> <input type="button" value="stop" onClick="myStopFunction()" /></pre>
clearTimeout()	<p>Clears a timer set with setTimeout()</p> <pre><script> varmyVar; function myFunction() { myVar = setTimeout(function(){ alert("Hello"); }, 1000); } function myStopFunction() { clearTimeout(myVar); }</pre>

```

</script>
<input type="button" value="Start" onClick="myFunction()" />
<input type="button" value="Stop" onClick="myStopFunction()" />

```

[close\(\)](#)

Closes the current window

```

<script>
varlitwindow= window.open("", "", "width=200, height=100");
litwindow.document.write("<p>A new window!</p>"); function
closewindow()
{
    litwindow.close();
}
</script>
<input type="button" value="close" onClick="closewindow()" />

```

[confirm\(\)](#)

Displays a dialog box with a message and an OK and a Cancel button.

```

<script>
window.confirm("Are you sure to delete?"); </script>

```

[createPopup\(\)](#)

Creates a pop-up window

```

<script> function show_popup() { var
p = window.createPopup(); varpbody
= p.document.body;
pbody.style.backgroundColor = "lime";
pbody.style.border = "solid black
1px";
pbody.innerHTML = "This is a pop-up! Click outside to close.";
p.show(150,150,200,50,document.body);
}
</script>
<input type="button" value="Show" onClick="show_popup()" />

```

As of Internet Explorer 11, the createPopup() method is no longer supported in other browser.

[focus\(\)](#)

Sets focus to the current window.

```

<script> varlitwindow= window.open("", "", "width=200,
height=100"); litwindow.document.write("<p>A new
window!</p>"); litwindow.focus();
</script>

```

[getComputedStyle\(\)](#)

Gets the current computed CSS styles applied to an element

```

<script>
function getTheStyle() { varelem =
document.getElementById("elem-container");
varcssprop = window.getComputedStyle(elem,null);}
</script>

```

[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. myWindow.moveBy(250,250);
moveTo()	Moves a window to the specified position myWindow.moveTo(100, 100);
open()	Opens a new browser window. window.open(URL, name, specs, replace) Here replace(optional): true - URL replaces the current document in the history list false - URL creates a new entry in the history list. window.open("", "", "width=200, height=100");
print()	Prints the content of the current window. window.print();
prompt()	Displays a dialog box that prompts the visitor for input. window.prompt("Enter name","LIT");
resizeBy()	Resizes the window by the specified pixels myWindow.resizeBy(250, 250);
resizeTo()	Resizes the window to the specified width and height myWindow.resizeTo(250, 250);
scroll()	Deprecated. This method has been replaced by the scrollTo() method. <script> x = 0; \$(document).ready(function(){ \$("div").scroll(function(){ \$("span").text(x+= 1); }); }); </script> <div>welcome to LIT</div>

[scrollBy\(\)](#) Scrolls the document by the specified number of pixels

```
<script> function
scrollWin() {
window.scrollBy(100, 0);
} </script>
```

[scrollTo\(\)](#) Scrolls the document to the specified coordinates

```
<script> function
scrollWin() {
window.scrollTo(500,
0);
}
</script>
```

[setInterval\(\)](#) Calls a function or evaluates an expression at specified intervals (in milliseconds) <script>

```
function myFunction() { setInterval(function(){
alert("Hello"); }, 3000);
}
</script>
```

[setTimeout\(\)](#) Calls a function or evaluates an expression after a specified number of milliseconds <script>

```
function myFunction() { setTimeout(function(){
alert("Hello"); }, 3000); }
</script>
```

[stop\(\)](#) Stops the window from loading. Means it does not display the content of window.

```
window.stop();
```

Window Object Properties

Property

Description

[closed](#) Returns a Boolean value indicating whether a window has been closed or not.

```
<script>
if (litwindow.closed) {

alert("window closed!");
} else {
alert("window is open!");
}
</script>
```

[defaultStatus](#) Sets or returns the default text in the statusbar of a window.

```
<script>
```

```
var lit=window.open("", "", "width=200, height=100");
lit.defaultStatus="LIT Susant K Rout";
</script>
```

(Not supported in any browser)

[document](#)

Returns the Document object for the window. The document is a part of the Window object and can be accessed as **window.document**.

```
<script>
window.document.write("hello student");
</script>
```

[frameElement](#)

Returns the <iframe> element in which the current window is inserted.

```
// If the window is in an <iframe>, change its source
<button onclick="myFunction()">Check IT</button>
<script>
function myFunction() { var frame =
window.frameElement; if (frame)
{ frame.src =
"http://www.litindia.in/";
}
}
</script>
```

[frames](#)

Returns all <iframe> elements in the current window.

```
<button onclick="myFunction()">Try it</button>
<br><br>
<iframe src="http://www.litindia.in"></iframe>
<script>
function myFunction() { window.frames[0].location =
"http://www.litsolution.in"; }
</script>
```

[history](#)

