

Day 7



Strings in JavaScript:

What is a String?

A string in JavaScript is a sequence of characters enclosed in quotes. Strings can include letters, numbers, symbols, or spaces.

Syntax

```
let str1 = "Hello"; // Double quotes
```

```
let str2 = 'World'; // Single quotes
```

```
let str3 = `JS`; // Backticks (Template literals)
```

Key Points

1. Strings can be enclosed in single (') or double (") quotes.
2. Backticks ` allow template literals — embedding variables and expressions.
3. Strings are immutable — cannot change individual characters directly.

String Methods (Basics)

Method	Description	Example
toUpperCase()	Converts to uppercase	"hello".toUpperCase() → "HELLO"
toLowerCase()	Converts to lowercase	"HELLO".toLowerCase() → "hello"
trim()	Removes spaces from both ends	" hi ".trim() → "hi"
includes()	Checks if substring exists	"Hello".includes("He") → true
indexOf()	Returns first index of substring	"Hello".indexOf("l") → 2
slice()	Extracts part of a string	"Hello".slice(1, 4) → "ello"
replace()	Replaces part of the string	"Hello".replace("H", "J") → "Jello"

Example: a very basic example of string, also 3 ways to declare and initialize

```
JS strings.js > ...
1 let str1 = "Hello"; // Double quotes
2 let str2 = 'World'; // Single quotes
3 let str3 = `JS`;    // Backticks (Template literals)
```

Indexing in string:

The string "Hello" has 5 characters: H, e, l, l, o.

Character	H	e	l	l	o
Index	0	1	2	3	4

Now, to get the length of the string, we use .length:

```
5 let s1 = "Aditya";
6 console.log(s1);
7 console.log(s1.length);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS E:\JavaScript\Day1-10\Day7> node .\strings.js
Aditya
6
```

Now, to get the character as per the index value:

```
9 let s1 = "Aditya";
10 console.log(s1[0]);
11 console.log(s1[1]);
12 console.log(s1[2]);
13 console.log(s1[3]);
14 console.log(s1[4]);
15 console.log(s1[5]);
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS E:\JavaScript\Day1-10\Day7> node .\strings.js
A
d
i
t
y
a
```

Example: index to be printed is 1 more than the length of string, then undefined is returned.

```
9   let s1 = "Aditya";
10  console.log(s1[6]);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day7> node .\strings.js
undefined
```

Example: string using backtick

```
18  //Using back tick to print
19  let s1 = `Aditya`;
20  let s2 = `Utsav`;
21  console.log(s1);
22  console.log(s2);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day7> node .\strings.js
Aditya
Utsav
```

Example: printing variable in backtick, called string interpolation

```
24  //Using backtick and $
25  let s1 = "Aditya";
26  let s2 = "Utsav";
27  console.log(`${s2} is a friend of ${s1}`);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day7> node .\strings.js
Utsav is a friend of Aditya
```

Example: template literal and quotes

```
30  //Template literal and "" and ''
31  let s1 = `Aditya is "good" boy not like 'Utsav'`;
32  console.log(s1);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day7> node .\strings.js
Aditya is "good" boy not like 'Utsav'.
```

Common Escape Sequences

Escape Sequence	Meaning	Example
\'	Single quote	'It\'s nice' → It's nice
\"	Double quote	"He said \"Hi\"" → He said "Hi"
\\	Backslash	"C:\\Users\\Admin" → C:\Users\Admin
\n	Newline	"Hello\nWorld" →

Example: valid writing of double and single quotes in JS without using escape sequence character

```
let s1 = "Aditya";  
let s2 = "Utsa'v"; //valid  
let s3 = 'Utsa"v'; //valid
```

Example: using the escape sequence character

```
41 let s2 = "Uts\\""; // Escapes the double quotes inside  
42 console.log(s2);    // Output: Uts"  
43 let str = 'It\'s nice';  
44 console.log(str);   // Output: It's nice
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day7> node .\strings.js  
Uts"  
It's nice
```

Also,

- Template literals are ES6 feature.
- Strings are 0-indexed.
- Strings are immutable.
- + operator can concatenate numbers as strings if one operand is a string.
- Multi-line strings cannot be created with single or double quotes.

Example: using + on strings

```
52 console.log(4+2);
53 let s1 = "Adi";
54 console.log(s1+2);
55 let s2 = 2;
56 console.log(s1+s2);
57 console.log(2+s2);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day7> node .\strings.js
6
Adi2
Adi2
4
```

```
59 let s3 = "2";
60 console.log(s3+2)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day7> node .\strings.js
22
```

Example: strings are immutable

```
62 //Strings are immutable
63 let s1 = "Aditya";
64 let s2 = s1;
65
66 console.log(s1);
67 console.log(s2);
68
69 s1 = "Adi";
70 console.log(s1);
71 console.log(s2);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\JavaScript\Day1-10\Day7> node .\strings.js
Aditya
Aditya
Adi
Aditya
```

--The End--