

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Part -1 JavaScript

Introduction to JavaScript

JavaScript ek powerful programming language hai jo mainly web pages ko interactive banane ke liye use hota hai. Jab hum kisi website par buttons click karte hain, forms fill karte hain, animations dekhte hain ya live updates milte hain—yeh sab JavaScript ki wajah se possible hota hai.

JavaScript browser ke andar hi run hoti hai, matlab aapko koi extra software install karne ki zarurat nahi hoti. Yeh HTML aur CSS ke saath milkar website ko structure, style aur behavior deti hai:

- HTML: Website ka structure (jaise building ka skeleton)
- CSS: Website ka design/style (colors, layout, fonts)
- JavaScript: Website ka behavior (interactivity, logic)

Why We Used JavaScript

- Web pages ko interactive banane ke liye
- Buttons, forms, menus, popups work karwane ke liye
- Page ko reload kiye bina content change karne ke liye
- Modern web apps (like YouTube, Gmail) banane ke liye
- Mobile apps banane ke liye (React Native)
- Backend servers banane ke liye (Node.js)
- Games aur animations banane ke liye
- Kyunki JavaScript easy to learn aur har browser me run hoti hai

JavaScript Output Methods

`console.log()`

Browser console me output show karta hai.

`console.log("Hello JavaScript!");`

`document.write()`

Direct web page par output likhta hai.

`document.write("This is JavaScript output");`

`innerHTML`

HTML element ke andar content show karta hai.

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

`document.getElementById("demo").innerHTML = "Output here!";`

What are Variables

Jaise dabba jisme hum koi value rakhte hain.

Types of Variables in JavaScript

- Let- Value change ho sakti hai

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Variable Example</title>
</head>
<body>
  <script>
    let age = 20; // variable declared
    age = 21;    // value changed
    document.write("Age is: " + age);
  </script>
</body>
</html>
```

- Var- Purana method, ab kam use hota hai

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Var Example</title>
</head>
<body>
  <script>
    var name = "Ali"; // variable declared using var
    document.write("Name is: " + name);
  </script>
</body>
</html>
```

- Const-Value change nahi ho sakti

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Const Example</title>
</head>
<body>
  <script>
    const pi = 3.14; // constant declared
    // pi = 3.15; // ✗ Not allowed, will cause error
    document.write("Value of pi is: " + pi);
  </script>
</body>
</html>
```

Popup boxes

Popup box ek chhota sa dialog box hota hai jo screen par user se interact karne ke liye aata hai.

Ye user ko message dikhata, confirmation mangta, ya input leta.

Jab popup open hota hai, user baaki page ke saath interact nahi kar sakta jab tak wo response na de.

Alert Box – Sirf message dikhata hai. **OK button** hota hai.

```
<!DOCTYPE html>
<html>
<head>
  <title>Alert Box Example</title>
</head>
<body>
  <script>
    // Ye ek alert box show karega
    alert("Hello! Welcome!");
  </script>
</body>
</html>
```

Confirm Box – Question puchta hai. **OK aur Cancel buttons** hote

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

hain.

```
<!DOCTYPE html>
<html>
<head>
  <title>Confirm Box Simple</title>
</head>
<body>
  <script>
    let check = confirm("Are you 18?");
    document.write("Your response is: " + check);
  </script>
</body>
</html>
```

Prompt Box – User se input leta hai. OK aur Cancel buttons hote hain.

```
<!DOCTYPE html>
<html>
<head>
  <title>Prompt Box Example</title>
</head>
<body>
  <script>
    // User se naam input lene ke liye prompt box
    let name = prompt("Enter your name:", "Guest");
    // Naam ko page par display karna
    document.write("Hello, " + name);
  </script>
</body>
</html>
```

Operators

Operators are special symbols jo values ke upar operations perform karte hain.

Types of JavaScript Operators

Arithmetic Operators (Math ke Operators)

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Operator	Meaning	Example
+	Addition	$5 + 2 = 7$
-	Subtraction	$5 - 2 = 3$
*	Multiplication	$5 * 2 = 10$
/	Division	$10 / 2 = 5$
%	Modulus (Reminder)	$5 \% 2 = 1$

```
<!DOCTYPE html>
<html>
<head>
  <title>Arithmetic Operators Example</title>
</head>
<body>
  <script>
    let a = 5;
    let b = 2;
    document.write("a + b = " + (a + b) + "<br>");
    document.write("a - b = " + (a - b) + "<br>");
    document.write("a * b = " + (a * b) + "<br>");
    document.write("a / b = " + (a / b) + "<br>");
    document.write("a % b = " + (a % b) + "<br>");
  </script>
</body>
</html>
```

Assignment Operators (Value Assign Karne ke Operators)

Operator	Meaning	Example
=	Assign value	$x = 5$

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Operator	Meaning	Example
<code>+=</code>	Add & assign	<code>x += 2 → x = x+2</code>
<code>-=</code>	Subtract & assign	<code>x -= 2 → x = x-2</code>
<code>*=</code>	Multiply & assign	<code>x = 2 → x = x2</code>
<code>/=</code>	Divide & assign	<code>x /= 2 → x = x/2</code>

```
<!DOCTYPE html>
<html>
<head>
  <title>Assignment Operators</title>
</head>
<body>
<script>
let x = 5;
document.write("x = " + x + "<br>");
x += 2; document.write("x += 2 → " + x + "<br>");
x -= 2; document.write("x -= 2 → " + x + "<br>");
x *= 2; document.write("x *= 2 → " + x + "<br>");
x /= 2; document.write("x /= 2 → " + x + "<br>");
</script>
</body>
</html>
```

Comparison Operators (Compare Karne ke Operators)

Operator	Meaning	Example
<code>==</code>	Equal	<code>5 == 5 → true</code>
<code>===</code>	Strict equal (type + value)	<code>5 === "5" → false</code>
<code>!=</code>	Not equal	<code>5 != 2 → true</code>
<code>!==</code>	Strict not equal	<code>5 !== "5" → true</code>
<code>></code>	Greater than	<code>5 > 2 → true</code>

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Operator	Meaning	Example
<	Less than	2 < 5 → true
>=	Greater or equal	5 >= 5 → true
<=	Less or equal	4 <= 5 → true

```
<!DOCTYPE html>
<html>
<head>
  <title>Comparison Operators</title>
</head>
<body>
<script>
let a = 5, b = 2, c = "5";
document.write("a == b → " + (a == b) + "<br>");
document.write("a == c → " + (a == c) + "<br>");
document.write("a === c → " + (a === c) + "<br>");
document.write("a != b → " + (a != b) + "<br>");
document.write("a !== c → " + (a !== c) + "<br>");
document.write("a > b → " + (a > b) + "<br>");
document.write("a < b → " + (a < b) + "<br>");
document.write("a >= b → " + (a >= b) + "<br>");
document.write("b <= a → " + (b <= a) + "<br>");
</script>
</body>
</html>
```

Logical Operators (Logic ke Operators)

Operator	Meaning	Example
&&	AND	true && false → false
!	NOT	!true → false

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<head>
  <title>Logical Operators</title>
</head>
<body>
<script>
let a = true;
let b = false;
document.write("a && b → " + (a && b) + "<br>"); // AND
document.write("a || b → " + (a || b) + "<br>"); // OR
document.write("!a → " + (!a) + "<br>"); // NOT
document.write("!b → " + (!b) + "<br>");
</script>
</body>
</html>
```

Increment / Decrement Operators

Operator	Meaning	Example
++	Increase by 1	x = 5 → x++ → 6
--	Decrease by 1	x = 5 → x-- → 4

```
<!DOCTYPE html>
<html>
<head>
  <title>Increment & Decrement Operators</title>
</head>
<body>
<script>
let x = 5;
document.write("Initial value: x = " + x + "<br>");
x++; // Increment by 1
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
document.write("After x++ → " + x + "<br>");  
x--; // Decrement by 1  
document.write("After x-- → " + x + "<br>");  
</script>  
</body>  
</html>
```

Ternary Operator (Shortcut if/else)

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Ternary Operator Example</title>  
</head>  
<body>  
  <h2>Ternary Operator Example</h2>  
  <script>  
    // Age check using ternary operator  
    let age = 18;  
    let status = (age >= 18) ? "Adult" : "Minor";  
    // Display result in console  
    console.log(status);  
    // Optional: Display result on the web page  
    document.write("Status: " + status);  
  </script>  
</body>  
</html>
```

String Operators

- + → Strings ko join karta hai (Concatenation)

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>String Operators</title>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</head>
<body>
<script>
let firstName = "Ali";
let lastName = "Khan";
document.write("Full Name: " + firstName + " " + lastName);
</script>
</body>
</html>
```

What are Conditional Statements

Conditional statements are instructions in JavaScript jo decision lene ke liye use hoti hain.

Yani program ko decide karna hota hai ki kya action lena hai based on some condition.

What Is an if Statement?

- if statement ek conditional statement hai
- Iska kaam hai decision lena: agar koi condition true ho, to wo ek block of code execute karega.
- Agar condition false ho, to wo code skip ho jayega (agar sirf if ho, else na ho).

```
if (condition) {
// Code yahan chalega agar condition true ho
}
<!DOCTYPE html>
<html>
<body>
<script>
let age = 20;
if (age >= 18) {
console.log("You are an adult");
}
// Agar age 18 se kam hoti, to kuch bhi aur nahi chalega is example mein
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</script>  
</body>  
</html>
```

If-Else

if-else ka matlab hai condition check karna aur uske hisaab se decision lena.

if → “agar ye condition sach hai, to ye karo”

else → “warna ye karo”

```
if (condition) {  
    // ye code tab chalega jab condition true ho  
} else {  
    // ye code tab chalega jab condition false ho  
}  
<!DOCTYPE html>  
<html lang="en">  
<head>  
<title>Leap year </title>  
</head>  
<body>  
<h1>Check Grade</h1>  
<script>  
let year = prompt("Enter a year:");  
if ((year % 4 === 0 && year % 100 !== 0) || (year % 400 === 0)) {  
    alert(year + " is a leap year.");  
} else {  
    alert(year + " is not a leap year.");  
}  
</script>  
</body>  
</html>
```

Else if

else if programming me use hota hai jab hume multiple conditions check karni ho.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- if → sabse pehli condition check karta hai.
- else if → agar pehli condition false ho, tab next condition check karta hai.
- else → agar sab conditions false ho, tab execute hota hai.

```
if (condition1) {
    // runs if condition1 is true
}
else if (condition2) {
    // runs if condition2 is true
}
else if (condition3) {
    // runs if condition3 is true
}
else {
    // runs if none of the above conditions are true
}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Grade Calculator</title>
</head>
<body>
  <h1>Check Grade</h1>
  <script>
    let marks = Number(prompt("Enter your marks (0-100):"));
    let grade;
    if (marks >= 90 && marks <= 100) {
      grade = "A";
    }
    else if (marks >= 80) {
      grade = "B";
    }
  </script>
</body>
</html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
else if (marks >= 70) {
    grade = "C";
}
else if (marks >= 60) {
    grade = "D";
}
else if (marks >= 0) {
    grade = "F";
}
else {
    grade = "Invalid marks!";
}
console.log("Grade:", grade);
alert("Grade: " + grade);
</script>
</body>
</html>
```

Switch statement

switch statement programming me use hota hai jab hume ek variable ke multiple possible values ke basis par alag-alag code run karna ho. Ye if-else if ke jagah cleaner aur easy-to-read hota hai jab options zyada ho.

```
switch (expression) {
    case value1:
        // ye code chalega agar expression === value1
        break;
    case value2:
        // ye code chalega agar expression === value2
        break;
    case value3:
        // ye code chalega agar expression === value3
        break;
    default:
        // ye code chalega agar kisi bhi case se match na ho
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

}

- expression evaluate hota hai.
- Uski value har case ke saath compare hoti hai.
- Agar match milta hai → us case ka code run hota hai.
- break statement ka matlab hai baaki ke cases skip kar do.
- default tab run hota hai jab koi match na mile (ye else ki tarah hai).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Day Name Switch Example</title>
</head>
<body>
  <h1>Day Name Example using Switch</h1>

  <script>
    // Variable for the day number
    let day = 3;
    let dayName;
    // Switch statement to find the day name
    switch (day) {
      case 1:
        dayName = "Monday";
        break;
      case 2:
        dayName = "Tuesday";
        break;
      case 3:
        dayName = "Wednesday";
        break;
      case 4:
        dayName = "Thursday";
        break;
      case 5:
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
        dayName = "Friday";
        break;
    case 6:
        dayName = "Saturday";
        break;
    case 7:
        dayName = "Sunday";
        break;
    default:
        dayName = "Invalid day";
    }
    // Output the result
    console.log(dayName); // Output in console
    alert("Day Name: " + dayName); // Popup in browser
</script>
</body>
</html>
```

Loops

JavaScript me loops use hote hain jab hume ek hi kaam baar-baar repeat karna ho. Instead of baar-baar same code likhne ke, loops se easy aur fast ho jata hai.

for loop

Use tab hota hai jab hume pata ho kitni baar repeat karna hai.

```
for (initialization; condition; increment/decrement) {
    // ye code repeat hoga
}
<!DOCTYPE html>
<html>
<head>
    <title>Counting Loop</title>
</head>
<body>
    <script>
        for (let i = 1; i <= 5; i++) {
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
        console.log("Count:", i);
    }
</script>
</body>
</html>
```

While loop

While loop ek aisa loop hota hai jo tab tak chalta rehta hai jab tak condition true ho.

Jab condition false ho jaati hai, loop ruk jaata hai.

```
while (condition) {
    // code to run again and again
}
```

- condition: Jab tak yeh true rahe, loop chalta rahega.
- Jab condition false ho jaaye, loop ruk jaata hai.

```
<!DOCTYPE html>
<html>
<head>
<title>While Loop Example</title>
</head>
<body>
<script>
let i = 1;
while (i <= 5) {
    console.log("Count:", i);
    i++;
}
</script>
</body>
</html>
```

Do...while loop

Do...while loop pehle code chalata hai, phir condition check karta hai.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Agar condition true ho, loop dobara chalega.
Agar false ho, loop ruk jaayega.

```
do {  
    // run this code first  
} while (condition);  
<!DOCTYPE html>  
<html>  
<head>  
    <title>Do While Loop Example</title>  
</head>  
<body>  
<script>  
    let i = 1;  
    do {  
        console.log("Count:", i);  
        i++;  
    } while (i <= 5);  
</script>  
</body>  
</html>
```

Loop Scope

- Loop scope ka matlab hai:
- Jo variables loop ke andar declare kiye gaye hain, woh sirf loop ke andar hi kaam karte hain.
- Loop ke bahar unko access nahi kar sakte (agar let ya const use kiya ho).

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>Loop Scope Example</title>  
</head>  
<body>  
    <h1>Loop Scope Example</h1>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<div id="output"></div>
<script>
  const output = document.getElementById("output");
  // [1] Using let (block scoped)
  output.innerHTML += "<h2>Using let (block scoped)</h2>";
  for (let i = 1; i <= 3; i++) {
    let msg = "Hello from let";
    output.innerHTML += "Inside loop: " + i + ", " + msg + "<br>";
  }
  // Variables declared with let are not accessible here
  output.innerHTML += "Outside loop: let variables cannot be
  accessed.<br>";
  // [2] Using var (function scoped)
  output.innerHTML += "<h2>Using var (function scoped)</h2>";
  for (var j = 1; j <= 3; j++) {
    var message = "Hello from var";
    output.innerHTML += "Inside loop: " + j + ", " + message + "<br>";
  }
  // var variables are accessible outside the loop
  output.innerHTML += "Outside loop j: " + j + "<br>";
  output.innerHTML += "Outside loop message: " + message + "<br>";
</script>
</body>
</html>
```

Break in Loops

- Jab break statement loop ke andar aata hai,
→ loop turant ruk jaata hai.
- Loop ka baaki code skip ho jaata hai.
- Program control loop ke baad ke statements pe chala jaata hai.

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript break</h1>
<h2>The break Statement in a Loop</h2>
<p>Break the loop when the loop counter is 3:</p>
<p id="demo"></p>
<script>
let text = "";
for (let i = 0; i < 10; i++) {
  if (i === 3) { break; }
  text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Continue Statement

- Jab continue statement loop ke andar aata hai,
→ current iteration skip ho jaati hai.
- Loop ka baaki code us iteration me execute nahi hota.
- Program next iteration se continue karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Loops</h1>
<h2>The continue Statement</h2>
<p>Skip the iteration step when i equals 3.</p>
<p id="demo"></p>

<script>
let text = "";
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
for (let i = 1; i < 10; i++) {  
  if (i === 3) { continue; }  
  text += i*10 + "<br>";  
}  
document.getElementById("demo").innerHTML = text;  
</script>  
</body>  
</html>
```

String

String ek sequence hoti hai characters ki.

JavaScript me string ko quotes (" ", ' ') ya backticks (` `) me likhte hain.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Strings</h1>  
<p>Strings are written inside quotes. You can use single or double quotes:</p>  
<p id="demo"></p>  
<script>  
let carName1 = "Volvo XC60"; // Double quotes  
let carName2 = 'Volvo XC60'; // Single quotes  
document.getElementById("demo").innerHTML =  
carName1 + " " + carName2;  
</script>  
</body>  
</html>
```

String Template

Template literals backticks (` `) me likhi jaati hain, double ya single quotes nahi.

- Iska main fayda: variables aur expressions ko direct string ke andar embed kar sakte hain.
- Hum `${...}` syntax ka use karte hain variables ya expressions ko string me insert karne ke liye.

```
<!DOCTYPE html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<html>
<head>
  <title>Template Literal Example</title>
</head>
<body>
  <h1>Template Literal Example</h1>
  <div id="output"></div>
  <script>
    const output = document.getElementById("output");
    let name = "Rahul";
    let age = 20;
    // Template literal
    let message = `My name is ${name} and I am ${age} years old.`;
    output.innerHTML = message;
  </script>
</body>
</html>
```

String Functions

Basic String Methods

Javascript strings are primitive and immutable: All string methods produce a new string without altering the original string.

<u>String.length</u>	<u>String.isWellFormed()</u>
<u>String.charAt()</u>	<u>String.toWellFormed()</u>
<u>String.charCodeAt()</u>	<u>String.trim()</u>
<u>String.codePointAt()</u>	<u>String.trimStart()</u>
<u>String.concat()</u>	<u>String.trimEnd()</u>
<u>String.at()</u>	<u>String.padStart()</u>
<u>String[.]</u>	<u>String.padEnd()</u>
<u>String.slice()</u>	<u>String.repeat()</u>
<u>String.substring()</u>	<u>String.replace()</u>
<u>String.substr()</u>	<u>String.replaceAll()</u>
<u>String.toUpperCase()</u>	<u>String.split()</u>
<u>String.toLowerCase()</u>	

Length property

ADDRESS: EKTA VIHAR, NEAR ASHOKA PALACE, SHAHABAD MARKANDA CALL- 7700090800

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Length property ek string ki lambai batata hai.”

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The length Property</h2>
<p>The length of the string is:</p>
<p id="demo"></p>
<script>
let text = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
document.getElementById("demo").innerHTML = text.length;
</script>
</body>
</html>
```

Extracting String Characters

String se character nikalne ke 4 tareeke hain:

1. at(position) method
2. charAt(position) method
3. charCodeAt(position) method
4. Property access [] (jaise arrays mein)

charAt() method ek string mein di gayi position (index) pe wale character ko return karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript String</h1>
<h2>The charAt() Method</h2>
<p>The charAt() method returns the character at a given position in a
string:</p>
<p id="demo"></p>
<script>
var text = "HELLO WORLD";
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML = text.charAt(0);  
</script>  
</body>  
</html>
```

charAt() method ek string mein di hui index pe character ka code (UTF-16 code unit) return karta hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript String</h1>  
<h2>The charCodeAt() Method</h2>  
<p>The charCodeAt() method returns the unicode of the character at a given  
position in a string:</p>  
<p id="demo"></p>  
<script>  
let text = "HELLO WORLD";  
document.getElementById("demo").innerHTML = text.charCodeAt(0);  
</script>  
</body>  
</html>
```

concat() method ko + operator ki jagah use kiya ja sakta hai. Ye dono lines ek hi kaam karti hain:

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript String</h1>  
<h2>The concat() Method</h2>  
<p>The concat() method joins two or more strings:</p>  
<p id="demo"></p>  
<script>  
let text1 = "Hello";
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
let text2 = "World!";  
let text3 = text1.concat(" ",text2);  
document.getElementById("demo").innerHTML = text3;  
</script>  
</body>  
</html>
```

at() method string me ek character return karta hai jo specified index (position) par ho.

string.at(index)

- Agar negative index use karo, toh string ke end se count hota hai.
- Modern browsers me March 2022 se supported hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Strings</h1>  
<h2>Bracket Notation</h2>  
<p>The bracketed notation [] returns an indexed element from a string:</p>  
<p id="demo"></p>  
<script>  
const name = "W3Schools";  
let letter = name[2];  
document.getElementById("demo").innerHTML = letter;  
</script>  
</body>  
</html>
```

Browser Support at() is an ES2022 feature.

				
Chrome 92	Edge 92	Firefox 90	Safari 15.4	Opera 78
Apr 2021	Jul 2021	Jul 2021	Mar 2022	Aug 2021

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

slice() method string ka ek part extract karta hai aur usse nayi string me return karta hai.

- start → starting position (include hota hai)
- end → ending position (exclude hota hai)

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The slice() Method</h2>
<p>The sliced (extracted) part of the string is:</p>
<p id="demo"></p>
<script>
let text = "Apple, Banana, Kiwi";
let part = text.slice(7,13);
document.getElementById("demo").innerHTML = part;
</script>
</body>
</html>
```

substring() method slice() ke similar hai, matlab string ka ek part extract karta hai aur nayi string return karta hai.

Difference:

- slice() me agar negative index use karo → string ke end se count hota hai.
- substring() me agar start ya end < 0 → 0 treat hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript String Methods</h1>
<p>The substring() method extract a part of a string and returns the extracted parts in a new string:</p>
<p id="demo"></p>
<script>
let str = "Apple, Banana, Kiwi";
document.getElementById("demo").innerHTML = str.substring(7,13);
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</script>
</body>
</html>
```

substr() method slice() ke similar hai, matlab string ka ek part extract karta hai aur nayi string return karta hai.

Difference:

- slice() me second parameter = end position hota hai (exclude).
- substr() me second parameter = length of extracted part hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript String Methods</h1>
<p>The substr() method extract a part of a string
and returns the extracted parts in a new string:</p>
<p id="demo"></p>
<script>
let str = "Apple, Banana, Kiwi";
document.getElementById("demo").innerHTML = str.substr(7,6);
</script>
</body>
</html>
```

Convert to Upper Case

String ko uppercase me convert karne ke liye toUpperCase() use hota hai.

```
<!DOCTYPE html>
<html>
<head>
<title>Uppercase Example</title>
</head>
<body>
<h1>Check the Console for Uppercase Result</h1>
<script>
// Original string
const text = "Hello World";
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
// Convert to uppercase and print
console.log(text.toUpperCase()); // Output: "HELLO WORLD"
</script>
</body>
</html>
```

Convert to Lower Case

String ko lowercase me convert karne ke liye toLowerCase() use hota hai.

```
<!DOCTYPE html>
<html>
<head>
  <title>Lowercase Example</title>
</head>
<body>
  <h1>Check the Console for Lowercase Result</h1>
  <script>
    const text = "Hello World";
    console.log(text.toLowerCase());
  </script>
</body>
</html>
```

isWellFormed() method check karta hai ki string well-formed Unicode hai ya nahi.

Return value:

- true → agar string well-formed Unicode hai.
- false → agar string well-formed nahi hai.

```
<!DOCTYPE html>
<html>
<body>
  <h1>JavaScript Strings</h1>
  <h2>The isWellFormed() Method</h2>
  <p>isWellFormed() returns true if a string is well formed.</p>
  <p id="demo"></p>
</script>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
let text = "Hello world!";
let result = text.isWellFormed();
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

toWellFormed() method string me jo “lone surrogates” hote hain, unhe Unicode replacement character (□ / U+FFFD) se replace karke nayi string return karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The toWellFormed() Method</h2>
<p>toWellFormed() returns a new string where all "lone surrogates" are
replaced with the Unicode replacement character U+FFFD.</p>
<p id="demo"></p>
<script>
let text = "Hello world \uD800";
let result = text.toWellFormed();
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

trim() method string ke dono sides (start aur end) se whitespace remove karta hai aur nayi string return karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

<h2>The trim() Method</h2>

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

<script>

```
let text1 = " Hello World! ";
```

```
let text2 = text1.trim();
```

```
document.getElementById("demo").innerHTML =
```

```
"Length text1 = " + text1.length + "<br>Length text2 = " + text2.length;
```

</script>

</body>

</html>

trimEnd() method trim() ke similar hai, lekin ye sirf string ke end se whitespace remove karta hai.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript Strings</h1>
```

```
<h2>The trimEnd() Method</h2>
```

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

```
<script>
```

```
let text1 = " Hello World! ";
```

```
let text2 = text1.trimEnd();
```

```
document.getElementById("demo").innerHTML =
```

```
"Length text1 = " + text1.length + "<br>Length text2 = " + text2.length;
```

</script>

</body>

</html>

trimStart() method trim() ke similar hai, lekin ye sirf string ke start se whitespace remove karta hai.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<h1>JavaScript Strings</h1>
<h2>The trimStart() Method</h2>
<p id="demo"></p>
<script>
let text1 = " Hello World! ";
let text2 = text1.trimStart();
document.getElementById("demo").innerHTML =
"Length text1 = " + text1.length + "<br>Length text2 = " + text2.length;
</script>
</body>
</html>
```

padStart() method string ko start se pad karta hai, matlab string ke left side me extra characters add karta hai jab tak string desired length tak na pahunch jaye.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The padStart() Method</h2>
<p>The padStart() method pads a string from the start.</p>
<p>It pads the string with another string (multiple times) until it reaches a given
length.</p>
<p id="demo"></p>
<script>
let text = "5";
text = text.padStart(4,"0");
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

padEnd() method string ko end se pad karta hai, matlab string ke right side me extra characters add karta hai jab tak string desired length tak na

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

pahunch jaye.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The padEnd() Method</h2>
<p>The padEnd() method pads a string at the end.</p>
<p>It pads the string with another string (multiple times) until it reaches a given
length.</p>
<p id="demo"></p>
<script>
let text = "5";
document.getElementById("demo").innerHTML = text.padEnd(4,"x");
</script>
</body>
</html>
```

repeat() method string ko multiple copies me repeat karke nayi string return karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The repeat() Method</h2>
<p>repeat() returns a new string with a number of copies of a string:</p>
<p id="demo"></p>
<script>
let text = "Hello world!";
let result = text.repeat(2);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

replace() method string me ek specified value ko dusri value se replace karta hai aur nayi string return karta hai.

```
<!DOCTYPE html>
<html>
<head>
<title>Replace Example</title>
</head>
<body>
<h1>Check the Console for Replace Result</h1>
<script>
const text = "Hello World";
const newText = text.replace("World", "Universe");
console.log(newText); // Output: "Hello Universe"
</script>
</body>
</html>
```

replaceAll() method string me specified value ya regular expression ko sabhi occurrences ke saath replace karta hai aur nayi string return karta hai.

Parameters:

- searchValue → string ya regular expression (agar RegExp ho → global flag g mandatory)
- newValue → nayi value jo replace karegi

```
<!DOCTYPE html>
<html>
<head>
<title>ReplaceAll Example</title>
</head>
<body>
<h1>Check the Console for ReplaceAll Result</h1>
<script>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
// Original string
const text = "Hello World World";

// Replace all occurrences of "World" with "Universe"
const newText = text.replaceAll("World", "Universe");
// Print the result
console.log(newText); // Output: "Hello Universe Universe"
</script>
</body>
</html>
```

split() ka matlab hai “string ko tod ke ek array bana do”. Matlab, ek lambi string ko chhote-chhote tukdo me baant do, jisse har tukda array ke ek element me aa jaye.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript String Methods</h1>
<p>Display the first array element, after a string split:</p>
<p id="demo"></p>
<script>
let text = "a,b,c,d,e,f";
const myArray = text.split(",");
document.getElementById("demo").innerHTML = myArray[0];
</script>
</body>
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

String Methods

Basic String Methods

Javascript strings are primitive and immutable: All string methods produce a new string without altering the original string.

<code>String.length</code>	<code>String.isWellFormed()</code>
<code>String.charAt()</code>	<code>String.toWellFormed()</code>
<code>String.charCodeAt()</code>	<code>String.trim()</code>
<code>String.codePointAt()</code>	<code>String.trimStart()</code>
<code>String.concat()</code>	<code>String.trimEnd()</code>
<code>String.at()</code>	<code>String.padStart()</code>
<code>String[.]</code>	<code>String.padEnd()</code>
<code>String.slice()</code>	<code>String.repeat()</code>
<code>String.substring()</code>	<code>String.replace()</code>
<code>String.substr()</code>	<code>String.replaceAll()</code>
<code>String.toUpperCase()</code>	<code>String.split()</code>
<code>String.toLowerCase()</code>	

indexOf() method string ke andar kisi word/letter ka pehla location (index number) batata hai. Agar wo word mil gaya, toh uski position dega. Agar nahi mila, toh -1 return karega.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The indexOf() Method</h2>
<p>The indexOf() method returns the position of the first occurrence of a string in a string.</p>
<p>The position of the first occurrence of "locate" is:</p>
<p id="demo"></p>
<script>
let text = "Please locate where 'locate' occurs!";
let index = text.indexOf("locate");
document.getElementById("demo").innerHTML = index;
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</script>  
</body>  
</html>
```

lastIndexOf() method string ke andar kisi word/letter ka last wala position batata hai — yani wo word aakhri baar kahan mila, uska index return karta hai. Agar word nahi milta, toh ye bhi -1 return karta hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Strings</h1>  
<h2>The lastIndexOf() Method</h2>  
<p>The position of the last occurrence of "locate" is:</p>  
<p id="demo"></p>  
<script>  
let text = "Please locate where 'locate' occurs!";  
let index = text.lastIndexOf("locate");  
document.getElementById("demo").innerHTML = index;  
</script>  
</body>  
</html>
```

search() method string ke andar kisi word ya regular expression (pattern) ko dhundta hai, aur jahan pe pehli baar match milta hai, uska position (index) return karta hai. Agar match nahi milta, toh -1 return karta hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Strings</h1>  
<h2>The search() Method</h2>  
<p>The search() method returns the position of the first occurrence of a string in a string.</p>  
<p>Return the position of the first occurrence of a regular expression:</p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<p id="demo"></p>
<script>
let text = "Please locate where 'locate' occurs!";
let index = text.search(/locate/);
document.getElementById("demo").innerHTML = index;
</script>
</body>
</html>
```

match() method string ke andar kisi word ya regular expression (pattern) ko match karta hai, aur jitne bhi matches milte hain unko array ke form me return karta hai.

