**What is the Difference Between Document and Window Objects?**

The “**Document Object**” represents a webpage structure that is inserted in the DOM tree hierarchy and when compiled the content gets displayed on the webpage. By getting access to the document object, the users can access the HTML webpage content along with the elements by which the webpage is built. On the other hand, the “**Window Object**” is at the head position of the DOM hierarchy showing a window browser or frame.

**What is the Document Object Model?**

The Document Object Model is the one on which the whole webpage is standing, it contains HTML elements that are placed in the form of a hierarchy tree. When it is compiled transforms the element into the form of a webpage. To use the Document object, the following syntax will be used:

document.property\_name;

Where the “**propertyName**” is the name of the property available in the DOM whose data is going to be retrieved. Now, let’s visit the properties and various methods of Document Object.

**Properties of Document Object**

In the below table, the properties along with the description for the Document are placed:

|  |  |
| --- | --- |
| **Properties** | **Description** |
| activeElement | It retrieves the currently used or active HTML element in the document. |
| body | Retrieves the content residing inside the “**<body>**” tag. |
| anchors | Retrieve all <a> anchor elements |
| baseURI | Retrieves a string value that identifies the base URI |
| cookie | Retrieves the currently executing document’s cookie information. |
| charSet | Retrieves a string that represents the character encoding scheme. |
| defaultView | Retrieves the current window object. |
| designMode | It sets the currently selected document to editable or non-editable by providing it a value of “**on**” and “**off**” respectively. |
| domain | Retrieves the domain name for the current document server. |
| embeds | Display the collection of all embedded components. |
| fullScreenElement | It retrieves the current element information which is in the full-screen mode. |
| lastModified | Retrieves the current document date and time that was last updated. |
| readyState | Retrieves the document loading status. |
| scripts | Retrieve script elements that are utilized in the current document. |
| strictErrorChecking | It sets or retrieves the information about the enforcement of strict error checking. |

## ****Methods of Document Object****

There are several methods that can be utilized to get information or perform specific tasks. These methods are presented below in tabular form:

[JavaScript](https://linuxhint.com/category/javascript/)

# What is the Difference Between Document and Window Objects?

3 months ago

by [Abdul Moeed](https://linuxhint.com/author/abdulmoeed/)

The “**Document Object**” represents a webpage structure that is inserted in the DOM tree hierarchy and when compiled the content gets displayed on the webpage. By getting access to the document object, the users can access the HTML webpage content along with the elements by which the webpage is built. On the other hand, the “**Window Object**” is at the head position of the DOM hierarchy showing a window browser or frame.

This blog explains the key points that differentiate document objects from window objects.

* [What is the Document Object Model?](https://linuxhint.com/difference-between-document-and-window-object/#1)
* [Properties of Document Object](https://linuxhint.com/difference-between-document-and-window-object/#2)
* [Methods of Document Object](https://linuxhint.com/difference-between-document-and-window-object/#3)
* [What is the Windows Object Model?](https://linuxhint.com/difference-between-document-and-window-object/#4)
* [Windows Object Model Methods](https://linuxhint.com/difference-between-document-and-window-object/#5)
* [Windows Object Model Properties](https://linuxhint.com/difference-between-document-and-window-object/#6)
* [How Do Document Objects Differ from Window Objects?](https://linuxhint.com/difference-between-document-and-window-object/#7)

## ****What is the Document Object Model?****

The Document Object Model is the one on which the whole webpage is standing, it contains HTML elements that are placed in the form of a hierarchy tree. When it is compiled transforms the element into the form of a webpage. To use the Document object, the following syntax will be used:

0 of 2 minutes, 28 secondsVolume 0%

Loading ad

document.property\_name;

Where the “**propertyName**” is the name of the property available in the DOM whose data is going to be retrieved. Now, let’s visit the properties and various methods of Document Object.

