JavaScript

Introduction

The JavaScript language

- **JavaScript** is a programming language for use in HTML pages.
- Invented in 1995 at Netscape Corporation (LiveScript).
- Script means small piece of code. JavaScript is designed to add interactivity to HTML pages.
- JavaScript programs are run by an interpreter built into the user's web browser.
- While JavaScript programs are commonly executed by the user's web browser, they can also be run on the server-side using platforms such as Node.js.
- JavaScript can modify the contents of a web page when the page is loaded or in response to a user action.

Using JavaScript in HTML Document

- JavaScript can be inserted into documents by using the **SCRIPT tag.**
- All java script statements end with a semicolon (;)

… Not necessarily!...

- While JavaScript doesn't require semicolons after every statement, using them is considered good practice to prevent potential issues with Automatic Semicolon Insertion.
- JavaScript is case sensitive language.
- JavaScript uses C-style comments: // and /* */

```
<html>
<head>
<title>Hello World in JavaScript</title>
</head>
<body>
   <script type="text/javascript">
         document.write("Hello World!");
  </script>
</body>
</html>
```

The **type** attribute used to be required.
But in **HTML5** it can be left out.

Using JavaScript in HTML Document

- You can have any number of scripts.
- Scripts can be placed in the HEAD or in the BODY.
  - In the HEAD, scripts are run before the page is displayed.
  - In the BODY, scripts are run as the page is displayed.
- Functions that may execute multiple times is typically placed in the <HEAD>. These are only interpreted when the relevant function or event-handler are called
- Also Scripts can be placed within an HTML tag, such as <body> or <form>—This is called an **event handler**, and it enables the script to work with HTML elements.

Note: When using JavaScript in event handlers, you don't need to use the <script> tag.

Example

```
<html>
<head>
 <title>Hello World in JavaScript</title>
     <script>
          function helloWorld( ) {
           document.write("Hello World!"); }
    </script>
</head>
<body>
<h1>Event handler</h1>
<button type="button”  onclick=" alert('You clicked the button.')"> Click</button>
    <script>
          helloWorld( );
    </script>
</body>
</html>
```

External Scripts

- Scripts can also be loaded from an external file.
- This is useful if you have a complicated script or set of subroutines that are used in several different documents.
- JavaScript files have the file extension **.js**.
- To use an external script, put the name of the script file in the **src** (source) attribute of a <script> tag:

```
<script src="myscript.js"> </script>
```

- You can place an external script reference in <head> or <body>.
- You can add several script files to one page.

Note:
External scripts cannot contain `<script>` tags.

JavaScript Data Types

- A value in JavaScript is always of a certain type. There are eight basic data types in JavaScript. They are:

Data Types	Description	Example
String	represents textual data	'hello', "hello world!"
Number	an integer or a floating-point number	3, 3.234, 3e-2 etc.
BigInt	an integer more than **$(2^{53} - 1)$**	900719925124740999n
Boolean	Any of two values: true or false	true and false
undefined	variable is declared but not initialized	var a;
null	denotes a null value	let a = null;
Object	key-value pairs of collection of data	let student = { };

JavaScript Data Types

- all data types except Object are primitive data types, whereas Object is non-primitive.
- Strings may be enclosed in either single quotes or double quotes

  \" (double quote), \’ (single quote), etc.
- JavaScript is a dynamically typed language. It means that a variable doesn’t associate with a type. In other words, a variable can hold a value of different types. For example:

```
let counter = 120;      // counter is a number
    counter = false;    // counter is now a Boolean
    counter = "foo";    // counter is now a string
```

JavaScript Variables

- A **variable** is a named container used for storing values.
- There are two kind of variables in JavaScript : **local variable** and **global variable**.
- A local variable is declared inside block or function.
- A global variable is accessible from any function.
- JavaScript variable can be declared in different ways by using different keywords: **var**, **let** and **const** keyword. Each keyword is used in some specific conditions.
- var has been available since 1995, but with the release of ES6, let and const are now available to declare variables.

