# DOM Objects

## Global Object

In JavaScript, the global object is an object that provides variables and functions that are available to all scripts running in the current context. The global object is the highest-level object in the JavaScript object hierarchy.

In a web browser, the global object is the window object, which represents the current browser window. It provides access to the browser's Document Object Model (DOM) and other browser-specific features such as the alert() function and the navigator object.

Here are a few examples of global objects and their properties and methods in JavaScript:

1. **window object:**

   The window object is the global object in a web browser, it represents the current browser window, and provides access to the browser's Document Object Model (DOM) and other browser-specific features such as the alert() function, setTimeout() method, and the navigator object.

2. **console object:**

   The console object provides methods for logging information to the browser's developer console. Examples of methods are console.log(), console.info(), console.error(), console.warn()

## History Object

In JavaScript, the history object provides properties and methods for interacting with the browser's history. The history object is a property of the window object, which represents the current browser window.
Here are a few examples of properties and methods of the history object:

1. **history.length:** Returns the number of items in the browser's history.

   ```
   console.log(history.length);
   ```

2. **history.back():** Navigates to the previous page in the browser's history.

   ```
   history.back();
   ```

3. **history.forward():** Navigates to the next page in the browser's history.

   history.forward();

## Navigator Object

In JavaScript, the navigator object provides properties and methods for interacting with the browser and getting information about the user's computer and browser. The navigator object is a property of the window object, which represents the current browser window.

Here are a few examples of properties and methods of the navigator object:

1. **navigator.userAgent:** Returns a string that represents the user-agent header sent by the browser to the server. This can be used to determine the type and version of the browser.

   console.log(navigator.userAgent);

2. **navigator.language:** Returns a string that represents the preferred language of the user's browser.

   console.log(navigator.language);

3. **navigator.onLine:** Returns a Boolean value indicating whether the browser is currently online or offline.

   console.log(navigator.onLine);

## Location Object

In JavaScript, the location object provides properties and methods for interacting with the current URL of the browser. The location object is a property of the window object, which represents the current browser window.

Here are a few examples of properties and methods of the location object:

1. **location.href:** Returns a string that represents the entire URL of the current page.
2. **location.protocol:** Returns a string that represents the protocol of the current URL (such as "http:" or "https:").
3. **location.host:** Returns a string that represents the hostname and port number of the current URL.
4. **location.pathname:** Returns a string that represents the pathname of the current URL.
5. location.assign("https://www.google.com"): It opens this new webpage but keeps the previous page's history as well. So, you can go back.

6. **location.replace("https://www.google.com"):** It is the same as assign(). Only the previous page is removed from the browser's history.

```
console.log(location.href);
console.log(location.protocol);
console.log(location.host);
console.log(location.pathname);
location.assign("https://www.google.com");
location.replace("https://www.google.com");
```

# Dialog Methods

In JavaScript, there are several built-in dialog methods that can be used to create dialog boxes and display them to the user:

1. **alert():** Displays an alert dialog box with an OK button and a message. This method blocks the execution of the script until the user clicks OK.

```
alert("This is an alert message");
```

2. **prompt():** Displays a dialog box that prompts the user for input. This method returns the user's input as a string.

```
const name = prompt("What is your name?");
console.log(`Hello, ${name}!`);
```

3. **confirm():** Displays a dialog box with a message and OK and Cancel buttons. This method returns a Boolean value indicating whether the user clicked OK or Cancel.

```
const confirmDelete = confirm("Are you sure you want to delete this item?");
if (confirmDelete) {
   // Perform delete action
} else {
   // Do not perform delete action
}
```