- Agar match mil gaya → array me result(s) aayega
- Agar match nahi mila → null return hota hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The match() Method</h2>
<p>Perform a search for "ain":</p>
<p id="demo"></p>
<script>
let text = "The rain in SPAIN stays mainly in the plain";
const myArr = text.match("ain");
document.getElementById("demo").innerHTML = myArr.length + " " + myArr;
</script>
</body>
</html>
```

matchAll() method string ke andar kisi word ya regular expression (pattern) ko match karta hai aur jitne bhi matches milte hain unko iterator ke form me return karta hai matlab ek-ek karke results nikal sakte ho.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Yeh match() jaisa hi hai, par difference: matchAll() saare matches ka full detail deta hai, including groups, index, etc. aur hamesha iterator return karta hai, array nahi.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The matchAll() Method</h2>
<p>ES2020 introduced the string method matchAll().</p>
<p id="demo"></p>
<script>
let text = "I love cats. Cats are very easy to love. Cats are very popular."
const iterator = text.matchAll("Cats");
document.getElementById("demo").innerHTML = Array.from(iterator);
</script>
</body>
</html>
```

includes() method check karta hai ki string ke andar koi specific word ya value present hai ya nahi. Agar value mil gayi → true. Agar nahi mili → false.

```
<!doctype html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The includes() Method</h2>
<p>Check if a string includes "world":</p>
<p id="demo"></p>
<script>
let text = "Hello world, welcome to the universe.";
document.getElementById("demo").innerHTML = text.includes("world");
</script>
</body>
</html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

startsWith() method check karta hai ki string kis value se shuru ho rahi hai. Agar string us specified value se start hoti hai → true. Agar nahi hoti → false.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Strings</h1>
<h2>The startsWith() Method</h2>
<p>Check if a string starts with "Hello":</p>
<p id="demo"></p>
<script>
let text = "Hello world, welcome to the universe.";
document.getElementById("demo").innerHTML = text.startsWith("Hello");
</script>
</body>
</html>
```

endsWith() method check karta hai ki string kis value pe end ho rahi hai. Agar string usi specified value se end hoti hai → true. Agar nahi hoti → false.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Strings</h2>
<p>Check if a string ends with "Doe":</p>
<p id="demo"></p>
<script>
let text = "John Doe";
document.getElementById("demo").innerHTML = text.endsWith("Doe");
</script>
</body>
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Numbers

JavaScript me sirf ek hi type ka number hota hai —
Alag-alag number types (jaise integer, float, double) separate nahi hote.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Numbers</h2>
<p>Extra large or extra small numbers can be written with scientific (exponent)
notation:</p>
<p id="demo"></p>
<script>
let x = 123e5;
let y = 123e-5;
document.getElementById("demo").innerHTML = x + "<br>" + y;
</script>
</body>
</html>
```

Basic Methods

Basic number methods can be used
on any number:

toString()
toExponential()
toFixed()
toPrecision()
valueOf()

Static Methods

Static methods can only be used **on**
Number:

Number.isFinite()
Number.isInteger()
Number.isNaN()
Number.isSafeInteger()
Number.parseInt()
Number.parseFloat()

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

The toString() Method

toString() method number ko string me convert karta hai.
Matlab jo number tha, usko text bana deta hai.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Number Methods</h2>
<p>The toString() method converts a number to a string.</p>
<p id="demo"></p>
<script>
let x = 123;
document.getElementById("demo").innerHTML =
  x.toString() + "<br>" +
  (123).toString() + "<br>" +
  (100 + 23).toString();
</script>
</body>
</html>
```

The toExponential() Method

toExponential() method number ko exponential (scientific) notation me convert karta hai

aur string ke form me return karta hai.

- Exponential notation: Jaise 1234 → 1.234e+3
- Parameter: Decimal ke baad kitne digits chahiye, ye specify kar sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Number Methods</h2>
<p>The toExponential() method returns a string, with the number rounded and written using exponential notation.</p>
<p>An optional parameter defines the number of digits behind the decimal point.</p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<p id="demo"></p>
<script>
let x = 9.656;
document.getElementById("demo").innerHTML =
  x.toExponential() + "<br>" +
  x.toExponential(2) + "<br>" +
  x.toExponential(4) + "<br>" +
  x.toExponential(6);
</script>
</body>
</html>
```

toFixed() ek JavaScript method hai jo numbers ke saath kaam karta hai.

- Purpose: Number ko string me convert karna aur specify kiye gaye decimal places tak round karna.
- Returns: string, number nahi.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Number Methods</h2>
<p>The toFixed() method rounds a number to a given number of digits.</p>
<p>For working with money, toFixed(2) is perfect.</p>
<p id="demo"></p>
<script>
let x = 9.656;
document.getElementById("demo").innerHTML =
  x.toFixed(0) + "<br>" +
  x.toFixed(2) + "<br>" +
  x.toFixed(4) + "<br>" +
  x.toFixed(6);
</script>
</body>
</html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

toFixed() ek JavaScript method hai jo number ko string me convert karta hai aur total significant digits specify karne deta hai.

- Purpose: Number ko exact total length ke saath display karna (not just decimal places).
- Returns: string, number nahi.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Number Methods</h2>
<p>The toPrecision() method returns a string, with a number written with a
specified length:</p>
<p id="demo"></p>
<script>
let x = 9.656;
document.getElementById("demo").innerHTML =
  x.toFixed() + "<br>" +
  x.toFixed(2) + "<br>" +
  x.toFixed(4) + "<br>" +
  x.toFixed(6);
</script>
</body>
</html>
```

Method	Description
Number()	Returns a number converted from its argument.
parseFloat()	Parses its argument and returns a floating point number
parseInt()	Parses its argument and returns a whole number

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Number() ek JavaScript function hai jo variables ko number me convert karta hai.

- Purpose: String, boolean, ya kisi other type ko number me badalna
- Returns: number (agar conversion possible ho), otherwise NaN

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Global Methods</h2>
<p>The Number() method converts variables to numbers:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
Number(true) + "<br>" +
Number(false) + "<br>" +
Number("10") + "<br>" +
Number(" 10") + "<br>" +
Number("10 ") + "<br>" +
Number(" 10 ") + "<br>" +
Number("10.33") + "<br>" +
Number("10,33") + "<br>" +
Number("10 33") + "<br>" +
Number("John");
</script>
</body>
</html>
```

Number() dates pe bhi kaam karta hai.

- Purpose: Date ko timestamp number me convert karna (milliseconds since 1 Jan 1970, UTC)
- Returns: number

```
<!DOCTYPE html>
<html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<body>
<h2>JavaScript Global Methods</h2>
<p>The Number() method can convert a date to a number:</p>
<p id="demo"></p>
<script>
let x = new Date("1970-01-01");
document.getElementById("demo").innerHTML = Number(x);
</script>
</body>
</html>
```

parseInt() ek JavaScript function hai jo string ko integer (whole number) me convert karta hai.

- Purpose: String me se first number extract karna
- Returns: number (integer)
- Special: Spaces ignore hote hain, aur sirf pehla valid number consider hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Global Functions</h2>
<h2>parseInt()</h2>
<p>The global JavaScript function parseInt() converts strings to numbers:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
parseInt("-10") + "<br>" +
parseInt("-10.33") + "<br>" +
parseInt("10") + "<br>" +
parseInt("10.33") + "<br>" +
parseInt("10 6") + "<br>" +
parseInt("10 years") + "<br>" +
parseInt("years 10");
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</script>  
</body>  
</html>
```

parseFloat() ek JavaScript function hai jo string ko floating-point number (decimal allowed) me convert karta hai.

- Purpose: String me se first valid number extract karna, decimal ke saath
- Returns: number
- Special: Spaces ignore hote hain, aur sirf pehla number consider hota hai

```
<!DOCTYPE html>  
<html>  
<body>  
<h2>JavaScript Global Methods</h2>  
<p>The parseFloat() method converts strings to numbers:</p>  
<p id="demo"></p>  
<script>  
document.getElementById("demo").innerHTML =  
  parseFloat("10") + "<br>" +  
  parseFloat("10.33") + "<br>" +  
  parseFloat("10 6") + "<br>" +  
  parseFloat("10 years") + "<br>" +  
  parseFloat("years 10");  
</script>  
</body>  
</html>
```

Number JavaScript ka built-in object hai jo numbers ke liye special methods provide karta hai. Ye methods numbers ko format ya convert karne me help karte hain.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Method	Description
Number.isInteger()	Returns true if the argument is an integer
Number.isNaN()	Returns true if the argument is NaN
Number.isFinite()	Returns true if the argument is not Infinity nor NaN
Number.isSafeInteger()	Returns true if the argument is a safe integer
Number.parseFloat()	Converts a string to a number
Number.parseInt()	Converts a string to a whole number

Number.isInteger() ek JavaScript method hai jo check karta hai ki given value ek integer hai ya nahi.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The isInteger() Method</h2>
<p>The isInteger() method returns true if the argument is an integer.</p>
<p>Otherwise it returns false.</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
Number.isInteger(10) + "<br>" + Number.isInteger(10.5);
</script>
</body>
</html>
```

Number.isFinite() ek JavaScript method hai jo check karta hai ki

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

given value finite number hai ya nahi.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The Number.isFinite() Method</h2>
<p>The isFinite() method returns true if the argument is not Infinity nor
Nan.</p>
<p id="demo"></p>
<script>
let result = Number.isFinite(123);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

Number.isNaN() ek JavaScript method hai jo check karta hai ki given value NaN (Not a Number) hai ya nahi.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The Number.isNaN() Method</h2>
<p>The isNaN() method returns true if the argument is NaN (Not a
Number).</p>
<p id="demo"></p>
<script>
let result = Number.isNaN(123);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Number.isSafeInteger()

Ye JavaScript method check karta hai ki given value safe integer hai ya nahi. Safe integer matlab: wo integer jo exactly represent ho sakta hai as a double-precision number (IEEE 754 standard).

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The isSafeInteger() Method</h2>
<p>The isSafeInteger() method returns true if the argument is a safe
integer.</p>
<p>Otherwise it returns false.</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
Number.isSafeInteger(10) + "<br>" +
Number.isSafeInteger(12345678901234567890);
</script>
</body>
</html>
```

Number.parseFloat()-Ye JavaScript method ek string ko number me convert karta hai (floating-point number).

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The parseFloat() Method</h2>
<p>Number.parseFloat() converts strings to numbers:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
Number.parseFloat("10") + "<br>" +
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
Number.parseFloat("10.33") + "<br>" +  
Number.parseFloat("10 20 30") + "<br>" +  
Number.parseFloat("10 years") + "<br>" +  
Number.parseFloat("years 10");  
</script>  
</body>  
</html>
```

Number.isInteger()-Ye JavaScript method check karta hai ki given value integer hai ya nahi.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Numbers</h1>  
<h2>The isInteger() Method</h2>  
<p>The isInteger() method returns true if the argument is an integer.</p>  
<p>Otherwise it returns false.</p>  
<p id="demo"></p>  
<script>  
document.getElementById("demo").innerHTML =  
Number.isInteger(10) + "<br>" + Number.isInteger(10.5);  
</script>  
</body>  
</html>
```

Number.isSafeInteger()

Ye JavaScript method check karta hai ki given value safe integer hai ya nahi. Safe integer matlab: wo integer jo exactly represent ho sakta hai as a double-precision number (IEEE 754 standard).

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Numbers</h1>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

<h2>The isSafeInteger() Method</h2>

<p>The isSafeInteger() method returns true if the argument is a safe integer.</p>

<p>Otherwise it returns false.</p>

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

<script>

document.getElementById("demo").innerHTML =

Number.isSafeInteger(10) + "
" +

Number.isSafeInteger(12345678901234567890);

</script>

</body>

</html>

Number Properties

Number Properties

Number.EPSILON

Number.MAX_VALUE

Number.MIN_VALUE

Number.MAX_SAFE_INTEGER

Number.MIN_SAFE_INTEGER

Number.POSITIVE_INFINITY

Number.NEGATIVE_INFINITY

Number.NaN

Number.EPSILON

Ye JavaScript property represent karta hai the smallest difference between 1 and the next representable floating-point number greater than 1.

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Numbers</h1>

<h2>The EPSILON Property</h2>

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

<p>The difference between 1 and the smallest floating point number greater than 1, in JavaScript is:</p>

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

<script>

let x = Number.EPSILON;

document.getElementById("demo").innerHTML = x;

</script>

</body>

</html>

Number.MAX_VALUE

Ye JavaScript constant represent karta hai sabse bada possible number jo JavaScript me store kiya ja sakta hai.

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Numbers</h1>

<h2>The MAX_VALUE Property</h2>

<p>The largest possible number in JavaScript is:</p>

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

<script>

let x = Number.MAX_VALUE;

document.getElementById("demo").innerHTML = x;

</script>

</body>

</html>

JavaScript Number.MIN_VALUE

Number.MIN_VALUE JavaScript ka ek built-in constant hai jo JavaScript mein represent kiya ja sakne wala sabse chhota *positive* number dikhata hai.

<!DOCTYPE html>

<html>

<body>

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<h1>JavaScript Numbers</h1>
<h2>The MIN_VALUE Property</h2>
<p>The smallest number possible in JavaScript is:</p>
<p id="demo"></p>
<script>
let x = Number.MIN_VALUE;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

JavaScript Number.MIN_SAFE_INTEGER

Number.MIN_SAFE_INTEGER JavaScript ka wo constant hai jo JavaScript mein safely represent kiya ja sakne wala sabse chhota (most negative) integer batata hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The MIN_SAFE_INTEGER Property</h2>
<p>The minimum safe integer in JavaScript is:</p>
<p id="demo"></p>
<script>
let x = Number.MIN_SAFE_INTEGER;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

JavaScript Number.MAX_SAFE_INTEGER

Number.MAX_SAFE_INTEGER JavaScript ka wo constant hai jo JavaScript mein safely represent kiya ja sakne wala sabse bada integer batata hai.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The MAX_SAFE_INTEGER Property</h2>
<p>The maximum safe integer in JavaScript is:</p>
<p id="demo"></p>
<script>
let x = Number.MAX_SAFE_INTEGER;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

JavaScript Number.POSITIVE_INFINITY

Number.POSITIVE_INFINITY JavaScript ka ek special numeric value hai jo positive infinity ko represent karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The POSITIVE_INFINITY Property</h2>
<p>POSITIVE_INFINITY is returned on overflow:</p>
<p id="demo"></p>
<script>
let x = 1 / 0;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

JavaScript Number.NEGATIVE_INFINITY

Number.NEGATIVE_INFINITY JavaScript ka ek special numeric value hai jo

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

negative infinity ko represent karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The NEGATIVE_INFINITY Property</h2>
<p></p>
<p id="demo"></p>
<script>
let x = Number.NEGATIVE_INFINITY;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

JavaScript NaN

NaN ka matlab hota hai Not a Number. JavaScript mein yeh ek special numeric value hai jo tab milti hai jab koi calculation ka result valid number nahi ban sakta.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Number Properties</h2>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Number.NaN;
</script>
</body>
</html>
```

JavaScript Number Reference

LEARNING HUB
SHAHABAD MARKANDA
 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Feature / Name	Meaning
constructor	Batata hai ki Number object kis function se bana
EPSILON	1 aur uske thoda-sa upar wale number ke beech ka difference
isFinite()	Check karta hai ki value finite (normal) number hai
isInteger()	Dekhta hai ki value integer hai ya nahi
isNaN()	Check karta hai ki value exactly NaN hai
isSafeInteger()	Check karta hai ki number safe integer range mein hai
MAX_SAFE_INTEGER	JS ka sabse bada safe integer ($2^{53} - 1$)
MIN_SAFE_INTEGER	JS ka sabse chhota safe integer ($-(2^{53} - 1)$)
MAX_VALUE	JS ka sabse bada representable number
MIN_VALUE	JS ka sabse chhota positive number (zero ke bohot paas)
NaN	"Not-a-Number" — invalid number result
NEGATIVE_INFINITY	Negative infinity (overflow ya divide by 0)
POSITIVE_INFINITY	Positive infinity (overflow ya divide by 0)
parseFloat()	String ko decimal number mein convert karta hai
parseInt()	String ko integer (whole number) mein convert karta hai
prototype	Object mein new properties/methods add karne deta hai
toExponential(x)	Number ko exponential/scientific form mein deta hai
toFixed(x)	Number ko decimal ke baad x digits tak format karta hai
toLocaleString()	Number ko local formatting rules ke saath string banata hai
toPrecision(x)	Number ko x total digits mein format karta hai
toString()	Number ko string bana deta hai (base specify kar sakte ho)
valueOf()	Number ka actual primitive (real) value deta hai

Constructor

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The constructor Property</h2>
<p>The constructor property returns the function that created the Number prototype:</p>
<p id="demo"></p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
let num = 134.5;
let text = num.constructor;
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

EPSILON

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The EPSILON Property</h2>
<p>The difference between 1 and the smallest floating point number greater than 1 is:</p>
<p id="demo"></p>
<script>
let x = Number.EPSILON;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

Number.isFinite()

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The Number.isFinite() Method</h2>
<p>Number.isFinite() returns true if the value is a finite, legal number:</p>
<p id="demo"></p>
<script>
let result = Number.isFinite(123);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

Number.isInteger()

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The Number.isInteger() Method</h2>
<p>Number.isInteger() returns true if the value is an integer of the datatype Number:</p>
<p id="demo"></p>
<p>Number.isInteger() is not supported in Internet Explorer 11 and earlier.</p>
<script>
let result =
"Is 123 an integer? " + Number.isInteger(123) + "<br>" +
"Is 123 an integer? " + Number.isInteger(-123) + "<br>" +
"Is '123' an integer? " + Number.isInteger('123');
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

The Number.isNaN()

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The Number.isNaN() Method</h2>
<p>Number.isNaN() returns true if a value is NaN and of datatype Number, otherwise false.</p>
<p id="demo"></p>
<p>Number.isNaN() is not supported in Internet Explorer.</p>
<script>
let result =
"Is 123 NaN? " + Number.isNaN(123) + "<br>" +
"Is -123 NaN? " + Number.isNaN(-1.23) + "<br>" +
"Is '123' NaN? " + Number.isNaN('123') + "<br>" +
"Is 0/0 NaN? " + Number.isNaN(0 / 0);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

The Number.isSafeInteger()

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The Number.isSafeInteger() Method</h2>
<p>Number.isSafeInteger() returns true if a value is a safe integer and of datatype
Number:</p>
<p id="demo"></p>
<p>Number.isSafeInteger() is not supported in Internet Explorer.</p>
<script>
let result =
"Is 123 a safe integer? " + Number.isSafeInteger(123) + "<br>" +
"Is +123 a safe integer? " + Number.isSafeInteger(-123) + "<br>" +
"Is '123' a safe integer? " + Number.isSafeInteger('123');
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

The MAX_SAFE_INTEGER

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The MAX_SAFE_INTEGER Property</h2>
<p>The maximum safe integer in JavaScript is:</p>
<p id="demo"></p>
<script>
let x = Number.MAX_SAFE_INTEGER;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

The MIN_SAFE_INTEGER

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<h2>The MIN_SAFE_INTEGER Property</h2>
<p>The minimum safe integer in JavaScript is:</p>
<p id="demo"></p>
<script>
let x = Number.MIN_SAFE_INTEGER;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

The MAX_VALUE

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The MAX_VALUE Property</h2>
<p>Number.MAX_VALUE returns the largest possible number in JavaScript:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Number.MAX_VALUE;
</script>
</body>
</html>
```

The MIN_VALUE

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The MIN_VALUE Property</h2>
<p>Number.MIN_VALUE returns the smallest possible number in JavaScript:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Number.MIN_VALUE;
</script>
</body>
</html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Nan

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The Nan Property</h2>
<p>NaN returns NaN ("Not a Number");</p>
<p id="demo"></p>
<script>
let x = NaN;
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

NEGATIVE_INFINITY

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The NEGATIVE_INFINITY Property</h2>
<p>Number.NEGATIVE_INFINITY returns negative Infinity:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Number.NEGATIVE_INFINITY;
</script>
</body>
</html>
```

POSITIVE_INFINITY

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The POSITIVE_INFINITY Property</h2>
<p>Number.POSITIVE_INFINITY returns Infinity:</p>
<p id="demo"></p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
document.getElementById("demo").innerHTML = Number.POSITIVE_INFINITY;
</script>
</body>
</html>
```

parseFloat()

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Number Methods</h1>
<h2>The parseFloat() Method</h2>
<p>Parse a string and returns the first number:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
Number.parseFloat(10) + "<br>" +
Number.parseFloat("10") + "<br>" +
Number.parseFloat("10.33") + "<br>" +
Number.parseFloat("34 45 66") + "<br>" +
Number.parseFloat("He was 40");
</script>
</body>
</html>
```

Number.parseInt()

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Number Methods</h1>
<h2>The Number.parseInt() Method</h2>
<p>Parse a string and returns the first integer:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
Number.parseInt("10") + "<br>" +
Number.parseInt("10.00") + "<br>" +
Number.parseInt("10.33") + "<br>" +
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
Number.parseInt("34 45 66") + "<br>" +  
Number.parseInt(" 60 ") + "<br>" +  
Number.parseInt("40 years") + "<br>" +  
Number.parseInt("He was 40");  
</script>  
</body>  
</html>
```

Prototype

```
<!DOCTYPE html>  
<html>  
<body>  
<h2>JavaScript Numbers</h2>  
<p>Call myMethod() to display a number's half value.:</p>  
<p id="demo"></p>  
<script>  
Number.prototype.myMethod = function() {  
  return this.valueOf() / 2;  
};  
let n = 55;  
document.getElementById("demo").innerHTML = n.myMethod();  
</script>  
</body>  
</html>
```

toExponential()

```
<!DOCTYPE html>  
<html>  
<body>  
<h2>JavaScript Numbers</h2>  
<p>The toExponential() method converts a number into an exponential notation:</p>  
<p id="demo"></p>  
<script>  
let num = 5.56789;  
let n = num.toExponential();  
document.getElementById("demo").innerHTML = n;  
</script>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</script>  
</body>  
</html>
```

toFixed()

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Numbers</h1>  
<h2>The toFixed() Method</h2>  
<p>toFixed() converts a number to a string, rounded to a specified number of decimals:</p>  
<p id="demo"></p>  
<script>  
let num = 5.56789;  
let n = num.toFixed();  
document.getElementById("demo").innerHTML = n;  
</script>  
</body>  
</html>
```

toLocaleString()

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Numbers</h1>  
<h2>The toLocaleString() Method</h2>  
<p>toLocaleString() returns a number as a string, using local language format.</p>  
<p id="demo"></p>  
<script>  
let num = 1000000;  
let text = num.toLocaleString();  
document.getElementById("demo").innerHTML = text;  
</script>  
</body>  
</html>
```

toPrecision()

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The toPrecision() Method</h2>
<p>toPrecision() formats a number to a specified length:</p>
<p id="demo"></p>
<script>
let num = 13.3714;
document.getElementById("demo").innerHTML = num.toPrecision(2);
</script>
</body>
</html>
```

toString()

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The toString() Method</h2>
<p>toString() returns a number as a string:</p>
<p id="demo"></p>
<script>
let num = 15;
let text = num.toString();
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

valueOf()

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Numbers</h1>
<h2>The valueOf() Method</h2>
<p>valueOf() returns the primitive value of a number:</p>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<p id="demo"></p>
<script>
let num = 15;
let n = num.valueOf();
document.getElementById("demo").innerHTML = n;
</script>
</body>
</html>
```

JavaScript Data Type

JavaScript mein data types do main categories mein aate hain:

1. Primitive Types (single value, immutable)
2. Non-Primitive / Reference Types (objects, arrays, functions etc.)

Primitive Data Types

Type	Example	Description
Number	10, 3.14, -5	Numbers, integers aur floating-point dono
BigInt	9007199254740992n	Bahut bada integer, safe integer range se bada
String	"Hello", 'JS'	Text ya characters ka data
Boolean	true, false	True/False value
Undefined	let a; console.log(a) → undefined	Variable declare hua hai, value assign nahi
Null	let a = null;	Empty value / intentionally no value
Symbol	Symbol('id')	Unique identifier, mostly object keys ke liye

Non-Primitive / Reference Types

Type	Example	Description
Object	{name: "Ali", age: 20}	Key-value pair ka collection
Array	[1, 2, 3]	Ordered collection of values
Function	function hello() {}	Executable code block

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Type	Example	Description
Date	new Date()	Date aur time ka data
RegExp	/abc/	Regular expressions for pattern matching

Special Values

Name	Meaning
NaN	Not-a-Number, invalid calculation
Infinity / -Infinity	Positive/Negative infinity
null	Intentional empty value
undefined	Variable exists but no value assigned

Functions

Functions programming ka fundamental building block hote hain.
Ye reusable blocks of code hote hain jo koi specific task perform karte hain.

Execution

Function tab run hota hai jab usko call ya invoke kiya jaata hai.

```
function functionName(parameters) {  
    // Code to perform task  
    // Optional return value  
}
```

Key Points

Point	Explanation
Reusable	Ek function ko baar-baar call kar sakte ho, code repeat nahi karna padta
Parameters	Input values jo function ke kaam ke liye diye jaate hain
Return Value	Function output bhi de sakta hai using return keyword

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Point	Explanation
Execution	Function tab run hota hai jab explicitly call kiya jaata hai

Function Invocation ()

Function invocation ka matlab hai: function ka code tab run hota hai jab function ko call kiya jaaye.

JavaScript mein function invoke karne ke liye parentheses () use hote hain.

- () operator hi function ko invoke karta hai
- Function tab tak run nahi hota jab tak explicitly call ya event se invoke na ho
- Parentheses ke andar arguments/parameters bhi pass kiye ja sakte hain

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Functions</h1>
<p>Invoke (call) a function that converts from Fahrenheit to Celsius:</p>
<p id="demo"></p>
<script>
function toCelsius(f) {
  return (5/9) * (f-32);
}
let value = toCelsius(77);
document.getElementById("demo").innerHTML = value;
</script>
</body>
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Parameters

- Definition: Variables jo function declaration mein define kiye jaate hain.
- Purpose: Function ke andar input values ko hold karna.
- Position: Function ke parentheses () ke andar.

Arguments

- Definition: Real values jo function call ke waqt pass ki jaati hain.
- Purpose: Function ke parameters ko actual data provide karna.
- Position: Function call ke parentheses () ke andar.

JavaScript Function Parameter Rules

No Data Type Specified

- JavaScript function parameters ka data type define nahi karta
- Matlab aap koi bhi type ka value pass kar sakte ho: Number, String, Boolean, Object, etc.

No Type Checking

- JavaScript arguments ki type check nahi karta
- Agar aap wrong type pass karte ho, function still run ho jaata hai
- Type errors sirf runtime pe aate hain agar operation incompatible ho

No Check on Number of Arguments

- JavaScript arguments ki count check nahi karta
- Agar kam arguments pass kiye → missing parameters = undefined
- Agar zyada arguments pass kiye → extra arguments ignore ho jaate hain (ya access kiye jaa sakte hain arguments object se)

JavaScript Default Parameters

- Agar function kam arguments ke saath call hota hai, missing parameters ka value undefined ho jaata hai.
- JavaScript (ES6+) mein aap default value assign kar sakte ho parameters ko.
- Agar function call ke waqt argument missing ho, default value use hoti hai.

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Functions</h1>
<h2>Default Parameter Values</h2>
<p>If y is not passed or undefined, then y = 10:</p>
<p id="demo"></p>
<script>
function myFunction(x, y = 10) {
  return x + y;
}
document.getElementById("demo").innerHTML = myFunction(5);
</script>
</body>
</html>
```

Function Rest Parameter

The rest parameter (...) allows a function to treat an indefinite number of arguments as an array:

- Iska use tab hota hai jab aapko unknown number of arguments handle karne ho.

