

동어 : 템플릿 리터럴(Template Literals), 템플릿 문자열(Template Strings), 문자열 템플릿(String Templates), Back-Ticks 구문(Back-Ticks Syntax)

1. Back-Ticks 구문

템플릿 리터럴은 따옴표("") 대신 백틱(``)을 사용하여 문자열을 정의합니다.

예시

```
let text = `Hello World!`;
```

1.1 문자열 내부의 따옴표

템플릿 리터럴을 사용하면 문자열 내에서 작은따옴표와 큰따옴표를 모두 사용할 수 있습니다.

예시

```
let text = `He's often called "Johnny"`;
```

1.2 여러 줄 문자열

템플릿 리터럴은 여러 줄 문자열을 허용합니다.

예시

```
let text =  
`The quick  
brown fox  
jumps over  
the lazy dog`;
```

2. Interpolation(보간, 내삽법)

템플릿 리터럴은 변수와 표현식을 문자열에 쉽게 삽입할 수 있는 방법을 제공합니다.

이 방법을 문자열 보간이라고 합니다.

구문

```
${...}
```

2.1 변수 대체

템플릿 리터럴은 문자열의 변수를 허용합니다.

예시

```
let firstName = "John";
let lastName = "Doe";

let text = `Welcome ${firstName}, ${lastName}!`;
```

변수를 실제 값으로 자동 대체하는 것을 문자열 보간 이라고 합니다 .

2.2 식 대체

템플릿 리터럴은 문자열의 표현식을 허용합니다.

예시

```
let price = 10;
let VAT = 0.25;

let total = `Total: ${price * (1 + VAT).toFixed(2)}`;
```

표현식을 실제 값으로 자동 대체하는 것을 문자열 보간 이라고 합니다 .

2.3 HTML 템플릿

예시

```
<p id="demo"></p>

<p>Template literals are not supported in Internet Explorer.</p>

<script>
let header = "Templates Literals";
let tags = ["template literals", "javascript", "es6"];

let html = `<h2>${header}</h2><ul>`;

for (const x of tags) {
  html += `<li>${x}</li>`;
}

html += `</ul>`;
document.getElementById("demo").innerHTML = html;
</script>
```