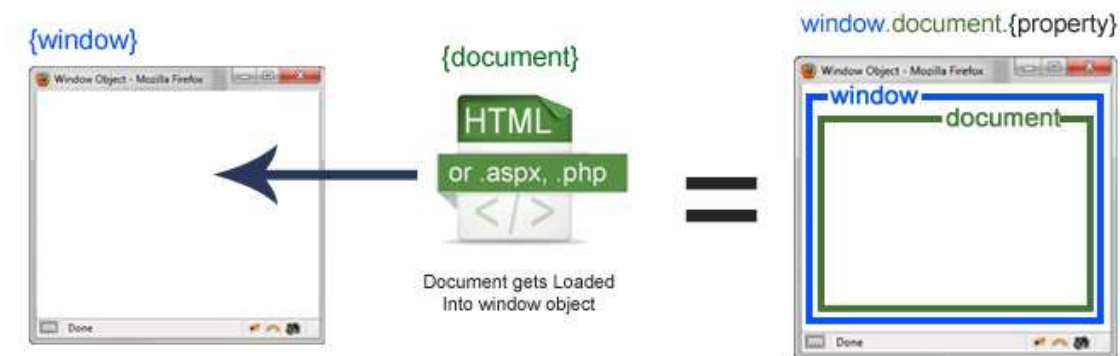


Blog on the difference between Document and Window Objects

Window object : It is the top most object and outermost element of the object hierarchy. It is the root level element in any web page. All the global variables are defined on the window object. The JavaScript window object sits at the top of the JavaScript Object hierarchy and represents the browser window. The window object is supported by all browsers.

All global JavaScript objects, functions, and variables automatically become members of the window object. The window is the first thing that gets loaded into the browser. This window object has the majority of the properties like length, innerWidth, innerHeight, name, if it has been closed, its parents, and more.

The window object represents the current browsing context. It holds things like window.location, window.history, window.screen, window.status, or the window.document. Each browser tab has its own top-level window object. Each of these windows gets its own separate global object. window.window always refers to window, but window.parent and window.top might refer to enclosing windows, giving access to other execution contexts. Moreover, the window property of a window object points to the window object itself.



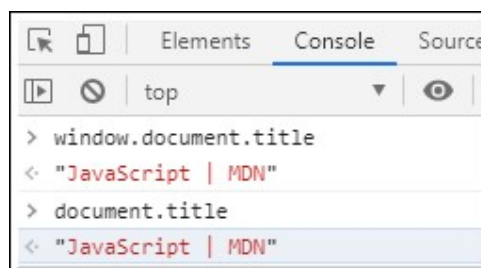
- **For example:** `alert()`, `confirm()` are methods defined on the window object.
- Writing `alert()` is similar to `window.alert()`
- Also, properties like `document`, `location` are properties of the window object.

Document object : It is the direct child of the window object. Each HTML document that gets loaded into a window becomes a document object. The document contains the contents of the page. Using document object, JavaScript can modify, add and delete the HTML elements, attributes CSS styles in the page. You can access it via `window.document` or `document`. Document object has many useful methods defined on it.

The *window object* represents a window/tab containing a DOM document where as *document object* is property of *window object* that points to the DOM document loaded in that window.

The Document interface represents any web page loaded in the browser and serves as an entry point into the web page's content, which is the DOM tree. When an HTML document is loaded into a web browser , it becomes a document object. It is the root node of the HTML document. The document actually gets loaded inside the window object and has properties available to it like title, URL, cookie, etc. HTML documents, served with the "text/html" content type, also implement the HTMLDocument interface, whereas XML and SVG documents implement the XMLDocument interface.

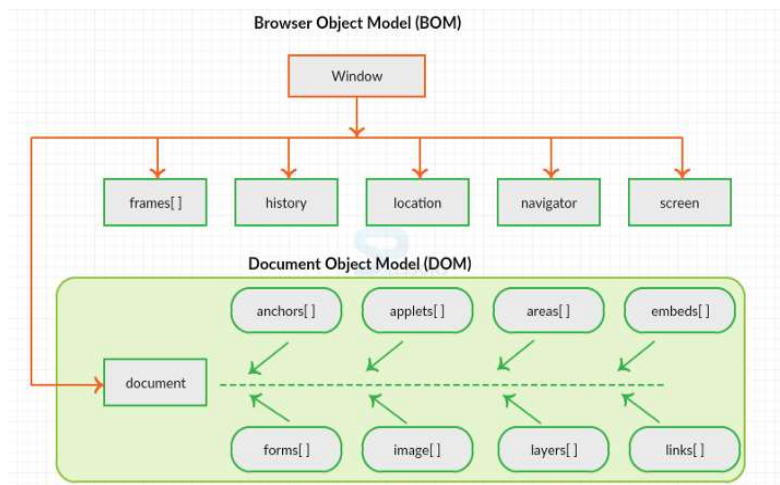
You can access a document object either using *window.document* property or using document object directly as window is global object. In the below example, title is the property of document object.



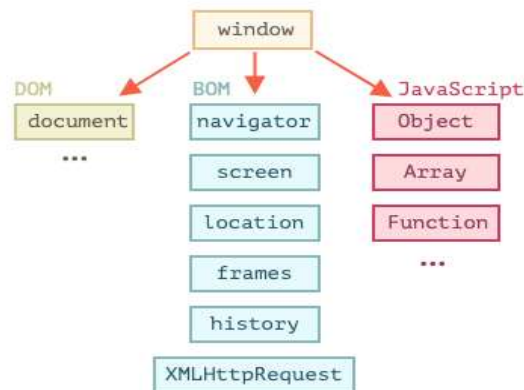
The other major difference is that both window object and document object have properties and methods. Few method names are same in both objects but with different behavior. In the below example *window.open()* opens a new tab or window and *document.open()* creates a blank document within the window.

- **For example:** `document.getElementById()`, `document.getElementsByTagName()`, `document.createElement()`, `document.querySelector()` and many more.

Flow Chart:



The Browser Object Model contains objects that represent the current browser window or tab. The topmost object in the BOM is the window object representing the window or tab or an iframe sometimes. Window object has properties like browser history, location and the device's screen etc. In case of multi tab browser, a window object represents a single tab, but some of its properties like `innerHeight`, `innerWidth` and methods like `resizeTo()` will affect the whole browser window.



When a web page is loaded, the browser creates a Document Object Model of the page. The document object represents the whole html document as a tree of Objects(HTML, HEAD, BODY, and other HTML tags). It is the root element that represents the html document.

```
alert('Hello World!')  
// Is the same as  
window.alert('Hello World')
```

Vs

```
✓ <html>  
✓ | <head>  
  | | <title>My Document</title>  
  | </head>  
✓ | <body>  
  | | <h1>Header</h1>  
  | | <p>Paragraph</p>  
  | </body>  
  </html>
```