**DIFFERENCE BETWEEN DOCUMENT AND WINDOWS OBJECTS**

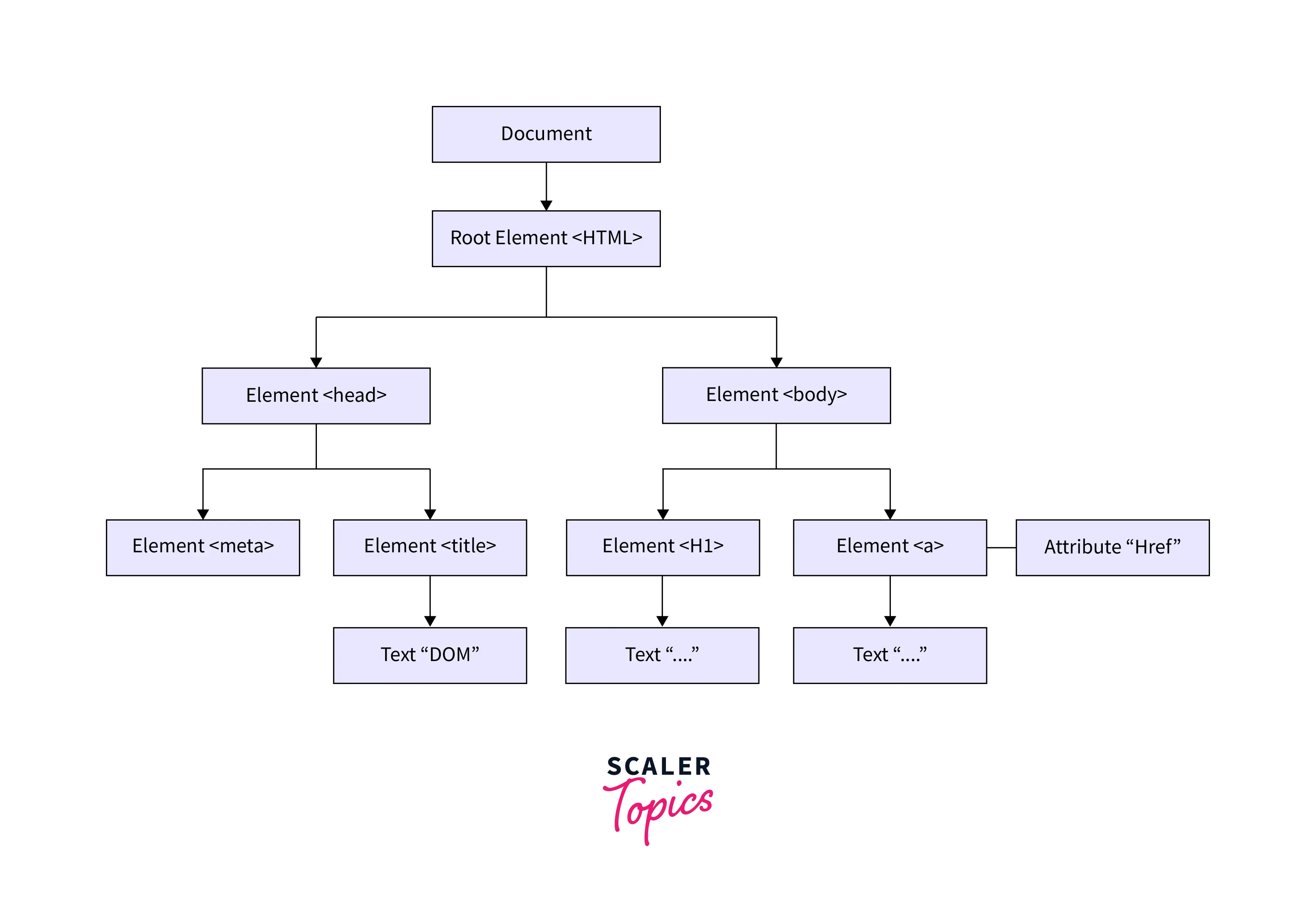
**DOCUMENT:**

Document object represents the HTML document that is displayed in that window. The Document object has various properties that refer to other objects which allow access to and modification of document content. The way a document content is accessed and modified is called Document Object Model, or DOM. The Objects are organized in a hierarchy and it applies to the organization of objects in a Web document.



DOM stands for the **Document Object Model** in javascript. It works as application programming interface for web documents (HTML and XML documents). Moreover, it is a "tree structure" representation built by the browser that allows the HTML structure to be easily accessed by programming languages such as javascript to change or manipulate the document structure, style, and content. It represents the document as nodes and objects in a tree structure.

Document object represent the whole HTML document. When web page is loaded in the browser, it is converted into document object and the window object is top element of the object hierarchy. So **document** and **window.document** both give the same result.



Key components of the document node tree structure are as follows:-

* **Document node:** The whole web document is a document node.
* **Element node:** Elements are the basic building blocks of the document, and they represent an element that exists in the DOM, such as , , are the element nodes.
* **Text node:** It is contained by the element node, representing the text content of that element node such as title element contains "DOM" text content.
* **Attribute node:** It provides more information about the element node, such as a tag containing a **href** attribute with the external link.

