

Strings in JavaScript

What is a String?

A **string** is a sequence of characters used to represent text. It can contain letters, numbers, symbols, or even be empty.

In JavaScript, strings are written inside **single quotes**, **double quotes**, or **backticks**.

```
let single = 'Hello';  
let double = "World";  
let template = `Hello World`;
```

All three are valid, but backticks (```) are useful for string interpolation (covered later).

Declaring Strings

```
let message = "Welcome to JavaScript!";  
let name = 'Salman';
```

String Length

You can check how many characters are in a string using the `.length` property.

```
let msg = "Hello";  
console.log(msg.length); // 5
```

Multiline Strings

Using `\n` (newline character):

```
let text = "Line 1\nLine 2\nLine 3";  
console.log(text);
```

Using template literals (backticks):

```
let multiline = `This is line 1  
This is line 2  
This is line 3`;  
console.log(multiline);
```

String Indexing

Strings are indexed like arrays — the first character is at position `0`.

```
let word = "JavaScript";  
console.log(word[0]); // "J"  
console.log(word[4]); // "S"
```

Strings are Immutable

You cannot change a specific character in a string directly.

```
let greeting = "Hello";  
greeting[0] = "Y"; // This will not work  
console.log(greeting); // Still "Hello"
```

To change a string, you have to **create a new one**.

Concatenation (Combining Strings)

Using `+` operator:

```
let firstName = "Salman";  
let lastName = "Khan";  
let fullName = firstName + " " + lastName;  
console.log(fullName); // "Salman Khan"
```

Summary

- Strings are sequences of characters.
- Use single, double, or backticks to define them.
- Strings are immutable.
- You can access characters using indices.
- Use `+` for concatenation or backticks for string interpolation.