

# JavaScript String Methods — Complete Master Guide

This document contains **all important JavaScript string methods** with:

- Definition (simple English)
- Syntax
- Input → Output
- Step-by-step explanation
- Real-time use cases

This is designed so students can clearly understand what each string method does.

---

## 1. length

### Definition

Returns how many characters are in a string.

### Syntax

```
string.length
```

### Input → Output

Input:

```
"hello".length
```

Output:

```
5
```

### Explanation

Counts characters one by one.

### Real-time use

- Validate password length.

---

## 2. charAt()

### Definition

Returns the character at a specific index.

### Syntax

```
string.charAt(index)
```

### Input → Output

Input:

```
"hello".charAt(1)
```

Output:

```
"e"
```

### Step-by-step

Index starts at 0, so index 1 is "e".

### Real-time use

- Extract specific characters.
- 

## 3. charCodeAt()

### Definition

Gives the ASCII/Unicode number of a character.

### Syntax

```
string.charCodeAt(index)
```

## **Input → Output**

Input:

```
"A".charCodeAt(0)
```

Output:

```
65
```

---

## **4. toUpperCase()**

### **Definition**

Converts all letters to CAPITALS.

### **Syntax**

```
string.toUpperCase()
```

## **Input → Output**

Input:

```
"hello".toUpperCase()
```

Output:

```
"HELLO"
```

---

## **5. toLowerCase()**

### **Definition**

Converts all letters to lowercase.

## Syntax

```
string.toLowerCase()
```

## Input → Output

```
"HELLO" → "hello"
```

---

## 6. trim()

### Definition

Removes spaces from start and end.

## Input → Output

```
" hello " → "hello"
```

---

## 7. trimStart() / trimEnd()

### Definition

Remove spaces only from start or end.

---

## 8. includes()

### Definition

Checks if a string contains a substring.

## Syntax

```
string.includes(value)
```

## **Input → Output**

```
"javascript".includes("script") → true
```

---

## **9. startsWith() / endsWith()**

### **Definition**

Checks if string begins or ends with value.

## **Input → Output**

```
"hello".startsWith("he") → true  
"hello".endsWith("lo") → true
```

---

## **10. indexOf()**

### **Definition**

Returns index where substring first appears.

## **Input → Output**

```
"hello".indexOf("l") → 2
```

---

## **11. lastIndexOf()**

### **Definition**

Returns last position where substring appears.

## **Input → Output**

```
"hello".lastIndexOf("l") → 3
```

## 12. slice()

### Definition

Cuts part of string and returns new string.

### Syntax

```
string.slice(start, end)
```

### Input → Output

```
"javascript".slice(0,4) → "java"
```

---

## 13. substring()

### Definition

Similar to slice but handles negatives differently.

### Input → Output

```
"javascript".substring(4,10) → "script"
```

---

## 14. substr()

### Definition (OLD method)

Takes a start index and length.

### Input → Output

```
"javascript".substr(4,6) → "script"
```

## 15. split()

### Definition

Breaks a string into an array using a separator.

### Syntax

```
string.split(separator)
```

### Input → Output

```
"a,b,c".split(",") → ["a", "b", "c"]
```

---

## 16. repeat()

### Definition

Repeats a string multiple times.

### Input → Output

```
"hi".repeat(3) → "hihihi"
```

---

## 17. replace()

### Definition

Replaces first matching substring.

### Input → Output

```
"hello world".replace("world", "JS") → "hello JS"
```

## 18. replaceAll()

### Definition

Replaces ALL matches.

### Input → Output

```
"a-a-a".replaceAll("-", "+") → "a+a+a"
```

---

## 19. concat()

### Definition

Joins two or more strings.

### Input → Output

```
"Hello".concat(" ", "World") → "Hello World"
```

---

## 20. padStart() / padEnd()

### Definition

Adds padding to start or end.

### Input → Output

```
"5".padStart(3, "0") → "005"
```

---

## 21. toString()

Converts value to string.

```
(100).toString() → "100"
```

---

## 22. valueOf()

Returns primitive value of String object.

```
new String("JS").valueOf() → "JS"
```

---

# Detailed Examples for All Methods

Below are **three detailed examples for each string method**, using variables instead of direct strings.

---

### 1. length — Examples

```
let name = "javascript";
console.log(name.length); // 10

let city = "Hyderabad";
console.log(city.length); // 9

let empty = "";
console.log(empty.length); // 0
```

---

### 2. charAt() — Examples

```
let word = "developer";
console.log(word.charAt(0)); // d
console.log(word.charAt(3)); // e
console.log(word.charAt(word.length - 1)); // r
```

---

### 3. charCodeAt() — Examples

```
let letter = "A";
console.log(letter.charCodeAt(0)); // 65

let ch = "z";
console.log(ch.charCodeAt(0)); // 122
```

```
let symbol = "@";
console.log(symbol.charCodeAt(0)); // 64
```

---

## 4. `toUpperCase()` — Examples

```
let lang = "javascript";
console.log(lang.toUpperCase()); // JAVASCRIPT

let country = "india";
console.log(country.toUpperCase()); // INDIA

let mix = "Hello World";
console.log(mix.toUpperCase()); // HELLO WORLD
```

---

## 5. `toLowerCase()` — Examples

```
let title = "WELCOME";
console.log(title.toLowerCase()); // welcome

let mode = "DaRK";
console.log(mode.toLowerCase()); // dark

let phrase = "GOOD MORNING";
console.log(phrase.toLowerCase()); // good morning
```

---

## 6. `trim()` — Examples

```
let text = "    hello    ";
console.log(text.trim()); // "hello"

let msg = "    welcome";
console.log(msg.trim()); // "welcome"

let pad = "bye    ";
console.log(pad.trim()); // "bye"
```

## 7. split() — Examples

```
let items = "apple,banana,grapes";
console.log(items.split(","));
// ["apple", "banana", "grapes"]

let sentence = "I love coding";
console.log(sentence.split(" "));
// ["I", "love", "coding"]

let chars = "ABC";
console.log(chars.split(""));
// ["A", "B", "C"]
```

## 8. join (Array → String)

```
let arr1 = ["a", "b", "c"];
console.log(arr1.join("-"));
// a-b-c

let words = ["Hello", "World"];
console.log(words.join(" "));
// Hello World

let digits = [1, 2, 3];
console.log(digits.join(""));
// 123
```

# Real-Time Company Use Cases

## 1. Clean user data

```
name.trim().toLowerCase()
```

## 2. Create URL slug

```
"Hello World".toLowerCase().replaceAll(" ", "-")
```

## 3. Validate email

```
email.includes("@")
```

#### 4. Extract file extension

```
"photo.png".split(".").pop() → "png"
```

---

---