**Properties of Document Object**

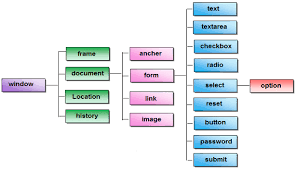
* **Window Object:** The window object is the top-level object in the hierarchy. It includes property methods like document, location, etc.
* **Document Object:** When the web page is loaded into the browser window, it becomes the document object and tree-like structure that includes all the web page content. It includes methods like **getElementById**, **getElementByName** to access the web content via the DOM.
* **Link Object:** Link objects are objects or attributes set using the link tag.
* **Anchor Object:** anchor objects are the objects that are represented by the href tag for reference of the links.
* **Form object:** A Form object is enclosed between <form></form> tags that represent a form on a web page. Examples of form tags include a checkbox, radio, etc.
* **Form Control Elements:** The form can have many elements to control the text input field, button, radio button, etc.

**Methods of Document Object:**

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Syntax** |
| getElementById() | This method returns the first element with the specified ID. | document.getElementById(id) |
| getElementsByClassName() | This method returns an object containing all the elements with the specified class names. | document.getElementsByClassName(name) |
| ByTagName() | This method returns all elements having the given tagname. | document.getElementsByTagName(tagname) |
| querySelector() | This method returns the first element that matches a CSS selector. | document.querySelector("Name") |
| querySelectorAll() | This method returns all elements in an array form that match the given CSS selector. | document.querySelectorAll("Name") |
| createElement() | This method is used to create a new element. | document.createElement("tagName") |

**WINDOW:**

* JavaScript Window Object is an already available global object which represents the currently opened window tab in the browser.
* Window object is nothing but the browser window which is opened in the browser different from the normal object since window is an object of browser.
* Every tab opened in the browser will have its window object associated with it and it will be available in every part of the application as it is global.
* It has multiple values and methods are available with it and we can call these methods directly to perform multiple operations over the browser window.



**Methods of JavaScript Window Object**

**alert()**

This method will display an alert box over the browser window.

**confirm()**

This method will display a dialogue box along with the ok and cancel button. We can pass the message to it and it will display it on the dialogue box.

**prompt()**

This method displays a dialogue box with the user input field, we can also pass the message to it and it will display it on the dialogue box.

**open()**

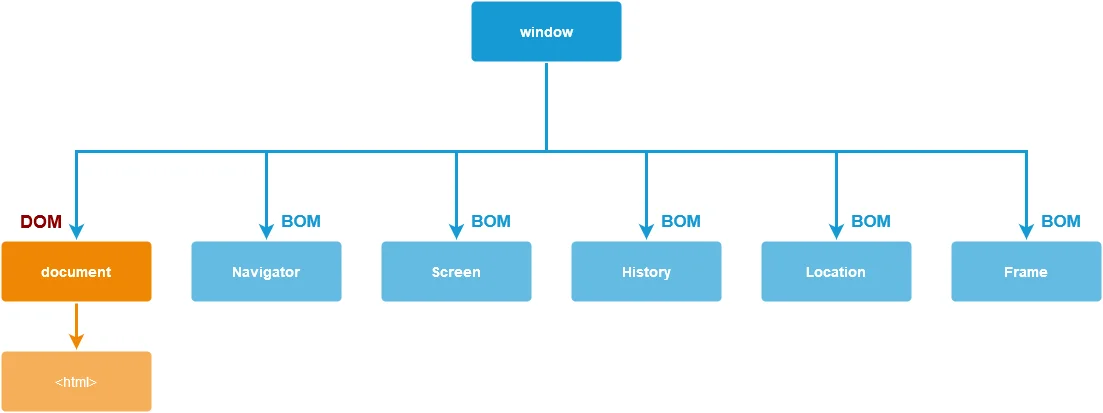
This method will open the new window with the specified URL address.

**close()**

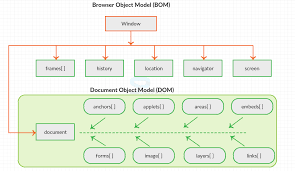
This method will close the window which is opened by the Window object.

**setTimeout()**

This method will perform the action specified after a time interval which is specified while calling it. We can perform any kind of activity in this way.



**Window vs Object:**



|  |  |
| --- | --- |
| **Document** | **Window** |
| It represents any HTML document or web page that is loaded in the browser. | It represents a browser window or frame that displays the contents of the webpage. |
| It is loaded inside the window. | It is the very first object that is loaded in the browser. |
| Object of window property. | Object of the browser. |
| All the tags, elements with attributes in HTML are part of the document. | Global objects, functions, and variables of JavaScript are members of the window object. |
| We can access the document from a window using the window. document | We can access the window from the window only. i.e. **window.window** |
| Document is part of BOM (browser object model) and dom (Document object model) | Window is part of BOM, not DOM. |
| Properties of document objects such as title, body, cookies, etc can also be accessed by a window like this window. document.title | Properties of the window object cannot be accessed by the document object. |
| **syntax:** document.propertyname; | **syntax:** window.propertyname; |