

Layer

Corresponds to a layer in an HTML page and provides a means for manipulating that layer.

Client-side object

Implemented in JavaScript 1.2

Created by The HTML `LAYER` or `ILAYER` tag, or using cascading style sheet syntax. The JavaScript runtime engine creates a **Layer** object corresponding to each layer in your document. It puts these objects in an array in the `document.layers` property. You access a **Layer** object by indexing this array.

To define a layer, use standard HTML syntax. If you specify the `ID` attribute, you can use the value of that attribute to index into the `layers` array.

For a complete description of layers, see *Dynamic HTML in Netscape Communicator*.

Some `layer` properties can be directly modified by assignment; for example, "`mylayer.visibility = hide`". A layer object also has methods that can affect these properties.

- Event handlers**
- `onMouseOver`
 - `onMouseOut`
 - `onLoad`
 - `onFocus`
 - `onBlur`

Property Summary

Property	Description
<code>above</code>	The layer object above this one in z-order, among all layers in the document or the enclosing window object if this layer is topmost.
<code>background</code>	The image to use as the background for the layer's canvas.
<code>bgColor</code>	The color to use as a solid background color for the layer's canvas.
<code>below</code>	The layer object below this one in z-order, among all layers in the document or null if this layer is at the bottom.
<code>clip.bottom</code>	The bottom edge of the clipping rectangle (the part of the layer that is visible.)

Property	Description
<code>clip.height</code>	The height of the clipping rectangle (the part of the layer that is visible.)
<code>clip.left</code>	The left edge of the clipping rectangle (the part of the layer that is visible.)
<code>clip.right</code>	The right edge of the clipping rectangle (the part of the layer that is visible.)
<code>clip.top</code>	The top edge of the clipping rectangle (the part of the layer that is visible.)
<code>clip.width</code>	The width of the clipping rectangle (the part of the layer that is visible.)
<code>document</code>	The layer's associated document.
<code>left</code>	The horizontal position of the layer's left edge, in pixels, relative to the origin of its parent layer.
<code>name</code>	A string specifying the name assigned to the layer through the <code>ID</code> attribute in the <code>LAYER</code> tag.
<code>pageX</code>	The horizontal position of the layer, in pixels, relative to the page.
<code>pageY</code>	The vertical position of the layer, in pixels, relative to the page.
<code>parentLayer</code>	The <code>layer</code> object that contains this layer, or the enclosing <code>window</code> object if this layer is not nested in another layer.
<code>siblingAbove</code>	The <code>layer</code> object above this one in z-order, among all layers that share the same parent layer, or null if the layer has no sibling above.
<code>siblingBelow</code>	The <code>layer</code> object below this one in z-order, among all layers that share the same parent layer, or null if layer is at the bottom.
<code>src</code>	A string specifying the URL of the layer's content.
<code>top</code>	The vertical position of the layer's top edge, in pixels, relative to the origin of its parent layer.
<code>visibility</code>	Whether or not the layer is visible.
<code>window</code>	The <code>window</code> or <code>Frame</code> object that contains the layer, regardless of whether the layer is nested within another layer.
<code>x</code>	A convenience synonym for <code>Layer.left</code> .
<code>y</code>	A convenience synonym for <code>Layer.top</code> .
<code>zIndex</code>	The relative z-order of this layer with respect to its siblings.

Method Summary

Method	Description
<code>captureEvents</code>	Sets the window or document to capture all events of the specified type.
<code>handleEvent</code>	Invokes the handler for the specified event.
<code>load</code>	Changes the source of a layer to the contents of the specified file, and simultaneously changes the width at which the layer's HTML contents will be wrapped.
<code>moveAbove</code>	Stacks this layer above the layer specified in the argument, without changing either layer's horizontal or vertical position.
<code>moveBelow</code>	Stacks this layer below the specified layer, without changing either layer's horizontal or vertical position.
<code>moveBy</code>	Changes the layer position by applying the specified deltas, measured in pixels.
<code>moveTo</code>	Moves the top-left corner of the window to the specified screen coordinates.
<code>moveToAbsolute</code>	Changes the layer position to the specified pixel coordinates within the page (instead of the containing layer.)
<code>releaseEvents</code>	Sets the layer to release captured events of the specified type, sending the event to objects further along the event hierarchy.
<code>resizeBy</code>	Resizes the layer by the specified height and width values (in pixels).
<code>resizeTo</code>	Resizes the layer to have the specified height and width values (in pixels).
<code>routeEvent</code>	Passes a captured event along the normal event hierarchy.

In addition, this object inherits the `watch` and `unwatch` methods from `Object`.

