Write a blog on the Difference between document & and window objects

JavaScript is a versatile and powerful programming language used to enhance the interactivity and functionality of web pages. When working with JavaScript in a web environment, two essential objects play a crucial role: the Document object and the Window object. These objects are distinct but closely related, serving distinct purposes in web development. In this blog, we'll explore the key differences between these two objects to help you better understand their roles and how to use them effectively.

**Document Object**

The Document object represents the web page itself, providing access to the document's content, structure, and elements. It serves as an interface to manipulate and interact with the elements within an HTML document. Here are some important characteristics of the Document object:

1. Accessing Document Elements: The document object, you can access and manipulate HTML elements on a web page. This includes elements like headings, paragraphs, forms, images, and more. You can use methods like `getElementById`, `getElementsByClassName`, and `querySelector` to select and interact with these elements.

2. DOM Manipulation: The Document object is the entry point to the Document Object Model (DOM), which represents the hierarchical structure of an HTML document. You can create, modify, or remove elements and their attributes within the DOM using JavaScript.

3. Events: Document-level events, such as `DOMContentLoaded` and `click`, are attached to the Document object. These events allow you to respond to user interactions and document loading.

**Window Object**

In contrast, the Window object represents the browser window or tab that displays a web page. It provides access to browser-related functionalities and global properties. Here are key aspects of the Window object:

1. Global Scope: Variables and functions declared in the global scope are attached to the Window object. This means that if you declare a variable without the `let`, `const`, or `var` keywords, it becomes a global variable attached to the Window object.

2. Browsing Context: The Window object manages the browsing context, including navigation, history, and location. You can use properties like `window. location` to retrieve or modify the current URL.

3. Timers: Functions like `setTimeout` and `setInterval` are part of the Window object, enabling you to schedule code execution at specified intervals.