```
function functionName(...restParam) {
  // restParam ek array ki tarah behave karta hai
}
```

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Functions</h1>
<h2>The Rest Parameter</h2>
<p>The rest parameter (...) allows a function to treat an indefinite number of arguments as an array:</p>
<p id="demo"></p>
<script>
function sum(...args) {
  let sum = 0;
  for (let arg of args) sum += arg;
}
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
return sum;
}
let x = sum(4, 9, 16, 25, 29, 100, 66, 77);
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

JavaScript Arguments Object

- JavaScript functions ke paas built-in object hota hai: arguments
- Ye object all arguments ko array-like format mein store karta hai
- Useful jab aapko unknown number of arguments process karne ho
- arguments array-like hota hai, par actual array nahi
- Isme length hota hai aur index se access hota hai
- Array methods direct nahi chalte (unless convert to array)
- Available inside every function (arrow functions ke liye nahi)
- Helps in processing variable number of arguments

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Functions</h2>
<p>Sum of all arguments:</p>
<p id="demo"></p>
<script>
function sumAll() {
  let sum = 0;
  for(let i = 0; i < arguments.length; i++) {
    sum += arguments[i];
  }
  return sum;
}
document.getElementById("demo").innerHTML = sumAll(1, 123, 500, 115, 44, 88);
</script>
</body>
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Arguments Are Passed by Value

- JavaScript mein function arguments pass by value hote hain.
- Matlab: function ko actual value milti hai, original variable ka memory location nahi.
- Agar function ke andar argument change ho jaaye → original variable par effect nahi padta.

Reference Types (Objects)

- Primitive types → pass by value
- Objects / Arrays → pass by reference (function changes reflect outside)

Objects Are Passed by Reference

- JavaScript mein objects ke references values hote hain.
- Matlab: Function ko object ka reference milta hai, na ki object ki copy.
- Agar function ke andar object property change hoti hai → original object par effect hota hai
- Objects and Arrays → passed by reference
- Function ke andar property add, delete, update → original object par effect
- Agar object completely replace kar diya (reassigned) → bahar effect nahi hota

Local Variables

- Local variable wo variable hota hai jo function ke andar declare kiya jaata hai.
- Iska scope sirf function ke andar hota hai → bahar access nahi kar sakte.
- Local variables function ke khatam hone ke baad destroy ho jaate hain.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Functions</h1>
<p>Outside myFunction() carName is undefined.</p>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
let text = "Outside: " + typeof carName;
document.getElementById("demo1").innerHTML = text;
function myFunction() {
  let carName = "Volvo";
  let text = "Inside: " + typeof carName + " " + carName;
  document.getElementById("demo2").innerHTML = text;
myFunction();
</script>
</body>
</html>
```

Functions Used as Variables

In programming, functions are not just things you call—they can also be treated like variables. Matlab, aap ek function ko ek variable ki tarah store kar sakte ho, aur phir use assign ya pass kar sakte ho, jaise normal data.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Functions</h1>
<p>Using a function as a variable:</p>
<p id="demo"></p>
<script>
let text = "The temperature is " + toCelsius(77) + " Celsius.";
document.getElementById("demo").innerHTML = text;
function toCelsius(fahrenheit) {
  return (5/9) * (fahrenheit-32);
}
</script>
</body>
</html>
```

Arrow Function Parameters

- Arrow function ek concise syntax hai function likhne ka, introduced in ES6.
- Use => (fat-arrow) likhne ko:

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- Agar ek hi expression ho function body mein, toh implicit return ho sakta hai (matlab return likhne ki zaroorat nahi):
- Agar body mein multiple statements hain, toh curly braces chahiye aur return likhna padega:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrow Functions</h1>
<p id="demo"></p>
<script>
// myFunction computes the product of x and y
let myFunction = (x, y) => { return x * y };
let result = myFunction(5, 4);
document.getElementById("demo").innerHTML = "The product is: " + result;
</script>
</body>
</html>
```

Browser Support

`= () => {}` is an [ES6 feature](#).

ES6 is fully supported in all modern browsers since June 2017:

				
Chrome 51	Edge 15	Firefox 54	Safari 10	Opera 38
May 2016	Apr 2017	Jun 2017	Sep 2016	Jun 2016

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Object

Ek object ek aisa variable hota hai jo bohot saare variables ko hold kar sakta hai.

Objects key-value pairs ka collection hote hain, jahan har key (jise property name bhi kehte hain) ka ek value hota hai.

Objects kisi bhi cheez ko describe kar sakte hain, jaise houses, cars, people, animals, ya koi bhi subject.

Car Object	Car Properties	Car Methods
	car.name = Fiat	car.start()
	car.model = 500	car.drive()
	car.weight = 850kg	car.brake()
	car.color = white	car.stop()

Alag-alag cars ke same properties hote hain, lekin property values har car mein alag ho sakti hain.

Alag-alag cars ke same methods hote hain, lekin ye methods alag-alag time par perform kiye ja sakte hain.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Creating an Object</h2>
<p id="demo"></p>
<script>
// Create an Object:
const car = {type:"Fiat", model:"500", color:"white"};
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
// Display Data from the Object:
document.getElementById("demo").innerHTML =
"The car type is " + car.type;
</script>
</body>
</html>
```

How to Create a JavaScript Object

JavaScript Object kaise banaye

Ek object literal ek simple aur short tareeka hai object create karne ka.

Object literal ek list hoti hai key : value pairs ki, jo curly braces { } ke andar likhi jaati hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>Creating JavaScript Objects</h1>
<h2>Using an Object Literal</h2>
<p id="demo"></p>
<script>
// Create an Object:
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};
// Display Data from the Object:
document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years old.";
</script>
</body>
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Using the new Keyword

JavaScript me new keyword ka use ek naya object create karne ke liye hota hai, jo ek constructor function se banta hai.

Jab hum new use karte hain:

- Ek empty object banta hai.
- Constructor function call hoti hai aur this us new object ko refer karta hai.
- Function ke andar jo bhi properties aur methods define hoti hain, wo us naye object me add ho jaati hain.
- Function automatically naya object return karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>Creating JavaScript Objects</h1>
<h2>Using the new Keyword</h2>
<p id="demo"></p>
<script>
// Create an Object
const person = new Object({
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
});
// Display Object Content
document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years old.";
</script>
</body>
</html>
```

Object Constructor Functions

Kabhi-kabhi hume same type ke bohot saare objects create karne padte hain.

Aise objects create karne ke liye hum object constructor function ka use karte hain.

Good practice ye hai ki constructor function ka naam hamesha capital letter se start ho.

```
<!DOCTYPE html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Object Constructors</h2>
<p id="demo"></p>
<script>
// Constructor Function for Person objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}
// Create a Person object
const myFather = new Person("John", "Doe", 50, "blue");
// Display age
document.getElementById("demo").innerHTML =
"My father is " + myFather.age + ".";
</script>
</body>
</html>
```

JavaScript Object.create()

Object.create() method ek naya object banata hai kisi existing object se.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.create() Method</h2>
<p>Create a new object from the prototype of another project.</p>
<p id="demo"></p>
<script>
// Create an Object:
const person = {
  firstName: "John",
  lastName: "Doe",
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
language: "EN"
};
// Create new Object
const man = Object.create(person);
man.firstName = "Peter";
// Display Properties
document.getElementById("demo").innerHTML = person.firstName + " and " +
man.firstName;
</script>
</body>
</html>
```

JavaScript Object.fromEntries()

ES2019 me JavaScript me Object.fromEntries() method add hua. fromEntries() method ek object banata hai iterable key/value pairs se.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.fromEntries() Method</h2>
<p>The number of pears are:</p>
<p id="demo"></p>
<script>
const fruits = [
  ["apples", 300],
  ["pears", 900],
  ["bananas", 500]
];
const myObj = Object.fromEntries(fruits);
document.getElementById("demo").innerHTML = myObj.pears;
</script>
</body>
</html>
```

JavaScript Object.assign()

The Object.assign() method copies properties from one or more source objects to a target object.

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.assign() Method</h2>
<p id="demo"></p>
<script>
// Create Target Object
const person1 = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};
// Create Source Object
const person2 = {firstName: "Anne",lastName: "Smith"};
// Assign Source to Target
Object.assign(person1, person2);
// Display Target
let text = Object.entries(person1);
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

In JavaScript, Objects are King.

Agar aap Objects samajh jao, to aap JavaScript samajh jaoge.
JavaScript me, almost “everything” ek object hai.

- Objects bhi objects hain
- Maths bhi objects hain
- Functions bhi objects hain
- Dates bhi objects hain
- Arrays bhi objects hain
- Maps bhi objects hain
- Sets bhi objects hain

JavaScript Primitives

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- A primitive data type is data type that can only store a single primitive value.
- JavaScript defines 7 types of primitive data types:

Type	Example value
string	"Hello"
number	3.14
boolean	true
bigint	12345678901234
null	null
undefined	undefined
symbol	symbol

Immutable

Primitive values immutable hote hain (ye hardcoded hote hain aur change nahi kiye ja sakte).
Jaise: agar $x = 3.14$, to aap x ka value change kar sakte ho, lekin 3.14 ka value change nahi kar sakte.

Value	Type	Comment
"Hello"	string	"Hello" is always "Hello"
3.14	number	3.14 is always 3.14
true	boolean	true is always true
false	boolean	false is always false
null	null	null is always null
undefined	undefined	undefined is always undefined

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

JavaScript Objects are Mutable

Objects mutable hote hain: ye reference se address hote hain, value se nahi.

Agar person ek object hai, to ye statement person ka copy create nahi karegi:

```
const x = person;
```

Yahaan, x object person ka copy nahi hai, balki x = person hai.

x aur person same memory address share karte hain.

Iska matlab: agar aap x me koi change karte ho, to person me bhi wahi change ho jayega.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Objects are Mutable</h2>
<p>Changes to a copy of an object will also change the original object:</p>
<p id="demo"></p>
<script>
// Create an Object
const person = {
  firstName: "John",
  lastName: "Doe",
  age:50,
  eyeColor: "blue"
};
// Try to create a Copy
const x = person;
// This will change age in person !!!
x.age = 10
document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years old.";
</script>
</body>
</html>
```

Object Constructor Functions

Kabhi-kabhi hume same type ke bohot saare objects create karne padte hain.

Aise objects create karne ke liye hum object constructor function ka use karte hain.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

Good practice ye hai ki constructor function ka naam hamesha capital letter se start ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Object Constructors</h2>
<p id="demo"></p>
<script>
// Constructor Function for Person objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}
// Create a Person object
const myFather = new Person("John", "Doe", 50, "blue");
// Display age
document.getElementById("demo").innerHTML =
"My father is " + myFather.age + ".";
</script>
</body>
</html>
```

Property Default Values

Jo value kisi property ko di jaati hai, wo default value ban jaati hai har object ke liye jo constructor se create hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Object Constructors</h2>
<p id="demo"></p>
<script>
// Constructor function for Person objects
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
  this.nationality = "English";
}
// Create 2 Person objects
const myFather = new Person("John", "Doe", 50, "blue");
const myMother = new Person("Sally", "Rally", 48, "green");
// Display nationality
document.getElementById("demo").innerHTML =
"My father is " + myFather.nationality + ". My mother is " + myMother.nationality;
</script>
</body>
</html>
```

Constructor Function Methods

Ek constructor function me methods bhi ho sakte hain.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Object Constructors</h2>
<p id="demo"></p>
<script>
// Constructor Function for Person objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
  this.fullName = function() {
    return this.firstName + " " + this.lastName
  };
};
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE:** <https://learninghubshahabad.in>

```
}  
// Create a Person object  
const myFather = new Person("John", "Doe", 50, "blue");  
// Display full name  
document.getElementById("demo").innerHTML =  
"My father is " + myFather.fullName();  
</script>  
</body>  
</html>
```

Adding a Method to an Object

Ek already created object me method add karna kaafi easy hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Objects</h1>  
<h2>Object Constructors</h2>  
<p id="demo"></p>  
<script>  
// Constructor function for Person Objects  
function Person(first, last, age, eye) {  
  this.firstName = first;  
  this.lastName = last;  
  this.age = age;  
  this.eyeColor = eye;  
}  
// Create 2 Person objects  
const myFather = new Person("John", "Doe", 50, "blue");  
const myMother = new Person("Sally", "Rally", 48, "green");  
// Add a Name Method  
myMother.changeName = function (name) {  
  this.lastName = name;  
}  
// Change Name  
myMother.changeName("Doe");  
// Display fullName  
document.getElementById("demo").innerHTML =  
"My mother's last name is " + myMother.lastName;
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</script>  
</body>  
</html>
```

Adding a Property to an Object

JavaScript me kisi object me naya property add karna bahut easy hota hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Objects</h1>  
<h2>Object Constructors</h2>  
<p id="demo"></p>  
<script>  
// Constructor function for Person objects  
function Person(first, last, age, eye) {  
  this.firstName = first;  
  this.lastName = last;  
  this.age = age;  
  this.eyeColor = eye;  
}  
// Create 2 Person objects  
const myFather = new Person("John", "Doe", 50, "blue");  
const myMother = new Person("Sally", "Rally", 48, "green");  
// Add nationality to first object  
myFather.nationality = "English";  
// Display nationality  
document.getElementById("demo").innerHTML =  
"My father is " + myFather.nationality;  
</script>  
</body>  
</html>
```

Adding a Property to a Constructor

Aap directly constructor function me baad me naya property add nahi kar sakte jo already ban chuke objects me automatically appear ho jaye.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Adding a Property to a Constructor</h2>
<p id="demo"></p>
<script>
// Constructor function for Person Objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}
// Create 2 Person Objects
const myFather = new Person("John", "Doe", 50, "blue");
const myMother = new Person("Sally", "Rally", 48, "green");

// Add a new Property
Person.prototype.nationality = "English";
// Display new Property
document.getElementById("demo").innerHTML =
"The nationality of my father is " + myFather.nationality;
</script>
</body>
</html>
```

What is this?

JavaScript me this ek special keyword hai jo kis object ko refer kar raha hai, yeh batata hai. Lekin this ka meaning situation ke hisaab se change hota rehta hai.

Situation	Meaning
Alone	Global object
Simple function	Global object
Strict mode function	undefined
Object method	Usi object ko

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Situation	Meaning
Event handler	HTML element (Button, div, etc.)
call/apply/bind	Jo object aap manually set karo

this in a Method

Jab this ko object ke method ke andar use karte ho to this us object ko refer karta hai jiska wo method hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>The JavaScript this Keyword</h1>
<p>In this example, this refers to the person object.</p>
<p>Because fullName is a method of the person object.</p>
<p id="demo"></p>
<script>
// Create an object:
const person = {
  firstName: "vivek",
  lastName: "sharma",
  id: 5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};
// Display data from the object:
document.getElementById("demo").innerHTML = person.fullName();
</script>
</body>
</html>
```

Object Method Binding

JavaScript me kabhi-kabhi aap kisi object ka method (function) **dusri jagah pass** kar dete ho—jaise event me, timeout me, ya variable me store karke.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>The JavaScript <i>this</i> Keyword</h1>
<p>In this example, <b>this</b> refers to the <b>person object</b>.</p>
<p id="demo"></p>
<script>
// Create an object:
const person = {
  firstName : "John",
  lastName  : "Doe",
  id        : 5566,
  myFunction : function() {
    return this;
  }
};
// Display data from the object:
document.getElementById("demo").innerHTML = person.myFunction();
</script>
</body>
</html>
```

Explicit Function Binding

JavaScript me call() aur apply() predefined methods hain jo aapko yeh allow karte hain ki: Kisi object ka method kisi doosre object ke saath run kara sako (yaani this ko manually set kar sakte ho). Isko hi explicit binding kehte hai

```
<!DOCTYPE html>
<html>
<body>
<h1>The JavaScript <i>this</i> Keyword</h1>
<p>In this example <strong>this</strong> refers to person2, even if it is a method of
person1:</p>
<p id="demo"></p>
<script>
const person1 = {
  fullName: function() {
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
return this.firstName + " " + this.lastName;
}
}
const person2 = {
  firstName:"John",
  lastName: "Doe",
}
let x = person1.fullName.call(person2);
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

Function Borrowing

JavaScript me function borrowing ka matlab hai. Ek object dusre object ke method ko use kar sakta hai chahe us object ke paas wo method ho hi na! Yeh hota hai bind(), call(), ya apply() se.

bind() se Function Borrowing

bind() method se aap **kisi method ka this dusre object par set kar dete ho.** Isse method dusre object ke liye bhi kaam karne lagta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Function bind()</h1>
<p>This example creates 2 objects (person and member).</p>
<p>The member object borrows the fullname method from person:</p>
<p id="demo"></p>
<script>
const person = {
  firstName:"John",
  lastName: "Doe",
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
}
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
const member = {
  firstName: "Hege",
  lastName: "Nilsen",
}
let fullName = person.fullName.bind(member);
document.getElementById("demo").innerHTML = fullName();
</script>
</body>
</html>
```

this Binding Order

Order	Situation	Why / Because
1	bind()	this function ko bind() se explicitly set kiya gaya hai → sabse strong binding
2	apply()	this ko apply() se function call karte waqt set kiya gaya
2	call()	call() bhi apply() jaisa hi hai → same priority
3	Object Method	Function object ka method hai → this object ko refer karta hai
4	Global Scope	Function ko normal tarike se global me call kara → this = global object (ya strict mode me undefined)

JavaScript Global This

In JavaScript, globalThis is a special built-in object that provides a standard way to access the global object, regardless of the environment your code runs in - whether it is a browser, Node.js, a Web Worker, or another JS runtime.

The global object is the top-level object in a JavaScript environment:

- In browsers, it is window.
- In Node.js, it is global.
- In Web Workers, it is self.

Each environment used to have its own name for this object. This caused compatibility issues for code meant to run everywhere.

Global. This was introduced in [ECMAScript 2020 \(ES11\)](#) to solve that problem. It is a unified reference to the global object, no matter the environmen

```
<!DOCTYPE html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<html>
<body>
<h1>JavaScript this</h1>
<h2>The globalThis Object</h2>
<p id="demo"></p>
<script>
let result = globalThis === window;
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

Object Destructuring

JavaScript me object destructuring ek shortcut hai jo object ke properties ko directly variables me assign kar deta hai.

Isse aapko baar-baar obj.property likhne ki zarurat nahi padti.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Object Destructuring</h2>
<p id="demo"></p>
<script>
// Create an Object
const person = {
  firstName: "sahil",
  lastName: "sharma",
  age: 50
};
// Destructuring
let {lastName, firstName} = person;
// Display Primitive Values
document.getElementById("demo").innerHTML = firstName + " " + lastName;
</script>
</body>
</html>
```

Object Default Values

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

JavaScript me kabhi-kabhi object me kuch properties missing hoti hain.
Aise cases me hum default values set kar sakte hain destructuring ke time.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Object Destructuring</h2>
<p id="demo"></p>
<script>
// Create an Object
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 50
};

// Destructuring
let {firstName, lastName, country = "US"} = person;
// Display Primitive Values
document.getElementById("demo").innerHTML =
firstName + " " + lastName + " " + country;
</script>
</body>
</html>
```

Object Property Alias

JavaScript me object destructuring ke time, aap object ke property ka naam change karke variable me store kar sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>Object Destructuring</h2>
<p id="demo"></p>
<script>
// Create an Object
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 50
};
// Destructuring
let {lastName : name} = person;
// Display Primitive Values
document.getElementById("demo").innerHTML = name;
</script>
</body>
</html>
```

String Destructuring

JavaScript me destructuring sirf objects aur arrays ke liye nahi, balki strings ke characters ko variables me unpack karne ke liye bhi use hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>String Destructuring</h2>
<p id="demo"></p>
<script>
let name = "W3Schools";
// Destructuring
let [a1, a2, a3, a4, a5] = name;
// Display Value
document.getElementById("demo").innerHTML = a1;
</script>
</body>
</html>
```

Array Destructuring

We can pick up array variables into our own variables:

```
<!DOCTYPE html>
<html>
<body>
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<h1>JavaScript Arrays</h1>
<h2>Array Destructuring</h2>
<p>Move array variables into single variables:</p>
<p id="demo"></p>
<script>
// Create an Array
const fruits = ["Bananas", "Oranges", "Apples", "Mangos"];
// Destructuring
let [fruit1, fruit2] = fruits;
// Display Primitive Values
document.getElementById("demo").innerHTML = fruit1 + " " + fruit2;
</script>
</body>
</html>
```

Skipping Array Values

We can skip array values using two or more commas:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Array Destructuring</h2>
<p>We can skip array values using two or more commas:</p>
<p id="demo"></p>
<script>
// Create an Array
const fruits = ["Bananas", "Oranges", "Apples", "Mangos"];
// Destructuring
let [fruit1,,fruit2] = fruits;
// Display Primitive Values
document.getElementById("demo").innerHTML = fruit1 + " " + fruit2;
</script>
</body>
</html>
```

Array Position Values

JavaScript me array destructuring se aap specific index ke values ko directly variables me assign kar sakte ho.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Array Destructuring</h2>
<p>We can pick up values from specific index locations of an array:</p>
<p id="demo"></p>
<script>
// Create an Array
const fruits = ["Bananas", "Oranges", "Apples", "Mangos"];

const {[0]:fruit1 ,[1]:fruit2} = fruits;
// Display Primitive Values
document.getElementById("demo").innerHTML = fruit1 + " " + fruit2;
</script>
</body>
</html>
```

The Rest Property

JavaScript me rest property (...) ka use karke aap destructuring me bachi hui values ko ek array me store kar sakte ho.

Yeh objects, arrays, aur strings ke liye kaam karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Array Destructuring</h2>
<p id="demo"></p>
<script>
// Create an Array
const numbers = [10, 20, 30, 40, 50, 60, 70]
// Destructuring
const [a,b, ...rest] = numbers;
// Display the Values
document.getElementById("demo").innerHTML =
"<p>a is " + a +
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
"<p>b is " + b +  
"<p>the rest is " + rest;  
</script>  
</body>  
</html>
```

Destructuring Maps

JavaScript me Map ek special object hota hai jo key-value pairs store karta hai. Map ko destructure karne ke liye hum for...of loop aur array destructuring ka use karte hain, kyunki Map iterable hota hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Maps</h1>  
<h2>Map Destructuring</h2>  
<p>Move array variables into single variables:</p>  
<p id="demo"></p>  
<script>  
// Create a Map  
const fruits = new Map([  
  ["apples", 500],  
  ["bananas", 300],  
  ["oranges", 200]  
]);  
// Destructuring  
let text = "";  
for (const [key, value] of fruits) {  
  text += "<p>" + key + " is " + value;  
}  
// Display the Values  
document.getElementById("demo").innerHTML = text;  
</script>  
</body>  
</html>
```

Swapping Variables with Destructuring

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Pehle hum variables swap karte the temporary variable use karke:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Variables</h1>
<h2>Swapping Variables</h2>
<p id="demo"></p>
<script>
let firstName = "John";
let lastName = "Doe";
// Destructing
[firstName, lastName] = [lastName, firstName];
// Display the Values
document.getElementById("demo").innerHTML = firstName + " " + lastName;
</script>
</body>
</html>
```

Objects Prototype

JavaScript me har object ke paas ek prototype hota hai.

Prototype ek parent object hai jahan se aap properties aur methods inherit kar sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<p>Using an object constructor:</p>
<p id="demo"></p>
<script>
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
  this.nationality = "English";
}
const myFather = new Person("John", "Doe", 50, "blue");
const myMother = new Person("Sally", "Rally", 48, "green");
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML =  
"The nationality of my father is " + myFather.nationality + ". The nationality of my mother is  
" + myMother.nationality;  
</script>  
</body>  
</html>
```

Prototype Inheritance

All JavaScript objects inherit properties and methods from a prototype:

- Date objects inherit from Date.prototype
- Array objects inherit from Array.prototype
- Person objects inherit from Person.prototype

The Object.prototype is on the top of the prototype inheritance chain:

Date objects, Array objects, and Person objects inherit from Object.prototype.

```
<!DOCTYPE html>  
<html>  
<body>  
<h2>JavaScript Objects</h2>  
<p id="demo"></p>  
<script>  
function Person(first, last, age, eye) {  
  this.firstName = first;  
  this.lastName = last;  
  this.age = age;  
  this.eyeColor = eye;  
Person.prototype.name = function() {  
  return this.firstName + " " + this.lastName  
};  
const myFather = new Person("John", "Doe", 50, "blue");  
document.getElementById("demo").innerHTML =  
"My father is " + myFather.name();  
</script>  
</body>  
</html>
```

Object Iteration

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

JavaScript me objects key-value pairs store karte hain.

Object ko iterate karne ke liye directly for...of loop nahi use hota (kyunki objects iterable nahi hote).

Instead, for...in loop, Object.keys(), Object.values(), aur Object.entries() use karte hain.

Object.assign()

Object.assign() method ka use karke aap ek ya zyada source objects ke properties ko ek target object me copy kar sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.assign() Method</h2>
<p id="demo"></p>
<script>
// Create Target Object
const person1 = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};
// Create Source Object
const person2 = {firstName: "Anne",lastName: "Smith"};
// Assign Source to Target
Object.assign(person1, person2)
// Display Target
let text = Object.entries(person1);
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Object.entries()

Object.entries() method ek object ke key/value pairs ko array me convert karta hai.

- Har element → [key, value] array

```
<!DOCTYPE html>
<html>
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.entries() Method</h2>
<p>Object.entries() makes it simple to convert an Object to a Map:</p>
<p id="demo"></p>
<script>
const fruits = {Bananas:300, Oranges:200, Apples:500};
const myMap = new Map(Object.entries(fruits));
document.getElementById("demo").innerHTML = myMap;
</script>
</body>
</html>
```

Object.fromEntries()

Object.fromEntries() method ek array of key/value pairs ko object me convert karta hai.

- Basically, ye Object.entries() ka reverse hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.fromEntries() Method</h2>
<p>The number of pears are:</p>
<p id="demo"></p>
<script>
const fruits = [
  ["apples", 300],
  ["pears", 900],
  ["bananas", 500]
];
const myObj = Object.fromEntries(fruits);
document.getElementById("demo").innerHTML = myObj.pears;
</script>
</body>
</html>
```

Object.values()

Object.values() method object ke sirf values ko ek array me return karta hai.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.values() Method</h2>
<p>Object.values() returns an array of values from an object:</p>
<p id="demo"></p>
<script>
const person = {
  firstName : "John",
  lastName  : "Doe",
  age       : 50,
  eyeColor  : "blue"
};

let text = Object.values(person)
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html><!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.values() Method</h2>
<p>Object.values() returns an array of values from an object:</p>
<p id="demo"></p>
<script>
const person = {
  firstName : "John",
  lastName  : "Doe",
  age       : 50,
  eyeColor  : "blue"
};

let text = Object.values(person)
document.getElementById("demo").innerHTML = text;
</script>
</body>
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

</html>

Object.groupBy()

Object.groupBy() is a new method in ES2024 that lets you group items of an object based on a function you provide.

- It does not change the original object.
- It returns a new object where the keys are the group names (strings) returned by your callback function.
- The values are arrays of the original items that belong to that group.
-

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.groupBy() Method</h2>
<p id="demo"></p>
<script>
// Create an Array
const fruits = [
  {name:"apples", quantity:300},
  {name:"bananas", quantity:500},
  {name:"oranges", quantity:200},
  {name:"kiwi", quantity:150}
];
// Callback function to select low volumes
function myCallback({ quantity }) {
  return quantity > 200 ? "ok" : "low";
}
// Group by ok and low
const result = Object.groupBy(fruits, myCallback);
// Display Results
let text = "These fruits are Ok: <br>";
for (let [x,y] of result.ok.entries()) {
  text += y.name + " " + y.quantity + "<br>";
}
text += "<br>These fruits are low: <br>";
for (let [x,y] of result.low.entries()) {
  text += y.name + " " + y.quantity + "<br>";
}
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
}  
document.getElementById("demo").innerHTML = text;  
</script>  
</body>  
</html>
```

Object.groupBy() vs Map.groupBy() Object.groupBy()

- Output = normal JS object {}
- Key = string only
- Value = array
- Access = obj.key
- Example:

Map.groupBy()

- Output = Map object new Map()
- Key = any type (string, number, object...)
- Value = array
- Access = map.get(key)

Object.keys()

What it does:

- Give all keys of an object
- Returns array of strings
- Original object? Safe, no change

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Objects</h1>  
<h2>The Object.keys() Method</h2>  
<p>Object.keys() returns an enumerable array of the keys of an object:</p>  
<p id="demo"></p>  
<script>  
// Create an Object  
const person = {  
  firstName: "John",  
  lastName: "Doe",
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE:** <https://learninghubshahabad.in>

```
age: 50,  
eyeColor: "blue"  
};  
// Get the Keys  
const keys = Object.keys(person);  
document.getElementById("demo").innerHTML = keys;  
</script>  
</body>  
</html>
```

Object Management -In JavaScript, an object is like a box with labeled compartments (keys) and things inside (values).
Object Management is basically all the stuff you do with these boxes:

Object.defineProperty()

What it does:

- Add a new property → put a new label+value in your box
- Change value → update the thing inside
- Change metadata → control if you can change/delete/list it
- Getters & setters → make “special doors” to read/write

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Object</h1>  
<h2>The defineProperty() Method</h2>  
<p>This example adds a new property to an object.</p>  
<p id="demo"></p>  
<script>  
// Create an Object:  
const person = {  
  firstName: "John",  
  lastName: "Doe",  
  language: "EN"  
};  
// Add a Property  
Object.defineProperty(person, "year", {value:"2008"})  
// Display the Property
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML = person.year;  
</script>  
</body>  
</html>
```

Changing a Property

In JavaScript:

- An object = box with keys (labels) and values (things inside)
- Changing a property = taking a key and replacing its value with something new

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Object</h1>  
<h2>The defineProperty() Method</h2>  
<p>This example changes a property value.</p>  
<p id="demo"></p>  
<script>  
// Create an Object:  
const person = {  
  firstName: "John",  
  lastName: "Doe",  
  language: "EN"  
};  
// Change a Property  
Object.defineProperty(person, "language", {value:"NO"})  
// Display the Property  
document.getElementById("demo").innerHTML = person.language;  
</script>  
</body>  
</html>
```

Property Attributes in JavaScript

JavaScript mein har property ka ek name aur ek value hota hai.

Value basically us property ka ek attribute hota hai.

Value ke alawa properties ke kuch aur attributes bhi hote hain:

- enumerable
- configurable

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- writable

Yeh attributes decide karte hain ki property ka behaviour kaisa hoga — jaise ki kya usse read kar sakte ho, write kar sakte ho, loop mein dikhai degi ya delete ho sakti hai.

By default, JavaScript mein tum saare attributes read kar sakte ho, lekin sirf value ko hi change kar sakte ho, woh bhi tabhi jab property writable ho. ECMAScript 5 se, JavaScript ne aise methods introduce kiye jinke through tum saare attributes get aur set kar sakte ho, sirf value hi nahi.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Object</h1>
<h2>The getOwnPropertyNames() Method</h2>
<p>This example list all the properties of an object.</p>
<p id="demo"></p>
<script>
// Create an Object:
const person = {
  firstName: "John",
  lastName: "Doe",
  language: "EN"
}
// Change Property
Object.defineProperty(person, "language", {enumerable:false});
// Display all Properties
document.getElementById("demo").innerHTML = Object.getOwnPropertyNames(person);
</script>
</body>
</html>
```

JavaScript Object.keys()

Object.keys() ek JavaScript method hai jo:

Object ke saare enumerable properties ki list return karta hai

(yani jo properties enumerable = true hoti hain, unke names ek array mein de deta hai)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

```
<h1>JavaScript Object</h1>
<h2>The Object.keys() Method</h2>
<p>This example list the enumerable properties of an object.</p>
<p id="demo"></p>
<script>
// Create an object:
const person = {
  firstName: "John",
  lastName: "Doe",
  language: "EN"
// Change the language Property
Object.defineProperty(person, "language", {enumerable:false});
// Display all Enumerable Properties
document.getElementById("demo").innerHTML = Object.keys(person);
</script>
</body>
</html>
```

Prototype Properties

JavaScript mein objects apne prototype se properties inherit karte hain.

