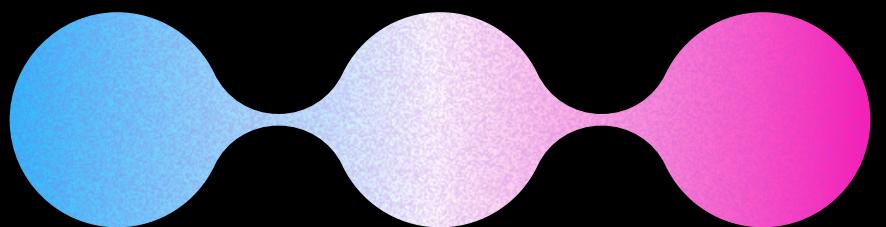
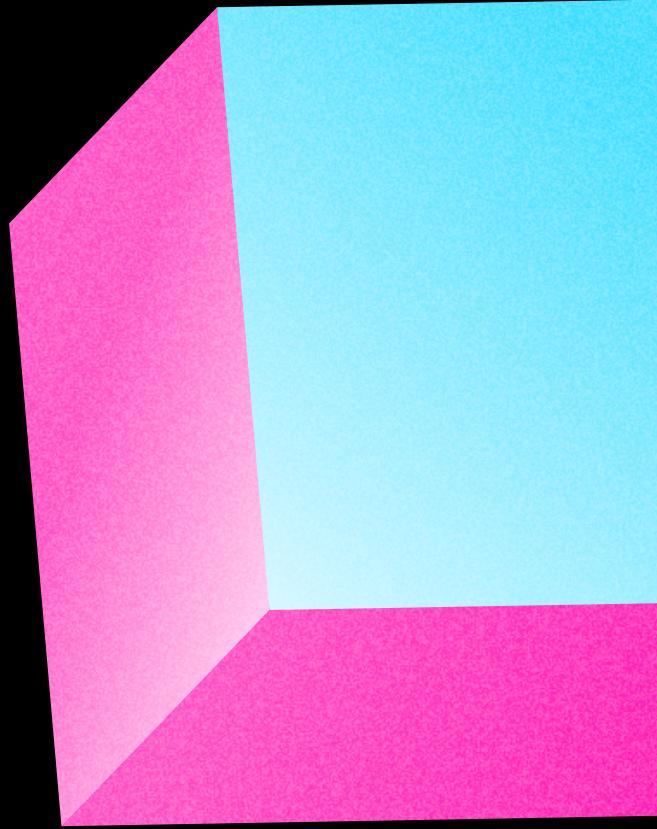


@rasi kumar

TEMPLATE LITERALS IN JAVASCRIPT





BRIEF INTRO

Template literals, introduced in **ECMAScript 6 (ES6)**, provide a way to create strings with embedded expressions in JavaScript. They are enclosed by backticks (` `) instead of single or double quotes. Template literals support multi-line strings and interpolation, making string manipulation more convenient and readable.

Here's how you can use **template literals** in JavaScript:

```
● ● ●  
1 const name = 'Max';  
2 const age = 30;  
3  
4 // Using template literals  
5 const greeting = `Hello, my name is ${name} and I am ${age} years old.`;  
6  
7 console.log(greeting);
```





EXAMPLE OF LITERALS

In this Example

- We define variables `'name'` and `'age'`.
- We create a string `'greeting'` using template literals, enclosing it within backticks (````).
- Within the template literal, we can embed variables and expressions using `${ }` syntax. These expressions are evaluated and concatenated into the resulting string.
- When we log `'greeting'` to the console, it will display: `'Hello, my name is John and I am 30 years old'`.

... ● ● ●



PROGRAM & OUTPUT OF LITERALS

Template literals also support **multi-line strings** without the need for special escape characters:

PROGRAM

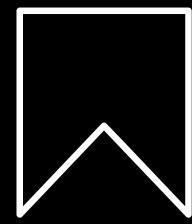
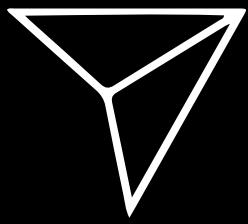
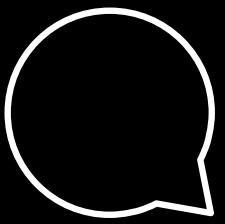
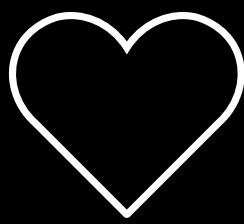
```
● ● ●  
1 const multiLineString = `  
2   This is a multi-line  
3   string using template literals.  
4   It preserves whitespace and line breaks.  
5 `;  
6  
7 console.log(multiLineString);  
8
```

OUTPUT

```
This is a multi-line  
string using template literals.  
It preserves whitespace and line breaks.
```

• • • •

FOLLOW FOR MORE TIPS!



@rasikumar