Template literals and tagged templates

Absolutely! Let's dive deep into **Template Literals** and **Tagged Templates** in JavaScript.

🔷 Template Literals

Template literals are **string literals** that allow:

- Multi-line strings
- Expression interpolation
- Embedded expressions
- Tagged templates (advanced use)

🔶 Syntax

Use **backticks** (` `) instead of quotes:

```
const name = "Abhi";
const greeting = `Hello, ${name}!`;

console.log(greeting); // "Hello, Abhi!"
```

🔶 Features of Template Literals

1. Multi-line Strings

```
const multiLine = `This is
a multi-line
string.`;
```

```
console.log(multiLine);
```

2. Expression Interpolation

You can embed any valid JS expression using `${}`:

```
const a = 5;
const b = 10;
console.log(`The sum is ${a + b}`); // "The sum is 15"
```

3. Function calls in interpolation

```
function greet() {
  return "👋";
}
console.log(`Hello ${greet()}`); // "Hello 👋"
```

🔷 Tagged Templates

Tagged templates are an advanced form of template literals where a **function processes** the template literal.

🔶 Syntax

```
function tag(strings, ...values) {
  console.log(strings); // Array of literal strings
  console.log(values);  // Interpolated values
}

const name = "Abhi";
const age = 22;

tag`My name is ${name} and I am ${age} years old.`;
```

Output:

```
strings = ["My name is ", " and I am ", " years old."]
values = ["Abhi", 22]
```

You can use this to:

- Sanitize input
- Translate content
- Format output
- Escape HTML

🔶 Example: Custom Formatter Tag

```
function highlight(strings, ...values) {
  return strings.reduce((acc, str, i) => `${acc}${str}<b>${values[i] ?? ''}</b>`,
"");
}

const lang = "JavaScript";
const result = highlight`I love ${lang} so much!`;

console.log(result); // I love <b>JavaScript</b> so much!
```

🔷 Real-world Use Cases

Use Case	Example
HTML escaping	Prevent XSS with custom `html` tag function
Localization	Create `i18n` tagged functions for translation
Logging/Debugging	Tagged loggers to format output
Styling in CSS-in-JS	Styled-components in React uses tagged templates

---

✅ Summary

Feature	Template Literal	Tagged Template
Basic Syntax	`Hello, ${name}`	tag`Hello, ${name}`
Multi-line support	✅	✅
Expression support	✅	✅ (via values array)
Function control	❌	✅ (your function processes values)
Use case	Readable strings	Custom parsing, formatting, escaping

---

Would you like a **quiz**, **code practice**, or an **example project** using tagged templates (e.g. styled-components in React)?