Window Interface

HTML-5.com is an HTML User's Guide and quick reference of HTML elements and attributes for web developers who code HTML web pages, not only for HTML 5 but for HTML coding in general, with demos and examples of HTML code plus a cheat sheet for web developers. TV Series & Actors and Actresses. Follow TV Series and HTML 5 on Google+.

HTML-5.com ➢ itemscopehttp://data-vocabulary.org/Breadcrumb<span itemprop="title">HTML 5</span> ➢ itemscopehttp://data-vocabulary.org/Breadcrumb<span itemprop="title">JavaScript Code</span> ➢ itemscopehttp://data-vocabulary.org/Breadcrumb<span itemprop="title">JavaScript DOM</span> ➢ itemscopehttp://data-vocabulary.org/Breadcrumb

Window Interface

Window Interface

The <dfn>Window interface</dfn> provides properties and methods for accessing the windows and documents in the object model. A <dfn>window</dfn> object is an instance that implements the Window interface and represents the area where a Document is loaded into the browser. In a tabbed browser, each tab has its own window object; one window does *not* have access to the window object in any other browser tab. There is also one separate window object for each browser window and some methods, such as resizeTo and resizeBy, are delegated to one of these because they apply to a whole application window rather than a single browser tab.

Superinterfaces:

AbstractView

Implements:

EventTarget

Use:

window object

DOM Properties	Supported	
get only	ApplicationCache	**window.applicationCache**
get only	boolean	**window.closed**
get or set	DOMString	**window.defaultStatus** **window.defaultStatus = *value***
get only	DOMObject	**window.dialogArguments**
get only	Document	**window.document**
get only	Element	**window.frameElement**
get only	WindowProxy	**window.frames**

get only	History	**window.history**
get only	int	**window.innerHeight**
get only	int	**window.innerWidth**
get only	unsigned long	**window.length**
get only	Storage	**window.localStorage**
get only	Location	**window.location**
get only	BarProp	**window.locationbar**
get only	BarProp	**window.menubar**
get or set	DOMString	**window.name** **window.name = *value***
get only	Navigator	**window.navigator**
get only	WindowProxy	**window.opener**

get only	Opera	**window.opera**
		addEventListener(DOMString type, any function, boolean capturePhase)
		buildNumber()
		buildNumber("inconspicuous")
		clearErrorMessages()
		collect()
		initiate garbage collection
		defineMagicFunction(DOMString functionName, function function)
		defines a function for a method
		defineMagicFunction(DOMString variableName, function getterFunction,

		function setterFunction)
		defines functions for getting and setting a property
		errorIndex()
		errorMessage(int index)
		getOverrideHistoryNavigatio
		getPreference(DOMString section, DOMString preferenceName)
		getPreferenceDefault(DOMS section, DOMString preferenceName)
		isRegistered()
		openInSourceViewer(DOMS url)
		popXSLTransform()
		postError(DOMString message, ...)
		pushXSLTransform(Node xslStyleSheetNode)
		setOverrideHistoryNavigatio mode)
		setPreference(DOMString section, DOMString preferenceName, DOMString value)
		SRPNumber()
		version()

get only	int	**window.outerHeight**
get only	int	**window.outerWidth**

get only	int	**window.pageXOffset**
get only	int	**window.pageYOffset**
get only	WindowProxy	**window.parent**
get only	BarProp	**window.personalbar**
get or set	any	**window.returnValue** **window.returnValue = *value*** A property of windows created with the showModalDialog method that can be used to set or get the value returned by the modal dialog window.
get only	BarProp	**window.scrollbars**
get only	WindowProxy	**window.self**
get only	Storage	**window.sessionStorage**
get only	DOMString	**window.status**
get only	BarProp	**window.statusbar**
get only	BarProp	**window.toolbar**
get only	WindowProxy	**window.top**
get only	WindowProxy	**window.window**

DOM Interface Methods	Supported
void	**window.addEventListener(DOMString *type*, EventListener *listener*, boolean *useCapture*)**
void	**window.alert(DOMString *message*)**
void	**window.blur()**
void	**window.clearInterval(int *intervalID*)**
void	**window.clearTimeout(int *timeoutID*)**
void	**window.close()**
boolean	**window.confirm(DOMString *message*)**
boolean	**window.dispatchEvent(Event *event*)**
void	**window.focus()**