Matlab agar prototype mein koi property hai, toh object usko use kar sakta hai—even agar object khud us property ko directly define na kare.

JavaScript Getters and Setters

Getters aur Setters JavaScript mein aise functions hote hain jo tumhe Object Accessors define karne ki permission dete hain.

Inka use karke tum computed properties bana sakte ho — yaani property ka value direct store nahi hota, balki function ke through read ya set hota hai.

JavaScript Getter (The get Keyword)

Getter tab call hota hai jab tum property read karte ho.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Getters and Setters</h2>
<p>Getters and setters allow you to get and set object properties via methods.</p>
<p>This example uses a lang property to get the value of the language property:</p>
<p id="demo"></p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
// Create an object:
const person = {
  firstName: "John",
  lastName: "Doe",
  language: "en",
  get lang() {
    return this.language;
  }
};
// Display data from the object using a getter:
document.getElementById("demo").innerHTML = person.lang;
</script>
</body>
</html>
```

JavaScript Setter (set keyword)

lang property input leti hai, aur us input se language property ko set karti hai.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Getters and Setters</h2>
<p>Getters and setters allow you to get and set properties via methods.</p>
<p>This example uses a lang property to set the value of the language property.</p>
<p id="demo"></p>
<script>
// Create an object:
const person = {
  firstName: "John",
  lastName: "Doe",
  language: "NO",
  set lang(value) {
    this.language = value;
  }
};
// Set a property using set:
person.lang = "en";
// Display data from the object:
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML = person.language;  
</script>  
</body>  
</html>
```

JavaScript Function or Getter

Getter

- Property ki tarah use hota hai
- Bina () call hota hai
- Value ko compute karke return karta hai
- Mostly read-only computed properties ke liye

Normal Function

- Function ki tarah call hota hai
- () lagana padta hai
- Use flexible operations ya logic ke liye

```
<!DOCTYPE html>  
<html>  
<body>  
<h2>JavaScript Object Methods</h2>  
<p>Object data can be accessed using a getter.</p>  
<p id="demo"></p>  
<script>  
// Create an object:  
const person = {  
  firstName: "vivek",  
  lastName: "sharma",  
  get fullName() {  
    return this.firstName + " " + this.lastName;  
  }  
};  
// Display data from the object using a getter:  
document.getElementById("demo").innerHTML = person.fullName;  
</script>  
</body>  
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Data Quality

JavaScript mein getters aur setters use karke tum apne data ki quality better bana sakte ho. Yeh ensure karte hain ki property ko read ya set karte time data correct format mein ho.

```
<!DOCTYPE html>
<html>
<head>
  <title>Getter Setter Example</title>
</head>
<body>
  <h2>JavaScript Getter & Setter - Data Quality Example</h2>
  <p>Language (uppercase only): <span id="output"></span></p>
  <script>
    const user = {
      _language: "", // private-like storage
      // Setter: Always store value in uppercase
      set language(value) {
        this._language = value.toUpperCase();
      },
      // Getter: Always return value in uppercase
      get language() {
        return this._language;
      }
    };
    // Using the setter
    user.language = "english"
    // Using the getter
    document.getElementById("output").textContent = user.language;
  </script>
</body>
</html>
```

Why Using Getters and Setters?

- Simpler Syntax
 - Properties ko normal variables ki tarah access aur update kar sakte ho, function call ki zarurat nahi.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

- Example: `obj.prop = 10; console.log(obj.prop);`
- **Uniform Syntax for Properties & Methods**
 - Methods jo calculations ya processing karte hain, unko property jaisa syntax use karke call kar sakte ho.
 - Code clean aur readable lagta hai.
- **Better Data Quality / Validation**
 - Control kar sakte ho ki value kaise store aur retrieve ho.
 - Example: hamesha uppercase store karna, ya invalid input block karna.
- **Do Things Behind-the-Scenes**
 - Property read ya set hone par extra logic automatically execute hota hai.
 - Example: formatting, logging, calculations, etc.

Object.defineProperty()

- `Object.defineProperty()` method ka use karke tum existing object me new property add kar sakte ho, ya existing property ke attributes modify kar sakte ho.
- Iske through tum getters aur setters bhi define kar sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Getters and Setters</h2>
<p>Perfect for creating counters:</p>
<p id="demo"></p>
<script>
// Define an object
const obj = {counter : 0};
// Define Setters and Getters
Object.defineProperty(obj, "reset", {
  get : function () {this.counter = 0;}
});
Object.defineProperty(obj, "increment", {
  get : function () {this.counter++;}
});
Object.defineProperty(obj, "decrement", {
  get : function () {this.counter--;}
});
Object.defineProperty(obj, "add", {
  set : function (value) {this.counter += value;}
});
});
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
Object.defineProperty(obj, "subtract", {  
  set : function (value) {this.counter -= value;}  
});
```

```
// Play with counter:  
obj.reset;  
obj.add = 5;  
obj.subtract = 1;  
obj.increment;  
obj.decrement;  
document.getElementById("demo").innerHTML = obj.counter;  
</script>  
</body>  
</html>
```

JavaScript Object Protection Methods

JavaScript me objects ko modify karne se protect karne ke liye kuch built-in methods hain:

```
// Prevents re-assignment  
const car = {type:"Fiat", model:"500", color:"white"};  
  
// Prevents adding object properties  
Object.preventExtensions(object)  
  
// Returns true if properties can be added to an object  
Object.isExtensible(object)  
  
// Prevents adding and deleting object properties  
Object.seal(object)  
  
// Returns true if object is sealed  
Object.isSealed(object)  
  
// Prevents any changes to an object  
Object.freeze(object)  
  
// Returns true if object is frozen  
Object.isFrozen(object)
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Using const for Objects

- const se declare kiya hua object reassign nahi ho sakta.
- Matlab: tum object ko ek naya object assign nahi kar sakte.
- Lekin, object ke andar properties change kar sakte ho:
- Property add kar sakte ho
- Property delete kar sakte ho
- Property value modify kar sakte ho

JavaScript: Object.preventExtensions()

- Object.preventExtensions(obj) ka use karke tum object me nayi properties add karne se rok sakte ho.
- Lekin:
- Existing properties modify ki ja sakti hain
- Existing properties delete ki ja sakti hain

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.preventExtensions() Method</h2>
<p id="demo"></p>
<script>
"use strict"
// Create Object
const person = {firstName:"John", lastName:"Doe"};
// Prevent Extensions
Object.preventExtensions(person)
// Test Error
let text = "";
try {
  person.nationality = "English";
}
catch (err) {
  text = err;
}
document.getElementById("demo").innerHTML = text;
</script>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</body>  
</html>
```

JavaScript: Object.isExtensible()

- Object.isExtensible(obj) ka use karke tum check kar sakte ho ki object me new properties add ki ja sakti hain ya nahi.
- Ye method true return karta hai agar object extensible hai, warna false.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Objects</h1>  
<h2>The Object.isExtensible() Method</h2>  
<p id="demo"></p>  
<script>  
// Create Object  
const person = {firstName:"John", lastName:"Doe"};  
// Prevent Extensions  
Object.preventExtensions(person);  
// This will return false  
let answer = Object.isExtensible(person)  
document.getElementById("demo").innerHTML =answer;  
</script>  
</body>  
</html>
```

JavaScript: Object.seal()

- Object.seal(obj) ka use karke tum object ko sealed bana sakte ho.
- Sealed object me:
 - New properties add nahi ki ja sakti
 - Existing properties delete nahi ki ja sakti
 - Existing properties non-configurable ho jati hain (unke attributes change nahi ho sakte)

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Objects</h1>  
<h2>The Object.seal() Method</h2>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
Object.seal(fruits);
// Test Error
let text = "";
try {
  fruits.push("Kiwi");
}
catch (err) {
  text = err;
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

JavaScript: Object.isSealed()

- Object.isSealed(obj) ka use karke tum check kar sakte ho ki object sealed hai ya nahi.
- Ye method true return karta hai agar object seal kiya gaya hai, warna false.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.isFrozen() Method</h2>
<p id="demo"></p>
<script>
// Create Array
const fruits = ["Banana", "Orange", "Apple", "Mango"];
// Seal Array
Object.seal(fruits);
// This will return true
let answer = Object.isSealed(fruits);
document.getElementById("demo").innerHTML = answer;
</script>
</body>
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

JavaScript: Object.freeze()

- Object.freeze(obj) ka use karke tum object ko completely read-only bana sakte ho.
- Frozen object me:
- Existing properties modify nahi ki ja sakti
- New properties add nahi ki ja sakti
- Existing properties delete nahi ki ja sakti
- Matlab: object fully immutable ho jata hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.freeze() Method</h2>
<p id="demo"></p>
<script>
// Create Array
const fruits = ["Banana", "Orange", "Apple", "Mango"];
// Freeze Array
Object.freeze(fruits);
// Test Error
let text
try {
  fruits.push("Kiwi");
  text = fruits;
}
catch (err) {
  text = err;
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

JavaScript: Object.isFrozen()

- Object.isFrozen(obj) ka use karke tum check kar sakte ho ki object frozen hai ya nahi.
- Ye method true return karta hai agar object freeze kiya gaya hai, warna false.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<h2>The Object.isFrozen() Method</h2>
<p id="demo"></p>
<script>
// Create Object
const fruits = ["Banana", "Orange", "Apple", "Mango"];
// Freeze Object
Object.freeze(fruits);
// This will return true
let answer = Object.isFrozen(fruits);
document.getElementById("demo").innerHTML = answer;
</script>
</body>
</html>
```

Object Methods and Properties

Name	Description
assign()	Ek source object ki properties ko target object me copy karta hai.
constructor	Return karta hai function jo object ka prototype create kiya.
create()	Ek naya object return karta hai jo existing object se create hota hai.
defineProperties()	Ek ya zyada properties add ya change karta hai.
defineProperty()	Ek property add ya modify karta hai.
delete	Object se ek property delete karta hai.
entries()	Object ki key/value pairs ka array return karta hai.
freeze()	Object ko completely read-only banata hai.
fromEntries()	Iterable list of key/value pairs se object create karta hai.
getOwnPropertyDescriptor()	Ek property ka descriptor object return karta hai.
getOwnPropertyDescriptors()	Object ki sabhi properties ke descriptors return karta hai.
getOwnPropertyNames()	Object ki keys ka array return karta hai.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Name	Description
groupBy()	Object elements ko callback ke return value ke basis par group karta hai.
isExtensible()	Check karta hai agar object extensible hai (new properties add ho sakti hain).
isFrozen()	Check karta hai agar object frozen hai.
isSealed()	Check karta hai agar object sealed hai.
keys()	Object ki keys ka array return karta hai.
preventExtensions()	Object me new properties add karne se rokta hai.
prototype	JavaScript objects me properties aur methods add karne ke liye use hota hai.
seal()	Object me new properties add ya existing delete nahi kar sakte.
toString()	Object ko string me convert karke return karta hai.
valueOf()	Object ka primitive value return karta hai.

JavaScript Dates

- JavaScript me Date object ka use date aur time handle karne ke liye hota hai.
- Date object automatically current date and time set kar sakta hai ya specific date/time bhi set ki ja sakti hai.
- Date object ke methods se tum date ko read, set, format aur calculate kar sakte ho.

Creating Date Objects

```
new Date()
new Date(date string)

new Date(year, month)
new Date(year, month, day)
new Date(year, month, day, hours)
new Date(year, month, day, hours, minutes)
new Date(year, month, day, hours, minutes, seconds)
new Date(year, month, day, hours, minutes, seconds, ms)

new Date(milliseconds)
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

JavaScript: new Date()

- new Date() ka use karke tum ek Date object create kar sakte ho.
- Agar kuch bhi argument nahi doge, ye current date aur time set karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>Using new Date()</h2>
<p>Create a new date object with the current date and time:</p>
<p id="demo"></p>
<script>
const d = new Date();
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

JavaScript: new Date(date string)

- new Date(dateString) ka use karke tum specific date create kar sakte ho.
- dateString ek valid date format me hona chahiye (ISO 8601 format is best: "YYYY-MM-DD").

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>Using new Date()</h2>
<p id="demo"></p>
<script>
const d = new Date("2022-03-25");
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

JavaScript: new Date(year, month, ...)

new Date(year, month, day, hour, minute, second, millisecond) ka use karke tum specific date aur time create kar sakte ho.

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

Parameters:

- year → full year (e.g., 2025)
- month → month (0–11, January = 0, December = 11)
- day → day of month (1–31), default = 1
- hour → 0–23, default = 0
- minute → 0–59, default = 0
- second → 0–59, default = 0
- millisecond → 0–999, default = 0

JavaScript: Using 6, 4, 3, or 2 Numbers in new Date()

- Tum new Date() me 2–7 numbers use karke specific date aur time create kar sakte ho.
- Agar all 7 numbers nahi doge, JS missing values ko default 0 ya 1 consider karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript new Date()</h2>
<p>One parameter will be interpreted as new Date(milliseconds).</p>
<p id="demo"></p>
<script>
const d = new Date(2018);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

JavaScript: Previous Century (1 or 2-digit years)

- Agar tum new Date() me 1-digit ya 2-digit year doge, JavaScript usko 1900s (19xx) me consider karega.
- Matlab:
- new Date(99, 0, 1) → year 1999
- new Date(9, 0, 1) → year 1909

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript new Date()</h2>
<p>One digit years will be interpreted as 19xx:</p>
<p id="demo"></p>
<script>
const d = new Date(9, 11, 24);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

JavaScript: new Date(milliseconds)

- new Date(milliseconds) ka use karke tum ek date object create kar sakte ho based on milliseconds since January 1, 1970, 00:00:00 UTC.
- Matlab: 1970-01-01 00:00:00 UTC = 0 milliseconds
- Positive number → future date, Negative number → past date

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>Using new Date()</h2>
<p>0 milliseconds from January 01 1970 UTC is:</p>
<p id="demo"></p>
<script>
const d = new Date(0);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

JavaScript: Date Methods

- Jab Date object create hota hai, uske saath methods aate hain jisse tum date aur time read ya modify kar sakte ho.
- Tum local time ya UTC (GMT) dono use kar sakte ho.
- Date methods year, month, day, hour, minute, second, millisecond ke liye kaam karte hain.

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

JavaScript: Displaying Dates

- Jab tum ek Date object console.log() ya directly output karte ho, JavaScript default me toString() method use karta hai.
- toString() → date ka string representation return karta hai, jisme date, time, aur time zone included hota hai.
- Format ECMAScript specification me defined hai, isliye consistent hota hai across browsers.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The toISOString() Method</h2>
<p>Convert a date to a string using the ISO standard:</p>
<p id="demo"></p>
<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.toISOString();
</script>
</body>
</html>
```

JavaScript: Date Input

- JavaScript me tum user se date input le sakte ho, usually HTML `<input type="date">` ke through.
- Browser automatically date picker provide karta hai, jisse user easily date select kar sakta hai.
- Selected date ko JavaScript me value property ke through access kiya ja sakta hai.
- Date input ka format YYYY-MM-DD hota hai.

Type	Example
ISO Date	"2015-03-25" (The International Standard)
Short Date	"03/25/2015"
Long Date	"Mar 25 2015" or "25 Mar 2015"

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

JavaScript: ISO Dates

- ISO 8601 → international standard for representing dates and times.
- JavaScript me ISO format ka use karna best practice hai, kyunki ye consistent aur unambiguous hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript ISO Dates</h2>
<p id="demo"></p>
<script>
const d = new Date("2015-03-25");
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

JavaScript: Time Zones

- Default behavior:
- Setting a date → agar timezone specify nahi kiya, JS browser ka local timezone use karega.
- Getting a date → agar timezone specify nahi kiya, JS result local timezone me convert karega.
- Matlab agar date/time GMT me create kiya gaya hai, aur user US Central Time me hai, to JS automatically adjust karega local timezone ke hisaab se.

JavaScript: Short Dates

- Short Date Format → ek compact version of date string, usually without time.
- Browser's locale settings ke hisaab se format change hota hai.
- Commonly use hota hai toLocaleDateString() method ke saath.

```
<!DOCTYPE html>
<html>
<body>
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE:** <https://learninghubshahabad.in>

```
<h2>JavaScript new Date()</h2>
<p id="demo"></p>
<script>
const d = new Date("03/25/2015");
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

JavaScript: Long Dates

- Long Date Format → more readable, human-friendly date format.
- Usually format hota hai: “MMM DD YYYY” (Month abbreviation, Day, Year)
- JavaScript me use hota hai `toDateString()` ya `toLocaleDateString()` with options.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript new Date()</h2>
<p id="demo"></p>
<script>
const d = new Date("JANUARY, 25, 2015");
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

Date.parse() in JavaScript

The `Date.parse()` method is used to convert a date string into a number representing the milliseconds since January 1, 1970, 00:00:00 UTC (this is known as the Unix epoch)

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Date.parse()</h2>
<p>Date.parse(string) returns milliseconds.</p>
<p>You can use the return value to convert the string to a date object:</p>
<p id="demo"></p>
<script>
let msec = Date.parse("March 21, 2012");
const d = new Date(msec);
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML = d;  
</script>  
</body>  
</html>
```

Date Get Methods

- Date object se date/time ka part nikalna.
- Maan lo, tumhare paas aaj ka date hai: 18 Nov 2025, 2:30 PM.
Agar tumhe sirf year, ya month, ya hour chahiye, to ye methods use karte ho.

Method	Description
getFullYear()	Get year as a four digit number (yyyy)
getMonth()	Get month as a number (0-11)
getDate()	Get day as a number (1-31)
getDay()	Get weekday as a number (0-6)
getHours()	Get hour (0-23)
getMinutes()	Get minute (0-59)
getSeconds()	Get second (0-59)
getMilliseconds()	Get millisecond (0-999)
getTime()	Get time (milliseconds since January 1, 1970)

getFullYear() method date object se sirf year nikalta hai, aur hamesha 4 digits me deta hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Dates</h1>  
<h2>The getFullYear() Method</h2>  
<p>Return the full year of a date object:</p>  
<p id="demo"></p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.getFullYear();
</script>
</body>
</html>
```

getMonth() method date object se month nikalta hai, lekin 0 se start hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>JavaScript getMonth()</h2>
<p>Return the month as a number.</p>
<p>You can use an array of names to return the month as a name:</p>
<p id="demo"></p>
<script>
const months
["January","February","March","April","May","June","July","August","September","October",
,"November","December"];
const d = new Date();
let month = months[d.getMonth()];
document.getElementById("demo").innerHTML = month;
</script>
</body>
</html>
```

The getDate()

- getDate() method ek Date object se month ka day (din) nikalta hai.
- Ye 1 se 31 ke beech ka integer return karta hai.
- Agar date invalid ho, to NaN bhi de sakta hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The getDate() Method</h2>
<p>Return the day of a date as a number (1-31):</p>
<p id="demo"></p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.getDate();
</script>
</body>
</html>
```

The getHours()

- getHours() method date object se “hour” part nikalta hai.
- Ye local time ke hisaab se hour return karta hai.
- Return value ek integer hota hai 0 se 23 ke beech.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The getHours() Method</h2>
<p>Return the hours of a date as a number (0-23):</p>
<p id="demo"></p>
<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.getHours();
</script>
</body>
</html>
```

The getMinutes()

- getMinutes() method ek Date object se minutes ka part nikalta hai.
- Ye local time ke hisaab se minutes return karta hai.
- Return value ek integer hota hai, jo 0 se 59 ke beech mein hota hai.
- Agar date object invalid ho, to ye NaN return kar sakta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The getMinutes() Method</h2>
<p>Returns the minutes of a date as a number (0-59):</p>
<p id="demo"></p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.getMinutes();
</script>
</body>
</html>
```

getSeconds() method date object ke seconds ko return karta hai as a number 0 se 59 ke beech.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The getSeconds() Method</h2>
<p>Return the seconds of a date as a number (0-59):</p>
<p id="demo"></p>
<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.getSeconds();
</script>
</body>
</html>
```

getMilliseconds()

getMilliseconds() method date object ke milliseconds ko return karta hai as a number 0 se 999 ke beech.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The getMilliseconds() Method</h2>
<p>Return the milliseconds of a date as a number (0-999):</p><p id="demo"></p>
<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.getMilliseconds();
</script>
</body>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

</html>

The `getDay()` Method

`getDay()` method date object ke weekday ko return karta hai as a number 0 se 6 ke

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript Dates</h1>
```

```
<h2>The getDay() Method</h2>
```

```
<p>Return the weekday as a number.</p>
```

```
<p>You can use an array of names to return the weekday as a name:</p>
```

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

```
<script>
```

```
const days =
```

```
["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"];
```

```
const d = new Date();
```

```
let day = days[d.getDay()];
```

```
document.getElementById("demo").innerHTML = day;
```

```
</script>
```

```
</body>
```

```
</html>
```

`getTime()` Method:

- Purpose: Ye method return karta hai number of milliseconds since 1 January 1970 (UTC).
- Use: Mostly time calculation ya date comparison ke liye use hota hai.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript Dates</h1>
```

```
<h2>The getTime() Method</h2>
```

```
<p></p>
```

```
<p>Return the number of milliseconds since midnight January 1, 1970:</p>
```

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

```
<script>
```

```
const d = new Date();
```

```
document.getElementById("demo").innerHTML = d.getTime();
```

```
</script>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</body>  
</html>
```

The Date.now() Method

Date.now() returns the number of milliseconds since January 1, 1970.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Dates</h1>  
<h2>Using Date.now()</h2>  
<p>Calculate the number of years since January 1, 1970:</p>  
<p id="demo"></p>  
<script>  
// Calculate milliseconds in a year  
const minute = 1000 * 60;  
const hour = minute * 60;  
const day = hour * 24;  
const year = day * 365;  
// Divide Date.now() with a year  
let years = Math.round(Date.now() / year);  
document.getElementById("demo").innerHTML = years;  
</script>  
</body>  
</html>
```

UTC Date Get Methods

UTC (world time, timezone independent) ka current year return karta hai.

Method	Same As (Local)	Description
getUTCDate()	getDate()	Month ka day return karta hai UTC time ke hisaab se.
getUTCFullYear()	getFullYear()	Current year UTC ke hisaab se return karta hai.
getUTCMonth()	getMonth()	Month number (0-11) UTC ke hisaab se. 0 = Jan, 11 = Dec.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Method	Same As (Local)	Description
getUTCDay()	getDay()	Week ka day (0–6) UTC ke hisaab se. 0 = Sunday.
getUTCHours()	getHours()	Hour (0–23) UTC ke hisaab se.
getUTCMinutes()	getMinutes()	Minutes (0–59) UTC ke hisaab se.
getUTCSeconds()	getSeconds()	Seconds (0–59) UTC ke hisaab se.
getUTCMilliseconds()	getMilliseconds()	Milliseconds (0–999) UTC ke hisaab se.

getTimezoneOffset() Method

- Purpose: Ye method return karta hai difference between local time and UTC in minutes.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The getTimezoneOffset() Method</h2>
<p>The time zone difference in minutes is:</p>
<p id="demo"></p>
<script>
const d = new Date();
document.getElementById("demo").innerHTML = d.getTimezoneOffset();
</script>
</body>
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Set Date Methods

Set Date Methods allow आपको existing Date object के parts को change करने के लिये use होने हैं। Basically, आप year, month, day, hour, etc. update कर सकते हैं।

Method	Description
setDate()	Month का day set करता है (1-31).
setFullYear()	Year set करता है (yyyy).
setHours()	Hour set करता है (0-23).
setMilliseconds()	Milliseconds set करता है (0-999).
setMinutes()	Minutes set करता है (0-59).
setMonth()	Month set करता है (0-11). 0 = Jan, 11 = Dec.
setSeconds()	Seconds set करता है (0-59).
setTime()	Date का complete timestamp set करता है in milliseconds since 1 Jan 1970.

setFullYear()

Purpose: Ye method Date object का year set करता है।

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The setFullYear() Method</h2>
<p>The setFullYear() method sets the year of a date object.</p>
<p id="demo"></p>
<script>
const d = new Date("January 01, 2025");
d.setFullYear(2020);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

setMonth() Method

Purpose: Ye method Date object का month set करता है।

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The setMonth() Method</h2>
<p>The setMonth() method sets the month of a date object.</p>
<p>Note that months count from 0. December is month 11.</p>
<p id="demo"></p>
<script>
const d = new Date("January 01, 2025");
d.setMonth(11);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

The setDate() Method

The setDate() method sets the day of a date object (1-31):

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The setDate() Method</h2>
<p>The setDate() method can be used to add days to a date.</p>
<p id="demo"></p>
<script>
const d = new Date("January 01, 2025");
d.setDate(d.getDate() + 50);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

setHours() Method

- Purpose: Ye method Date object ke hours set karta ha.

```
<!DOCTYPE html>
<html>
<body>
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<h1>JavaScript Dates</h1>
<h2>The setHours() Method</h2>
<p>The setHours() method can also be used to set minutes and seconds.</p>
<p id="demo"></p>
<script>
const d = new Date("January 01, 2025");
d.setHours(22, 10, 20);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

setMinutes() Method

- Purpose: Ye method Date object ke minutes set karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The setMinutes() Method</h2>
<p>The setMinutes() method sets the minutes of a date object (0-59).</p>
<p id="demo"></p>
<script>
const d = new Date("January 01, 2025");
d.setMinutes(30);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

setSeconds() Method

- Purpose: Ye method Date object ke seconds set karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Dates</h1>
<h2>The setSeconds() Method</h2>
<p>The setSeconds() method sets the seconds of a date object (0-59).</p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<p id="demo"></p>
<script>
const d = new Date("January 01, 2025");
d.setSeconds(30);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

JavaScript Date Methods

Method / Property	Explanation
new Date()	Naya Date object create karta hai, ya current date & time le aata hai
constructor	Ye function return karta hai jo Date object banata hai (basically Date)
getDate()	Month ka day (1–31)
getDay()	Week ka day (0 = Sunday, 6 = Saturday)
getFullYear()	Year (4-digit, e.g., 2025)
getHours()	Hour (0–23)
getMinutes()	Minutes (0–59)
getSeconds()	Seconds (0–59)
getMilliseconds()	Milliseconds (0–999)
getMonth()	Month (0 = Jan, 11 = Dec)
getTime()	Jan 1, 1970 se milliseconds count
getTimezoneOffset()	Local aur UTC ke beech ka time difference (minutes me)
getUTCDate()	UTC ke hisaab se month ka day
getUTCDay()	UTC ke hisaab se week ka day
getUTCFullYear()	UTC ke hisaab se year
getUTCHours()	UTC ke hisaab se hour
getUTCMinutes()	UTC ke hisaab se minutes
getUTCSeconds()	UTC ke hisaab se seconds
getUTCMilliseconds()	UTC ke hisaab se milliseconds

LEARNING HUB
SHAHABAD MARKANDA
 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Method / Property	Explanation
getUTCMonth()	UTC ke hisaab se month
getYear()	Old method, ab getFullYear() use karo
now()	Current time ka milliseconds since Jan 1, 1970
parse(dateString)	Date string ko milliseconds me convert karta hai
prototype	Object me naye methods/properties add karne ka tareeka
setDate()	Date ka day set karo
setMonth()	Month set karo
setFullYear()	Year set karo
setHours()	Hour set karo
setMinutes()	Minutes set karo
setSeconds()	Seconds set karo
setMilliseconds()	Milliseconds set karo
setTime()	Date ko specific milliseconds set karo
setUTCDate()	UTC me day set karo
setUTCMonth()	UTC me month set karo
setUTCFullYear()	UTC me year set karo
setUTCHours()	UTC me hour set karo
setUTCMinutes()	UTC me minutes set karo
setUTCSeconds()	UTC me seconds set karo
setUTCMilliseconds()	UTC me milliseconds set karo
setYear()	Old method, ab setFullYear() use karo
toDateString()	Sirf date readable string me convert karo
toTimeString()	Sirf time readable string me convert karo
toString()	Full date + time readable string me
toUTCString()	UTC ke hisaab se date + time string
toISOString()	ISO standard date string (2025-11-18T15:45:00.000Z)
toJSON()	JSON format me date string
toLocaleDateString()	Local format me date

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

Method / Property	Explanation
toLocaleTimeString()	Local format me time
toLocaleString()	Local format me full date + time
toGMTString()	Old method, use toUTCString()
UTC()	UTC ke hisaab se milliseconds since Jan 1, 1970
valueOf()	Primitive value of Date (milliseconds)

What is an Array

- Array = ek container jo multiple values ko ek saath store karta hai.
- Example: let fruits = ["apple", "banana", "mango"];
- Arrays indexed hote hain, starting from 0.
- fruits[0] → "apple"
- fruits[1] → "banana"

Characteristic	Explanation
Elements	Array me values ka list hota hai, jise elements kehte hain. Example: [1, "hello", true]
Ordered	Array elements ka order fix hota hai, jo unke index pe depend karta hai.
Zero Indexed	Pehla element ka index 0 hota hai, doosra 1, aur aise hi aage. Example: arr[0] first item
Dynamic Size	Array ka size badal sakta hai. Elements add/remove kar sakte ho.
Heterogeneous	Ek hi array me different types ke elements store kar sakte ho — numbers, strings, objects, ya arrays. Example: [1, "text", [2,3]]

Why Use Arrays

1. Store Multiple Values – Ek hi variable me bohot saari values store kar sakte ho.
2. Ordered Data – Elements index ke according arranged hote hain.
3. Zero-Indexed – Pehla element 0 index pe hota hai, easy access ke liye.
4. Dynamic Size – Array ka size badh ya ghat sakta hai.
5. Heterogeneous – Different data types ek hi array me store kar sakte ho.
6. Easy Iteration – Loops aur methods se sab elements pe easily kaam kar sakte ho.
7. Better Than Multiple Variables – Alag-alag variables ke bajaye arrays simple aur scalable solution hai.

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Creating an Array

Ye sabse easy aur common way hai array create karne ka.
Elements ko square brackets [] me likhte hain, comma se separate karke.

