**The document Object**

The **document** object represents the HTML document loaded in the current browser window. It serves as the entry point to access and manipulate the content of a web page. Here are some key aspects of the **document** object:

1. **Hierarchy**: The **document** object is a property of the **window** object, which makes it a part of the global object hierarchy in JavaScript. You can access it directly as **document** or via **window.document**.
2. **DOM (Document Object Model) Manipulation**: The **document** object is primarily used to interact with the DOM, which is a tree-like representation of the web page's structure. You can access and modify elements, attributes, and content using the **document** object. For example, you can change the text of an element, modify styles, add or remove elements, and much more.
3. **Example**: To access an element with an ID of "myElement," you can use **document.getElementById("myElement")**. This method allows you to manipulate the corresponding HTML element.

const element = document.getElementById("myElement");

element.textContent = "New Text Content";

### The window Object

The **window** object represents the browser window or tab that contains the web page. It serves as the global object in the browser's JavaScript environment. Here are some key aspects of the **window** object:

1. **Global Scope**: The **window** object is at the top level of the JavaScript object hierarchy. Variables and functions defined in the global scope are properties of the **window** object. For instance, if you declare a variable like **var x = 10;** in the global scope, you can access it as **window.x**.
2. **Browser Properties and Methods**: The **window** object provides access to various browser-related properties and methods, such as **window.location** (to get or set the current URL), **window.alert()** (to display alert dialogs), and **window.open()** (to open new browser windows or tabs).
3. **Global Events**: The **window** object is used for registering global event listeners, such as **window.addEventListener()** for handling events like **DOMContentLoaded** and **resize**.
4. **Example**: To open a new browser window, you can use **window.open()**:

const newWindow = window.open("https://www.example.com", "Example Window", "width=500,height=400");

### Key Differences

Now that we've explored the **document** and **window** objects, let's summarize their key differences:

1. **Scope**: The **document** object is primarily concerned with the content and structure of the web page (DOM manipulation), while the **window** object handles global browser-related properties and methods.
2. **Hierarchy**: The **document** object is a property of the **window** object and is nested within it.
3. **Purpose**: The **document** object is used for working with the web page's content, elements, and structure, while the **window** object is used for browser-related tasks and global JavaScript functionality.
4. **Access**: You can access the **document** object directly or via **window.document**, but the **window** object is always accessed as **window**.