

ChatGPT: JavaScript Template Literals

In JavaScript, what does ``${expression}`` do?

In JavaScript, `${expression}` is used for **string interpolation** inside **template literals** (also known as template strings).

How It Works

Template literals use **backticks** (```) instead of **quotes** (`'` or `"`), and `${expression}` allows you to embed JavaScript expressions inside a string.

1. String Interpolation

You can insert variables or expressions inside a string:

```
const name = "Alice";  
console.log(`Hello, ${name}!`); // Output: Hello, Alice!
```

2. Evaluate Expressions

You can insert **mathematical expressions** inside `${}`:

```
const a = 5, b = 10;  
console.log(`Sum: ${a + b}`); // Output: Sum: 15
```

3. Multi-line Strings

Template literals allow multi-line strings **without needing** `\n`:

```
const message = `This is a  
multi-line string.`;  
console.log(message);
```

4. Calling Functions Inside `${}`

You can execute functions directly within template literals:

```
function greet() {  
  return "Welcome!";  
}
```

```
}  
  
console.log(` ${greet()} Have a great day!`);  
  
// Output: Welcome! Have a great day!
```

5. Using Objects Inside Template Strings

```
const user = { name: "Bob", age: 30 };  
  
console.log(` User: ${user.name}, Age: ${user.age}`);  
  
// Output: User: Bob, Age: 30
```

6. Nesting Template Literals

You can **nest** template literals inside `${}`:

```
const name = "Charlie";  
  
const message = `Hello, ${`dear ${name}`}`;  
  
console.log(message); // Output: Hello, dear Charlie
```

When to Use `${}`

- ✅ When you need **string interpolation**
- ✅ When working with **multi-line strings**
- ✅ When **dynamically inserting values** inside a string

Would you like an advanced example using `${}` inside a tagged template function? 🚀