```
!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<p id="demo"></p>
<script>
const cars = [
  "Saab",
  "Volvo",
  "BMW"
];
document.getElementById("demo").innerHTML = cars;
</script>
</body>
</html>
```

Using the JavaScript Keyword new

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<p id="demo"></p>
<script>
const cars = new Array("Saab", "Volvo", "BMW");
document.getElementById("demo").innerHTML = cars;
</script>
</body>
</html>
```

Accessing Array Elements

Concept:

- Array me elements ko access karne ke liye index number ka use karte hain.
- Indexing starts from 0.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Bracket Indexing</h2>
<p>JavaScript array elements are accessed using numeric indexes (starting from 0).</p>
<p id="demo"></p>
<script>
const cars = ["Saab", "Volvo", "BMW"];
cars[0] = "Opel";
document.getElementById("demo").innerHTML = cars;
</script>
</body>
</html>
```

Converting an Array to a String

toString() method array ko string me convert karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The toString() Method</h2>
<p>The toString() method returns the elements of an array as a comma separated string:</p>
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
let myList = fruits.toString();
document.getElementById("demo").innerHTML = myList;
</script>
</body>
</html>
```

Access the Full Array

With JavaScript, the full array can be accessed by referring to the array name:

```
<!DOCTYPE html>
<html>
<body>
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<h1>JavaScript Arrays</h1>
<p id="demo"></p>
<script>
const cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars;
</script>
</body>
</html>
```

Arrays are Objects

Concept:

- JavaScript me arrays ek special type ke objects hote hain.
- Agar aap typeof operator use karoge, array ke liye "object" return hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Objects</h1>
<p>JavaScript uses names to access object properties.</p>
<p id="demo"></p>
<script>
const person = {firstName:"Muskan", lastName:"Sharma", age:46};
document.getElementById("demo").innerHTML = person.firstName;
</script>
</body>
</html>
```

Array Elements Can Be Objects

Concept:

- JavaScript me variables objects ho sakte hain.
- Arrays bhi special objects hain.
- Isliye array ke elements me different types store kar sakte ho: numbers, strings, objects, functions, aur even other arrays.

Array Properties and Methods

The length Property

Concept:

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- length property array ke total elements ki count return karta hai.
- Ye read-only nahi, aap isse modify karke array ko resize bhi kar sakte ho

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The length Property</h2>
<p>The length property returns the length of an array:</p>
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
let size = fruits.length;
document.getElementById("demo").innerHTML = size;
</script>
</body>
</html>
```

Accessing the First Array Element

Concept:

- Array ke first element ko access karne ke liye index 0 use karte hain.
- JavaScript arrays zero-indexed hote hain, matlab first element ka index 0 hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Bracket Indexing</h2>
<p>JavaScript array elements are accesses using numeric indexes (starting from 0).</p>
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits[0];
</script>
</body>
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Accessing the Last Array Element

Concept:

- Array ka last element access karne ke liye length - 1 index use karte hain.
- JavaScript arrays zero-indexed hote hain, matlab last element ka index total elements - 1 hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Bracket Indexing</h2>
<p>JavaScript array elements are accessed using numeric indexes (starting from 0).</p>
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits[fruits.length-1];
</script>
</body>
</html>
```

Looping Array Elements

Concept:

- Array ke saare elements par repeat action perform karne ke liye loops ka use karte hain.
- Commonly for loop, for...of, forEach use hote hain.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The forEach() Method</h2>
<p>Call a function for each array element:</p>
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
let text = "<ul>";
fruits.forEach(myFunction);
text += "</ul>"
document.getElementById("demo").innerHTML = text;
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

```
function myFunction(value) {  
text += "<li>" + value + "</li>";  
}  
</script>  
</body>  
</html>
```

Associative Arrays

Concept:

- Kuch programming languages me arrays ke indexes names (strings) ho sakte hain.
- Aise arrays ko associative arrays ya hashes kehte hain.
- JavaScript me arrays hamesha numbered indexes use karte hain.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Arrays</h1>  
<p id="demo"></p>  
<script>  
const person = [];  
person[0] = "John";  
person[1] = "Doe";  
person[2] = 46;  
document.getElementById("demo").innerHTML =  
person[0] + " " + person.length;  
</script>  
</body>  
</html>
```

The Difference Between Arrays and Objects

- Array: Numbered indexes, ordered, best for lists.
- Object: Named indexes (keys), unordered, best for key-value data.

JavaScript new Array()

Concept:

- JavaScript me array constructor hai: new Array()
- Lekin modern JS me array literal [] use karna easier aur safer hai.

```
<!DOCTYPE html>
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<html>
<body>
<h1>JavaScript Arrays</h1>
<p>Avoid using new Array().</p>
<p id="demo"></p>
<script>
var points = new Array(40);
document.getElementById("demo").innerHTML = points[0];
</script>
</body>
</html>
```

How to Recognize an Array

- JS me arrays ek special type ka object hote hain.
- Matlab typeof ko pata nahi ki yeh array hai, bas kehta hai "yeh object hai."

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Nested JavaScript Objects and Arrays.</h2>
<p id="demo"></p>
<script>
let x = "";
const myObj = {
  name: "John",
  age: 30,
  cars: [
    {name:"Ford", models:["Fiesta", "Focus", "Mustang"]},
    {name:"BMW", models:["320", "X3", "X5"]},
    {name:"Fiat", models:["500", "Panda"]}
  ]
}
for (let i in myObj.cars) {
  x += "<h2>" + myObj.cars[i].name + "</h2>";
  for (let j in myObj.cars[i].models)
    x += myObj.cars[i].models[j] + "<br>";
}
}
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML = x;  
</script>  
</body>  
</html>
```

Basic Array Methods

Basic Array Methods

<u>Array_length</u>	<u>Array_isArray()</u>
<u>Array_toString()</u>	<u>Array_delete()</u>
<u>Array_at()</u>	<u>Array_concat()</u>
<u>Array_join()</u>	<u>Array_copyWithin()</u>
<u>Array_pop()</u>	<u>Array_flat()</u>
<u>Array_push()</u>	<u>Array_slice()</u>
<u>Array_shift()</u>	<u>Array_splice()</u>
<u>Array_unshift()</u>	<u>Array_toSpliced()</u>

JavaScript Array length Property

length property array ka size (kitne elements hain) return karta hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Arrays</h1>  
<h2>The length Property</h2>  
<p>The length property can be used to set the length of an array:</p>  
<p id="demo"></p>  
<script>  
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.length = 2;  
document.getElementById("demo").innerHTML = fruits;  
</script>  
</body>  
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

JavaScript Array toString() Method

toString() method array ke elements ko ek string me convert kar deta hai, aur comma se separate karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The toString() Method</h2>
<p>The toString() method returns the elements of an array as a comma separated string:</p>
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
let myList = fruits.toString();
document.getElementById("demo").innerHTML = myList;
</script>
</body>
</html>
```

Array.at() Kya Hota Hai?

at() JavaScript ka **array method** hai jo array ke kisi specific **index** par jo value hoti hai, usko return karta hai.

Lekin iska special feature ye hai ki **negative index** bhi use kar sakte ho — jiska matlab array ke end se count karna.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Bracket Notation</h2>
<p>The bracket notation [] returns an indexed element from an array:</p>
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
let fruit = fruits[2];
document.getElementById("demo").innerHTML = fruit;
</script>
</body>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

</html>

Array.join() Kya Hota Hai?

join() method array ke saare elements ko mila kar ek string bana deta hai.

Ye toString() jaisa hi hota hai, bas difference ye hai ki tum separator decide kar sakte ho (elements ke beech kya aaye).

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript Arrays</h1>
```

```
<h2>The join() Method</h2>
```

```
<p>The join() method joins array elements into a string.</p>
```

```
<p>In this example we have used " * " as a separator between the elements:</p>
```

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

```
<script>
```

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
document.getElementById("demo").innerHTML = fruits.join(" * ");
```

```
</script>
```

```
</body>
```

```
</html>
```

pop() Kya Karta Hai?

- pop() array ka last element remove kar deta hai aur us removed element ko return bhi karta hai.
- Yaani:
- array ka aakhri element toot ke nikal jaata hai
- jo element nikla, pop() wapas return karta hai
- original array change ho jaata hai (mutate hota hai)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript Arrays</h1>
```

```
<h2>The pop() Method</h2>
```

```
<p>The pop() method returns the value that was "popped out":</p>
```

```
<p id="demo1"></p>
```

```
<p id="demo2"></p>
```

```
<script>
```

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo1").innerHTML = fruits.pop();
document.getElementById("demo2").innerHTML = fruits;
</script>
</body>
</html>
```

JavaScript Array push()

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The push() Method</h2>
<p>The push() method returns the new array length:</p>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits.push("Kiwi");
document.getElementById("demo2").innerHTML = fruits;
</script>
</body>
</html>
```

JavaScript Array shift()

- shift() method array ka pehla element remove kar deta hai aur baaki saare elements ko left shift kar deta hai (index 1 → 0, 2 → 1, etc).
- Ye removed element ko return bhi karta hai.
- First element delete
- Array modify hota hai (mutate)
- Removed value return hoti hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The shift() Method</h2>
<p>The shift() method returns the element that was shifted out.</p>
<p id="demo1"></p>
<p id="demo2"></p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits.shift();
document.getElementById("demo2").innerHTML = fruits;
</script>
</body>
</html>
```

unshift() method

array ke beginning (start) mein

ek ya zyada elements add karta hai.

Aur jab ye new elements start me add hote hain,

toh purane elements right shift ho jaate hain (index 0 → 1, 1 → 2 ...)

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The unshift() Method</h2>
<p>The unshift() method returns the length of the new array:</p>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits.unshift("Lemon");
document.getElementById("demo2").innerHTML = fruits;
</script>
</body>
</html>
```

Array.isArray()

- Array.isArray() method check karta hai ki given value array hai ya nahi.
- Agar value array hogi → true return karega
Agar array nahi hogi → false return karega

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The isArray() Method</h2>
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple"];
document.getElementById("demo").innerHTML = Array.isArray(fruits);
</script>
</body>
</html>
```

JavaScript Array delete()

JavaScript me delete koi array method nahi hota. Ye ek operator hai jo kisi array element ko delete to kar deta hai, lekin uski jagah empty hole (undefined / empty slot) chhod deta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The delete Method</h2>
<p>Deleting elements leaves undefined holes in an array:</p>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML =
"The first fruit is: " + fruits[0];
delete fruits[0]
document.getElementById("demo2").innerHTML =
"The first fruit is: " + fruits[0];
</script>
</body>
</html>
```

JavaScript Array concat()

concat() method do ya zyada arrays ko ek saath jodkar naya array bana deta hai. Original arrays change nahi hote. Return ek naya array hota hai jisme sab elements mil jaate hain.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

<h2>The concat() Method</h2>

<p>The concat() method merges (concatenates) arrays:</p>

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

<script>

```
const myGirls = ["Cecilie", "Lone"];
```

```
const myBoys = ["Emil", "Tobias", "Linus"];
```

```
const myChildren = myGirls.concat(myBoys);
```

```
document.getElementById("demo").innerHTML = myChildren;
```

</script>

</body>

</html>

copyWithin()

- copyWithin() method array ke andar hi elements ko ek jagah se uthakar doosri jagah copy karta hai.
- Original array modify hota hai (mutate)
- Naya array create nahi hota
- Sirf copy hota hai — length same rehti hai
- Purane elements overwrite ho jaate hain

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Arrays</h1>

<h2>The copyWithin() Method</h2>

<p>copyWithin() copies array elements to another position in an array, overwriting existing values.</p><p>Copy to index 2, the elements from index 0 to 2:</p>

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

<script>

```
const fruits = ["Banana", "Orange", "Apple", "Mango", "Kiwi", "Papaya"];
```

```
document.getElementById("demo").innerHTML = fruits.copyWithin(2,0,2);
```

</script>

</body>

</html>

flat() method

nested arrays (arrays ke andar arrays) ko flatten karta hai matlab sub-array elements ko main array me la deta hai.

- Naya array return karta hai

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- Original array change nahi hota
- Depth specify karke decide kar sakte ho kitna flatten karna hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The flat() Method</h2>
<p id="demo"></p>
<script>
const myArr = [[1,2],[3,4],[5,6]];
const newArr = myArr.flat();
document.getElementById("demo").innerHTML = newArr;
</script>
</body>
</html>
```

flatMap()

- flatMap() method do kaam ek saath karta hai:
- Map: Array ke har element par function apply karta hai
- Flatten: Agar function nested array return karta hai, to 1 level flatten kar deta hai
- Hamesha 1-level flatten hoti hai
- Original array change nahi hota
- Return ek naya array hota ha

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The flatMap() Method</h2>
<p id="demo"></p>
<script>
const myArr = [1, 2, 3, 4, 5,6];
const newArr = myArr.flatMap(x => [x, x * 10]);
document.getElementById("demo").innerHTML = newArr;
</script>
</body>
</html>
```

Array.splice()

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

splice() method array ke elements ko remove, add ya replace karne ke liye use hota hai.

- Original array modify (mutate) hota hai
- Return value: removed elements ka array

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The splice() Method</h2>
<p>The splice() method adds new elements to an array, and returns an array with the deleted
elements (if any):</p>
<p id="demo1"></p>
<p id="demo2"></p>
<p id="demo3"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = "Original Array:<br> " + fruits;
let removed = fruits.splice(2, 2, "Lemon", "Kiwi");
document.getElementById("demo2").innerHTML = "New Array:<br>" + fruits;
document.getElementById("demo3").innerHTML = "Removed Items:<br> " + removed;
</script>
</body>
</html>
```

Using splice() to Remove Elements

- **start** → index jahan se removal start karna hai
- **deleteCount** → kitne elements remove karne hain

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The splice() Method</h2>
<p>The splice() methods can be used to remove array elements:</p>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits;
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
fruits.splice(0, 1);
document.getElementById("demo2").innerHTML = fruits;
</script>
</body>
</html>
```

slice() method array ka ek part nikalta hai aur naya array return karta hai.

- Original array change nahi hota
- Negative index allowed
- End index excluded hota hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The slice() Method</h2>
<p>Slice out a part of an array starting from array element 2 ("Lemon"):</p>
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];
const citrus = fruits.slice(2);
document.getElementById("demo").innerHTML = fruits + "<br><br>" + citrus;
</script>
</body>
</html>
```

Array Search Methods

<u>Array indexOf()</u>	<u>Array find()</u>
<u>Array lastIndexOf()</u>	<u>Array findIndex()</u>
<u>Array includes()</u>	<u>Array findLast()</u>
	<u>Array findLastIndex()</u>

JavaScript Array indexOf()

indexOf() method array me element ka index return karta hai.

- Agar element array me hai → index return hota hai (0-based)
- Agar element nahi hai → -1 return hota hai

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

JavaScript Array lastIndexOf()

lastIndexOf() method array me element ka last occurrence ka index return karta hai.

- Agar element array me hai → last index return hota hai
- Agar element nahi hai → -1 return hota hai
- Ye indexOf() ke opposite hai jo first occurrence return karta hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The lastIndexOf() Method</h2>
<p id="demo"></p>
<script>
const fruits = ["Apple", "Orange", "Apple", "Mango"];
let position = fruits.lastIndexOf("Apple") + 1;
document.getElementById("demo").innerHTML = "Apple is found in position " + position;
</script>
</body>
</html>
```

JavaScript Array includes()

- includes() method check karta hai koi element array me exist karta hai ya nahi.
- Returns true agar element array me ho
- Returns false agar element array me na ho
- NaN bhi detect kar sakta hai (jo indexOf nahi kar pata)
- Original array mutate nahi hota

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The includes() Method</h2>
<p>Check if the fruit array contains "Mango":</p>
<p id="demo"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.includes("Mango");
</script></body>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

</html>

JavaScript Array find() method array ka pehla element return karta hai jo condition satisfy kare.

- Condition callback function me define hoti hai
- Agar koi element match nahi kare → undefined return hota hai
- Original array mutate nahi hota

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The find() Method</h2>
<p id="demo"></p>
<script>
const numbers = [4, 9, 16, 25, 29];
let first = numbers.find(myFunction);
document.getElementById("demo").innerHTML = "First number over 18 is " + first;
function myFunction(value, index, array) {
return value > 18;
}
</script>
</body>
</html>
```

JavaScript Array findIndex()

The findIndex() method returns the index of the first array element that passes a test function. This example finds the index of the first element that is larger than 18:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The findIndex() Method</h2>
<p id="demo"></p>
<script>
const numbers = [4, 9, 16, 25, 29];
document.getElementById("demo").innerHTML = "First number over 18 has index " +
numbers.findIndex(myFunction);
function myFunction(value, index, array) {
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
return value > 18;
    }
</script>
</body>
</html>
```

JavaScript Array findLast() Method

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The findLast() Method</h2>
<p id="demo"></p>
<script>
const temp = [27, 28, 30, 40, 42, 35, 30];
let high = temp.findLast(x => x > 40);
document.getElementById("demo").innerHTML = "The last temperature over 40 was " +
high;
</script>
</body>
</html>
```

JavaScript Array Sort

Alphabetic Sort

[Array.sort\(\)](#)
[Array.reverse\(\)](#)
[Array.toSorted\(\)](#)
[Array.toReversed\(\)](#)
[Sorting Objects](#)

Numeric Sort

[Numeric Sort](#)
[Random Sort](#)
[Math.min\(\)](#)
[Math.max\(\)](#)
[Home made Min\(\)](#)
[Home made Max\(\)](#)

Sorting Arrays in JavaScript

JavaScript me arrays ko sort karne ke liye sort() method use hota hai.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

- Default behavior: array elements ko strings ke hisaab se alphabetically sort karta hai
- Original array mutate hota hai (naya array nahi banata)

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The sort() Method</h2>
<p>The sort() method sorts an array alphabetically:</p>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits;
fruits.sort();
document.getElementById("demo2").innerHTML = fruits;
</script>
</body>
</html>
```

Reversing an Array

The reverse() method reverses the elements in an array:
reverse() method array ke elements ka order ulta kar deta hai.

- Original array mutate hota hai
- Return value = reversed array

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Sort in Reverse</h2>
<p>The reverse() method reverses the elements in an array.</p>
<p>By combining sort() and reverse() you can sort an array in descending order:</p>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
// Create and display an array:
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo1").innerHTML = fruits;
// First sort the array
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
fruits.sort();
// Then reverse it:
fruits.reverse();
document.getElementById("demo2").innerHTML = fruits;
</script>
</body>
</html>
```

Numeric Sort

Default sort() behavior:

- Sorts values as strings by default.
- Works fine for words ("Apple" comes before "Banana").
- Fails for numbers: [100, 25, 5, 50].sort() → [100, 25, 5, 50]

Why numbers sort incorrectly:

- "25" vs "100" → compares first character "2" vs "1"
- Alphabetical/string comparison, not numeric

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The sort() Method</h2>
<p>Sort the array in descending order:</p>
<p id="demo1"></p>
<p id="demo2"></p>
<script>
const points = [40, 100, 1, 5, 25, 10];
document.getElementById("demo1").innerHTML = points;
points.sort(function(a, b){return b - a});
document.getElementById("demo2").innerHTML = points;
</script>
</body>
</html>
```

Compare ka matlab hota hai do ya zyada cheezon ko dekhna aur unke similarities aur differences batana.

Example: Cats aur Dogs ko compare karna = kya similar hai, kya alag hai.

2. Kyun Compare karte hain:

- Cheezein better samajhne ke liye

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

- Differences aur similarities dikhane ke liye
- Essay ya report me explanation clear karne ke liye

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The sort() Method</h2>
<p>Click the buttons to sort the array alphabetically or numerically.</p>
<button onclick="myFunction1()">Sort Alphabetically</button>
<button onclick="myFunction2()">Sort Numerically</button>
<p id="demo"></p>
<script>
const points = [40, 100, 1, 5, 25, 10];
document.getElementById("demo").innerHTML = points;
function myFunction1() {
  points.sort();
  document.getElementById("demo").innerHTML = points;
}
function myFunction2() {
  points.sort(function(a, b){return a - b});
  document.getElementById("demo").innerHTML = points;
}
</script>
</body>
</html>
```

Sorting an Array in Random Order

- Normal sort me elements order me arrange hote hain (chhota→bada, A→Z).
- Random order me elements randomly mix ho jaate hain, matlab har baar alag order me dikhte hain.
- Is process ko shuffle karna kehte hain.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The sort() Method</h2>
<p>Click the button (again and again) to sort the array in random order.</p>
<button onclick="myFunction()">Try it</button>
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<p id="demo"></p>
<script>
const points = [40, 100, 1, 5, 25, 10];
document.getElementById("demo").innerHTML = points;
function myFunction() {
  points.sort(function() {return 0.5 - Math.random()});
  document.getElementById("demo").innerHTML = points;
}
</script>
</body>
</html>
```

The Fisher Yates Method

- Ye ek algorithm hai jo array ko randomly shuffle karta hai.
- Matlab har permutation (order) equally possible hota hai.
- Computer science me isko Knuth shuffle bhi kehte hain.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Array Sort</h1>
<h2>The Fisher Yates Method</h2>
<p>Click the button (again and again) to sort the array in random order.</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
const points = [40, 100, 1, 5, 25, 10];
document.getElementById("demo").innerHTML = points;
function myFunction() {
  for (let i = points.length - 1; i > 0; i--) {
    let j = Math.floor(Math.random() * (i+1));
    let k = points[i];
    points[i] = points[j];
    points[j] = k;
  }
  document.getElementById("demo").innerHTML = points;
}
</script>
</body>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

</html>

Find the Lowest (or Highest) Array Value

- Array me lowest value = sabse chhota number
- Array me highest value = sabse bada number
- Array ko sort karo aur first/last element read karo
- Math.min() ya Math.max() use karo
- Apni khud ki function likho (loop karke elements compare karo)

Find Min or Max with sort()

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The sort() Method</h2>
<p>The highest number is <span id="demo"></span>.</p>
<script>
const points = [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return b-a});
document.getElementById("demo").innerHTML = points[0];
</script>
</body>
</html>
```

Math.min() directly array ke elements par work nahi karta, isliye spread operator

(...) use karna padta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Array Sort</h1>
<p>The lowest number is <span id="demo"></span>.</p>
<script>
const points = [40, 100, 1, 5, 25, 10];
document.getElementById("demo").innerHTML = myArrayMin(points);
function myArrayMin(arr) {
  return Math.min.apply(null, arr);
}
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</script>  
</body>  
</html>
```

Using Math.max() on an Array

Math.max() directly array par kaam nahi karta, spread operator (...) use karna padta hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Array Sort</h1>  
<p>The highest number is <span id="demo"></span>.</p>  
<script>  
const points = [40, 100, 1, 5, 25, 10];  
document.getElementById("demo").innerHTML = myArrayMax(points);  
function myArrayMax(arr) {  
  return Math.max.apply(null, arr);  
}  
</script>  
</body>  
</html>
```

JavaScript Array Minimum Method

- JavaScript me direct built-in function for minimum value exist nahi karta.
- Fastest way = home-made function.
- Idea: Loop through the array and compare each element with the current minimum.

```
<!DOCTYPE html>  
<html>  
<body>  
<h2>JavaScript Array Sort</h2>  
<p>The lowest number is <span id="demo"></span>.</p>  
<script>  
const points = [40, 100, 1, 5, 25, 10];  
document.getElementById("demo").innerHTML = myArrayMin(points);  
function myArrayMin(arr) {  
  let len = arr.length;  
  let min = Infinity;  
  while (len--) {  
    if (arr[len] < min) {
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
    min = arr[len];
  }
}
return min;
}
</script>
</body>
</html>
```

JavaScript Array Maximum Method

JavaScript me direct built-in function for maximum value exist nahi karta.

Fastest way = home-made function.

Idea: Loop through the array and compare each element with the current maximum.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript Array Sort</h2>
```

```
<p>The highest number is <span id="demo"></span>.</p>
```

```
<script>
```

```
const points = [40, 100, 1, 5, 25, 10];
```

```
document.getElementById("demo").innerHTML = myArrayMax(points);
```

```
function myArrayMax(arr) {
```

```
  let len = arr.length;
```

```
  let max = -Infinity;
```

```
  while (len--) {
```

```
    if (arr[len] > max) {
```

```
      max = arr[len];
```

```
    }
```

```
  }
```

```
  return max;
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Stable Array sort()

- Stable sort ka matlab:
Agar array me equal elements hain, to unka original order preserve hota hai.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

- JavaScript me sort() method mostly stable hai (modern browsers me).

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript Stable Sort</h1>
```

```
<p>From ES2019, browsers must use a stable sorting algorithm.</p>
```

```
<p>When sorting elements on a key, the elements must keep their relative position to other objects with the same key.</p>
```

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

```
<script>
```

```
const myArr = [
```

```
  {name:"X00",price:100 },
```

```
  {name:"X01",price:100 },
```

```
  {name:"X02",price:100 },
```

```
  {name:"X03",price:100 },
```

```
  {name:"X04",price:110 },
```

```
  {name:"X05",price:110 },
```

```
  {name:"X06",price:110 },
```

```
  {name:"X07",price:110 },
```

```
  {name:"X08",price:120 },
```

```
  {name:"X09",price:120 },
```

```
  {name:"X10",price:120 },
```

```
  {name:"X11",price:120 },
```

```
  {name:"X12",price:130 },
```

```
  {name:"X13",price:130 },
```

```
  {name:"X14",price:130 },
```

```
  {name:"X15",price:130 },
```

```
  {name:"X16",price:140 },
```

```
  {name:"X17",price:140 },
```

```
  {name:"X18",price:140 },
```

```
  {name:"X19",price:140 }]
```

```
];
```

```
myArr.sort( (p1, p2) => {
```

```
  if (p1.price < p2.price) return -1;
```

```
  if (p1.price > p2.price) return 1;
```

```
  return 0;
```

```
});
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
let txt = "";
myArr.forEach(myFunction);
function myFunction(value) {
  txt += value.name + " " + value.price + "<br>";
}
document.getElementById("demo").innerHTML = txt;
</script>
</body>
</html>
```

Array Iteration

- Ye methods array ke har element pe ek function apply karne ka ek structured tarika dete hain.
- Instead of writing a manual for loop, aap forEach(), map(), filter(), reduce(), some(), every(), etc. jaise built-in methods ka use kar sakte hain.
- Ye methods readability badhate hain aur functional programming style ko support karte hain.

Array Iteration Methods

Array iteration methods operate on every array item.

Array forEach

Array map()

Array flatMap()

Array filter()

Array reduce()

Array reduceRight()

Array every()

Array some()

Array from()

Array keys()

Array entries()

Array with()

Array Spread (...)

Array Rest (...)

JavaScript Array forEach()

JavaScript ka forEach() method ek array ke har element ke liye ek function ko call karta hai. Simple words me: Agar tumhare paas ek array hai, to forEach() usme se ek-ek element utha kar tumhare diye hue function ko chalata hai.

