Write a blog on the difference between document & window objects.

* **The Document Object :**

The Document object represents the web page itself. It serves as an interface to the HTML content displayed in a web browser. Here are some essential characteristics of the document object:

1. **Accessing HTML Elements**: The `document` object allows you to access and manipulate elements on the web page. You can query elements using methods like `getElementById`, `getElementsByClassName`, or `querySelector`.
2. **Modifying Content**: You can change the content and structure of the web page using the `document` object. This includes altering text, attributes, and adding or removing elements.
3. **Event Handling**: The `document` object enables you to attach event listeners to specific elements, such as buttons or forms. This is essential for creating interactive web applications.
4. **DOM** (Document Object Model): The `document` object is at the heart of the DOM, a structured representation of the web page. It allows you to traverse and manipulate the hierarchical structure of HTML elements.
5. **No Global Scope:** Unlike the `window` object, the `document` object is not globally accessible. It is specific to the current web page and cannot be accessed from frames or iframes.

* **The Window Object :**

The Window object represents the web browser window or tab that contains the web page. It provides access to various properties and methods related to the browser environment. Here are the key features of the window object:

1. **Global Scope**: The `window` object is globally accessible in JavaScript. Variables and functions declared without the `var`, `let`, or `const` keyword become properties of the `window` object. For example, `var x = 10;` creates `window.x`.
2. **Controlling the Browse**r : You can use the `window` object to control browser behavior, such as opening new windows or tabs (`window.open`), navigating to different URLs (`window.location`), and setting timeouts (`window.setTimeout`) or intervals (`window.setInterval`).
3. **Storing Data**: The `window` object can be used to store temporary data across different parts of your application. This is often done with the `localStorage` and `sessionStorage` properties.
4. **Handling Events** : Global events like resizing the window or closing the tab are managed using the `window` object. Event listeners can be attached to these events for specific functionalities.
5. **Frames and Iframes** :The `window` object also manages frames and iframes within a web page. Each frame or iframe has its own `window` object, allowing for communication between them.