1. 1. `document` Object:

The `document` object represents the HTML document itself. It provides an interface to interact with the content of a webpage, allowing developers to manipulate and modify the structure, style, and content of the document.

- **Properties and Methods:**
  - `document.getElementById()`: Retrieves an element based on its unique ID.
  - `document.getElementsByTagName()`: Returns a collection of elements with the specified tag name.
  - `document.createElement()`: Creates a new HTML element.
  - `document.querySelector()`: Returns the first element that matches a specified CSS selector.
  - `document.getElementById().innerHTML`: Gets or sets the HTML content within an element.
- **Usage:**
  - Manipulating the content of a webpage.
  - Accessing and modifying HTML elements and attributes.

2. `window` Object:

The `window` object represents the browser window that contains the document. It is the top-level object in the browser's JavaScript object hierarchy and serves as a global context for all JavaScript code running in the browser.

- **Properties and Methods:**
  - `window.innerWidth` and `window.innerHeight`: Provide the dimensions of the browser window.
  - `window.location`: Gives information about the current URL.
  - `window.alert()`: Displays an alert box with a specified message.
  - `window.setTimeout()`: Invokes a function or evaluates a code snippet after a specified delay.
- **Usage:**
  - Managing the overall browser environment.
  - Controlling the browser window, including navigation and resizing.
  - Setting timeouts and intervals for executing functions.

Key Differences:

1. **Scope:**
   - The `document` object is focused on the content and structure of the HTML document.

- The `window` object is broader and encompasses the entire browser window and its properties.

2. **Hierarchy:**
   - The `document` object is nested within the `window` object, as the document is a part of the browser window.
3. **Use Cases:**
   - The `document` object is mainly used for accessing and manipulating the content of the HTML document.
   - The `window` object is used for controlling the overall browser environment, including navigation, resizing, and handling global events.
4. **Global Context:**
   - When JavaScript code is executed in a browser, the `window` object serves as the global object. This means that variables and functions declared globally become properties and methods of the `window` object.