

String methods that every **js** dev should know

RAJ BHENSDADIYA

- Strings are fundamental in JavaScript, and knowing how to manipulate them is crucial.
- Let's dive into some powerful string methods with real-world examples!

String length

- This method determines the number of characters in a string.



JS stringMethods.js

```
const myString = 'Hello, World!';  
const length = myString.length;  
console.log(`Length of the string: ${length}`);  
// Output: Length of the string: 13
```

String slice

- Slice is used to extract a portion of a string and return it.



JS stringMethods.js

```
const sentence = 'JavaScript is amazing!';
const sliced = sentence.slice(0, 10);
console.log(`sliced: ${sliced}`);
// Output: Sliced: JavaScript
```

String substring

- Similar to slice, it extracts a part of a string but doesn't accept negative indexes.



JS stringMethods.js

```
const text = 'OpenAI is innovative!';
const sub = text.substring(7, 12);
console.log(`Substring: ${sub}`);
// Output: Substring: is in
```

String substr

- Extracts a specific number of characters from a string, starting from a given index.



JS stringMethods.js

```
const phrase = 'Web development is fun!';
const substr = phrase.substr(4, 12);
console.log(`Substr: ${substr}`);
// Output: Substr: development
```

String replace

- Replaces a specified substring with another string.

JS stringMethods.js

```
const original = 'I love JavaScript!';
const replaced = original.replace('JavaScript', 'coding');
console.log(`Replaced: ${replaced}`);
// Output: Replaced: I love coding!
```

String replaceAll

- Replaces all occurrences of a substring with another string.



JS stringMethods.js

```
const textWithDots = 'Replace all the dots with hyphens... please!';  
const replacedAll = textWithDots.replaceAll('.', '-');  
console.log(`Replaced All: ${replacedAll}`);  
// Output: Replaced All: Replace all the dots with hyphens--- please!
```

String toUpperCase

- Converts a string to uppercase.



JS stringMethods.js

```
const mixedCase = 'HeLLo WoRLD!';
console.log(mixedCase.toUpperCase());
// Output: HELLO WORLD!
```

String toLowerCase

- Converts a string to lowercase.



JS stringMethods.js

```
const mixedCase = 'HeLLo WoRLD!';
console.log(mixedCase.toUpperCase());
// Output: HELLO WORLD!
```

String trim

- Removes whitespace from both ends of a string.



JS stringMethods.js

```
const paddedString = '    Trim me!    ';
const trimmed = paddedString.trim();
console.log(`Trimmed: "${trimmed}"`);
// Output: Trimmed: "Trim me!"
```

String trimStart

- Removes whitespace from the beginning of a string.



JS stringMethods.js

```
const paddedString = '    Trim me!    ';
const trimmedStart = paddedString.trimStart();
console.log(`Trimmed Start: "${trimmedStart}"`);
// Output: Trimmed Start: "Trim me!"
```

String trimEnd

- Removes whitespace from the end of a string.

JS stringMethods.js

```
const paddedString = '    Trim me!    ';
const trimmedEnd = paddedString.trimEnd();
console.log(`Trimmed End: "${trimmedEnd}"`);
// Output: Trimmed End: "    Trim me!"
```

String padStart

- Adds padding to the beginning of a string.



JS stringMethods.js

```
const number = '42';
const paddedNumber = number.padStart(5, '0');
console.log(`Padded Number: ${paddedNumber}`);
// Output: Padded Number: 00042
```

String padEnd

- Adds padding to the end of a string.



JS stringMethods.js

```
const number = '42';
const paddedNumber = number.padEnd(5, '0');
console.log(`Padded Number: ${paddedNumber}`);
// Output: Padded Number: 42000
```

String charAt

- Returns the character at a specified index in the string.



JS stringMethods.js

```
const text = 'Hello, World!';
const character = text.charAt(7);
console.log(`Character at index 7: ${character}`);
// Output: Character at index 7: W
```

String charCodeAt

- Returns the Unicode value of the character at a specified index in the string.



JS stringMethods.js

```
const text = 'Hello, World!';
const unicodeValue = text.charCodeAt(0);
console.log(`Unicode value at index 0: ${unicodeValue}`);
// Output: Unicode value at index 0: 72
```

String split

- Splits a string into an array of substrings based on a specified delimiter.



JS stringMethods.js

```
const sentence = 'JavaScript is amazing!';
const words = sentence.split(' ');
console.log(`Split into words: ${words}`);
// Output: Split into words: JavaScript,is,amazing!
```

I'm Raj.

Software Engineer;
passionate for building and shipping
scalable software.

I am open to full-stack development
projects. (JavaScript, Python, AI/ML,
DevOps)

Let's connect and expand our
professional network together.

I'm excited to collaborate, exchange
ideas, and contribute to impactful
projects.