JavaScript Variables

- **var** keyword used to declare a variable that can accessible globally and changeable.

```
var x = 10;
```

- **let** keyword used to declare a variable that can accessible locally and changeable.

```
let counter = 120;
```

- **const** keyword used to declare a variable that can accessible locally and it can’t be reassigned.

```
const PI = 3.14;
```

JavaScript Operators

- JavaScript has most of the operators we're used to from C/Java
  - Arithmetic (+, - , *, /, %)
  - Assignment (=, +=, -=, *= /=, %=, ++, --)
  - Logical (&&, ||, !)
  - Comparison (<, >, <=, >=, ==)

JavaScript Output

- JavaScript output defines the ways to display the output of a given code.
- The output can be display by using four different ways which are listed below:
  - **document.write()** and **document.writeln()** //methods used to display some information on the HTML document.
  - **window.alert()**. //pop-up warning box.
  - **innerHTML**. //to write HTML elements.
  - **console.log()**. //is written to the browser console.

Using document.write()

- Used to write any output into the HTML webpage.
- The write() method writes the HTML expressions or JavaScript code to a document.
- This method is mostly used for testing purposes.

```
document.write(" any message");
document.writeln(" any message");
```

- The document is object name and write() or writeln() are methods.
- The difference between these two methods is carriage form feed character that is new line character automatically added into the document.writeln(),but it is not included in document.write ().

Using document.write()

- JavaScript Date Methods

```
<body>
  <p>
    <script>
      var today= new Date();
      document.write("Current date: <br/>" );
      document.write(today.toDateString());

      document.write("<br/>");
      document.write("&copy; ");
      document.write(today.getFullYear());
    </script>
  </p>
</body>
```

HTML Character Entities:

-copyright
- non-breaking space

Current date:
Sun May 12 2025
© 2025

Using window.alert()

- The alert() method in JavaScript is used to display a virtual alert box. It is mostly used to give a warning message to the users.
- The alert() method displays an alert box with a message and an OK button. When the dialog box pops up, we have to click "OK" to proceed.

```
window.alert("Welcome to\nJavaScript\nProgramming!");
```

- The window is object name and alert() is method.

Note: The window.alert() method can be written without the window prefix, like: alert()

Using innerHTML

- JavaScript lets you write into an HTML element by using innerHTML property.
- you can add anything, it can be a text message, some HTML element or anything else.
- To do that first you need to provide a specific Id to the **HTML element** that you want to access by the JavaScript code.
- To access an HTML element use document.getElementById(id) method, where **id** is the value of the id attribute of the HTML tag.

```
<p id="demo"></p>
<script>
    document.getElementById("demo").innerHTML ="JavaScript";
</script>
```

Using innerHTML

```
<!DOCTYPE html>
<html>
<head>
      <title>JavaScript innerHTML</title>
</head>
<body>
   <div id="main"></div>
   <script src="app.js"></script>
</body>
</html>
```

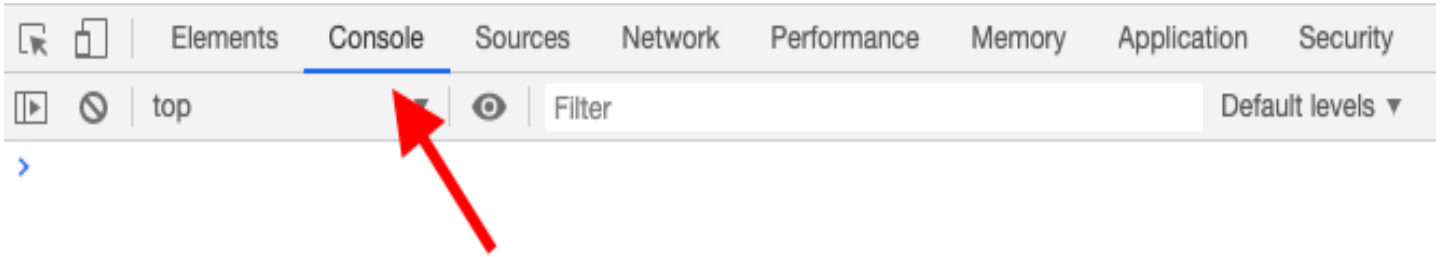
app.js file

```
let  mainDiv=document.getElementById('main');
mainDiv.innerHTML = "<h1>Welcome to JavaScript Programming</h1>";
mainDiv.style.color="orange";
```

Using console.log()

- JavaScript also lets you create console logs which can be seen in the **browser's developers' tools(Console)** for debugging purposes.
- The statement written inside a console log will be executed but would not be displayed in the browser instead it will be displayed inside the console of the browser.

