

JavaScript Browser Object Model

JavaScript provides WebAPIs and Interfaces(object types) that we can use while developing web application or website. These APIs and objects help us in controlling the lifecycle of the webpage and performing various actions like getting browser information, managing screen size, opening and closing new browser window, getting URL information or updating URL, getting cookies and local storage, etc.

The Interfaces (object types) which help us **interact with the browser window** are known as **Browser objects**. Browser object is not an official term but its a group of objects which belongs to different WebAPIs but are used for managing various browser related information and actions.

For example, when an HTML document is opened in a browser window, the browser interprets the document as a collection of hierarchical objects(HTML tags) and accordingly displays the data contained in these objects(HTML page rendering). The browser parses the document and creates a collection of objects, which defines the documents and its details. We have shown the various objects that can be used to access various components of the browser window in the picture below:



The browser objects are of various types, used for interacting with the browser and belongs to different APIs. The collection of these Browser objects is also known as **Browser object Model(BOM)**.

The default object of browser is Window which means you can call its functions directly.

Browser Objects:

The objects listed below are called browser objects.

- Window – part of DOM API
- Navigator

- Document – part of DOM API
- Screen – property of Window object
- History – property of Window object
- Location – property of Window and Document object

Window Object

It is used to interact with the browser window which displays the web page. It generally **represents a tab in browser**, but some actions like window width and height will affect the complete browser window.

We have covered JavaScript Window Object separately, in detail.

Navigator Object

It acts as a storehouse of all the data and **information about the Browser software used to access the webpage** and this object is used to fetch information related to the browser for example, whether the user is using Chrome browser or Safari browser, which version of browser is being used etc.

We have covered JavaScript Navigator Object separately, in detail.

Document Object

This object **represent the HTML document** that is loaded into the browser. A document object is an object that provides access to all HTML elements of a document(webpage). We can use this object to append a new HTML tag to the webpage, modify any existing HTML tag, etc.

We have covered JavaScript Document Object separately, in detail.

History Object

It stores Uniform Resource Locator(URLs) visited by a user in the browser. It is a built-in object which is used to get browser history. This object is a property of the JavaScript Window object.

We have covered JavaScript History Object separately, in detail.

Screen Object

It is a built-in object which is used to **fetch information related to the browser screen**, like the screen size, etc. It is also obtained from the Window object.

We have covered JavaScript Screen Object separately, in detail.

Location Object

Location is a built-in object which **represent the location of the object to which it is linked**, which can be Window or Document. Both the Document and Window interface have a linked location property.

JavaScript - Multimedia

The JavaScript **navigator** object includes a child object called **plugins**. This object is an array, with one entry for each plug-in installed on the browser. The navigator.plugins object is supported only by Netscape, Firefox, and Mozilla only.

Example

Here is an example that shows how to list down all the plug-on installed in your browser –

```
<html>
<head>
<title>List of Plug-Ins</title>
</head>

<body>
<tableborder="1">
<tr>
<th>Plug-in Name</th>
<th>Filename</th>
<th>Description</th>
</tr>

<scriptlanguage="JavaScript"type="text/javascript">
for(i=0;i<navigator.plugins.length;i++){
document.write("<tr><td>");
document.write(navigator.plugins[i].name);
document.write("</td><td>");
document.write(navigator.plugins[i].filename);
document.write("</td><td>");
```

```

document.write(navigator.plugins[i].description);
document.write("</td></tr>");
}
</script>
</table>
</body>
</html>

```

Output

Plug-in Name	Filename	Description
PDF Viewer	internal-pdf-viewer	Portable Document Format
Chrome PDF Viewer	internal-pdf-viewer	Portable Document Format
Chromium PDF Viewer	internal-pdf-viewer	Portable Document Format
Microsoft Edge PDF Viewer	internal-pdf-viewer	Portable Document Format
WebKit built-in PDF	internal-pdf-viewer	Portable Document Format

Controlling Multimedia

Let us take one real example which works in almost all the browsers –

```

<html>
<head>
<title>Using Embedded Object</title>

<scripttype="text/javascript">
<!--
functionplay(){
if(!document.demo.IsPlaying()){
document.demo.Play();
}
}

```

```
}  
function stop() {  
    if (document.demo.IsPlaying()) {  
        document.demo.StopPlay();  
    }  
}  
function rewind() {  
    if (document.demo.IsPlaying()) {  
        document.demo.StopPlay();  
    }  
    document.demo.Rewind();  
}  
//-->  
</script>  
</head>  
  
<body>  
<embed id="demo" name="demo"  
src="http://www.amrood.com/games/kumite.swf"  
width="318" height="300" play="false" loop="false"  
pluginspage="http://www.macromedia.com/go/getflashplayer"  
swliveconnect="true">  
  
<form name="form" id="form" action="#" method="get">  
<input type="button" value="Start" onclick="play();" />  
<input type="button" value="Stop" onclick="stop();" />  
<input type="button" value="Rewind" onclick="rewind();" />  
</form>  
</body>  
</html>
```