

# Template Literals and String Methods in Javascript

## Examples of using Template Literals:

### standard method - variables and concatenation:

```
const name = "John";
```

```
const age = 30;
```

```
let x = "Hello, my name is " + name + "and I am " + age + "years old."
```

```
console.log(x);
```

### template literals - variables and concatenation:

(preferred method, more concise)

```
const name = "John";
```

```
const age = 30;
```

```
let x = `Hello, my name is ${name} and I am ${age} years old.`;
```

```
console.log(x);
```

## A note on how string methods work:

Javascript within the browser and within Node will wrap strings inside a string object and then use the method on that object to perform this work.

Strings have methods that can be used on them to transform them.

JavaScript provides many built-in methods for working with strings. The most common methods are used for tasks such as extracting parts of a string, modifying its content or case, and searching for substrings.

### Common JavaScript String Methods:

Here are some of the most frequently used string methods in JavaScript:

- **length property:** Returns the length of a string (not a method, but a crucial property).
- **slice(start, end):** Extracts a section of a string and returns a new string. It accepts negative indices.
- **substring(start, end):** Similar to slice(), it extracts characters between two specified indices, but does not accept negative indices.
- **substr(start, length):** Extracts a specified number of characters from a starting index (less commonly used in modern JS).
- **replace(searchValue, newValue):** Replaces the first occurrence of a specified value or a regular expression with another value. Use `replaceAll()` to replace all occurrences.
- **toUpperCase():** Converts all the characters in a string to uppercase letters.
- **toLowerCase():** Converts all the characters in a string to lowercase letters.
- **concat(string1, ...):** Joins two or more strings together, returning a new combined string. The `+` operator is often used as a simpler alternative.
- **trim():** Removes whitespace from both the beginning and the end of a string.
- **split(separator):** Splits a string into an array of substrings based on a specified separator.
- **indexOf(searchValue):** Returns the index of the first occurrence of a specified substring. If not found, it returns `-1`.
- **lastIndexOf(searchValue):** Returns the index of the last occurrence of a specified substring.
- **includes(searchValue):** Checks if a string contains a specified substring and returns `true` or `false`.
- **startsWith(searchValue):** Checks if a string begins with a specified substring.
- **endsWith(searchValue):** Checks if a string ends with a specified substring.
- **charAt(index):** Returns the character at a specified index (position) within the string.

### **Example of using the length property:**

```
const h ="Hello World";  
  
let z = "Number of Letters in Hello World: " + h.length;  
  
console.log(z);
```

### **Example of using the typeof property:**

```
let y = "The type of the first value is: " + typeof h;  
  
console.log(y);  
let w = "The type of the second value is: " + typeof age;  
  
console.log(w);
```

### **Example of using toUpperCase method:**

```
let j = "hello there new world."  
console.log(j.toUpperCase());
```

### **Example of using toLowerCase() method:**

```
let k = "SCREAMER SAYS, I HATE SMALL LETTERS!";  
console.log(k.toLowerCase());
```