Javascript

Javascript

- JavaScript is a programming language that executes on the browser.
-
- It turns static HTML web pages into interactive web pages by dynamically updating content, validating form data, controlling multimedia, animate images, and almost everything else on the web pages.

JavaScript is a **scripting language** used to create and control dynamic website content.

Runs in the **browser**.

Used for:

- Interactive forms
- Animations
- Dynamic content updates
- Web apps and games

How JavaScript Works

- Runs in the browser (client-side)
- Interacts with HTML and CSS
- Can also run on the server (Node.js)

🧠 **Note**: JavaScript is not the same as Java.

IDEs for JavaScript Application Development

- Visual Studio Code (Free, cross-platform)
- Eclipse (Free, cross-platform)
- Atom (Free, cross-platform)
- Notepad++ (Free, Windows)
- Code Lobster (Free, cross-platform)
- WebStorm (Paid, cross-platform)

Variables

- **var keyword** is used to declare variables since JavaScript was created. It is confusing and error-prone when using variables declared using var.
- **let keyword** removes the confusion and error of var. It is the new and recommended way of declaring variables in JavaScript.
- **const keyword** is used to declare a constant variable that cannot be changed once assigned a value.

`var` – Function Scoped

```
function testVar() {
  var x = 10;
  if (true) {
    var x = 20;      // Same variable!
    console.log(x); // 20
  }
  console.log(x); // 20 (var is function-scoped)
}
```

`let` – Block Scoped

```
function testLet() {
  let x = 10;
  if (true) {
    let x = 20; // Different variable
    console.log(x); // 20
  }
  console.log(x); // 10 (let is block-scoped)
}
```

`const` – Constant (Block Scoped)

```
const name = "Alice";
name = "Bob";   // ❌ Error: Assignment to constant variable
```

Use `let` for values that may change.

Use `const` for values that should never be reassigned.

Avoid `var` in modern JavaScript (can cause unexpected behavior due to hoisting and scope).

Feature	`var`	`let`	`const`
Scope	Function-scoped	Block-scoped (`{}`)	Block-scoped (`{}`)
Re-declaration	Allowed	❌ Not allowed in same scope	❌ Not allowed in same scope
Re-assignment	Allowed	Allowed	❌ Not allowed
Hoisting	Yes (initialized as `undefined`)	Yes (not initialized)	Yes (not initialized)
Use case	Legacy code (avoid if possible)	Preferred for changing values	Preferred for constants

Datatypes

Type	typeof return value	
Null	"object"	
Undefined	"undefined"	
Boolean	"boolean"	
Number	"number"	
BigInt	"bigint"	
String	"string"	
Symbol	"symbol"	

Example

- var a='hello' //string
- var b="hello“
- var t=22;
- var l="10"+22
- var m= true
- console.log(typeof(a))

JavaScript Strict Mode (With Examples)

- JavaScript is a loosely typed (dynamic) scripting language
- JavaScript allows strictness of code using "use strict" with ECMAScript 5 or later. Write "use strict" at the top of JavaScript code or in a function.

Example

- "use strict";

- var x = 1; // valid in strict mode
- y = 1; // invalid in strict mode

Types of JavaScript Operators

- Arithmetic Operators
- Assignment Operators
- Comparison Operators
- Logical Operators
- Conditional Operators
- Type Operators

Arithmetic operator

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (Division Remainder)

Assignment operator

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

Comparison operator

Operator	Description
==	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

Logical operator

Operator	Description
&&	logical and
||	logical or
!	logical not

Category of operator

- Unary operator
  - Increment & Decrement operator
    - Prefix & Postfix operator
- Binary operator
- Ternary operator（?）

Ternary operator

- condition ? value if true : value if false

- Example
- Input: let result = (10 > 0) ? true : false;
- Output: true

- Input: let message = (20 > 15) ? "Yes" : "No";
- Output: Yes

Example 2(Ternary operator)

- let age = 60
- let result = (age > 59)? "Senior Citizen":"Not a Senior Citizen";
-
- console.log(result);

Example 3 (Ternary Operator)

```
let marks = 95;
let result = (marks < 40) ? "Unsatisfactory" :
        (marks < 60) ? "Average" :
        (marks < 80) ? "Good" : "Excellent" ;

console.log(result);
```

Logical operator Example

- And
- Or
- Not

Generate a Random Number between 0 and 1

- // generating a random number
- const a = Math.random();
- console.log(a);

Type conversion

- type conversion is the process of converting data of one type to another. For example: converting String data to Number.