CSSStyleDeclaration	**window.getComputedStyle(Element *element*)**
Selection	**window.getSelection()**
void	**window.moveBy(int *deltaX*, int *deltaY*)**
void	**window.moveTo(int *x*, int *y*)**
WindowProxy	**window.open(DOMString *url*, DOMString *target*, DOMString *features*, DOMString *replace*)**
void	**window.postMessage(DOMString *message*, URI *targetOrigin*)**
void	**window.print()**
DOMString	**window.prompt(DOMString *promptText*, DOMString *defaultValue*)**
void	**window.removeEventListener(DOMString *type*, EventListener *listener*, boolean *useCapture*)**
void	**window.resizeBy(int *deltaX*, int *deltaY*)**
void	**window.resizeTo(int *width*, int *height*)**
void	**window.scroll(int *x*, int *y*)**
void	**window.scrollBy(int *deltaX*, int *deltaY*)**
void	**window.scrollTo(int *x*, int *y*)**
int intervalID	**window.setInterval(DOMString *code*, int *delayInMilliseconds*)**
int timeoutID	**window.setTimeout(DOMString *code*, int *delayInMilliseconds*)**
any	**window.showModalDialog(DOMString *url*, any *argument*)**
void	**window.stop()**

DOM Events		
Event Type	Property	Supported
abort	Function	onabort
afterprint	Function	onafterprint
beforeprint	Function	onbeforeprint
beforeunload	Function	onbeforeunload

blur	Function	onblur
canplay	Function	oncanplay
canplaythrough	Function	oncanplaythrough
change	Function	onchange
click	Function	onclick
contextmenu	Function	oncontextmenu
cuechange	Function	oncuechange
dblclick	Function	ondblclick
drag	Function	ondrag
dragend	Function	ondragend
dragenter	Function	ondragenter
dragleave	Function	ondragleave
dragover	Function	ondragover
dragstart	Function	ondragstart
drop	Function	ondrop
durationchange	Function	ondurationchange
emptied	Function	onemptied
ended	Function	onended
error	Function	onerror
focus	Function	onfocus
formchange	Function	onformchange
forminput	Function	onforminput
hashchange	Function	onhashchange
input	Function	oninput
invalid	Function	oninvalid
keydown	Function	onkeydown
keypress	Function	onkeypress
keyup	Function	onkeyup
load	Function	onload
loadeddata	Function	onloadeddata
loadedmetadata	Function	onloadedmetadata
loadstart	Function	onloadstart
message	Function	onmessage
mousedown	Function	onmousedown

mousemove	Function	onmousemove
mouseout	Function	onmouseout
mouseover	Function	onmouseover
mouseup	Function	onmouseup
mousewheel	Function	onmousewheel
offline	Function	onoffline
online	Function	ononline
pause	Function	onpause
play	Function	onplay
playing	Function	onplaying
pagehide	Function	onpagehide
pageshow	Function	onpageshow
popstate	Function	onpopstate
progress	Function	onprogress
ratechange	Function	onratechange
readystatechange	Function	onreadystatechange
redo	Function	onredo
reset	Function	onreset
resize	Function	onresize
scroll	Function	onscroll
seeked	Function	onseeked
seeking	Function	onseeking
select	Function	onselect
show	Function	onshow
stalled	Function	onstalled
storage	Function	onstorage
submit	Function	onsubmit
suspend	Function	onsuspend
timeupdate	Function	ontimeupdate
undo	Function	onundo
unload	Function	onunload
volumechange	Function	onvolumechange
waiting	Function	onwaiting

Window onload Example

```
<script type="text/javascript">
window.addEventListener("load", function init() {
   alert('"Hello, World!" from Window load event');
}, false);
</script>
```

<script type="text/javascript"> window.addEventListener("load", function props(e) { function addCell(row, text) { var cell = row.insertCell(-1); cell.appendChild(document.createTextNode(text)); } var e = e || window.event; document.getElementById('eventType').innerHTML = e.type; var table = document.createElement("table"); if (! (table instanceof HTMLTableElement)) alert("Table should be [object HTMLTableElement] but is " + table); table.setAttribute ("class", "border"); var thead = table.createTHead(); var row = thead.insertRow(-1); var lableList = ['#', 'Property', 'Value']; var len = lableList.length; for (var i = 0; i < len; i++) { addCell(row, lableList[i]); } var tbody = document.createElement("tbody"); table.appendChild(tbody); for (var p in e) { row = tbody.insertRow(-1); row.className = (row.rowIndex % 2)? 'odd':'even'; addCell(row, row.rowIndex); addCell(row, p); addCell(row, e[p]); } document.getElementById("eventProps").appendChild(table); }, false); </script>
Event: <span id="eventType"></span>

back to top

THE END