```
<!DOCTYPE html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The forEach() Method</h2>
<p>Call a function once for each array element:</p>
<p id="demo"></p>
<script>
const numbers = [45, 4, 9, 16, 25];
let txt = "";
numbers.forEach(myFunction);
document.getElementById("demo").innerHTML = txt;
function myFunction(value, index, array) {
  txt += value + "<br>";
}
</script>
</body>
</html>
```

JavaScript Array map()

JavaScript ka map() method ek naya array create karta hai jisme har element par function apply karke uska result store hota hai.

Important Points

- map() har element par function apply karta hai.
- Ye naya array return karta hai (original array ko change nahi karta).
- Ye un elements ke liye function execute nahi karta jinki value exist nahi karti (empty slots).

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The map() Method</h2>
<p>Create a new array by performing a function on each array element:</p>
<p id="demo"></p>
<script>
const numbers1 = [45, 4, 9, 16, 25];
const numbers2 = numbers1.map(myFunction);
document.getElementById("demo").innerHTML = numbers2;
function myFunction(value) {
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
return value * 2;
}
</script>
</body>
</html>
```

JavaScript Array flatMap()

flatMap() JavaScript me ES2019 (ECMAScript 2019) me add hua tha. Ye method 2 kaam ek saath karta hai:

1. Pehle array ke har element par map() jaisa function apply karta hai
2. Phir result ko 1-level deep flatten (flat) kar deta hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The flatMap() Method</h2>
<p id="demo"></p>
<script>
const myArr = [1, 2, 3, 4, 5,6];
const newArr = myArr.flatMap(x => [x, x * 10]);
document.getElementById("demo").innerHTML = newArr;
</script>
</body>
</html>
```

JavaScript Array filter()

JavaScript ka filter() method ek naya array banata hai, jisme sirf wohi elements aate hain jo given test (condition) pass karte hain.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The filter() Method</h2>
<p>Create a new array with all array elements that passes a test.</p>
<p id="demo"></p>
<script>
const numbers = [45, 4, 9, 16, 25];
const over18 = numbers.filter(myFunction);
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML = over18;
function myFunction(value) {
  return value > 18;
}
</script>
</body>
</html>
```

JavaScript Array reduce()

JavaScript ka reduce() method array ke har element par ek function run karta hai aur aakhri me ek single value return karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The reduce() Method</h2>
<p>Find the sum of all numbers in an array:</p>
<p id="demo"></p>
<script>
const numbers = [45, 4, 9, 16, 25];
let sum = numbers.reduce(myFunction);
document.getElementById("demo").innerHTML = "The sum is " + sum;
function myFunction(total, value) {
  return total + value;
}
</script>
</body>
</html>
```

JavaScript Array reduceRight()

reduceRight() bilkul reduce() jaisa hi hota hai, bas ek difference:

Ye array ko right-to-left process karta hai (matlab last element se start karta hai).

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The reduceRight() Method</h2>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<p>Find the sum of all numbers in an array:</p>
<p id="demo"></p>
<script>
const numbers = [45, 4, 9, 16, 25];
let sum = numbers.reduceRight(myFunction);
document.getElementById("demo").innerHTML = "The sum is " + sum;
function myFunction(total, value) {
  return total + value;
}
</script>
</body>
</html>
```

JavaScript Array every()

every() method array ke har element ko check karta hai aur dekhta hai ki kya saare elements given condition (test) ko pass karte hain ya nahi. Agar saare elements test pass kar lein, to true return hota hai. Agar ek bhi element fail ho jaye, to false return hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The every() Method</h2>
<p>The every() method checks if all array values pass a test.</p>
<p id="demo"></p>
<script>
const numbers = [45, 4, 9, 16, 25];
let allOver18 = numbers.every(myFunction);
document.getElementById("demo").innerHTML = "All over 18 is " + allOver18;
function myFunction(value) {
  return value > 18;
}
</script>
</body>
</html>
```

JavaScript Array some()

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

some() method array me check karta hai ki kya kam se kam ek element given condition (test) pass karta hai. Agar ek bhi element condition pass kar de, to true return hota hai. Agar ek bhi element pass na kare, to false return hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The some() Method</h2>
<p>The some() method checks if some array values pass a test:</p>
<p id="demo"></p>
<script>
const numbers = [45, 4, 9, 16, 25];
let someOver18 = numbers.some(myFunction);
document.getElementById("demo").innerHTML = "Some over 18 is " + someOver18;
function myFunction(value) {
  return value > 98;
}
</script>
</body>
</html>
```

JavaScript Array.from()

Array.from() method kisi bhi iterable object (jaise string, set, map) ya kisi bhi object jisme length property ho (jaise array-like objects) ko real array me convert kar deta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The from() Method</h2>
<p>The from() method can return an array from any variable with a length property.</p>
<p>It can return a string as an array:</p>
<p id="demo"></p>
<script>
const myNumbers = [1,2,3,4];
const myArr = Array.from(myNumbers, (x) => x * 2);
document.getElementById("demo").innerHTML = myArr;
</script>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</body>  
</html>
```

JavaScript Array keys()

Array.keys() method ek Array Iterator object return karta hai jisme array ke keys (yaani indexes) hote hain.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Arrays</h1>  
<h2>The keys() Method</h2>  
<p>Return an Array Iterator object with the keys of the array:</p>  
<p id="demo"></p>  
<script>  
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
const keys = fruits.keys();  
let text = "";  
for (let x of keys) {  
  text += x + "<br>";  
}  
document.getElementById("demo").innerHTML = text;  
</script>  
</body>  
</html>
```

JavaScript Array entries()

Array.entries() method ek Array Iterator object return karta hai jisme har element ka [index, value] pair hota hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Arrays</h1>  
<h2>The entries() method</h2>  
<p>entries() returns an Array Iterator object with key/value pairs:</p>  
<p id="demo"></p>  
<script>  
const fruits = ["Banana", "Orange", "Apple", "Mango"];  
const f = fruits.entries();
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
for (let x of f) {
  document.getElementById("demo").innerHTML += x + "<br>";
}
</script>
</body>
</html>
```

JavaScript Array with() Method

Ye method array ka ek naya copy banata hai aur usme specified index ka value replace karta hai, original array ko change kiye bina.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The with() Method</h2>
<p id="demo"></p>
<script>
const months = ["Januar", "Februar", "Mar", "April"];
const myMonths = months.with(2, "March");
document.getElementById("demo").innerHTML = myMonths;
</script>
</body>
</html>
```

JavaScript Array Spread (...)

JavaScript me spread operator ... ka use array ko individual elements me expand karne ke liye hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Operators</h1>
<h2>The ... Operator</h2>
<p>The spread operator (...) can be used to join arrays:</p>
<p id="demo"></p>
<script>
const arr1 = [1, 2, 3];
const arr2 = [4, 5, 6];
const arr3 = [...arr1, ...arr2];
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML = arr3;  
</script>  
</body>  
</html>
```

JavaScript Array Rest (...)

JavaScript me rest operator ... ka use array destructuring ke saath bacha hua (leftover) elements ko ek nayi array me collect karne ke liye hota hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Arrays</h1>  
<h2>Rest Elements</h2>  
<p>The rest operator (...) allows us to destruct an array and collect the leftovers:</p>  
<p id="demo"></p>  
<script>  
let a, b, rest;  
const arr1 = [1,2,3,4,5,6,7,8];  
[a, ...rest] = arr1;  
document.getElementById("demo").innerHTML = "The rest is: " + rest;  
</script>  
</body>  
</html>
```

JavaScript Array Reference

Method / Property	Description
[]	Naya array banata hai
new Array()	Naya array banata hai
at()	Array ka indexed element return karta hai
concat()	Arrays ko join karke naya array return karta hai
constructor	Wo function return karta hai jisne Array prototype banaya
copyWithin()	Array ke elements ko copy karke ek specified position me daalta hai
entries()	Key/value pair ka Array Iterator object return karta hai
every()	Check karta hai ki sab elements test pass karte hain ya nahi
fill()	Array ke elements ko ek static value se fill karta hai

LEARNING HUB
SHAHABAD MARKANDA
 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Method / Property	Description
filter()	Naya array banata hai jisme sirf elements jo test pass karte hain
find()	Pehla element return karta hai jo test pass kare
findIndex()	Pehla element ka index return karta hai jo test pass kare
findLast()	Last element return karta hai jo test pass kare
findLastIndex()	Last element ka index return karta hai jo test pass kare
flat()	Sub-array elements ko flatten karta hai
flatMap()	Har element ko map karke ek new flattened array banata hai
forEach()	Har element ke liye ek function call karta hai
from()	Iterable object ya array-like object se array banata hai
includes()	Check karta hai ki array me specified element hai ya nahi
indexOf()	Array me element search karke uska position return karta hai
isArray()	Check karta hai ki object array hai ya nahi
join()	Array ke sab elements ko ek string me join karta hai
keys()	Array Iterator return karta hai jisme indexes (keys) hote hain
lastIndexOf()	Array me end se search karke element ka position return karta hai
length	Array ke elements ka number set ya return karta hai
map()	Har element par function apply karke naya array banata hai
of()	Arguments se array banata hai
pop()	Last element remove karke return karta hai
prototype	Array object me properties aur methods add karne deta hai
push()	Array ke end me elements add karta hai aur new length return karta hai
reduce()	Array ko single value me reduce karta hai (left-to-right)
reduceRight()	Array ko single value me reduce karta hai (right-to-left)
reverse()	Array ke elements ka order reverse karta hai
shift()	First element remove karke return karta hai
slice()	Array ka ek part select karke naya array return karta hai
some()	Check karta hai ki koi element test pass karta hai ya nahi
sort()	Array ke elements sort karta hai

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Method / Property	Description
splice()	Array me elements add ya remove karta hai
toReversed()	Naya array return karta hai jisme elements reversed order me hote hain
toSorted()	Naya array return karta hai jisme elements sorted hote hain
toSpliced()	Naya array return karta hai jisme elements add ya remove hote hain
toString()	Array ko string me convert karta hai
unshift()	Array ke beginning me elements add karta hai aur new length return karta hai
valueOf()	Array ka primitive value return karta hai
with()	Naya array return karta hai jisme specified elements update hote hain

JavaScript Array Const

JavaScript const with Arrays

- const ka matlab hai variable ka reference constant hai, value immutable nahi hoti.
- Agar array const se declare ho, to array ka reference change nahi ho sakta, lekin array ke elements ko modify karna allowed hai.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript const</h2>
<p id="demo"></p>
<script>
const cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars;
</script>
</body>
</html>
```

Cannot be Reassigned

- Array ka reference fix ho jata hai
- Aap poore array ko naye array se replace nahi kar sakte

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- Lekin elements ko modify karna allowed hai

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript const</h2>
<p>You can NOT reassign a constant array:</p>
<p id="demo"></p>
<script>
try {
  const cars = ["Saab", "Volvo", "BMW"];
  cars = ["Toyota", "Volvo", "Audi"];
}
catch (err) {
  document.getElementById("demo").innerHTML = err;
}
</script>
</body>
</html>
```

Arrays are Not Constants

JavaScript me const keyword thoda misleading ho sakta hai const array ko constant nahi banata. Ye sirf array ka reference constant banata hai Matlab: array ka reference change nahi ho sakta, lekin elements ko modify karna allowed hai.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript const</h2>
<p>Declaring a constant array does NOT make the elements unchangeable:</p>
<p id="demo"></p>
<script>
// Create an Array:
const cars = ["Saab", "Volvo", "BMW"];
// Change an element:
cars[0] = "Toyota";
// Add an element:
cars.push("Audi");
// Display the Array:
document.getElementById("demo").innerHTML = cars;
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
</script>  
</body>  
</html>
```

Assigned when Declared

JavaScript me const variables ko declare karte waqt value assign karna mandatory hai.

- Matlab: const array ko declare karte hi initialize karna padta hai
- Baad me reassign allowed nahi hai

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Arrays</h1>  
<h2>JavaScript Hoisting</h2>  
<p id="demo"></p>  
<script>  
cars = ["Saab", "Volvo", "BMW"];  
var cars;  
document.getElementById("demo").innerHTML = cars[0];  
</script>  
</body>  
</html>
```

Const Block Scope

const se declared array ka scope block-level hota hai:

- Matlab agar array block ke andar declare kiya gaya hai ({ }), to wo sirf usi block ke andar accessible hota hai
- Block ke bahar uska reference exist nahi karta

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Arrays</h1>  
<h2>Declaring an Array Using const</h2>  
<p id="demo"></p>  
<script>  
const cars = ["Saab", "Volvo", "BMW"];  
// Here cars[0] is "Saab"  
{  
  const cars = ["Toyota", "Volvo", "BMW"];
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
// Here cars[0] is "Toyota"
}
// Here cars[0] is "Saab"
document.getElementById("demo").innerHTML = cars[0];
</script>
</body>
</html>
```

Redeclaring Arrays

Agar array var se declare kiya gaya hai, to aap program me kahin bhi same name se dobara declare kar sakte ho

Ye hoisting aur global/function scope ki wajah se possible hai

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Array Demo</title>
</head>
<body>
  <script>
    // VAR: Redeclaration allowed
    var arrVar = [1,2,3];
    var arrVar = [4,5,6];
    console.log("VAR:", arrVar); // [4,5,6]
    // LET: Redeclaration NOT allowed, modify elements allowed
    let arrLet = [10,20,30];
    arrLet[0] = 100;
    console.log("LET:", arrLet); // [100,20,30]
    // CONST: Redeclaration NOT allowed, modify elements allowed
    const arrConst = ["apple","banana"];
    arrConst.push("mango");
    arrConst[0] = "orange";
    console.log("CONST:", arrConst); // ["orange","banana","mango"]
    // Block scope demo
    const blockArr = [1,2,3];
    {
      const blockArr = [4,5,6];
      console.log("Inside block:", blockArr); // [4,5,6]
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
}  
  console.log("Outside block:", blockArr); // [1,2,3]  
</script>  
</body>  
</html>
```

Typed Arrays

Typed Arrays JavaScript me binary data handle karne ke liye design kiye gaye hain.

Key Points:

- **Fixed Length:** Typed array ka size declare karte waqt fix ho jata hai aur baad me change nahi hota.
- **Fixed Type:** Har element ka type fixed hota hai, jaise Int8, Uint8, Float32, etc.
- **Binary Data:** Ye arrays raw binary data ko efficiently store aur manipulate karte hain.
- **Underlying Buffer:** Typed arrays ArrayBuffer ke upar kaam karte hain.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Typed Arrays</h1>  
<h2>The Int8Array Object</h2>  
<p>Create a typed array of 5 8-bit integers (byte format):</p>  
<p id="demo"></p>  
<script>  
const myArr = Uint8Array.from([0,1,2,3,4]);  
document.getElementById("demo").innerHTML = "Array: " + myArr + "<br><br>Bytes per  
element: " + myArr.BYTES_PER_ELEMENT;  
</script>  
</body>  
</html>
```

Typed Array Benefits

Fixed Type & Fixed Length

- Typed arrays me sab elements same type ke hote hain (jaise Int8, Float32)
- Size declare karte waqt fix ho jata hai

Raw Memory

- Ye directly binary memory use karte hain
- JavaScript functions ko without conversion pass kiya ja sakta hai

High Performance

- Normal arrays se kaafi faster

- ALL COMPUTER COURSES
- ENGLISH LANGUAGE COURSES
- Brand Marketing & Custom Software Solution
- TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- Binary data ke saath intensive computation me efficient

Differences from Regular Arrays

- Fixed Length- Typed Arrays ka size fix hota hai. Matlab, push() ya pop() use karke aap uska size change nahi kar sakte. Agar zyada elements chahiye, toh naya typed array create karna padega. Normal arrays me aap kabhi bhi size badha ya ghata sakte ho, par typed arrays me nahi.
- Type Restriction- Typed Arrays me sirf ek hi type ka data store hota hai. Jaise Int8Array me sirf integers store kar sakte ho, strings ya floats allowed nahi. Normal arrays me aap mix type ka data store kar sakte ho, jaise numbers, strings, objects sab ek saath.
- Underlying Buffer- Typed Arrays ek ArrayBuffer ka view hote hain. Matlab aap directly memory ke binary data ko access aur manipulate kar sakte ho. Normal arrays me ye direct memory access possible nahi hota, aur wo thoda slow bhi hota hai large data ke liye.

Typed Array Types

Name	Min	Max	Bytes	Type	Explanation
Int8Array	-128	127	1 byte	Signed integer	Negative aur positive numbers store karta hai, 1 byte me.
Uint8Array	0	255	1 byte	Unsigned integer	Sirf positive numbers store karta hai, 0–255 range me.
Uint8ClampedArray	0	255	1 byte	Unsigned integer (clamped)	Positive numbers store karta hai, aur agar value 0–255 ke bahar ho, to automatically clamp ho jaata hai. Mostly pixels ke liye use hota hai.
Int16Array	-32768	32767	2 bytes	Signed short integer	Thode bade negative aur positive numbers store kar sakta hai.

LEARNING HUB
SHAHABAD MARKANDA
 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Name	Min	Max	Bytes	Type	Explanation
UInt16Array	0	65535	2 bytes	Unsigned short integer	Sirf positive numbers store karta hai, 2 bytes me.
Int32Array	-2147483648	2147483647	4 bytes	Signed long integer	Aur bade numbers store karne ke liye, 4 bytes use karta hai.
UInt32Array	0	4294967295	4 bytes	Unsigned long integer	Sirf positive numbers, 4 billion tak store kar sakta hai.
BigInt64Array	-2^{63}	$2^{63}-1$	8 bytes	BigInt	Bahut bade signed integers store kar sakta hai, 8 bytes me.
BigUint64Array	0	$2^{64}-1$	8 bytes	Unsigned BigInt	Sirf positive bahut bade integers store karta hai.
Float16Array	-65504	65504	2 bytes	Half-precision float	Decimal numbers store karta hai, thoda less precise, 2 bytes me.
Float32Array	$-3.4e38$	$3.4e38$	4 bytes	Float (single precision)	Normal decimal numbers store karta hai, 4 bytes me.
Float64Array	$-1.8e308$	$1.8e308$	8 bytes	Double precision float	Highly precise decimal numbers store karta hai, scientific calculations ke liye.

8 Bit Integers

s	Data Type	Range	Explanation (Hinglish)
Int8Array	Signed integer (byte)	-128 to 127	Ye negative aur positive numbers store karta hai, 1 byte me.

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

s	Data Type	Range	Explanation (Hinglish)
Uint8Array	Unsigned integer (octet)	0 to 255	Sirf positive numbers store karta hai, 1 byte me.
Uint8ClampedArray	Unsigned integer (octet)	0 to 255	Positive numbers store karta hai aur agar value 0–255 ke bahar ho, to automatically clamp ho jaata hai (0 ya 255 pe set). Mostly pixels/images ke liye use hota hai.

```

<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The Uint8Array Object</h2>
<p>Create a typed array of 10 unsigned 8-bit integers (octet format):</p>
<p id="demo"></p>
<script>
const myArr = new Uint8Array(10);
document.getElementById("demo").innerHTML = "Array: " + myArr + "<br>Bytes per
element: " + myArr.BYTES_PER_ELEMENT;
</script>
</body>
</html>

```

Uint8Array vs Uint8ClampedArray

- Ye typed array hai jo 0–255 ke values store karta hai.
- Agar aap koi value set karte ho jo 0–255 ke range ke bahar hai, ye first 8 bits ko hi consider karega.
- Matlab, extra bits ignore ho jaati hain.

16-Bits Integers

Typed Array	Type	Bits	Range	Behavior / Notes
Uint8Array	Unsigned 8-bit integer	8	0 – 255	Agar value 0–255 ke bahar ho, last 8 bits hi store hote hain
Uint8ClampedArray	Unsigned 8-bit integer (clamped)	8	0 – 255	Agar value <0 ho → 0, agar >255 ho → 255 (clamped)

LEARNING HUB
SHAHABAD MARKANDA
 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Typed Array	Type	Bits	Range	Behavior / Notes
Int16Array	Signed 16-bit integer	16	-32,768 – 32,767	First bit sign ke liye, negative numbers allowed
Uint16Array	Unsigned 16-bit integer	16	0 – 65,535	Negative values nahi, agar value >65,535 → wrap around

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The Uint16Array Object</h2>
<p id="demo"></p>
<script>
const myArr = new Uint16Array(10);
document.getElementById("demo").innerHTML = "Array: " + myArr + "<br>Bytes per
element: " + myArr.BYTES_PER_ELEMENT;
</script>
</body>
</html>
```

32-Bit Integers

Typed Array	Type	Bits	Range	Behavior / Notes
Int32Array	Signed 32-bit int	32	-2,147,483,648 → 2,147,483,647	Positive aur negative numbers dono allowed
Uint32Array	Unsigned 32-bit int	32	0 → 4,294,967,295	Sirf positive numbers, overflow ya wrap-around ho sakta hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The Uint32Array Object</h2>
<p>Create a typed array of 10 unsigned 32-bit integers (unsigned long format):</p>
<p id="demo"></p>
<script>
const myArr = new Uint32Array(10);
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML = "Array: " + myArr + "<br>Bytes per
element: " + myArr.BYTES_PER_ELEMENT;
</script>
</body>
</html>
```

64-Bit Integers

Typed Array	Type	Bits	Range	Behavior / Notes
BigInt64Array	Big signed integer	64	$-2^{63} \rightarrow 2^{63} - 1$	Positive aur negative numbers dono allowed
BigUint64Array	Big unsigned integer	64	$0 \rightarrow 2^{64}$	Sirf positive numbers, large integers store kar sakta hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The BigInt64Array Object</h2>
<p>Create a typed array of 10 signed 64-bit integers (bigint format):</p>
<p id="demo"></p>
<script>
const myArr = new BigInt64Array(10);
document.getElementById("demo").innerHTML = "Array: " + myArr + "<br>Bytes per
element: " + myArr.BYTES_PER_ELEMENT;</script>
</body>
</html>
```

Floating Point Numbers

Name	Description	Range
Float16Array	Half precision float. Matlab ye chhoti memory use karta hai aur ~3 significant decimal digits accurate hota hai.	-65504 to 65504
Float32Array	Normal precision float. Matlab 7 significant decimal digits tak accurate hota hai. Ye commonly use hota hai.	-3.4×10^{38} to 3.4×10^{38}
Float64Array	Double precision float. Matlab 15 significant decimal digits tak accurate. Large calculations ke liye best hai.	-1.8×10^{308} to 1.8×10^{308}

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The Float64Array Object</h2>
<p>Create a typed array of 10 floating point numbers in (double precision) 64-bit
format:</p>
<p id="demo"></p>
<script>
const myArr = new Float64Array(10);
document.getElementById("demo").innerHTML = "Array: " + myArr + "<br>Bytes per
element: " + myArr.BYTES_PER_ELEMENT;
</script>
</body>
</html>
```

Typed Array Methods

The from() Method

from() method creates a new typed array from any iterable object (like array, string, or another array-like object).

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The from() Method</h2>
<p id="demo"></p>
<script>
// Create a Typed Array
const myArr = Int16Array.from([1,2,3,4,5,6,7,8,9,0]);
document.getElementById("demo").innerHTML = myArr;
</script>
</body>
</html>
```

The of() Method

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

Purpose: of() method creates a new typed array directly from a list of numbers (arguments) you provide.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The of() Method</h2>
<p id="demo"></p>
<script>
// Create a Typed Array
const myArr = Int32Array.of(1,2,3,4,5,6)
document.getElementById("demo").innerHTML = myArr;
</script>
</body>
</html>
```

The constructor.name Property

- Purpose: constructor.name property typed array ka type (name) batata hai.
- Use: Jab aapko check karna ho ki array kis typed array ka instance hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The constructor.name Property</h2>
<p id="demo"></p>
<script>
// Create a Typed Array
const myArr = new Int32Array(10);
// Display constructor.name
document.getElementById("demo").innerHTML = myArr.constructor.name;
</script>
</body>
</html>
```

BYTES_PER_ELEMENT Property

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE:** <https://learninghubshahabad.in>

Purpose: BYTES_PER_ELEMENT batata hai har element ko store karne ke liye kitne bytes memory use ho rahe hain.

Use: Useful jab aapko memory consumption samajhna ho, ya arrays ka size calculate karna ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The BYTES_PER_ELEMENT Property</h2>
<p>The number of bytes used to store one Int32Array element is:</p>
<p id="demo"></p>
<script>
// Create a Typed Array
const myArr = new Int32Array(10);
// Display the number of bytes per element
document.getElementById("demo").innerHTML = myArr.BYTES_PER_ELEMENT;
</script>
</body>
</html>
```

Common Array Methods

Iteration (Array ke elements ko loop karna)

- `forEach()` → Array ke har element pe function apply karta hai, return kuch nahi karta.
- `map()` → Har element pe function apply karke naya array return karta hai.
- `filter()` → Condition match karne wale elements ka naya array return karta hai.
- `reduce()` → Array ke elements ko combine karke ek single value deta hai.
- `reduceRight()` → Reduce jaise hi hai, lekin right se left direction me kaam karta hai.
- `every()` → Check karta hai ki sab elements condition satisfy karte hain ya nahi.
- `some()` → Check karta hai ki koi element condition satisfy karta hai ya nahi.
- `find()` → Pehla element return karta hai jo condition satisfy kare.
- `findIndex()` → Pehla element ka index return karta hai jo condition satisfy kare.
- `findLast()` → Last element return karta hai jo condition satisfy kare.
- `findLastIndex()` → Last element ka index return karta hai jo condition satisfy kare.

Searching (Array me element dhundna)

- `includes()` → Check karta hai ki element array me exist karta hai ya nahi.
- `indexOf()` → Pehla index return karta hai jaha element milta hai.
- `lastIndexOf()` → Last index return karta hai jaha element milta hai.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

Manipulation (Array ko modify karna)

- `at()` → Specific index ka element return karta hai (negative index se reverse bhi).
- `copyWithin()` → Array ke ek part ko dusre part me copy karta hai without changing array size.
- `fill()` → Array ke elements ko ek value se replace karta hai.
- `reverse()` → Array ke elements ka order reverse kar deta hai.
- `set()` → Typed Array me values set karne ke liye.
- `slice()` → Array ka subset return karta hai, original array change nahi hota.
- `sort()` → Array ke elements ko sort karta hai.
- `subarray()` → Typed Array ka view return karta hai, original data bhi modify ho sakta hai.

Conversion (Array ko string ya dusre format me convert karna)

- `join()` → Array ke elements ko string me join karta hai.
- `toLocaleString()` → Array ke elements ko local format me string me convert karta hai.
- `toString()` → Array ko string me convert karta hai.

Non-mutating methods (Original array change nahi hota)

- `toReversed()` → Reverse karke naya array return karta hai.
- `toSorted()` → Sort karke naya array return karta hai.
- `with()` → Specific index ka value change karke naya array return karta hai.

The fill() Method

Purpose: Typed Array ke saare elements ko ek hi value se replace kar deta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The fill() Method</h2>
<p id="demo"></p>
<script>
// Create Typed Array
const myArr = new Int16Array(10);
// Fill the Array
myArr.fill(200, 0, 3);
document.getElementById("demo").innerHTML = myArr;
</script>
</body>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

</html>

The find() Method

Purpose: Typed Array me se pehla element return karta hai jo given condition (test) satisfy kare.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The find() Method</h2>
<p>The value of the first element that contains a value over 18 is:</p>
<p id="demo"></p>
<script>
// Create a Typed Array
const myArr = Int16Array.from([10,15,20,25,30,35,40,45,50]);
document.getElementById("demo").innerHTML = myArr.find((x) => x > 18);
</script>
</body>
</html>
```

The some() Method

- Purpose: Typed Array me check karta hai ki koi bhi element aisi condition satisfy karta hai ya nahi.
- Agar koi element condition satisfy kare, to true return karta hai, warna false.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Typed Arrays</h1>
<h2>The some() Method</h2>
<p>Does the array contain a value over 18?</p>
<p id="demo"></p>
<script>
// Create a Typed Array
const myArr = Int16Array.from([10,15,20,25,30,35,40,45,50]);
document.getElementById("demo").innerHTML = myArr.some((x) => x > 18);
</script>
</body>
</html>
```

LEARNING HUB
SHAHABAD MARKANDA
 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Array Methods

Method	Array	Typed Array
pop()	Yes	NO
push()	Yes	NO
shift()	Yes	NO
unshift()	Yes	NO
splice()	Yes	NO
flat()	Yes	NO
flatMap()	Yes	NO
concat()	Yes	NO
toSpliced()	Yes	NO

Typed Array Types

Name	Range	Type
Int8Array	-128 to 127	1 byte signed integer (negative aur positive dono)
Uint8Array	0 to 255	1 byte unsigned integer (sirf positive)
Uint8ClampedArray	0 to 255	1 byte unsigned integer, overflow ko clamp karta hai 0-255 me
Int16Array	-32,768 to 32,767	2 bytes signed short
Uint16Array	0 to 65,535	2 bytes unsigned short
Int32Array	-2,147,483,648 to 2,147,483,647	4 bytes signed long
Uint32Array	0 to 4,294,967,295	4 bytes unsigned long

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Name	Range	Type
BigInt64Array	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	8 bytes signed big integer
BigUint64Array	0 to 18,446,744,073,709,551,615	8 bytes unsigned big integer

JavaScript Sets

Set ek aisa collection hai jo unique values ko store karta hai.
Matlab, agar aap same value ko dobara add karoge, Set usko ignore kar dega.

How to Create a Set

Agar tumhare paas array hai aur tumhe unique values ka collection chahiye, toh tum Set use kar sakte ho.

The new Set() Method

Set ek special type ka object hai jo sirf unique values store karta hai.
Agar tumhare paas array hai aur tum unique elements chahte ho, tum us array ko new Set() ke andar pass kar sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<h2>The add() Method</h2>
<p>The add() method adds values to a set:</p>
<p id="demo"></p>
<script>
// Create a Set
const letters = new Set();
// Create Variables
const a = "a";
const b = "b";
const c = "c";
// Add Values as Variables to the Set
letters.add(a);
letters.add(b);
letters.add(c);
// Display the Size
document.getElementById("demo").innerHTML = "The set has " + letters.size + " values.";
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</script>  
</body>  
</html>
```

The add() Method

add() method ka use Set mein ek nayi value add karne ke liye hota hai.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Sets</h1>  
<h2>The add() Method</h2>  
<p>The add() method adds values to a set:</p>  
<p id="demo"></p>  
<script>  
// Create a Set  
const letters = new Set();  
// Add Values to the Set  
letters.add("a");  
letters.add("b");  
letters.add("c");  
letters.add("c");  
letters.add("c");  
letters.add("c");  
letters.add("c");  
letters.add("c");  
letters.add("c");  
// Display the Size  
document.getElementById("demo").innerHTML = "The set has " + letters.size + " values.";  
</script>  
</body>  
</html>
```

Listing Set Elements

Set ke elements directly access nahi ho sakte jaise array mein arr[0].

Par for..of loop use karke easily saare elements iterate kar sakte ho.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Sets</h1>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<p>Ypu can iterate set values with for..of:</p>
<p id="demo"></p>
<script>
// Create a Set
const letters = new Set(["a","b","c"]);
// List all Elements
let text = "";
for (const x of letters) {
  text += x + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Sets are Objects

JavaScript mein Set bhi ek object hai.

- Set ke andar tum values store kar sakte ho.
- Set ke paas methods hote hain jaise .add(), .delete(), .has(), .clear() etc.
- Keys aur values same hote hain, kyunki Set sirf unique values store karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1><h2>The instanceof Operator</h2>
<p>The instanceof Set returns true:</p>
<p id="demo"></p>
<script>
// Create a new Set
const letters = new Set(["a","b","c"]);
// Display typeof
document.getElementById("demo").innerHTML = letters instanceof Set;
</script>
</body>
</html>
```

Basic Set Methods

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Basic Set Methods

- `new Set()`
- `add()`
- `clear()`
- `delete()`
- `entries()`
- `forEach()`
- `has()`
- `keys()`
- `values()`

Set Properties

- `size`

The new Set() Method

JavaScript mein Set ek special type ka object hai jo sirf unique values store karta hai. Agar tumhare paas ek array hai jisme duplicates ho sakte hain, toh tum usko new Set() constructor ke andar pass karke unique values ka Set bana sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<p>Create a set from an array:</p>
<p id="demo"></p>
<script>
// Create a Set
const letters = new Set(["a","b","c"]);
// Display set.size
document.getElementById("demo").innerHTML = "The set has " + letters.size + " values.";
</script>
</body>
</html>
```

The add() Method

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

add() method ka use JavaScript Set ke andar nayi value add karne ke liye hota hai.

Important Points

- Agar value pehle se Set mein exist karti hai, toh add() duplicate add nahi karega.
- add() method chainable hota hai — ek hi line mein multiple values add kar sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<h2>The add() Method</h2>
<p>The add() method adds values to a Set:</p>
<p id="demo"></p>
<script>
// Create a new Set
const letters = new Set(["a","b","c"]);
// Add a new Element
letters.add("d");
letters.add("e");
// Display set.size
document.getElementById("demo").innerHTML = "The set has " + letters.size + " values.";
</script>
</body>
</html>
```

The size Property

size ek property hai (method nahi), jo Set ke kitne elements hain wo batati hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<h2>The size Property</h2>
<p id="demo"></p>
<script>
// Create a new Set
const mySet = new Set(["a","b","c"]);
// Display size
document.getElementById("demo").innerHTML = "The set size is: " + mySet.size;
</script>
</body>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

</html>

Listing Set Elements

JavaScript Set mein values ko directly index se access nahi kar sakte (jaise set[0]). Isliye Set ke saare elements list karne ka simplest method hai — for..of loop.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<p>Ypu can iterate set values with for..of:</p>
<p id="demo"></p>
<script>
// Create a Set
const letters = new Set(["a","b","c"]);
// List all Elements
let text = "";
for (const x of letters) {
  text += x + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

has() Method

has() ek method hai jo check karta hai ki koi value ya key data structure ke andar present hai ya nahi.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<h2>The has() Method</h2>
<p>The has() method returns true if a set contains a spesific value:</p>
<p id="demo"></p>
<script>
// Create a new Set
const letters = new Set(["a","b","c"]);
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

```
// Does the Set contain "d"?
answer = letters.has("d");
document.getElementById("demo").innerHTML = "The answer is " + answer;
</script>
</body>
</html>
```

The forEach() Method

forEach() ek method hai jo Set ke har element par ek function call karta hai. Matlab, Set ke andar jitne elements hain, un sab ko ek-ek karke process karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<h2>The forEach() Method</h2>
<p>forEach() calls a function for each element in a Set:</p>
<p id="demo"></p>
<script>
// Create a Set
const letters = new Set(["a","b","c"]);
// List all Elements
let text = "";
letters.forEach (function(value) {
  text += value + "<br>";
})
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

The values() Method (Set)

values() method Set ke sabhi values ka ek Iterator object return karta hai. Is Iterator ko use karke aap Set ke elements ko one-by-one access kar sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

<h2>The values() Method</h2>

<p>The values() method returns an Iterator object with all the values in a Set.</p>

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

<script>

// Create a Set

const letters = new Set(["a","b","c"]);

// List all Elements

let text = "";

for (const x of letters.values()) {

text += x + "
";

}

document.getElementById("demo").innerHTML = text;

</script>

</body>

</html>

The keys() Method (Set)

JavaScript Set me keys aur values dono same hote hain.

