JavaScript

What is JavaScript

- Widely-used programming language that is primarily used for web development.
- It allows developers to add interactivity and dynamic behavior to websites.
- JavaScript is a client-side scripting language.

JS Role in Web Development

- By enabling the creation of dynamic and interactive web applications.
- It enhances the user experience by providing features like .

1. form validation
2. responsive design
3. real-time updates,

First JS code

JavaScript within an HTML document using <script> tags

```
<!DOCTYPE html>
<html>
<head>
    <title>My First JavaScript</title>
</head>
<body>
    <script>
        // Your JavaScript code goes here
        alert('Hello, world!');
    </script>
</body>
</html>
```

Output:

"Hello, world!"

JS Syntax Overview

Variable declaration

```
var greeting = 'Hello, world!';
```

Function definition

```
function sayHello() {
    console.log(greeting);
}
```

Function call

```
sayHello();
```

JS Output

- innerHTML.
- document.write().
- window.alert().
- console.log().

Inner HTML

Input:

```
<script>
    var element = document.getElementById("output");
    element.innerHTML = "Hello, World!";
</script>
```

Output:

"Hello, world!"

Document Write

Input:

```
<script>
    document.write("Hello, World!");
</script>
```

Output:

"Hello, world!"

Window Alert

Input:

```
<script>
    window.alert("Hello, World!");
</script>
```

Output:

"Hello, world!"

Window Alert

Input:

```
<script>
    console.log("Hello, World!");
    console.log(123);
    console.log(document);
    console.log("Hello World", 123);
</script>
```

Output:

Hello, world!
123
Hello world 123

JS Basics

- Variables and Data Types.
- Operators.
- Conditional Statements.
- Loops.
- Arrays.
- Objects.

Variables

- Automatically
- Using var
- Using let
- Using const

Automatically

Input:

```
<script>
    var x = 5;
    var y = 6;
    var z = x + y;
</script>
```

Output:

11

Var

Input:

```
var x = 5;
var y = 6;
var z = x + y;
```

Output:

11

Let

Input:

```
let x = 5;
let y = 6;
let z = x + y;
```

Output:

11

Const

Input:

```
const x = 5;
const y = 6;
const z = x + y;
```

Output:

11

Datatypes

1. String
2. Number
3. Bigint
4. Boolean
5. Undefined
6. Object

Numbers

```
let length = 16;
let weight = 7.5;
```

Strings

```
let color = "Yellow";
let lastName = "Johnson";
```

Bigint

```
let x = BigInt("12345678901234567890123456789O");
```

Booleans

```
let x = true;
let y = false;
```

Undefined

```
let x;
x = 5;
x = "John";
```

Object

```
const person = {firstName:"John", lastName:"Doe"};
```

Operators

1. Arithmetic Operators
2. Assignment Operators
3. Comparison Operators
4. String Operators
5. Logical Operators
6. Bitwise Operators
7. Ternary Operators
8. Type Operators

Arithmetic operators

Addition (+)
Subtraction (-)
Multiplication (*)
Exponentiation (ES2016) (**)
Division (/)
Modulus (Division Remainder) (%)
Increment (++)
Decrement (--)

Assignment operators

(=) Assignment operator
(+=) Addition assignment
(-=) Subtraction Assignment
(*=) Multiplication Assignment
(/=) Division Assignment
(%=) Remainder Assignment
(**=) Exponentiation Assignment

Comparision operators

(==) equal to
(===) equal value and equal type
(!=) not equal
(!==) not equal value or not equal type
(>) greater than
(<) less than
(>=) greater than or equal to
(<=) less than or equal to
(?) ternary operator

Logical operators

(&&) logical and

(||) logical or

(!) logical not

Conditional Statements

Conditional statements like if, else if, and else allow you to make decisions in your code based on conditions.

Loops

Loops like for and while enable you to execute code repeatedly. They are useful for iterating through arrays, lists, or performing tasks a specific number of times.