

# 템플릿 리터럴

ES6는 템플릿 리터럴(Template literal)이라고 불리는 새로운 문자열 표기법을 도입하였다. 템플릿 리터럴은 일반 문자열과 비슷해 보이지만, ' 또는 " 같은 통상적인 따옴표 문자 대신 백틱(backtick) 문자 ```를 사용한다.

```
const template = `템플릿 리터럴은 '작은따옴표(single quotes)'과  
"큰따옴표(double quotes)"를 혼용할 수 있다.`;  
  
console.log(template);
```

일반적인 문자열에서 줄바꿈은 허용되지 않으며 공백(white-space)를 표현하기 위해서는 백슬래시(\)로 시작하는 이스케이프 시퀀스(Escape Sequence)를 사용하여야 한다. ES6 템플릿 리터럴은 일반적인 문자열과 달리 여러 줄에 걸쳐 문자열을 작성할 수 있으며 템플릿 리터럴 내의 모든 white-space는 있는 그대로 적용된다.

```
const template = `

  
  <li><a href="#home">Home</a></li>  
  <li><a href="#news">News</a></li>  
  <li><a href="#contact">Contact</a></li>  
  <li><a href="#about">About</a></li>  
</ul>`;  
  
console.log(template);
```

템플릿 리터럴은 + 연산자를 사용하지 않아도 간단한 방법으로 새로운 문자열을 삽입할 수 있는 기능을 제공한다. 이를 문자열 인터폴레이션(String Interpolation)이라 한다.

```
const first = 'Ung-mo';  
const last = 'Lee';
```

```
// ES5: 문자열 연결
console.log('My name is ' + first + ' ' + last + '.');
// "My name is Ung-mo Lee."

// ES6: String Interpolation
console.log(`My name is ${first} ${last}.`);
// "My name is Ung-mo Lee."
```

문자열 인터플레이션은 `${ ... }` 으로 표현식을 감싼다. 문자열 인터플레이션 내의 표현식은 문자열로 강제 타입 변환된다.

```
console.log(`1 + 1 = ${1 + 1}`); // "1 + 1 = 2"
```