DIFFERENCE BETWEEN DOCUMENT AND WINDOW OBJECTS

- The document and window objects are two of the most important objects in the JavaScript DOM. The document object represents the HTML document loaded in a browser window, while the window object represents the browser window itself.
- The document object has many properties and methods that allow you to manipulate the HTML document, such as getElementById(), querySelector(), and createElement(). The window object has many properties and methods that allow you to manipulate the browser window, such as innerWidth, innerHeight, and location.

Here is a table that summarizes the key differences between the document and window objects:

Property	**Document Object**	**Window Object**
Represents	HTML document	Browser window
Properties	`getElementById()`, `querySelector()`, `createElement()`	`innerWidth`, `innerHeight`, `location`
Methods	`appendChild()`, `removeChild()`, `createTextNode()`	`resizeTo()`, `scrollBy()`, `alert()`

In general, you can think of the document object as the root of the HTML document, and the window object as the root of the browser window.

The terms "**Property**" in the context of the Document Object and "**Property**" in the context of the Window Object refer to different things in web development:

- **Property in Document Object**
  - In the Document Object Model (DOM), the Document Object represents the HTML document loaded in the browser.
  - The properties of the Document Object are related to the document itself, such as \`title\`, \`URL\`, and methods like \`getElementById()\`.

For example:

\`document.title\` represents the title of the HTML document, and \`document.URL\` represents the URL of the page.

- **Property in Window Object :**
  - The Window Object represents the browser window or tab that contains the document. It's the top-level object in the client-side JavaScript environment.

- The properties of the Window Object are more related to the browser environment and the window itself. Common properties include `window.innerWidth`, `window.innerHeight`, and `window.location`.

For example:

window.innerWidth represents the width of the browser window, and **window.location** allows you to access and modify the URL of the page.

The key difference is that properties of the Document Object pertain to the HTML document loaded in the browser, while properties of the Window Object are more related to the browser window and its environment. Both are essential for web development, allowing you to interact with the document and control the behavior of the browser.

- Represents HTML document (DOM):
  - This refers to the Document Object Model, which represents the structured content of the HTML document.

Example:

If you have an HTML document with a `<div>` element like this:

HTML

```
<div id="myDiv">This is a div.</div>
```

You can access and manipulate this element in the DOM using JavaScript:

JAVASCRIPT

```
const myDiv = document.getElementById("myDiv");

myDiv.innerHTML = "Modified content";

// Changes the content of the div
```

- Represents Browser window (Window Object) :
  - This represents the entire browser window or tab and provides access to browser-specific properties and methods.

Example:

You can use the Window Object to open a new browser window or tab:

javascript

```
const newWindow = window.open("https://www.example.com", "_blank");
```

- Here, `window.open()` is a method that opens a new browser window/tab, and `window` represents the current browser window.

  - when you interact with the DOM, you're working with the structured content of the HTML document (e.g., accessing and modifying elements). On the other hand, when you use the Window Object, you're interacting with the browser window itself, which allows you to control the browser environment (e.g., opening new windows or tabs).

- The terms "Properties getElementById(), querySelector(), createElement()" and "Properties innerWidth, innerHeight, location" refer to different aspects of web development:
- **Properties getElementById(), querySelector(), createElement() :**
  - These are related to the Document Object Model (DOM) and are used for manipulating the content and structure of a web page.
    - `getElementById()`: This is a method to select an HTML element by its unique "id" attribute.
    - `querySelector()`: This is a method to select HTML elements using CSS-style selectors.
    - `createElement()`: This method is used to create a new HTML element.

Example:

HTML

```
<div id="myDiv">This is a div.</div>
```

JAVASCRIPT

```
// Using getElementById to select the element

  const elementById = document.getElementById("myDiv");

// Using querySelector to select the element

const querySelector = document.querySelector("#myDiv");

// Using createElement to create a new element

const newElement = document.createElement("p");

   newElement.textContent = "This is a new paragraph.";
```

- **Properties innerWidth, innerHeight, location**
  - These are properties of the Window Object and are used to interact with the browser window and its properties.
    - `innerWidth`: Represents the width of the browser's content area (viewport).
    - `innerHeight`: Represents the height of the browser's content area (viewport).
    - `location`: Represents the current URL of the browser window.

Example:

JAVASCRIPT

```
// Accessing the innerWidth and innerHeight properties

  const viewportWidth = window.innerWidth;

   const viewportHeight = window.innerHeight;

   // Accessing the location property to get the current URL

   const currentURL = window.location.href;
```

- The Element usage `getElementById()`, `querySelector()`, and `createElement()` are used to interact with the DOM to select and manipulate HTML elements, while `innerWidth`, `innerHeight`, and `location` are properties of the Window Object used to access information about the browser window and its environment.

- The terms "Methods appendChild(), removeChild(), createTextNode()" and "Methods resizeTo(), scrollBy(), alert()" refer to different methods in web development:
- **Methods appendChild(), removeChild(), createTextNode():**
  - These methods are used to manipulate the Document Object Model (DOM), which represents the structure and content of a web page.
    - `appendChild()`: Used to add a new child node (element or text node) to an existing element.
    - `removeChild()`: Used to remove a child node from an element.
    - `createTextNode()`: Used to create a new text node with the specified text content.

Example:

JAVASCRIPT

```
// Create a new paragraph element

Const newParagraph = document.createElement("p");

// Create a text node

Const textNode = document.createTextNode("This is some text.");

// Append the text node to the paragraph element

newParagraph.appendChild(textNode);

// Remove the paragraph element from the document

Const parentElement = document.getElementById("parent");

parentElement.removeChild(newParagraph);
```

- **Methods resizeTo(), scrollBy(), alert() :**
  - These methods are related to the Window Object and are used to interact with the browser window and its behavior.
    - `resizeTo()`: Used to resize the browser window to the specified dimensions.
    - `scrollBy()`: Used to scroll the content within the browser window by a specified number of pixels.
    - `alert()`: Used to display a simple dialog box with a message to the user.

Example:

JAVASCRIPT

```
// Resize the browser window to a specific width and height

Window.resizeTo(800, 600);

// Scroll the content by 100 pixels horizontally and 200 pixels vertically

Window.scrollBy(100, 200);

// Display an alert to the user

Alert(“This is an alert message!”);
```

- The Methods of usage `appendChild()`, `removeChild()`, and `createTextNode()` are methods for manipulating the DOM by adding, removing, and creating elements and text nodes, while `resizeTo()`, `scrollBy()`, and `alert()` are methods of the Window Object used for controlling the browser window’s size, scrolling, and displaying alert messages to the user.