Note Just as in the case of a document, if you want to define mouse click response for a layer, you must capture `onMouseDown` and `onMouseUp` events at the level of the layer and process them as you want.

For details about capturing events, see the *Client-Side JavaScript Guide*.

If an event occurs in a point where multiple layers overlap, the topmost layer gets the event, even if it is transparent. However, if a layer is hidden, it does not get events.

above

The `layer` object above this one in z-order, among all layers in the document or the enclosing window object if this layer is topmost.

Property of **Layer**

Read-only

Implemented in JavaScript 1.2

background

The image to use as the background for the layer's canvas (which is the part of the layer within the clip rectangle).

Property of **Layer**

Implemented in JavaScript 1.2

Description Each layer has a background property, whose value is an image object, whose `src` attribute is a URL that indicates the image to use to provide a tiled backdrop. The value is null if the layer has no backdrop. For example:

```
layer.background.src = "fishbg.gif";
```

below

The `layer` object below this one in z-order, among all layers in the document or null if this layer is at the bottom.

Property of **Layer**

Read-only

Implemented in JavaScript 1.2

bgColor

A string specifying the color to use as a solid background color for the layer's canvas (the part of the layer within the clip rectangle).

Property of **Layer**

Implemented in JavaScript 1.2

Description The `bgColor` property is expressed as a hexadecimal RGB triplet or as a string literal (see the *Client-Side JavaScript Guide*). This property is the JavaScript reflection of the `BGCOLOR` attribute of the `BODY` tag.

You can set the `bgColor` property at any time.

If you express the color as a hexadecimal RGB triplet, you must use the format `rrggbb`. For example, the hexadecimal RGB values for salmon are `red=FA`, `green=80`, and `blue=72`, so the RGB triplet for salmon is `"FA8072"`.

Examples The following example sets the background color of the `myLayer` layer's canvas to aqua using a string literal:

```
myLayer.bgColor="aqua"
```

The following example sets the background color of the `myLayer` layer's canvas to aqua using a hexadecimal triplet:

```
myLayer.bgColor="00FFFF"
```

See also `Layer.bgColor`

captureEvents

Sets the window or document to capture all events of the specified type.

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `captureEvents(eventType)`

Parameters

`eventType` Type of event to be captured. Available event types are listed in the *Client-Side JavaScript Guide*.

Description When a window with frames wants to capture events in pages loaded from different locations (servers), you need to use `captureEvents` in a signed script and precede it with `enableExternalCapture`. For more information and an example, see `enableExternalCapture`.

`captureEvents` works in tandem with `releaseEvents`, `routeEvent`, and `handleEvent`. For information on handling events, see the *Client-Side JavaScript Guide*.

clip.bottom

The bottom edge of the clipping rectangle (the part of the layer that is visible.) Any part of a layer that is outside the clipping rectangle is not displayed.

Property of **Layer**

Implemented in JavaScript 1.2

clip.height

The height of the clipping rectangle (the part of the layer that is visible.) Any part of a layer that is outside the clipping rectangle is not displayed.

Property of **Layer**

Implemented in JavaScript 1.2

clip.left

The left edge of the clipping rectangle (the part of the layer that is visible.) Any part of a layer that is outside the clipping rectangle is not displayed.

Property of **Layer**

Implemented in JavaScript 1.2

clip.right

The right edge of the clipping rectangle (the part of the layer that is visible.) Any part of a layer that is outside the clipping rectangle is not displayed.

Property of **Layer**

Implemented in JavaScript 1.2

clip.top

The top edge of the clipping rectangle (the part of the layer that is visible.) Any part of a layer that is outside the clipping rectangle is not displayed.

Property of **Layer**

Implemented in JavaScript 1.2

clip.width

The width of the clipping rectangle (the part of the layer that is visible.) Any part of a layer that is outside the clipping rectangle is not displayed.

Property of **Layer**

Implemented in JavaScript 1.2

document

The layer's associated document.

Property of **Layer**

Read-only

Implemented in JavaScript 1.2

Description Each `layer` object contains its own `document` object. This object can be used to access the images, applets, embeds, links, anchors and layers that are contained within the layer. Methods of the `document` object can also be invoked to change the contents of the layer.

handleEvent

Invokes the handler for the specified event.

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `handleEvent(event)`

Parameters

`event` Name of an event for which the specified object has an event handler.

Description `handleEvent` works in tandem with `captureEvents`, `releaseEvents`, and `routeEvent`. For information on handling events, see the *Client-Side JavaScript Guide*.

left

The horizontal position of the layer's left edge, in pixels, relative to the origin of its parent layer.