Isliye keys() method bhi ek Iterator object return karta hai jo Set ke values ko return karta hai.

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Sets</h1>

<h2>The keys() Method</h2>

<p>The keys() method returns a Set Iterator with all the values from a Set.</p>

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

<script>

// Create a Set

const letters = new Set(["a","b","c"]);

// List all Elements

let text = "";

for (const x of letters.keys()) {

text += x + "
";

}

document.getElementById("demo").innerHTML = text;

</script>

</body>

</html>

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

The entries() Method

The entries() method returns an Iterator with [value,value] pairs from a Set.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<h2>The entries() Method</h2>
<p>The entries() method returns an Iterator with [value,value] pairs from a Set.</p>
<p id="demo"></p>
<script>
// Create a Set
const letters = new Set(["a","b","c"]);
// List all Elements
let text = "";
for (const x of letters.entries()) {
  text += x + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Logic Methods

Logic methods basically mathematical set operations hote hain. Ye check karte hain ki elements ka relationship kya hai do sets ke beech.

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

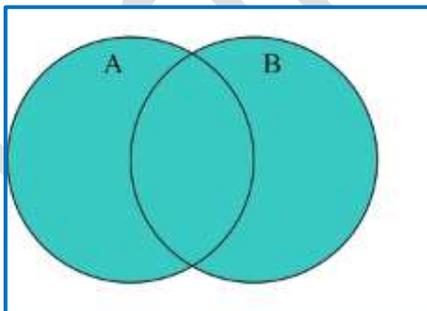
Logic Methods

In JavaScript 2025, 7 new logical methods were added to the Set object:

union()
difference()
intersection()
isDisjointFrom()
isSubsetOf()
isSupersetOf()
symmetricDifference()

The union() Method

- union() method do Sets ke elements ko combine karta hai aur ek naya Set return karta hai.
- Naya Set me sabhi unique elements honge jo:
- Pehle Set me hain ya
- Argument Set me hain ya
- Dono me hai



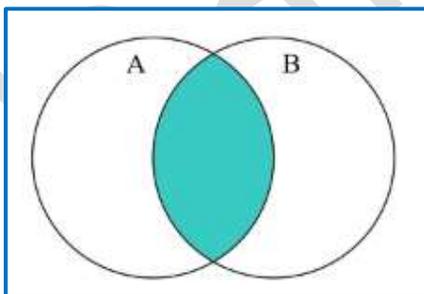
- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<h2>The Set.union() Method</h2>
<p>The <b>Set.union()</b> method returns a new set containing the elements which are in
this set,
or in the argument set, or in both:</p>
<p id="demo"></p>
<script>
const A = new Set(['a','b','c']);
const B = new Set(['b','c','d']);
const C = A.union(B);
let text = "";
for (const x of C) {
  text += x;
}
document.getElementById("demo").innerHTML = "The union is: " + text;
</script>
</body>
</html>
```

The intersection() Method

intersection() method do Sets ke common elements ko return karta hai aur naya Set banata hai.

Naya Set me sirf woh elements honge jo dono Sets me present hain.



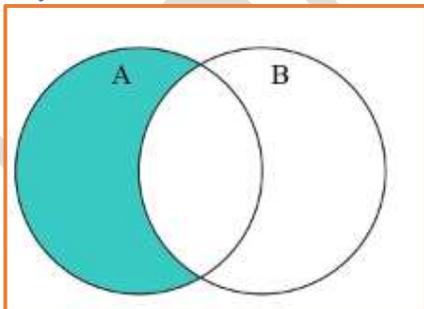
- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<h2>The intersection() Method</h2>
<p>The intersection() method returns a new set containing the elements which are in this set
and in the argument set:</p>
<p id="demo"></p>
<script>
const A = new Set(['a','b','c']);
const B = new Set(['b','c','d']);
const C = A.intersection(B);
let text = "";
for (const x of C) {
  text += x;
}
document.getElementById("demo").innerHTML = "The intersection is: " + text;
</script>
</body>
</html>
```

The difference() Method

difference() method do Sets ke elements ka difference return karta hai aur naya Set banata hai.

Naya Set me sirf woh elements honge jo pehle Set me hain aur argument Set me nahi hain.

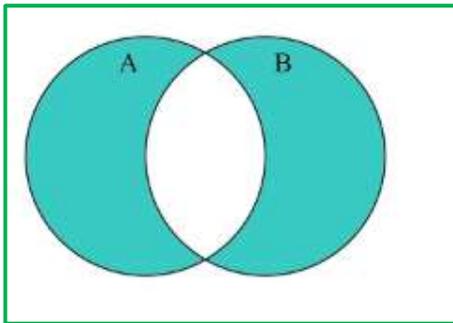


The symmetricDifference() Method

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- symmetricDifference() method do Sets ke elements ka exclusive difference return karta hai aur naya Set banata hai.
- Naya Set me sirf woh elements honge jo sirf ek Set me hain, dono me common elements exclude hote hain.



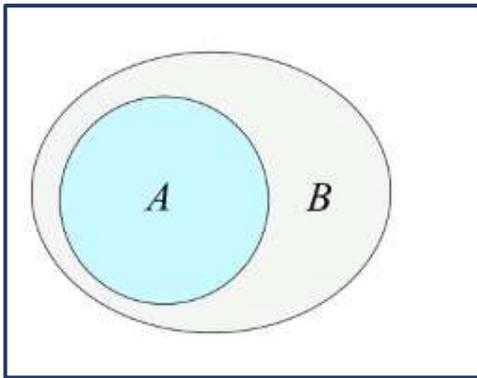
```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Sets</h1>
<h2>The symmetricDifference() Method</h2>
<p>The symmetricDifference() method returns a new set containing elements which are in
this set or in the argument set, but not in both:</p>
<p id="demo"></p>
<script>
const A = new Set(['a','b','c']);
const B = new Set(['b','c','d']);
const C = A.symmetricDifference(B);

let text = "";
for (const x of C) {
  text += x;
}
document.getElementById("demo").innerHTML = "Symmetric difference is: " + text;
</script>
</body>
</html>
```

The isSubsetOf() Method

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

- `isSubsetOf()` method check karta hai ki ek Set ka saara elements doosre Set me exist karte hain ya nahi.
- Agar saare elements argument Set me ho → true
Agar koi element missing ho → false



The `isSupersetOf()` Method

- `isSupersetOf()` method check karta hai ki ek Set me argument Set ke saare elements exist karte hain ya nahi.
- Agar argument Set ke saare elements is Set me ho → true
- Agar koi element missing ho → false

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript Sets</h1>
```

```
<h2>The isSupersetOf() Method</h2>
```

```
<p>The isSupersetOf() method returns true if all elements in the argument set are also in this set:</p>
```

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

```
<script>
```

```
const A = new Set(['a','b','c']);
```

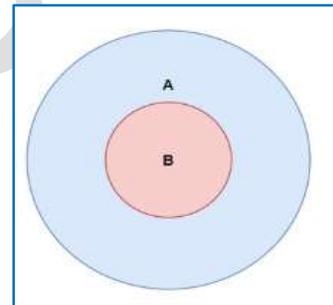
```
const B = new Set(['b','c','d']);
```

```
let answer = A.isSupersetOf(B);
```

```
document.getElementById("demo").innerHTML = answer;
```

```
</script>
```

```
</body>
```

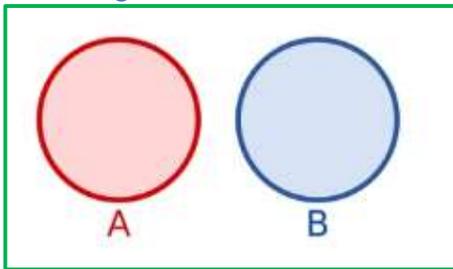


- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

</html>

The isDisjointFrom() Method

- isDisjointFrom() method check karta hai ki do Sets ke beech koi common element exist karta hai ya nahi.
- Agar koi element common na ho → true
Agar koi element common ho → false



```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>JavaScript Sets</h1>
```

```
<h2> the isDisjointFrom() Method</h2>
```

```
<p>The isDisjointFrom() method returns true if this set has no elements in common with the argument set:</p>
```

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

```
<script>
```

```
const A = new Set(['a','b','c']);
```

```
const B = new Set(['b','c','d']);
```

```
answer = A.isDisjointFrom(B);
```

```
document.getElementById("demo").innerHTML = answer;
```

```
</script>
```

```
</body>
```

```
</html>
```

JavaScript WeakSet

A WeakSet ek special type ka Set hai jo sirf objects ko store karta hai.

Key Features:

1. Values must be objects – primitive values (number, string, boolean) allowed nahi

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

2. Weak references – agar object kahin aur reference nahi hai, JavaScript automatically memory free kar deta hai (garbage collection)
3. No iteration – WeakSet me aap forEach, values(), etc. use nahi kar sakte

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript WeakSet</h1>
<p id="demo"></p>
<script>
// Create a WeakSet
let mySet = new WeakSet();
// Create an Object
let myObj = {fname:"John", lname:"Doe"};
// Add the Object
mySet.add(myObj);
// Delete the Object
mySet.delete(myObj);
// Do I have myObj in the mySet?
let answer = mySet.has(myObj);
document.getElementById("demo").innerHTML = "Do I have myObj? is " + answer
</script>
</body>
</html>
```

Garbage Collection (GC)

Garbage Collection ek automatic memory management mechanism hai jo JavaScript me memory ko efficiently handle karta hai.

JavaScript developer ko manually memory free karne ki zarurat nahi hoti — engine khud hi unused memory ko clean kar deta hai.

Weak References

- Weak References ka matlab hai ki ek object ko WeakSet ya WeakMap me store karne se uska memory garbage collection se bachta nahi hai.
- Agar object ka program me koi aur reference nahi hai, to GC (Garbage Collector) automatically us object ko free kar deta hai.

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE:** <https://learninghubshahabad.in>

WeakSet automatically us object ko remove kar deta hai jab object garbage collected ho jata hai.

Values Must Be Objects

- WeakSet me weak references store hote hain → Garbage Collection ke liye
- Garbage Collector sirf objects ko track karta hai, primitives ko nahi
- Isliye, WeakSet ka memory management sirf objects ke liye kaam karta hai

Tracking Objects

WeakSet sirf objects store karta hai aur unhe weakly hold karta hai. Agar object ka program me koi aur reference nahi hai, to Garbage Collector us object ko automatically delete kar deta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript WeakSet</h1>
<p id="demo"></p>
<script>
let text = "";
// Create a WeakSet to track persons
const persons = new WeakSet();
// Create Visitor Objects
const John = {name:"John", age:40};
const Paul = {name:"Paul", age:41};
const Ringo = {name:"Ringo", age:42};
const George = {name:"George", age:43}
// Track visits
track(Paul);
track(Ringo);
track(Paul);
// Function to track visitors
function track(visitor) {
  if (persons.has(visitor)) {
    text += visitor.name + " is visiting again.<br>";
  } else {
    persons.add(visitor);
    text += visitor.name + ", age" + visitor.age + ", is visiting for the first time<br>";
  }
}
```

LEARNING HUB
SHAHABAD MARKANDA
 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```

}
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>

```

Limited Methods

- add(obj) → object ko WeakSet me add karo
- has(obj) → check karo object WeakSet me hai ya nahi
- delete(obj) → object ko WeakSet se remove karo
- Numbers, strings ya primitives store nahi kar sakte.
- Items ko iterate nahi kar sakte (no forEach, no size).
- Memory safe: agar object kahin aur use nahi ho raha, toh automatically WeakSet se remove ho jata hai.

Method / Syntax	Description (Hinglish)
new WeakSet()	Ek naya WeakSet object create karta hai
add(object)	Object ko WeakSet me add karta hai
delete(object)	Object ko WeakSet se remove karta hai
has(object)	Check karta hai agar object WeakSet me exist karta hai; true/false return karta hai

Feature / Method	Description (Hinglish)
size property	WeakSet me total items count check karne ka option nahi hai
Spread operator ...	WeakSet ko spread karke array ya dusre structures me use nahi kar sakte
clear() method	WeakSet ko ek saath empty karne ka method nahi hai
Logic methods (union, difference, intersection)	Set jaise logic operations WeakSet me nahi hote
Iteration methods (forEach, keys, values, entries)	WeakSet ke items ko loop ya iterate nahi kar sakte

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Feature / Method	Description (Hinglish)
size property	WeakSet me total items count check karne ka option nahi hai
Spread operator ...	WeakSet ko spread karke array ya dusre structures me use nahi kar sakte
clear() method	WeakSet ko ek saath empty karne ka method nahi hai
Logic methods (union, difference, intersection)	Set jaise logic operations WeakSet me nahi hote
Iteration methods (forEach, keys, values, entries)	WeakSet ke items ko loop ya iterate nahi kar sakte

JavaScript Maps

Map object ek aisi cheez hai jo key-value pairs store karta hai.

Map me keys kisi bhi datatype ki ho sakti hain (number, string, object, etc.).

Map insertion order yaad rakhta hai, matlab jis order me keys add hui thi, wahi order me iterate kar sakte ho.

JavaScript Objects vs Maps

Feature	Object	Map
Iterability	Directly iterate nahi kar sakte	Directly iterable
size property	Nahi hai	Haan, size property available
Key types	Sirf Strings ya Symbols	Kisi bhi datatype ke keys
Key order	Well ordered nahi	Insertion order yaad rakhta hai
Default keys	Haan, prototype ke keys hote hain	Nahi, default keys nahi hote

JavaScript Map Methods

The new Map() Method

- new Map() method se ek naya Map create hota hai.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

- Agar aap array pass karte ho, jisme [key, value] pairs hote hain, toh wo saare pairs Map me add ho jate hain.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Maps</h1>
<h2>The new Map() Method</h2>
<p>Creating a map from an array:</p>
<p id="demo"></p>
<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);
let numb = fruits.get("apples");
document.getElementById("demo").innerHTML = "There are " + numb + " apples.";
</script>
</body>
</html>
```

Map.get → Kisi key ka value Map se retrieve karne ke liye use hota hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Maps</h1>
<h2>The get() Method</h2>
<p id="demo"></p>
<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);
document.getElementById("demo").innerHTML = fruits.get("apples");
</script>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</body>
</html>
```

Map.set(key, value) → Map me elements add karne ke liye use hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Maps</h1>
<h2>The set() Method</h2>
<p id="demo"></p>
<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);
fruits.set("apples", 200);

let numb = fruits.get("apples");
document.getElementById("demo").innerHTML = "There are " + numb + " apples.";
</script>
</body>
</html>
```

Map.size → Map me total elements ki counting return karta hai

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Maps</h1>
<h2>The size Property</h2>
<p id="demo"></p>
<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
]);  
document.getElementById("demo").innerHTML = fruits.size;  
</script>  
</body>  
</html>
```

Map.delete → Map se ek element remove karta hai

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Maps</h1>  
<h2>The delete() Method</h2>  
<p id="demo"></p>  
<script>  
// Create a Map  
const fruits = new Map([  
  ["apples", 500],  
  ["bananas", 300],  
  ["oranges", 200]  
]);  
// Delete an Element  
fruits.delete("apples");  
document.getElementById("demo").innerHTML = fruits.size;  
</script>  
</body>  
</html>
```

Map.clear() → Map ke saare elements remove kar deta hai

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>JavaScript Maps</h1>  
<h2>The clear() Method</h2>  
<p id="demo"></p>  
<script>  
// Create a Map  
const fruits = new Map([  
  ["apples", 500],
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
["bananas", 300],
["oranges", 200]
]);
// Clear the Map
fruits.clear();
document.getElementById("demo").innerHTML = fruits.size;
</script>
</body>
</html>
```

Map.has(key) → Check karta hai ki key Map me exist karti hai ya nahi

Returns true or false.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Maps</h1>
<h2>The has() Method</h2>
<p id="demo"></p>
```

```
<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);
// Delete an Element
fruits.delete("apples");
document.getElementById("demo").innerHTML = fruits.has("apples");
</script>
</body>
</html>
```

Map.forEach(callback) → Map ke har key/value pair ke liye callback

function run karta hai.

```
<!DOCTYPE html>
<html>
<body>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<h1>JavaScript Maps</h1>
<h2>The forEach() Method</h2>
<p id="demo"></p>
<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);
let text = "";
fruits.forEach (function(value, key) {
  text += key + ' = ' + value + "<br>"
})
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Map.entries() → Map ke [key, value] pairs ka iterator return karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Maps</h1>
<h2>The entries() Method</h2>
<p id="demo"></p>
<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);
let text = "";
for (const x of fruits.entries()) {
  text += x + "<br>";
}
</script>
</body>
</html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Map.keys() Jab aap map.keys() use karte ho, to ye aapko keys ka ek iterator object deta hai. Iterator ka matlab hota hai ki aap ek-ek karke values (yaha keys) ko loop ke through nikaal sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Maps</h1>
<h2>The keys() Method</h2>
<p id="demo"></p>
<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);
let text = "";
for (const x of fruits.keys()) {
  text += x + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Map.values()-Jab aap map.values() use karte ho, to ye aapko Map me stored saari values ek iterator form me deta hai. Iterator ka matlab hota hai ki aap loop laga kar ek-ek value ko access kar sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Maps</h1>
<h2>The values() Method</h2>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

```
<p id="demo"></p>
<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);
let total = 0;
for (const x of fruits.values()) {
  total += x;
}
document.getElementById("demo").innerHTML = total;
</script>
</body>
</html>
```

Objects as Keys

JavaScript me Map ka ek special feature hai ki aap objects ko keys ke roop me use kar sakte ho. Normal JavaScript objects {} me keys hamesha string ban jaati hain, lekin Map me aap objects, arrays, functions — sabko key bana sakte ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Maps</h1>
<p id="demo"></p>
<script>
// Create Objects
const apples = {name: 'Apples'};
const bananas = {name: 'Bananas'};
const oranges = {name: 'Oranges'};
// Create a Map
const fruits = new Map();
// Add Elements to the Map
fruits.set(apples, 500);
fruits.set(bananas, 300);
fruits.set(oranges, 200);
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>**

```
document.getElementById("demo").innerHTML = fruits.get("apples");
</script>
</body>
</html>
```

JavaScript Map.groupBy()

ES2024 ne JavaScript me ek naya feature add kiya: Map.groupBy()

Ye method kisi array ya iterable ke elements ko group karta hai, based on ek callback function ke return kiye gaye string value ke according.

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Maps</h1>
<h2>The Map.groupBy() Method</h2>
<p id="demo"></p>
<script>
// Create a Map
const fruits = [
  {name:"apples", quantity:300},
  {name:"bananas", quantity:500},
  {name:"oranges", quantity:200},
  {name:"kiwi", quantity:150}
];
// Callback function to select low volumes
function myCallback({ quantity }) {
  return quantity > 200 ? "ok" : "low";
}
// Group by ok and low
const result = Map.groupBy(fruits, myCallback)
// Display Results
let text ="These fruits are Ok: <br>";
for (let x of result.get("ok")) {
  text += x.name + " " + x.quantity + "<br>";
}
text += "<br>These fruits are low: <br>";
for (let x of result.get("low")) {
  text += x.name + " " + x.quantity + "<br>";
}
</script>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
document.getElementById("demo").innerHTML = text;  
</script>  
</body>  
</html>
```

Object.groupBy() vs Map.groupBy()

Object.groupBy()

- Output me plain JavaScript object deta hai {}.
- Groups ki keys string hoti hain.
- Key ordering not guaranteed (object ke rules follow hote hain).

Map.groupBy()

- Output me Map object deta hai.
- Map me keys string hi nahi, koi bhi value ho sakti hai.
- Keys ka insertion order preserved hota hai.
- Better for complex data structures.

JavaScript Map Reference

Method	Description
new Map()	Ek naya Map object banata hai.
clear()	Map ke saare elements ko delete kar deta hai (empty kar deta hai).
delete(key)	Diye gaye key ka element Map se hata deta hai.
entries()	Map ke saare [key, value] pairs ka iterator return karta hai.
forEach()	Map ke har key/value pair ke liye callback function run karta hai.
get(key)	Diye gaye key ki value return karta hai.
groupBy() (ES2024)	Elements ko group karta hai callback ke return kiye hue value ke hisaab se.
has(key)	Check karta hai ki key Map me exist karti hai ya nahi (true/false).
keys()	Map ke saare keys ka iterator return karta hai.
set(key, value)	Map me new key/value pair add karta hai, ya existing key ko update karta hai.
size	Map ke andar kitne elements hain, wo number return karta hai.
values()	Map ke saare values ka iterator return karta hai.

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

JavaScript Math Object

- Math object ek calculator toolkit jaisa hota hai.
- Aapko koi Math ka object create nahi karna padta.
- Aap simply Math.methodName() likh kar mathematical operations kar sakte ho.

Number to Integer

There are 4 common methods to round a number to an integer:

Method	Kya karta hai?	Example	Result
Math.round(x)	Nearest integer tak round karta hai	Math.round(4.6)	5
Math.ceil(x)	Hamesha upar wali integer tak round karta hai	Math.ceil(4.1)	5
Math.floor(x)	Hamesha neeche wali integer tak round karta hai	Math.floor(4.9)	4
Math.trunc(x)	Sirf integer part return karta hai (decimal hata deta hai)	Math.trunc(4.7)	4

Math.round()

Math.round(x) number x ko nearest integer tak round karta hai.

- Agar decimal part 0.5 ya zyada ho → number upar round hota hai
- Agar decimal part 0.4 ya kam ho → number neeche round hota hai

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.round()</h2>
<p>Math.round(x) returns the value of x rounded to its nearest integer:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.round(4.4);
</script>
</body>
</html>
```

Math.ceil()

Math.ceil(x) number x ko hamesha upar wali nearest integer tak round karta hai, chahe decimal kitna bhi ho.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.ceil()</h2>
<p>Math.ceil() rounds a number <strong>up</strong> to its nearest integer:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.ceil(4.4);
</script>
</body>
</html>
```

Math.floor()

Math.floor(x) number x ko hamesha neeche wali nearest integer tak round karta hai, chahe decimal kitna bhi ho.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.floor()</h2>
<p>Math.floor(x) returns the value of x rounded <strong>down</strong> to its nearest integer:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.floor(4.7);
</script>
</body>
</html>
```

Math.trunc()

Math.trunc(x) number x ka sirf integer part return karta hai.

- Decimal part remove ho jata hai.
- Na upar round, na neeche — bas integer part.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.trunc()</h2>
<p>Math.trunc(x) returns the integer part of x:</p>
<p id="demo"></p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
document.getElementById("demo").innerHTML = Math.trunc(4.7);
</script>
</body>
</html>
```

Math.sign()

Math.sign(x) check karta hai ki number positive, negative, ya zero hai, aur accordingly 1, -1, ya 0 return karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.sign()</h2>
<p>Math.sign(x) returns if x is negative, null or positive:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.sign(4);
</script>
</body>
</html>
```

Math.pow()

Math.pow(x, y) number x ko y ke power tak raise karta hai.

- Matlab: x^y
- Returns: Result as a number

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Math.pow()</h1>
<p>Math.pow(x,y) returns the value of x to the power of y:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.pow(8,2);
</script>
</body>
</html>
```

Math.sqrt()

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

Math.sqrt(x) number x ka square root return karta hai.

- Matlab: Jo number apne aap se multiply karke x de, wahi square root hai.
- Returns: Result as a number

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.sqrt()</h2>
<p>Math.sqrt(x) returns the square root of x:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.sqrt(64);
</script>
</body>
</html>
```

Math.abs()

Math.abs(x) number x ka absolute value return karta hai.

- Matlab: Result hamesha positive hota hai.
- Negative numbers ka sign hat jaata hai.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.abs()</h2>
<p>Math.abs(x) returns the absolute (positive) value of x:</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.abs(-4.7);
</script>
</body>
</html>
```

Math.sin()

- Math.sin(x) angle x ka sine return karta hai.
- Value -1 se 1 ke beech hoti hai.
- Angle radians me hona chahiye.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.sin()</h2>
<p>Math.sin(x) returns the sin of x (given in radians):</p>
<p>Angle in radians = (angle in degrees) * PI / 180.</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
"The sine value of 90 degrees is " + Math.sin(90 * Math.PI / 180);
</script>
</body>
</html>
```

Math.cos()

Math.cos(x) angle x ka cosine return karta hai.

- Value -1 se 1 ke beech hoti hai.
- Angle radians me hona chahiye.

Math.min() and Math.max()

- Math.min(a, b, c, ...) → Diye gaye numbers me sabse chhota return karta hai
- Math.max(a, b, c, ...) → Diye gaye numbers me sabse bada return karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.max()</h2>
<p>Math.max() returns the highest value in a list of arguments.</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
Math.max(0, 150, 30, 20, -8, -200);
</script>
</body>
</html>
```

Math.random()

- Math.random() ek random number return karta hai:

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- 0 (inclusive) se start hota hai
- 1 (exclusive) tak ja sakta hai
- Matlab: Result $0 \leq x < 1$

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.random()</h2>
<p>Math.random() returns a random number between 0 and 1:</p>
<p id="demo"></p>
<p>Tip: Click on "Run" several times.</p>
<script>
document.getElementById("demo").innerHTML = Math.random();
</script>
</body>
</html>
```

The Math.log() Method

- Math.log(x) number x ka natural logarithm (ln) return karta hai.
- Natural logarithm base e (≈ 2.718) par calculate hota hai.
- Matlab: ye batata hai ki e ko kitni power tak raise karna padega taaki x mile.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.log()</h2>
<p>How many times must we multiply Math.E to get 10?</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.log(10);
</script>
</body>
</html>
```

Math.log2()

Math.log2(x) number x ka base 2 logarithm return karta hai.

- Matlab: ye batata hai ki 2 ko kitni power tak raise karna padega taaki x mile.

```
<!DOCTYPE html>
<html>
<body>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<h2>JavaScript Math.log2()</h2>
<p>Math.log2() returns the base 2 logarithm of a number.</p>
<p>How many times must we multiply 2 to get 8?</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.log2(4);
</script>
</body>
</html>
```

Math.log10()

- Math.log10(x) number x ka base 10 logarithm return karta hai.
- Matlab: ye batata hai ki 10 ko kitni power tak raise karna padega taaki x mile.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Math.log10()</h2>
<p>Math.log10() returns the base 10 logarithm of a number.</p>
<p>How many times must we multiply 10 to get 1000?</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = Math.log10(1000);
</script>
</body>
</html>
```

Regular Expressions

Definition:

- Ek sequence of characters jo ek search pattern banata hai.
- Regex = shorthand for Regular Expression.
- JavaScript me RegExp object ke through use kiya jaata hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<p>Search a string for position of the match:</p>
<p id="demo"></p>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
let text = "Vk jain school!";
let n = text.search(/jain/i);
document.getElementById("demo").innerHTML = n;
</script>
</body>
</html>
```

Using String Methods

Method	Description (Hinglish)	Example
match(regex)	Regex ke saare matches return karta hai ek array me	"Hello 123".match(/\d+/g) → ["123"]
replace(regex, replacement)	Regex ke pattern ko replace karke new string return karta hai	"Hello 123".replace(/\d+/, "456") → "Hello 456"
search(regex)	Regex ke first match ka index return karta hai	"Hello 123".search(/\d+/) → 6

String.match()

Definition:

string.match(regex) string me regex pattern ke matches return karta hai.

- Agar global flag (g) use kiya ho → returns array of all matches
- Agar global flag nahi → returns array with first match and details

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The match() Method</h2>
<p>Search for "w3Schools" and display the first match:</p>
<p id="demo"></p>
<script>
let text = "Visit W3Schools!";
const myArr = text.match(/W3Schools/);
document.getElementById("demo").innerHTML = myArr;
</script>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

</body>
</html>

Using String replace()

replace() ek String method hai jo ek string ke specific part ko doosre text se replace karne ke liye use hoti hai.

Result Size: 486 x 495

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The replace() Method</h2>
<p>Replace "Microsoft" with "W3Schools" in the paragraph below:</p>
<p id="demo">Please visit Microsoft!</p>
<script>
let text = document.getElementById("demo").innerHTML;
text = text.replace(/Microsoft/, "W3Schools");

document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Using String search()

search() ek string method hai jo kisi string ke andar pattern ya substring ko dhundta hai.

Ye method pattern ka index return karta hai agar milta hai, aur -1 return karta hai agar nahi milta.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The search() Method</h2>
<p>Search for "W3Schools" and display the position of the match:</p>
<p id="demo"></p>
<script>
let text = "Visit W3Schools!";
let n = text.search(/W3Schools/);
document.getElementById("demo").innerHTML = n;
</script>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</body>  
</html>
```

RexExp Alteration (OR)

Regular expression me alteration ka matlab hota hai ek choice dena.

