1.Write a blog on a difference between **Document object** and **window object**:

**Ans:**

**Document Object:**

The Document object represents the entire HTML document within a web page. It serves as an entry point to the content and provides methods and properties to access and manipulate various elements present in the document. Here are some key characteristics of the Document object:

1. Hierarchy: The Document object sits at the top of the DOM hierarchy. It contains properties and methods to access elements like HTML tags, IDs, classes, etc.

2. Access to Content: It provides access to various elements like `<body>`, `<head>`, `<title>`, and more through methods like `getElementById()`, `getElementsByClassName()`, `getElementsByTagName()`, etc.

3. Methods for Manipulation: It offers methods to modify the document's content dynamically. For instance, `createElement()`, `appendChild()`, `removeChild()`, `innerHTML`, etc., enable the addition, removal, or modification of elements and content.

4. Event Handling: Document object handles events occurring in the document, such as `DOMContentLoaded`, `click`, `keydown`, etc. Event listeners can be attached to the document to respond to these events.

5. Properties: It holds properties related to the document, like `title`, `URL`, `body`, `doctype`, etc., providing information and access to document-related details.

**Window Object:**

The Window object represents the browser window containing the DOM and acts as the global object for JavaScript in a web browser. Here are its notable characteristics:

1. Global Scope: All global JavaScript variables, functions, and objects become properties of the Window object. It provides access to numerous methods and properties, like `localStorage`, `sessionStorage`, `setTimeout()`, `clearTimeout()`, `setInterval()`, etc.

2. Controls Browser Window: The Window object manages various aspects of the browser window, including methods to manipulate its dimensions (`resizeTo()`, `moveTo()`), control navigation (`open()`, `close()`), and handle history (`history` object).

3. Event Handling: Similar to the Document object, the Window object handles events specific to the browser window, such as `load`, `resize`, `scroll`, etc. These events can be captured and handled through event listeners.

4. Timers and Intervals: It facilitates the scheduling of code execution at specified intervals using `setTimeout()` and `setInterval()`.

5. Storage Access: Window object provides access to storage mechanisms like `localStorage` and `sessionStorage` for storing data persistently or for the duration of a session.