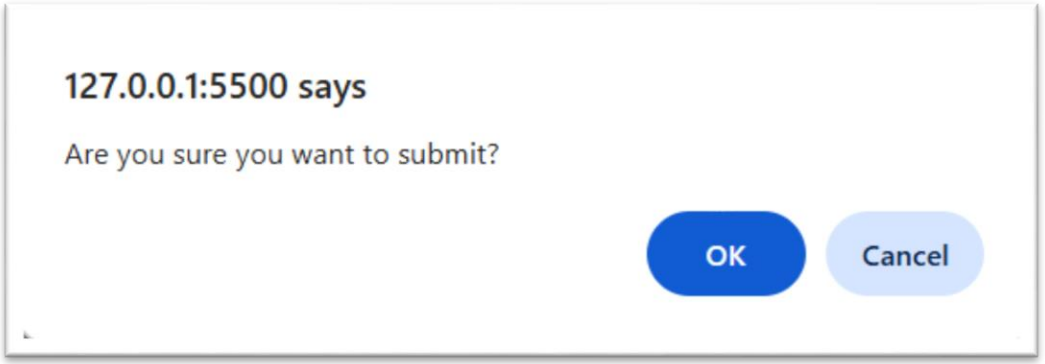
```
console.log(2+3);
```



JavaScript Popup Boxes

- In additional to **alert(msg) box** JavaScript has another two kind of popup boxes: Confirm box and Prompt box.
- **confirm(msg) box**: asks the user to confirm (or cancel) something.

```
window.confirm( "Are you sure you want to submit?" );
```

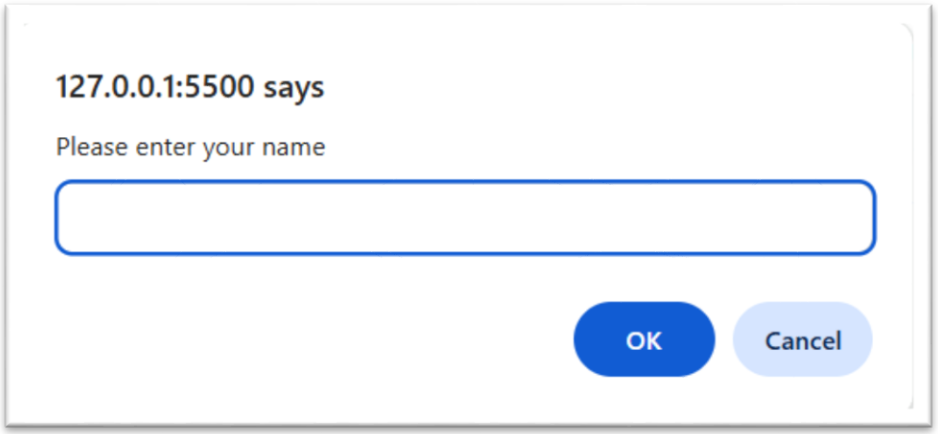


Note: The window.confirm() method can be written without the window prefix, like: confirm()

JavaScript Popup Boxes

- **prompt(msg, default) box**: asks the user to enter some text.
- The **prompt()** method returns the input value if the user clicks "OK", otherwise it returns **null** (if the user clicks “Cancel").

```
let name;
name = window.prompt( "Please enter your name" );
```



Example: Write a java script addition of two numbers.

```
<html>
<head>
<title>addition</title>
</head>
<body>
    <script>
      let a,b,c;
       a= parseInt(window.prompt("enter the a value"));
       b= parseInt(window.prompt("enter the b value"));
        c=a+b;
            document.write("The result="+c);
</script>
</body>
</html>
```

The **parseInt** method parses a value as a string and returns an integer.

Example:

```
<body>

   <p id="demo"></p>

 <script>

     let person = prompt("Please enter your name");
     if (person != null) {
       document.getElementById("demo").innerHTML = "Hello " +
           person + "! How are you today?";
     }
 </script>
</body>
```