Property of **Layer**

Implemented in JavaScript 1.2

The `Layer.x` property is a convenience synonym for the `left` property.

See also `Layer.top`

load

Changes the source of a layer to the contents of the specified file and simultaneously changes the width at which the layer's HTML contents are wrapped.

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `load(sourcestring, width)`

Parameters

`sourcestring` A string indicating the external file name.

`width` The width of the layer as a pixel value.

moveAbove

Stacks this layer above the layer specified in the argument, without changing either layer's horizontal or vertical position. After re-stacking, both layers will share the same parent layer.

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `moveAbove(aLayer)`

Parameters

`aLayer` The layer above which to move the current layer.

moveBelow

Stacks this layer below the specified layer, without changing either layer's horizontal or vertical position. After re-stacking, both layers will share the same parent layer.

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `moveBelow(aLayer)`

Parameters

`aLayer` The layer below which to move the current layer.

moveBy

Changes the layer position by applying the specified deltas, measured in pixels.

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `moveBy(horizontal, vertical)`

Parameters

`horizontal` The number of pixels by which to move the layer horizontally.

`vertical` The number of pixels by which to move the layer vertically.

moveTo

Moves the top-left corner of the window to the specified screen coordinates.

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `moveTo(x-coordinate, y-coordinate)`

Parameters

`x-coordinate` An integer representing the top edge of the window in screen coordinates.

`y-coordinate` An integer representing the left edge of the window in screen coordinates.

Security To move a window offscreen, call the `moveTo` method in a signed script. For information on security, see the *Client-Side JavaScript Guide*.

Description Changes the layer position to the specified pixel coordinates within the containing layer. For `ILayers`, moves the layer relative to the natural inflow position of the layer.

See also `Layer.moveBy`

moveToAbsolute

Changes the layer position to the specified pixel coordinates within the page (instead of the containing layer.)

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `moveToAbsolute(x, y)`

Parameters

`x` An integer representing the top edge of the window in pixel coordinates.

`y` An integer representing the left edge of the window in pixel coordinates.

Description This method is equivalent to setting both the `pageX` and `pageY` properties of the `layer` object.

Layer.name

name

A string specifying the name assigned to the layer through the `ID` attribute in the `LAYER` tag.

Property of **Layer**

Read-only

Implemented in JavaScript 1.2

pageX

The horizontal position of the layer, in pixels, relative to the page.

Property of **Layer**

Implemented in JavaScript 1.2

pageY

The vertical position of the layer, in pixels, relative to the page.

Property of **Layer**

Implemented in JavaScript 1.2

parentLayer

The `layer` object that contains this layer, or the enclosing `window` object if this layer is not nested in another layer.

Property of **Layer**

Read-only

Implemented in JavaScript 1.2

releaseEvents

Sets the window or document to release captured events of the specified type, sending the event to objects further along the event hierarchy.

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `releaseEvents(eventType)`

Parameters

`eventType` Type of event to be captured.

Description If the original target of the event is a window, the window receives the event even if it is set to release that type of event. `releaseEvents` works in tandem with `captureEvents`, `routeEvent`, and `handleEvent`. For more information, see the *Client-Side JavaScript Guide*.

resizeBy

Resizes the layer by the specified height and width values (in pixels).

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `resizeBy(width, height)`

Parameters

`width` The number of pixels by which to resize the layer horizontally.

`height` The number of pixels by which to resize the layer vertically.

Description This does not layout any HTML contained in the layer again. Instead, the layer contents may be clipped by the new boundaries of the layer. This method has the same effect as adding `width` and `height` to `clip.width` and `clip.height`.

resizeTo

Resizes the layer to have the specified height and width values (in pixels).

Method of **Layer**

Implemented in JavaScript 1.2

Description This does not layout any HTML contained in the layer again. Instead, the layer contents may be clipped by the new boundaries of the layer.

Syntax `resizeTo(width, height)`

Parameters

`width` An integer representing the layer's width in pixels.

`height` An integer representing the layer's height in pixels.

Description This method has the same effect setting `clip.width` and `clip.height`.

routeEvent

Passes a captured event along the normal event hierarchy.

Method of **Layer**

Implemented in JavaScript 1.2

Syntax `routeEvent(event)`

Parameters

`event` The event to route.

Description If a sub-object (document or layer) is also capturing the event, the event is sent to that object. Otherwise, it is sent to its original target.

`routeEvent` works in tandem with `captureEvents`, `releaseEvents`, and `handleEvent`. For more information, see the *Client-Side JavaScript Guide*.

siblingAbove

The layer object above this one in z-order, among all layers that share the same parent layer or null if the layer has no sibling above.

