

문자열 리터럴 : 문자열 (``),  
문자열 리터럴 (\$ { })

## # 문자열 리터럴 (Template Literal)

ES6에서 문자열 리터럴을  
문자열 리터럴로, 문자열 리터럴

```
var str_01 = `hello world`;
```

```
//문자열 리터럴  
const x='x';  
const y='y';  
const z='z';  
  
console.log(`${x} x, ${y} y, ${z} z`); //x x, y y, z z
```

▶ 문자열 리터럴은 JS ES6(ECMAScript 2015)에서 도입되었습니다.

▶ 문자열 리터럴은 문자열 리터럴을

▶ 문자열 리터럴을 문자열 리터럴로

## # 문자열 리터럴 - 문자열 리터럴

문자열 리터럴: Multi-line strings)

▶ 문자열 리터럴은 문자열 리터럴, 문자열 리터럴을

```
var str_01 = `  
Hi!  
  It's me!  
    JavaScript!  
So Cool!`;
```

<문자열 리터럴>

image.png

▶ 在字符串中，反斜杠（\）用于转义特殊字符，如换行符（\n）、制表符（\t）等。

```
var str_01 = "Hi! \n\t It's me! \n\t\t JavaScript! \n So Cool!";
```

▶ \n 换行符 (Line Feed), \t 制表符 (Tab)

## # 模板字符串

模板字符串使用反引号（`）包裹，可以包含换行符、制表符等特殊字符，无需转义。

例如，以下代码展示了如何定义和使用模板字符串：

```
function a(x){  
  return x+1;  
}
```

```
a(3); //31  
a`4`; //41
```

## # 模板字符串插值 (Expression interpolation)

▶ 在模板字符串中，使用 \${} 语法可以在字符串中插入变量或表达式。

```
var name = `张三`  
var price = 100  
var num = 5;  
  
console.log(`${name} 的价格是 ${price * num} 元。`);
```

<模板字符串>

image.png

▶ 在模板字符串中，使用 \${} 语法可以在字符串中插入变量或表达式。

▶ 變數 name 儲存字串, + 變數 price 與 變數 num

```
var name = `  `
var price = 100
var num = 5;

console.log(name + "  " + (price * num) + "  " );
```

▶ 變數 price 儲存字串. 變數 num 儲存字串

```
console.log(`${5+10}  `) // 15  
```

```
console.log(`${5+10}  `) // 15  
```

---

Revision #4

Created 3 January 2025 18:38:11 by DainChoi

Updated 19 April 2025 08:19:50 by DainChoi