In the Document Object Model (DOM), the **window** object is a global object that represents the browser window or tab in which a web page is displayed. It is the top-level object in the browser's JavaScript environment and provides access to various properties and methods that relate to the browser window and its environment.

The **window** object has a hierarchical relationship with the **document** object. Here's how they relate to each other:

1. **Global Scope**: The **window** object is the global object for JavaScript in a browser environment. This means that any variable or function declared without the **var**, **let**, or **const** keyword in a JavaScript file is automatically added as a property of the **window** object. For example, if you declare a variable like this:

javascriptCopy code

myVar = 42;

It is equivalent to:

javascriptCopy code

window.myVar = 42;

1. **Access to the Document**: The **window** object provides access to the **document** object. The **document** object represents the web page's content, including the HTML structure and the elements within it. You can access and manipulate the document using properties and methods of the **document** object, which is a property of the **window** object.

For example, to access the title of the current document, you can use:

javascriptCopy code

var title = window.document.title;

The **document** object is a property of the **window** object, so you access it through **window.document**.

1. **Event Handling**: The **window** object also handles global events such as resizing the window, navigating away from the page, and loading of resources. You can attach event handlers to the **window** object to respond to these events. For example, you can use **window.addEventListener** to listen for the **load** event to know when the page has finished loading.

In summary, the **window** object is the top-level object in the DOM that represents the browser window and provides access to various properties and methods, including the **document** object, which represents the web page's content. The **window** object is where global variables and functions reside, and it is also responsible for handling various browser-related events.

**WEB API’s are functionality provided to the engine ,accessible on window objects**

**JS runtime consists of engine,api,callback queue and event loop**

**So basically the event loop takes callback functions**

**from the callback queue**

**and puts them in the call stack**

**so that they can be executed.**

**Different javascript run times exist like that of node js,in node js run time instead of web apis there will c++ bindings and pool**

Refer notebook and Jonas notes and some videos for more