Property of **Layer**

Read-only

Implemented in JavaScript 1.2

siblingBelow

The **layer** object below this one in z-order, among all layers that share the same parent layer or null if layer is at the bottom.

Property of **Layer**

Read-only

Implemented in JavaScript 1.2

src

A URL string specifying the source of the layer's content. Corresponds to the **SRC** attribute.

Property of **Layer**

Implemented in JavaScript 1.2

top

The vertical position of the layer's left edge, in pixels, relative to the origin of its parent layer.

Property of **Layer**

Implemented in JavaScript 1.2

The **Layer.y** property is a convenience synonym for the **top** property.

See also **Layer.left**

visibility

Whether or not the layer is visible.

Property of **Layer**

Implemented in JavaScript 1.2

Description A value of **show** means show the layer; **hide** means hide the layer; **inherit** means inherit the visibility of the parent layer.

window

The **window** or **Frame** object that contains the layer, regardless of whether the layer is nested within another layer.

Property of **Layer**

Read-only

Implemented in JavaScript 1.2

x

The horizontal position of the layer's left edge, in pixels, relative to the origin of its parent layer.

Property of **Layer**

Implemented in JavaScript 1.2

The **x** property is a convenience synonym for the **Layer.left** property.

See also **Layer.y**

y

The vertical position of the layer's left edge, in pixels, relative to the origin of its parent layer.

Property of **Layer**

Implemented in JavaScript 1.2

The **y** property is a convenience synonym for the **Layer.top** property.

See also **Layer.x**

zIndex

The relative z-order of this layer with respect to its siblings.

Method of **Layer**

Implemented in JavaScript 1.2

Description Sibling layers with lower numbered z-indexes are stacked underneath this layer. The value of **zIndex** must be 0 or a positive integer.

layers

The `layers` property is an array containing an entry for each layer within the document.

Property of `document`

Implemented in JavaScript 1.2

Description You can refer to the layers in your code by using the `layers` array. This array contains an entry for each `Layer` object (`LAYER` or `ILAYER` tag) in a document; these entries are in source order. For example, if a document contains three layers whose `NAME` attributes are `layer1`, `layer2`, and `layer3`, you can refer to the objects in the `layers` array either as:

```
document.layers["layer1"]
document.layers["layer2"]
document.layers["layer3"]
```

or as:

```
document.layers[0]
document.layers[1]
document.layers[2]
```

When accessed by integer index, array elements appear in z-order from back to front, where 0 is the bottommost layer and higher layers are indexed by consecutive integers. The index of a layer is not the same as its `zIndex` property, as the latter does not necessarily enumerate layers with consecutive integers. Adjacent layers can have the same `zIndex` property values.

These are valid ways of accessing layer objects:

```
document.layerName
document.layers[index]
document.layers["layerName"]
// example of using layers property to access nested layers:
document.layers["parentlayer"].layers["childlayer"]
```

Elements of a `layers` array are JavaScript objects that cannot be set by assignment, though their properties can be set. For example, the statement

```
document.layers[0]="music"
```

is invalid (and ignored) because it attempts to alter the `layers` array. However, the properties of the objects in the array are readable and some are writable. For example, the statement

```
document.layers["suspect1"].left = 100;
```

document.linkColor

is valid. This sets the layer's horizontal position to 100. The following example sets the background color to blue for the layer `bluehouse` which is nested in the layer `houses`.

```
document.layers["houses"].layers["bluehouse"].bgColor="blue";
```

To obtain the number of layers in a document, use the `length` property:
`document.layers.length`.

linkColor

A string specifying the color of the document hyperlinks.

Property of `document`

Implemented in JavaScript 1.0

Description The `linkColor` property is expressed as a hexadecimal RGB triplet or as a string literal (see the *Client-Side JavaScript Guide*). This property is the JavaScript reflection of the `LINK` attribute of the `BODY` tag. The default value of this property is set by the user with the preferences dialog box. You cannot set this property after the HTML source has been through layout.

If you express the color as a hexadecimal RGB triplet, you must use the format `rrggbb`. For example, the hexadecimal RGB values for salmon are `red=FA`, `green=80`, and `blue=72`, so the RGB triplet for salmon is `"FA8072"`.

Examples The following example sets the color of document links to aqua using a string literal:

```
document.linkColor="aqua"
```

The following example sets the color of document links to aqua using a hexadecimal triplet:

```
document.linkColor="00FFFF"
```

See also `document.alinkColor`, `document.bgColor`, `document.fgColor`,
`document.vlinkColor`