## ****Properties of Document Object****

In the below table, the properties along with the description for the Document are placed:

|  |  |
| --- | --- |
| **Properties** | **Description** |
| activeElement | It retrieves the currently used or active HTML element in the document. |
| body | Retrieves the content residing inside the “**<body>**” tag. |
| anchors | Retrieve all <a> anchor elements |
| baseURI | Retrieves a string value that identifies the base URI |
| cookie | Retrieves the currently executing document’s cookie information. |
| charSet | Retrieves a string that represents the character encoding scheme. |
| defaultView | Retrieves the current window object. |
| designMode | It sets the currently selected document to editable or non-editable by providing it a value of “**on**” and “**off**” respectively. |
| domain | Retrieves the domain name for the current document server. |
| embeds | Display the collection of all embedded components. |
| fullScreenElement | It retrieves the current element information which is in the full-screen mode. |
| lastModified | Retrieves the current document date and time that was last updated. |
| readyState | Retrieves the document loading status. |
| scripts | Retrieve script elements that are utilized in the current document. |
| strictErrorChecking | It sets or retrieves the information about the enforcement of strict error checking. |

## ****Methods of Document Object****

There are several methods that can be utilized to get information or perform specific tasks. These methods are presented below in tabular form:

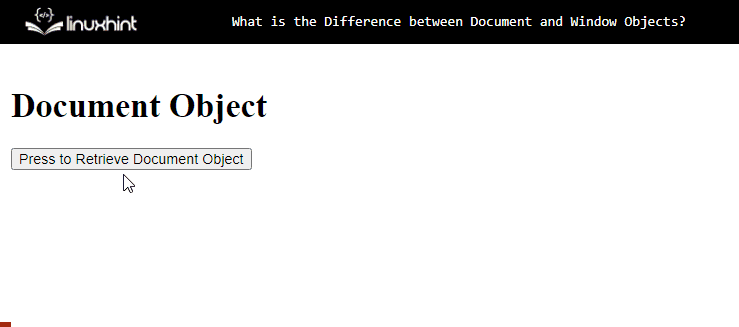
|  |  |
| --- | --- |
| **Methods** | **Description** |
| addEventListener() | It is utilized to call or invoke a function or method when the user performs some specified action like clicking a button. |
| adoptNode() | This method gets a node or object from another document and provides a node or parent object that represents the adopted node. |
| createAttribute() | It creates a specified attribute and returns the corresponding object. |
| execCommand() | It executes a command specified by the programmer on the editable section of the document. |
| fullscreenEnabled() | It checks if the current document can be shown in full-screen mode or not. |
| getElementByID() | Retrieves the reference of the HTML element by using its id. |
| getElementByClassName() | Retrieves the reference of the HTML element by using class attribute value. |
| getElementByTagName() | It retrieves an object of all HTML elements that contain a specified tag name. |
| normalize() | Removes the empty nodes and combines the adjacent text nodes along the parent or first node. |
| open() | Utilize to open the output stream to receive or create a collection for the output. |
| querySelector() | Returns the element that matches the specified CSS selector. |
| write() | This method inserts the content or JavaScript code into the document. |
| removeEventListener() | It removes the attached event handler from an element. |
| writeln() | It inserts document data at a new line after the compilation of each statement. |

h1> [Document](https://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+document) Object</h1>  
  
<button onclick="retriever()">Press to Retrieve [Document](https://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+document) Object</button>  
  
<p id="target"></p>  
  
<script>  
  
function retriever() {  
  
var docTitle = document.title;  
  
var docBody = document.body;  
  
document.getElementById("target").innerHTML =  
  
"Title of the document is: " + docTitle + "<br>" + "body : " + docBody;  
  
}  
  
</script>

In the above code:

* First, the button is created that utilizes the “**onclick**” event listener to call the “**retriever()**” function.
* Next, the “**retriever()**” function body is created that retrieves the “**title**” and “**body**” information by using the “**document.title**” and “**document.body**” properties.
* Then, this retrieved information is stored in “**docTitle**” and “**docBody**” and displayed on the web page using the “**innerHTML**” property.

After the compilation:



he output shows the required values using the Document object that has been retrieved.

## ****What is the Windows Object Model?****

The Windows Object Model or “WOM” is a framework used by Microsoft Windows operating systems to represent and interact with various system components, like Windows, controls, menus, and other GUI elements. It is based on a binary interface standard used in Windows for inter-process communication and component interaction.

