**Document and Window Objects in JavaScript**

When working with JavaScript in web development, we typically encounter two main elements: Documents and windows. While they both play an important role in transforming and interacting with the web, they serve different purposes. Let’s explore the difference between a Document and window object.

**“Document” Object: -**

The document object represents the entire HTML document within a webpage. It serves as an interface to interact with the structure and content of the document, allowing you to manipulate, query, and update elements.

Key Features: -

DOM Manipulation:

* Provides methods to manipulate the Document Object Model (DOM), allowing you to dynamically change the content and structure of the webpage.
* Examples include selecting elements, modifying their content, or adding/removing elements.
* Example:
  + // Example: Changing the text content of an element
  + document. getElementById('myElement').textContent = 'New Content';

Element Selection:

* Offers various methods to select HTML elements, such as getElementById, getElementsByClassName, getElementsByTagName, and newer methods like querySelector and querySelectorAll.
* Example:
* // Example: Selecting elements by class name

const elements = document.getElementsByClassName('myClass');

Events Handling:

* Allows the attachment of event listeners to elements, enabling you to respond to user interactions (clicks, keypresses, etc.).

// Example: Adding a click event listener

document. getElementById('myButton').addEventListener('click', function() { alert ('Button Clicked!'); });

**“Window” Object: -**

The Window object represents the browser window or global environment in which your JavaScript code executes. This includes not only the present document, but also various features and methods related to the browser environment.

Key features: -

Global Object:

* Serves as the global object in client-side JavaScript, meaning variables and functions declared without the var, let, or const keyword become properties of the window object.

// Example: Creating a global variable

window. myGlobalVar = 'Hello World!';

Browser Interaction:

* Provides properties and methods related to the browser environment, such as window.location for managing the current URL, window.alert for displaying alerts, and window.confirm for user confirmation dialogs.

// Example: Displaying an alert

window.alert('Welcome to the Website!');

Timers and Intervals:

* Offers functions like setTimeout and setInterval for executing code after a delay or at intervals.

// Example: Setting a timeout

window.setTimeout(function() {

console.log('Delayed Code');

}, 2000);

**Summary:**

* **Document:** Focuses on the content and structure of the HTML document. It allows manipulation of the DOM, element selection, and event handling.
* **Window:** Represents the browser environment and serves as the global object. It provides properties and methods for browser interaction, global variables, and timers/intervals.