Returns the History object for the window.

```
<script>
var x = history.length;//properties
document.write(x);
</script>
```

The methods of history objects are
History.go(), history.forward(), history.back()

[innerHeight](#)

Returns the inner height of a window's content area.

```
window.innerHeight;
```

[innerWidth](#)

Returns the inner width of a window's content area

```
window.innerWidth;
```

[length](#)

Returns the number of <iframe> elements in the current window.

```
<iframe src="x.php"></iframe>
<iframe src="y.php"></iframe>

<script>
var x = window.length;
document.write(x)
;
</script>
```

[location](#)

Returns the Location object for the window.
window.location="lit.php";

[name](#)

Sets or returns the name of a window.
<button onclick="myFunction()">Try it</button>
<script> function myFunction() { varmyWindow =
window.open("", "MsgWindow", "width=200, height=100");
myWindow.document.write("<p>This window's name is: " +
myWindow.name + "</p>");
}
</script>

[navigator](#)

Returns the Navigator object for the window.
<script>
var x = "Platform: " + window.navigator.platform;
document.write(x);
</script>

[opener](#)

Returns a reference to the window that created the window.
<button onclick="myFunction()">Try it</button>
<script> function myFunction() { varmyWindow =
window.open("", "myWindow", "width=200, height=100");
myWindow.document.write("<p>This is 'myWindow'</p>");
myWindow.opener.document.write("<p>This is the source
window!</p>");
}
</script>

[outerHeight](#)

Returns the outer height of a window, including toolbars/scrollbars.
window.outerHeight;
<script>
function myFunction() {

```
var w = window.outerWidth; var h = window.outerHeight;
document.getElementById("demo").innerHTML = "Width: " + w
+ "<br>Height: " + h;
}
</script>
```

[outerWidth](#)

Returns the outer width of a window, including toolbars/scrollbars.

```
window.outerWidth;
<script> function myFunction() { var w = window.outerWidth;
var h = window.outerHeight;
document.getElementById("demo").innerHTML = "Width: " + w
+
"<br>Height: " + h;
}
</script>
```

[pageXOffset](#)

Returns the pixels the current document has been scrolled (horizontally) from the upper left corner of the window.

```
window.pageXOffset;
<script> function myFunction() { window.scrollTo(100, 100);
alert("pageXOffset: " + window.pageXOffset + ", pageYOffset: " +
window.pageYOffset);
}
</script>
```

[pageYOffset](#)

Returns the pixels the current document has been scrolled (vertically) from the upper left corner of the window.

```
window.pageYOffset;
```

[parent](#)

Returns the parent window of the current window.

```
<script>
window.parent.document.body.style.backgroundColor = "red"; </script>
```

[screen](#)

Returns the Screen object for the window.

```
<script> var x = "Total Width: " +
screen.width; document.write(x);
</script>
```

[screenLeft](#)

Returns the horizontal coordinate of the window relative to the screen

```
<script>
function myFunction() { varmyWindow = window.open("", "myWin");
myWindow.document.write("<p>This is 'myWin'");
myWindow.document.write("<br>ScreenLeft: " + myWindow.screenLeft);
myWindow.document.write("<br>ScreenTop: " + myWindow.screenTop
+

```

```
"</p>");
}
</script>
```

[screenTop](#) Returns the vertical coordinate of the window relative to the screen

[screenX](#) Returns the horizontal coordinate of the window relative to the screen

```
<script>
function myFunction() { varmyWindow = window.open("", "myWin");
myWindow.document.write("<p>This is 'myWin'");
myWindow.document.write("<br>ScreenX: " + myWindow.screenX);
myWindow.document.write("<br>ScreenY: " + myWindow.screenY
+ "</p>");
}
</script>
```

[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)

```
<script>
// Check browser support
if (typeof(Storage) !== "undefined") {
    // Store
    localStorage.setItem("lastname", "Smith");
    // Retrieve
    document.getElementById("result").innerHTML =
    localStorage.getItem("lastname");
} else {
    document.getElementById("result").innerHTML = "Sorry, your browser
does not support Web Storage...";
}
</script>
```

[scrollX](#) An alias of [pageXOffset](#)

[scrollY](#) An alias of [pageYOffset](#)

[self](#) Returns the current window

```
<script>
function myFunction() {    if (window.top !== window.self) {
document.getElementById("demo").innerHTML = "This window is not the
topmost window! Am I in a frame?";
    } else {
document.getElementById("demo").innerHTML = "This window is the
```



```
topmost window!";  
    }  
}</script>
```

[status](#)

Sets or returns the text in the statusbar of a window

```
<script>  
window.status = "Some text in the status bar!!";  
</script>
```

[top](#)

Returns the topmost browser window

```
<script>  
function myFunction() {    if (window.top != window.self) {  
document.getElementById("demo").innerHTML = "This window is not the  
topmost window! Am I in a frame?";  
    } else {  
document.getElementById("demo").innerHTML = "This window is the  
topmost window!";  
    }  
}</script>
```