## ****Windows Object Model Properties****

The below table contains the Windows Object Model properties and their description:

|  |  |
| --- | --- |
| **Properties** | **Description** |
| Closed | It represents whether the window is closed or not. |
| Document | Returns a document object reference. |
| controllers | Retrieve the XUL controller objects. |
| defaultStatus | Elaborates the default message to be shown in the status bar. |
| console | Retrieves an object containing the reference to the console that offers access to browsers. |
| DOMMartrix | Retrieves the DOMMatrix object representing 4×4 matrices. |
| History | Provides information on the visited URLs. |
| DOMPoint | Refers to a DOMPoint that represents a 2D or 3D point coordination |
| DOMRect | Returns a DOMRect object reference. |
| Length | Represents the number of frames |

## ****Windows Object Model or WOW Methods****

The methods that come with the Windows Object Model are inserted below along with the description in tabular form:

|  |  |
| --- | --- |
| **Methods** | **Explanation** |
| alert() | It displays an alert box to show the message or warning along with the “**OK**” and “**Cancel**” buttons. |
| blur() | Diverges the focus from the current window |
| clearInterval() | Clears the interval set by the function “setInterval(). |
| getSelection() | Retrieves an object that identifies the range of selected text. |
| matchMedia() | Returns a MediaQueryList object that shows the outcome of a specific CSS query string. |
| moveTo() | It is utilized to perform the movement for the window from the left to the top position. |
| resizeTo() | Resizes a window to specific width or height |
| scrollBy() | Scroll the document by providing pixels. |
| scrollTo() | Scrolls to particular coordinates residing inside the document. |
| setInterval() | Repeats a function after every provided or specified interval of time |
| stop() | Stops the window from loading the resources in the browsing context. |

## ****How Does Document Object Differ from Window Object?****

To get information about the difference between the Document object and the window, object visit the below table:

|  |  |
| --- | --- |
| **Methods** | **Explanation** |
| alert() | It displays an alert box to show the message or warning along with the “**OK**” and “**Cancel**” buttons. |
| blur() | Diverges the focus from the current window |
| clearInterval() | Clears the interval set by the function “setInterval(). |
| getSelection() | Retrieves an object that identifies the range of selected text. |
| matchMedia() | Returns a MediaQueryList object that shows the outcome of a specific CSS query string. |
| moveTo() | It is utilized to perform the movement for the window from the left to the top position. |
| resizeTo() | Resizes a window to specific width or height |
| scrollBy() | Scroll the document by providing pixels. |
| scrollTo() | Scrolls to particular coordinates residing inside the document. |
| setInterval() | Repeats a function after every provided or specified interval of time |
| stop() | Stops the window from loading the resources in the browsing context. |

## ****How Does Document Object Differ from Window Object?****

To get information about the difference between the Document object and the window, object visit the below table:

|  |  |
| --- | --- |
| **Document Object** | **Window Object** |
| It represents the content for the web page that gets loaded on the browser and provides the methods to query and modify the structure of the document. | It identifies the browser tab that contains the webpage and handles events related to the window, such as resizing, scrolling, or closing. |
| Its scope is limited to the web pages it represents. | The scope can extend beyond the current document to a tab of the browser. |
| The various properties of the DOM can be returned or get by utilizing the document object. | Properties for window objects are not accessible using the document object. |
| Document Object is part of the Browser Object Model and Document Object Model. | It is only part of the Browser Object Model |
| The programmer can access the document from the window with the help of “**window.document**”. | It can also access by utilizing the “**window.window**” |
| Loaded in the window | Loaded in browser |
| The syntax for Document Object is:  document.propertyName | The syntax for Window Object is:  window.propertyName |

That is all about the difference between Document and Windows objects

## ****Conclusion****

The **“Document Object**” is a part of the Browser Object Model and the Document Object Model, it handles the rendering of content on the webpage and is located inside the window. On the other hand, the “**Window Object**” handles events related to the window like scrolling, resizing, and so on, and is located in the system browser. There are several properties and methods that are provided by both Document Object and Window Object which helps a lot in various purposes.