- There are two types of type conversion in JavaScript.
- **Implicit Conversion** - automatic type conversion
- **Explicit Conversion** - manual type conversion

Implicit type conversion

- let result;
- result = '3' + 2;
- console.log(result) // "32"
- result = '3' + true;
- console.log(result); // "3true"
- result = '3' + undefined;
- console.log(result); // "3undefined"
- result = '3' + null;
- console.log(result); // "3null"

Implicit Conversion to Number

- // numeric string used with - , / , * results number type
- let result;
- result = '4' - '2';
- console.log(result); // 2
- result = '4' - 2;
- console.log(result); // 2
- result = '4' * 2;
- console.log(result); // 8
- result = '4' / 2;
- console.log(result); // 2

Non-numeric String Results to NaN

- let result;
- result = 'hello' - 'world';
- console.log(result); // NaN

- result = '4' - 'hello';
- console.log(result); // NaN

Implicit Boolean Conversion to Number

- // if boolean is used, true is 1, false is 0
- let result;
- result = '4' - true;
- console.log(result); // 3
- result = 4 + true;
- console.log(result); // 5
- result = 4 + false;
- console.log(result); // 4

null Conversion to Number

- // null is 0 when used with number
- let result;

- result = 4 + null;
- console.log(result); // 4

- result = 4 - null;
- console.log(result); // 4

JavaScript Explicit Conversion

Example of explicit conversion

- //wap to ask two number and print their sum
- var e= prompt('enter number');
- var e1= prompt('enter number');
- console.log(Number(e) +Number(e1));

Convert to String Explicitly

- //number to string
- let result;
- result = String(324);
- console.log(result); // "324“
- result = String(null);
- console.log(result); // "null"

Conversion table

Value	String Conversion	Number Conversion	Boolean Conversion
1	"1"	1	true
0	"0"	0	false
"1"	"1"	1	true
"0"	"0"	0	true
"ten"	"ten"	NaN	true
true	"true"	1	true
false	"false"	0	false
null	"null"	0	false
undefined	"undefined"	NaN	false

Conditional operator

- If ..
- If else
- If else if else
- Switch case

Lab

- Check if a number is odd or even in JavaScript
- Find the largest of two number
- Find the largest of three number
- Find the a number is present in given range
- Find check if a year is leap year or not

- **Check if input variable is a number or not**

```
let input = "123";

if (typeof Number(input) === "number" && !isNaN(Number(input))) {
    console.log("It is a number");
} else {
    console.log("It is NOT a number");
}
```

Check if input variable is a number or not

- var num = "33d";
- if (isNaN(num))
- {
- console.log(`${num} is not a number`)
- }

Conditions for leap year

- If year is divisible by 4 and not divisible by 100 then print “leap year”.
- Or if year is divisible by 400 then print “leap year”.
- Else print “not a leap year”.

```
if(((year%4 == 0) && (year%100 != 0))){
    console.log(`Year ${year} is a leap year`);
}
```

Lab :Find number of days in a given month

- If month is outside the range of 1 and 12 print “Invalid month”.
- If month is equal to 2 ie, February print “28 days”
- Else if month is equal to 4, 6, 9 or 11 print “30 days”.
- Else print “31 days”.

solution

```
let month = 2; // example input

if (month < 1 || month > 12) {
    console.log("Invalid month");
} else if (month === 2) {
    console.log("28 days");
} else if (month === 4 || month === 6 || month === 9 || month === 11) {
    console.log("30 days");
} else {
    console.log("31 days");
}
```

What is ECMAScript?

- ECMAScript is the **standard** on which JavaScript is based.
- ES6 is one of the most important updates, introducing modern features that made JavaScript **cleaner, faster, and easier** to write.