## 1️⃣ JavaScript Strings

### What is a String?

A **string** is text written inside **quotes**.

### Create Strings

```
let name = "Jaspreet";

let city = 'Delhi';
```

✓ JavaScript allows **single (' ')** and **double (" ")** quotes.

---

### String Length

```
let text = "Hello World";

console.log(text.length); // 11
```

---

## 2️⃣ JavaScript String Templates (Template Literals)

Template literals use **backticks ( )**
They allow:

- Variables inside strings
- Multi-line strings

### Example: Variables inside string

```
let name = "Aman";

let age = 20;


let msg = `My name is ${name} and I am ${age} years old`;

console.log(msg);
```

### Output

My name is Aman and I am 20 years old

---

**Example: Multi-line string**

let text = `Hello

Welcome

To JavaScript`;

console.log(text);

✓ Much easier than normal strings.

---

## 3️⃣ JavaScript String Methods

String methods help **modify or read strings**.

---

### 🔹 toUpperCase()

let text = "hello";

console.log(text.toUpperCase());

Output:

HELLO

---

### 🔹 toLowerCase()

let text = "HELLO";

console.log(text.toLowerCase());

---

### 🔹 trim()

Removes extra spaces

let text = "  hello  ";

console.log(text.trim());

---

🔹 **slice(start, end)**

Extract part of string

let text = "JavaScript";

console.log(text.slice(0, 4)); // Java

---

🔹 **replace()**

let text = "I like PHP";

console.log(text.replace("PHP", "JavaScript"));

---

🔹 **split()**

Converts string to array

let text = "apple,banana,mango";

let arr = text.split(",");

console.log(arr);

---

🔹 **charAt()**

let text = "Hello";

console.log(text.charAt(1)); // e

---

🔠 **JavaScript String Search**

Used to **find text inside a string**.

---

🔹 **indexOf()**

let text = "Hello World";

console.log(text.indexOf("World")); // 6

Returns **-1 if not found**

---

🔹 **lastIndexOf()**

let text = "Hello Hello";

console.log(text.lastIndexOf("Hello")); // 6

---

🔹 **includes()**

let text = "JavaScript is easy";

console.log(text.includes("easy")); // true

---

🔹 **startsWith()**

let text = "Hello World";

console.log(text.startsWith("Hello")); // true

---

🔹 **endsWith()**

console.log(text.endsWith("World")); // true

---

## 5️⃣ JavaScript String Reference (Important Points)

🔸 **Strings are Immutable**

You **cannot change** characters directly.

let text = "Hello";

text[0] = "Y";  //  won't change

Correct way:

text = "Yello";

---

### 🔸 Strings as Objects

```
let x = "John";          // string
let y = new String("John"); // object
```

⚠ Avoid new String() — causes confusion.

---

### 🔸 Escape Characters

```
let text = "He said \"Hello\"";
```

| Escape | Meaning |
|---|---|
| ' | single quote |
| " | double quote |
| \ | backslash |

---

**Real-Life Use Example**

**Check email format**

```
let email = "test@gmail.com";


if (email.includes("@")) {
  console.log("Valid email");
} else {
  console.log("Invalid email");
}
```

---

# JavaScript Number

## 1️⃣ JavaScript Numbers

### What is a Number?

In JavaScript, **numbers include integers and decimals**.

```
let a = 10;     // integer
let b = 10.5;   // decimal
```

✓ JavaScript has **only one number type** (no int, float, double separately).

---

### Numbers with Exponent

```
let x = 5e3;   // 5 × 1000
let y = 5e-3;  // 0.005
```

---

### Numbers as Strings

```
let x = "10";
let y = 5;
console.log(x + y); // "105"
```

---

## 2️⃣ JavaScript Number Methods

Used to **convert and format numbers**.

---

### 🔹 toString()

```
let x = 10;
console.log(x.toString());
```

---

- **toFixed(n)**

Rounds number to n decimals

```
let x = 3.456;

console.log(x.toFixed(2)); // 3.46
```

---

- **toPrecision(n)**

```
let x = 123.456;

console.log(x.toPrecision(4)); // 123.5
```

---

- **Number()**

Convert value to number

```
Number("10");   // 10

Number(true);   // 1

Number(false);  // 0
```

---

- **parseInt()**

```
parseInt("10.5"); // 10
```

---

- **parseFloat()**

```
parseFloat("10.5"); // 10.5
```

---

- **isNaN()**

Checks Not-a-Number

```
isNaN("Hello"); // true
```

---

🔹 **isFinite()**

```
isFinite(10);      // true
isFinite(Infinity); // false
```

---

## 3️⃣ JavaScript Number Properties

Properties belong to the **Number object**.

---

🔹 **Number.MAX_VALUE**

```
Number.MAX_VALUE;
```

---

🔹 **Number.MIN_VALUE**

```
Number.MIN_VALUE;
```

---

🔹 **Number.POSITIVE_INFINITY**

```
Number.POSITIVE_INFINITY;
```

---

🔹 **Number.NEGATIVE_INFINITY**

```
Number.NEGATIVE_INFINITY;
```

---

🔹 **Number.NaN**

```
Number.NaN;
```

---

🔹 **Number.EPSILON**

Smallest difference between numbers

Number.EPSILON;

---

## 4️⃣ JavaScript Number Reference (Important Rules)

---

### 🔸 Precision Problem

let x = 0.1 + 0.2;

console.log(x); // 0.30000000000000004

✓ Solution:

let x = (0.1 * 10 + 0.2 * 10) / 10;

---

### 🔸 NaN

let x = 100 / "Hello";

console.log(x); // NaN

---

### 🔸 Infinity

let x = 2 / 0;

console.log(x); // Infinity

---

### 🔸 Numbers as Objects (avoid)

let x = new Number(10);

⚠ Use only:

let x = 10;

---

## 5️⃣ JavaScript Bitwise Operators

Bitwise works on **binary (0 & 1)**.

| Operator | Name |
|---|---|
| & | AND |
| \| | OR |
| ^ | XOR |
| ~ | NOT |
| << | Left Shift |
| >> | Right Shift |

---

**Examples**

**AND (&)**

5 & 1; // 1

**OR (|)**

5 | 1; // 5

**XOR (^)**

5 ^ 1; // 4

**Left Shift (<<)**

5 << 1; // 10

**Right Shift (>>)**

5 >> 1; // 2

---

**Real Use**

Used in:

- Permissions

- Low-level calculations
- Performance-critical apps

---

### 6️⃣ JavaScript BigInt

Used for **very large numbers**.

---

**Create BigInt**

Add n at end:

let big = 1234567890123456789012345678 90n;

---

**BigInt Operations**

let a = 10n;

let b = 20n;

console.log(a + b); // 30n

---

**BigInt + Number not allowed**

10n + 5; // ❌ Error

✓ Convert:

10n + BigInt(5);

---