- Agar aap | use karte ho, to pattern kisi bhi side ka match ho sakta hai

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>Regular Expressions</h1>  
<h2>RexExp Alteration (OR)</h2>  
<p>A global search for "red" or "green" or "blue":</p>  
<p id="demo"></p>  
<script>  
let text = "Black, white, red, green, blue, yellow.";  
let result = text.match(/red|green|blue/g);  
document.getElementById("demo").innerHTML = result;  
</script>  
</body>  
</html>
```

JavaScript Regex Flags

Regular expression flags extra parameters hote hain jo pattern ka behavior modify karte hain.

- Matlab, ye decide karte hain ki pattern kaise search hoga.

Flag	Description	Example	Output
/g	Global match – poore string me sab matches dhundho	"banana".match(/a/g)	["a", "a", "a"]
/i	Case-insensitive – capital/small letters ignore karo	"Hello".match(/hello/i)	"Hello"
/u	Unicode support – emoji ya special characters handle karo	"😊".match(/^{\u{1F600}}/u)	"😊"

The /g Flag (Global)

- /g ka matlab hai global search.
- Normally, RegExp sirf pehla match return karta hai.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- Agar /g use karo, to poore string me sab matches return honge

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The /g Flag</h2>
<p>A global search for "is" in a string:</p>
<p id="demo"></p>
<script>
let text = "Is this all there is?";
const pattern = /is/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

The /i Flag (Insensitive)

- /i ka matlab hai ignore case.
- Normally, RegExp case-sensitive hota hai: "abc" sirf "abc" se match hota, "ABC" ya "AbC" se nahi.
- /i flag use karne se capital aur small letters ignore ho jate hain.

RexExp Metacharacters

- Metacharacters wo characters hote hain jo special meaning rakhte hain RegExp me.
- Ye normal letters ya digits se alag kaam karte hain: match patterns, positions, groups, etc.

Metacharacter	Description	Example	Matches
\d	Digits (0-9)	^d/	"0", "5", "9"
\D	Non-digits	^D/	"a", "#", " "
\w	Word characters (letters, digits, _)	^w/	"a", "Z", "9", "_ "
\W	Non-word characters	^W/	"!", "@", " "

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Metacharacter	Description	Example	Matches
\s	Spaces / whitespace (space, tab, newline)	^s/	" ", "\t", "\n"
\S	Non-space characters	^S/	"a", "1", "@"

RegExp \d (digits) Metacharacter

- \d ek metacharacter hai jo kisi bhi digit (0-9) ko match karta hai.
- Ye exactly 0 se 9 ke numbers dhundta hai string me.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The \d Metacharacter</h2>
<p>A global search for digits in a string:</p>
<p id="demo"></p>
<script>
let text = "Give 100%!";
const pattern = /\d/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp \w (word) Metacharacter

- The \w metacharacter matches word characters.
- A word character is a character a-z, A-Z, 0-9, including _ (underscore).

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The \w Metacharacter</h2>
<p>A global search for word characters:</p>
<p id="demo"></p>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
let text = "Give 100%!";
const pattern = /\w/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

JavaScript RegExp Quantifiers

Code (Quantifier)	Description	Example	Matches
x*	Zero ya more occurrences of x	/a*/	"", "a", "aa", "aaa"
x?	Zero ya one occurrence of x	/a?/	"", "a"
x{n}	Exactly n occurrences of x	/a{3}/	"aaa"

The n? Quantifier

- x? ka matlab hai zero ya one occurrence of x.
- Matlab: x ho sakta hai ya nahi bhi ho sakta.
- Agar x ek se zyada baar repeat ho → wo sirf pehla match consider hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The ? Quantifier</h2>
<p>A global search for a "1", followed by zero or one "0" characters:</p>
<p id="demo"></p>
<script>
let text = "1, 100 or 1000?";
const pattern = /10?/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

Regular Expression Assertions

LEARNING HUB
SHAHABAD MARKANDA
 **CALL- 77000 90800**

- ALL COMPUTER COURSES
- ENGLISH LANGUAGE COURSES
- Brand Marketing & Custom Software Solution
- TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- Assertions wo patterns hain jo match ka position check karte hain.
- Ye characters match nahi karte, sirf position ya context check karte hain

Syntax	Name	Description	Example
^	String boundary	String ke start se match karta hai	/^Hello/ matches "Hello world"
\$	String boundary	String ke end pe match karta hai	/world\$/ matches "Hello world"
\b	Word boundary	Word ke start ya end pe match karta hai	/\bword\b/ matches "a word here"
(?=...)	Lookahead	Check karta hai ki aage specified string ho	/foo(=\d+)/ matches "foo123"
(?<=...)	Lookbehind	Check karta hai ki pehle specified string ho	/(?<=\d+)foo/ matches "123foo"

RegExp ^ Metacharacter

- The ^ metacharacter matches the beginning of a string.
- ^ ek string boundary metacharacter hai.
- Ye string ke beginning ko match karta hai.
- Note: Ye characters match nahi karta, sirf position check karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The ^ Metacharacter</h2>
<p>Test if a text starts with W3Schools:</p>
<p id="demo"></p>
<script>
const pattern = /^W3Schools/;
let text = "Hello W3Schools";
let result = pattern.test(text);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

RegExp \$ Metacharacter

- \$ ek string boundary metacharacter hai.
- Ye string ke end ko match karta hai.
- Note: Ye characters match nahi karta, sirf position check karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The $ Metacharacter</h2>
<p>Test if a text ends with with W3Schools:</p>
<p id="demo"></p>
<script>
const pattern = /W3Schools$/;
let text = "W3Schools tutorial";
let result = pattern.test(text);

document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

JavaScript RegExp Character Classes

- Character Classes square brackets [] me likhe jaate hain.
- Ye string me se koi bhi ek character match karte hain jo brackets ke andar diya ho.

Class	Description (Hinglish)	Example	Matches
[a]	Brackets ke andar diya hua ek character match kare	/[a]/	"a"
[abc]	Brackets ke andar diya hua koi bhi character match kare	/[abc]/	"a", "b", "c"
[a-z]	Specified range ke characters match kare (a se z tak)	/[a-z]/	"a", "b", "c", ... "z"
[0-9]	Specified range ke digits match kare (0 se 9 tak)	/[0-9]/	"0", "1", "2", ... "9"

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<h2>Using [] to find a range of characters</h2>
<p>A global search for the characters 0 to 9 in a string:</p>
<p id="demo"></p>
<script>
let text = "More than 1000 times";
let pattern = /[0-9]/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

JavaScript RegExp Flags

Flag	Description
/d	Substring matches karta hai (new 2022)
/g	Global match karta hai – string me saare matches dhundta hai
/i	Case-insensitive match – upper/lower case ignore karta hai
/m	Multiline matching – ^ aur \$ ko line ke start/end pe bhi match karta hai
/s	Dotall flag – . (dot) line terminators ko bhi match kare (new 2018)
/u	Unicode support enable karta hai (new 2015)
/v	Upgrade to /u – better Unicode support (new 2025)
/y	Sticky search – search current position se start hota hai, string ke aage hi match karta hai (new 2015)

Flag Syntax: /pattern/flags

Part	Description	Example
/	Opening delimiter – regular expression start mark karta hai	/abc/
pattern	Regular expression – search criteria ya pattern	abc
/	Closing delimiter – regular expression end mark karta hai	/abc/
flags	Optional modifiers – pattern ka behaviour change karte hain	/abc/g, /abc/i

RegExp /g Flag (Global)

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

/g Flag (Global)

- Normally, jab aap RegExp use karte ho without /g, ye sirf first match find karta hai.
- Agar aap /g flag use karte ho, ye string me saare matches ko find karega, na ki sirf pehla.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The /g Flag</h2>
<p>A global search for "is" in a string:</p>
<p id="demo"></p>
<script>
let text = "Is this all there is?";
const pattern = /is/g;

let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp /i Flag (Insensitive)

/i Flag (Case-Insensitive)

- Normally, RegExp case-sensitive hota hai. Matlab "abc" sirf "abc" se match karega, "ABC" ya "AbC" se nahi.
- Agar aap /i flag use karte ho, to RegExp case ko ignore kar deta hai

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The /i Flag</h2>
<p>A case-insensitive search for "w3schools" in a string:</p>
<p id="demo"></p>
<script>
let text = "Visit W3Schools";
const pattern = /w3schools/i;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
```

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
</script>  
</body>  
</html>
```

RegExp /d Flag

JavaScript RegExp me /d flag exist nahi karta. Ho sakta aap confuse ho rahe ho \d ke saath, jo digit character ke liye hota hai, ya phir ^ aur \$ ke saath, jo start aur end of string match karte hain.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>Regular Expressions</h1>  
<h2>The /d Flag</h2>  
<p>Match all that starts or ends with aa or bb:</p>  
<p id="demo"></p>  
<script>  
let text = "aaaabb";  
const pattern = /(aa)(bb)/d;  
let result = text.match(pattern);  
document.getElementById("demo").innerHTML = result;  
</script>  
</body>  
</html>
```

RegExp /s Flag (Single line/DotAll)

/s Flag (Single-line / DotAll)

- Normally, . (dot) RegExp me koi bhi character except newline \n match karta hai.
- Agar aap /s flag lagate ho, to . newline \n ko bhi match karega.

```
<!DOCTYPE html>  
<html>  
<body>  
<h1>Regular Expressions</h1>  
<h2>The /s Flag</h2>  
<p>The /s (dotAll) flag allows the . metacharacter to match any character, including line terminators.</p>  
<p id="demo"></p>  
<script>  
let text = "Line\nLine.";
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
const pattern = /Line./gs;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp /y Flag (Sticky):

- /y ka matlab hai sticky search.
- Sticky search ka matlab: RegExp sirf usi position se match karega jahan last match khatam hua tha.
- Ye normal /g flag se thoda alag hai: /g search aage se next match dhundta hai, chahe beech me koi characters miss ho gaye ho. /y sirf exact position pe match karega.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The /y Flag</h2>
<p>The /y flag must be set to allow matches from a position.</p>
<p id="demo"></p>
<script>
let text = "abc def ghi";
const pattern = ^w+;
// Start match from position 4
pattern.lastIndex = 4;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp Character Classes

Ye aapko allow karte hain ek ya ek se zyada specific characters match karne ke liye.

Class	Meaning (Hinglish)	Example	Match Example
[a]	Sirf "a" ko match karega	/[a]/.test("apple")	true ("a")
[^a]	"a" ke alawa koi bhi character match karega	/[^a]/.test("apple")	true ("p")

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Class	Meaning (Hinglish)	Example	Match Example
[abc]	a, b, ya c me se koi ek character match karega	/[abc]/.test("bat")	true ("b")
[^abc]	a, b, c ke alawa koi character match karega	/[^abc]/.test("dog")	true ("d")
[a-z]	Small letters a se z me se koi match karega	/[a-z]/.test("moon")	true ("m")
[^a-z]	Small letters ke alawa koi character match karega	/[^a-z]/.test("Moon123")	true ("M")
[0-9]	Digits 0 se 9 me se koi match karega	/[0-9]/.test("year2025")	true ("2")
[^0-9]	Digits ke alawa koi character match karega	/[^0-9]/.test("year2025")	true ("y")

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>Using [] to find a range of characters</h2>
<p>A global search for the characters A to Z:</p>
<p id="demo"></p>
<script>
let text = "This is W3Schools";
const pattern = /[A-Z]/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

Regular Expression Metacharacters

- Metacharacters woh characters hote hain jo special meaning rakhte hain RegExp me.
- Ye normal characters jaise a ya 1 se alag behave karte hain.
- Inka use hum patterns define karne ke liye karte hain: digits, letters, spaces, position of string, etc.

Meta	Meaning (Hinglish)	Example	Matches Example
\d	Digit match karta hai (0-9)	/\d/.test("7")	true

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Meta	Meaning (Hinglish)	Example	Matches Example
\D	Non-digit match karta hai	^D/.test("a")	true
\w	Word characters (letters, digits, underscore)	^w/.test("A")	true
\W	Non-word characters (special symbols, space)	^W/.test("!")	true
\s	Spaces match karta hai (space, tab, newline)	^s/.test(" ")	true
\S	Non-space characters	^S/.test("a")	true
\ddd	Character by Octal number ddd	^101/.test("A")	true (A)
\xhh	Character by Hexadecimal number hh	^x41/.test("A")	true (A)
\uhhhh	Unicode character by hex number hhhh	^u0041/.test("A")	true (A)

\d Metacharacter

- \d ka matlab hai digit match karna.
- Digit = 0 se 9 tak ka koi number.
- Ye single digit ko match karta hai, lekin agar aap + lagaye, to ek ya zyada digits match ho sakte hain.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The \D Metacharacter</h2>
<p>A global search for non-digit characters:</p>
<p id="demo"></p>
<script>
let text = "Give 100%!";
const pattern = ^D/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML=result;
</script>
</body>
</html>
```

RegExp \w (word) Metacharacter

- \w ka matlab hai word character match karna.
- Word characters me include hain:

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

- Lowercase letters: a-z
- Uppercase letters: A-Z
- Digits: 0-9
- Underscore: _
- Basically, ye alphanumeric + underscore match karta hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The \w Metacharacter</h2>
<p>A global search for word characters:</p>
<p id="demo"></p>
<script>
let text = "Give 100%!";
const pattern = /\w/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

The \s (space) Metacharacter

- \s ka matlab hai whitespace character match karna.
- Whitespace characters me include hote hain:
- Space " "
- Tab \t
- Newline \n
- Carriage return \r

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The \s Metacharacter</h2>
<p>A global search for whitespace characters in a string:</p>
<p id="demo"></p>
<script>
let text = "Is this all there is?";
```

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
const pattern = /\s/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp \xhh (hexadecimal)

\xhh Metacharacter

- \xhh ka matlab hai character ko hexadecimal number ke through match karna.
- hh = 2-digit hexadecimal number (00 se FF).
- Ye ASCII characters ya koi specific symbol match karne ke liye use hota hai.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The \xhh Metacharacters</h2>
<p>A global replace of the hexadecimal character 6F (o):</p>

<p id="demo"></p>
<script>
let text = "Visit W3Schools. Hello World!";
let pattern = /\x6F/g;
let result = text.replace(pattern, "*");
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp \uhhhh (unicode hex)

- \uhhhh Metacharacter
- \uhhhh ka matlab hai Unicode character ko hex code ke through match karna.
- hhhh = 4-digit hexadecimal number jo Unicode character ko represent karta hai.
- Ye ASCII ke alawa bhi characters match kar sakta hai, jaise emojis, accented letters, symbols, etc

```
<!DOCTYPE html>
<html>
<body>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<h1>Regular Expressions</h1>
<h2>The \uhhhh Metacharacter</h2>
<p>A global search for a Unicode character with the hexadecimal code 0057 (W):</p>
<p id="demo"></p>
<script>
let text = "Visit W3Schools. Hello World!";
const pattern = /\u0057/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

Regular Expression Methods

JavaScript me search, match, replace, aur test ke liye alag-alag methods use hote hain.

String Methods

Method	Description	Example	Result Example
match(regex)	Pattern ke saare matches ka array return karta hai	"apple 123".match(/\d/g)	["1", "2", "3"]
matchAll(regex)	Pattern ke saare matches ka iterator return karta hai	[... "apple 123".matchAll(/\d/g)]	[["1"], ["2"], ["3"]]
replace(regex)	Pattern ko replace karke nayi string return karta hai	"cat bat".replace(/cat/, "dog")	"dog bat"
replaceAll(regex)	Pattern ke saare occurrences replace karke string return karta hai	"cat bat".replaceAll(/cat/g, "dog")	"dog bat dog"
search(regex)	Pattern ka index return karta hai (pehla match)	"Hello world".search(/world/)	6
split(regex)	Pattern ke basis par string split karke array return karta hai	"a,b,c".split(/,/)	["a", "b", "c"]

RegExp Methods

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Method	Description	Example	Result Example
regex.exec()	Pattern ko string me search karta hai aur detailed match info ka iterator return karta hai	^d/g.exec("123")	["1", index: 0, input: "123"]
regex.test()	Check karta hai ki pattern string me exist karta hai ya nahi	^d/.test("123")	true

Regular Expression Assertions

- Assertions ek tarah ke conditions hote hain jo match hone se pehle ya baad check karte hain.
- Ye characters ko match nahi karte, bas context check karte hain.
- Mainly do types hain: Lookahead aur Lookbehind.

JavaScript Regex Assertions

Syntax	Name	Meaning (Hinglish)
^	String boundary	String ke start me match karta hai
\$	String boundary	String ke end me match karta hai
\b	Word boundary	Word ke start ya end me match karta hai
\B	Word boundary	Word ke start ya end ke alawa match karta hai
(?=...)	Lookahead	Match kare agar specified pattern ke baad string ho
(?!...)	Lookahead	Match kare agar specified pattern ke baad nahi string ho
(?<=...)	Lookbehind	Match kare agar specified pattern ke pehle string ho
(?<!...)	Lookbehind	Match kare agar specified pattern ke pehle nahi string ho

^RegExp metacharacter ka simple matlab hota hai: string ki shuruat se match karna.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<h2>The ^ Metacharacter</h2>
<p>Test if a text starts with W3Schools:</p>
<p id="demo"></p>
<script>
const pattern = /^W3Schools/;
let text = "Hello W3Schools";
let result = pattern.test(text);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp \$ Metacharacter -Jab aap regular expression me \$ use karte ho, to aap computer ko ye bolte ho:"Pattern ko string ke bilkul end me check karo."

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The $ Metacharacter</h2>
<p>Test if a text ends with with W3Schools:</p>
<p id="demo"></p>
<script>
const pattern = /W3Schools$/;
let text = "W3Schools tutorial";
let result = pattern.test(text);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

The \b Metacharacter

Jab aap regex me \b use karte ho, to aap computer ko bolte ho.Match tabhi karo jab pattern word ke start ya end par ho. Isse pata chalta hai ki ek word kahaan se start hota hai aur kah

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

<h2>The \b Metacharacter</h2>

<p>Search for the characters "LO" in the end of a word in the phrase:</p>

<p>"HELLO, LOOK AT YOU!"</p>

<p>Found in position: </p>

<script>

let text = "HELLO, LOOK AT YOU!";

let result = text.search(/LO\b/);

document.getElementById("demo").innerHTML = result;

</script>

</body>

</html>

RegExp Lookahead x(?=y)

"x ko tabhi match karo jab uske turant baad y ho."

<!DOCTYPE html>

<html>

<body>

<h1>Regular Expressions</h1>

<h2>Lookahead Assertion</h2>

<p>Match "W3schools" if "W3Schools" is followed by "Tutorials".</p>

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

<script>

let text = "W3Schools Tutorials";

let pattern = new RegExp("W3Schools(?=)Tutorials");

let result = pattern.test(text);

document.getElementById("demo").innerHTML = result;

</script>

</body>

</html>

Negative Lookahead x(?!y)

x(?!y) matches "x" if "x" is NOT followed by "y".

<!DOCTYPE html>

<html>

<body>

<h1>Regular Expressions</h1>

<h2>Negative Lookahead Assertion</h2>

<p>Match "W3schools" only if "W3Schools" is NOT followed by "Tutorials".</p>

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
<p id="demo"></p>
<script>
let text = "W3Schools Tutorials";
let pattern = /W3Schools(?: Tutorials)/;
let result = pattern.test(text);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp Lookbehind (?<=y)x

x ko tabhi match karo jab x se pehle y ho.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The Lookahead Assertion</h2>
<p>Match "W3Schools" if "W3Schools" is preceded by "Hello ".</p>
<p id="demo"></p>
<script>
let text = "Hello W3Schools";
let pattern = /(?!Hello )W3Schools/;
let result = pattern.test(text);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

Negative Lookbehind (?<!y)x

Negative lookbehind matlab pehle check karo ki x ke immediately pehle y nahi hai.
Agar y nahi mila, tab x ko match karo.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The Negative Lookahead Assertion</h2>
<p>Match "W3Schools" only if "W3Schools" is NOT preceded by "Hello ".</p>
<p id="demo"></p>
```

LEARNING HUB
SHAHABAD MARKANDA
 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<script>
let text = "Hello W3Schools";
let pattern = /(?!Hello )W3Schools/;
let result = pattern.test(text);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

Regular Expression Groups

Char / Syntax	Description	Hinglish	Example
(x)	Matches x and saves it (capturing group)	Pattern ko brackets me daal kar save kar deta hai, baad me use kar sakte ho (\1, \2 etc.)	(cat) and \1 → matches "cat and cat"
(?x)	Matches x and labels it n (named capturing group)	Group ko naam de deta hai, taaki baad me naam se refer kar sako	(?<word>\w+)\s+k<word> → matches "hello hello"
(?flag:x)	Enables flag(s) only inside x	Is group ke andar specific flags ON ho jayenge (jaise i, m, s)	(?i:cat) → matches "Cat", "CAT"
(?flag-flag:x)	Enables some flags and disables others inside x	Ek hi group me kuch flags ON, kuch OFF kar sakte ho	(?i-m:hello) → case-insensitive match, multiline OFF

RegExp Capturing Groups (x)

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>Capturing Groups (x)</h2>
<p>Search for "haha" in a string:</p>
<p id="demo"></p>
<script>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- **ALL COMPUTER COURSES**
 - **ENGLISH LANGUAGE COURSES**
 - **Brand Marketing & Custom Software Solution**
 - **TUITION NURSERY TO 10TH (ALL SUBJECTS)**
- WEBSITE: <https://learninghubshahabad.in>

```
let text = "Haha, haha, haha.";
const pattern = /(haha)+/;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

Named Capturing Groups (?<n>)

- Ye capturing group hai, lekin number ke bajaye naam se refer kiya ja sakta hai.
- Match hone par group ka value name ke through access hota hai.<!DOCTYPE html>

```
<html>
<body>
<h1>Regular Expressions</h1>
<h2>Named Capturing Groups</h2>
<p id="demo"></p>
<script>
const text = "Name: John Doe";
// Using named capturing groups
const regex = /(?<firstName>\w+) (?<lastName>\w+)/;
const match = text.match(regex);
let fName = match.groups.firstName;
let lName = match.groups.lastName;
document.getElementById("demo").innerHTML = "Last Name: " + match.groups.lastName;
</script>
</body>
</html>
```

RegExp Group Modifiers (?flag)

- Group modifier ka matlab: aap flags ko ek specific group me apply kar sakte ho, instead of poore regex.
- Ye inline likha jata hai regex ke andar, brackets me.

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>Groups Modifier (?flag)</h2>
<p>Search for "w3schools tutorials" (with only W3Schools case-intensitive):</p>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<p id="demo"></p>
<script>
let text = "W3Schools Tutorials.";
const pattern = /(?:W3Schools) tutorials/;
let result = pattern.test(text);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp Quantifiers

Quantifiers batate hain kitni baar koi character, group, ya character class appear ho sakta hai. Yani, ye repeat ka rule set karte hain regex me.

JavaScript RegExp Quantifiers

Code	Description (Hinglish)
x+	x kam se kam 1 baar match hoga
x*	x 0 ya more baar match hoga
x?	x 0 ya 1 baar match hoga (optional)
x{n}	x exactly n baar match hoga
x{n,m}	x n se m baar match hoga
x{n,}	x kam se kam n baar match hoga

RegExp + Quantifier

- + ka matlab hai: at least one occurrence.
- Ye check karta hai ki koi character, group, ya character class kam se kam 1 baar appear ho.
- Agar wo 0 baar appear ho, to match fail ho jayega.
- Agar 1 ya usse zyada baar appear ho, to match success hoga.

```
<!DOCTYPE html>
<html>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<body>
<h1>Regular Expressions</h1>
<h2>The + Quantifier</h2>
<p>A global search for at least one "o":</p>
<p id="demo"></p>
<script>
let text = "Hellooo World! Hello W3Schools!";
const pattern = /o+/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp * Quantifier

x^* matches zero or more occurrences of x .

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The * Quantifier</h2>
<p>A global search for an "l", followed by zero or more "o" characters:</p>
<p id="demo"></p>
<script>
let text = "Hellooo World! Hello W3Schools!";
const pattern = /lo*/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

RegExp ? Quantifier

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<h2>The ? Quantifier</h2>
```

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

```
<p>A global search for a "1", followed by zero or one "0" characters:</p>
<p id="demo"></p>
<script>
let text = "1, 100 or 1000?";
const pattern = /10?/g;
let result = text.match(pattern);
document.getElementById("demo").innerHTML = result;
</script>
</body>
</html>
```

Full RegExp Flag Reference

Flag	Description (Hinglish)
/d	Substring match perform karta hai (naya 2022)
/g	Global match karta hai (saare matches find karta hai)
/i	Case-insensitive match karta hai (capital ya small letter ignore)
/m	Multiline matching enable karta hai (^ aur \$ har line ke liye)
/s	Dot (.) ko line terminators ke saath match karne deta hai (naya 2018)
/u	Unicode support enable karta hai (naya 2015)
/v	/u ka upgrade for better Unicode support (naya 2025)
/y	Sticky search perform karta hai (match string ke current position se hi start) (naya 2015)

Full Character Classes Reference

Class	Description
[a]	Brackets ke andar ka character match karega (yahan sirf a)
[^a]	Brackets ke andar ka character match nahi karega (yahan a ke alawa sab)
[abc]	Brackets ke andar ke characters me se koi bhi match karega (a, b, c)
[^abc]	Brackets ke andar ke characters me se koi bhi match nahi karega (a, b, c ke alawa)
[a-z]	Lowercase letters a se z me se koi bhi match karega
[^a-z]	Lowercase letters a se z me se koi bhi match nahi karega
[0-9]	Digits 0 se 9 me se koi bhi match karega
[^0-9]	Digits 0 se 9 me se koi bhi match nahi karega

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Full Metacharacter Reference

Metacharacter	Description
`a`	b`
.	Kisi bhi character ko match karega, except line terminators
\w	Word characters match karega (a-z, A-Z, 0-9, _)
\W	Non-word characters match karega (word ke alawa)
\d	Digits 0-9 match karega
\D	Non-digit characters match karega
\s	Whitespace characters match karega (space, tab, newline)
\S	Non-whitespace characters match karega
[\b]	Backspace character match karega
\0	NULL character match karega
\n	New line character match karega
\f	Form feed character match karega
\r	Carriage return match karega
\t	Tab character match karega
\v	Vertical tab character match karega
\p{ }	Unicode character with given property match karega (new 2018)
\P{ }	Unicode character with given property ke alawa match karega (new 2018)
\ddd	Octal number ddd ke character ko match karega
\xhh	Hexadecimal number hh ke character ko match karega
\uhhhh	Unicode character with hex number hhhh match karega

JavaScript RegExp Objects

- JavaScript me RegExp ek regular expression object hai.
- Ye predefined properties aur methods ke saath aata hai, jisse aap pattern matching, searching aur replacing kar sakte ho strings me.

Using test()

- test() RegExp object ka method hai.
- Ye string me pattern search karta hai.

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

- Agar match milta hai → returns true
- Agar match nahi milta → returns false

Using exec()

- exec() RegExp object ka method hai.
- Ye string me pattern search karta hai aur match hone par detailed result object return karta hai.
- Agar match nahi milta → returns null

```
<!DOCTYPE html>
<html>
<body>
<h1>Regular Expressions</h1>
<p id="demo"></p>
<script>
const obj = /e/.exec("The best things in life are free!");
document.getElementById("demo").innerHTML =
"Found " + obj[0] + " in position " + obj.index + " in the text: " + obj.input;
</script>
</body>
</html>
```

RegExp.escape()

- RegExp.escape() method ek string return karta hai jisme special regex characters escape ho jate hain.
- Iska matlab ye characters literal treat hote hain, regex ke special meaning ke bajaye.
- Special characters: +, *, ?, ^, \$, (,), [,], {, }, |, \

Complete RegExp Reference

Name	Description
compile()	Regex ko compile karta hai (Deprecated)
constructor	Wo function return karta hai jo RegExp prototype create kiya
dotAll	s flag set hai to true return karta hai (new 2018)
escape()	String return karta hai jisme regex special characters escape ho gaye hain (new 2025)
exec()	Regex match ke liye result array return karta hai

LEARNING HUB
SHAHABAD MARKANDA
📞 **CALL- 77000 90800**

- ALL COMPUTER COURSES
 - ENGLISH LANGUAGE COURSES
 - Brand Marketing & Custom Software Solution
 - TUITION NURSERY TO 10TH (ALL SUBJECTS)
- WEBSITE: <https://learninghubshahabad.in>

Name	Description
flags	Regex me set modifiers/flags return karta hai (new 2015)
global	Agar g flag set hai to true return karta hai
hasIndices	Agar d flag set hai to true return karta hai (new 2022)
ignoreCase	Agar i flag set hai to true return karta hai
lastIndex	Next match ke start index ko specify karta hai
multiline	Agar m flag set hai to true return karta hai
source	Regex ka pattern text return karta hai
sticky	Agar y flag set hai to true return karta hai (new 2015)
test()	String me match check karta hai → returns true ya false
toString()	Regex ka string value return karta hai
unicode	Agar u flag set hai to true return karta hai (new 2018)
unicodeSets	Agar v flag set hai to true return karta hai (new 2023)

RegExp String Methods

Method	Description
match(regex)	String me regex ke matches ka Array return karta hai
matchAll(regex)	String me regex ke matches ka Iterator return karta hai (sab results ke liye)
replace(regex, s)	Regex ke matches ko replace karke nayi string return karta hai
replaceAll(regex, s)	Regex ke saare matches replace karke nayi string return karta hai
search(regex)	String me pehle match ka index return karta hai
split(regex)	Regex ke basis pe string split karke Array return karta hai