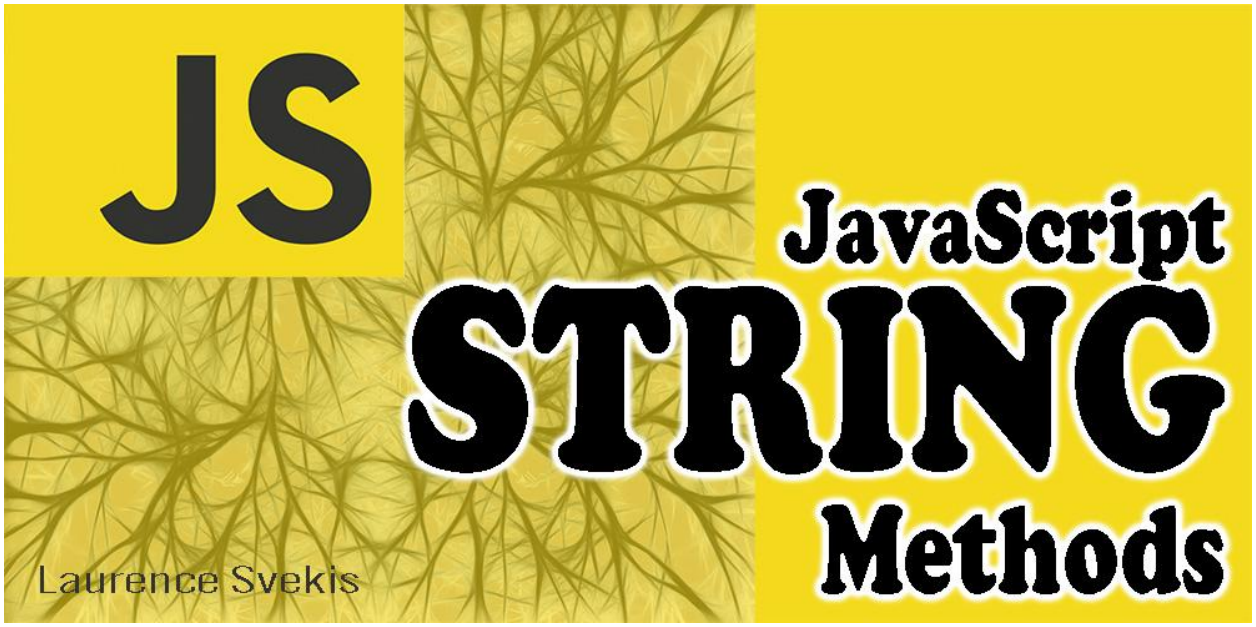
String Methods in JavaScript



Introduction to String Methods in JavaScript

JavaScript, one of the core technologies for web development, allows developers to work with text and strings extensively. Strings are sequences of characters enclosed in single or double quotes. JavaScript provides a rich set of built-in methods that make it easy to manipulate, search, extract, and transform strings. These methods are part of the String object and can be used to perform a wide range of tasks efficiently.

In this introduction, we'll explore some of the most commonly used string methods in JavaScript and provide examples to illustrate their usage.
length: The length property of a string returns the number of characters in the string. It's a property, not a method, but it's an essential part of working with strings.

```
const message = "Hello, JavaScript!";
const length = message.length; // 17
```

charAt(index):

The charAt() method returns the character at a specified index in the string. String indices are zero-based.

```
const word = "JavaScript";
const char = word.charAt(2); // "v"
```

substring(start, end):

The substring() method extracts a portion of a string between two specified indices (start and end).

```
const text = "Hello, World!";
const result = text.substring(0, 5); // "Hello"
```

split(separator):

The split() method divides a string into an array of substrings based on a specified separator.

```
const fruits = "apple,banana,cherry";
const fruitArray = fruits.split(","); // ["apple",
"banana", "cherry"]
```

indexOf(searchValue):

The indexOf() method returns the index of the first occurrence of a specified value within the string.

```
const sentence = "JavaScript is amazing!";
```

```
const index = sentence.indexOf("amazing"); // 13
```

replace(searchValue, newValue):

The replace() method replaces the first occurrence of a specified value with another value.

```
const message = "Hello, World!";
const newMessage = message.replace("World",
"Universe"); // "Hello, Universe!"
```

These are just a few of the many string methods available in JavaScript. Whether you're working on web applications, data processing, or any JavaScript project, having a solid understanding of these methods will be immensely beneficial. String methods enable you to manipulate and work with text effectively, making your JavaScript code more powerful and versatile.

Coding exercises for practicing JavaScript string methods:

Exercise 1: String Length

Write a program that takes a user's input and calculates the length of the input string using the length property.

Exercise 2: Character at Index

Create a program that prompts the user for a string and an index. Then, use the charAt() method to display the character at that index.

Exercise 3: Extract Substring

Write a program that takes a string and two indices (start and end). Use the substring() method to extract and display the substring between the given indices.

Exercise 4: Split and Count Words

Take a sentence as input and split it into words using the split() method. Count and display the number of words in the sentence.

Exercise 5: Search and Replace

Create a program that takes a sentence and allows the user to search for a word or phrase and replace it with another word or phrase using the replace() method.

Exercise 6: Find Index of Character

Write a program that takes a string and a character as input. Use the indexOf() method to find and display the index of the first occurrence of that character in the string.

Exercise 7: Reverse a String

Implement a program that reverses a given string using string manipulation methods.

Exercise 8: Remove Whitespace

Take a sentence with extra whitespace and use string methods to remove the extra spaces, leaving only single spaces between words.

Exercise 9: Check Palindrome

Write a program that checks if a given string is a palindrome (reads the same forwards and backwards) using string methods.

Exercise 10: Title Case Conversion

Create a program that takes a sentence and converts it to title case (the first letter of each word capitalized) using string methods.

Here are the solutions to these exercises:

Exercise 1: String Length

```
const userInput = prompt("Enter a string:");
const length = userInput.length;
console.log(`Length of the string: ${length}`);
```

Exercise 2: Character at Index

```
const userInput = prompt("Enter a string:");
const index = parseInt(prompt("Enter an index:"));
const charAtIndex = userInput.charAt(index);
```

```
console.log(`Character at index ${index}:
${charAtIndex}`);
```

Exercise 3: Extract Substring

```
const userInput = prompt("Enter a string:");
const start = parseInt(prompt("Enter the start
index:"));
const end = parseInt(prompt("Enter the end index:"));
const substring = userInput.substring(start, end);
console.log(`Substring: ${substring}`);
```

Exercise 4: Split and Count Words

```
const sentence = prompt("Enter a sentence:");
const words = sentence.split(" ");
console.log(`Number of words: ${words.length}`);
```

Exercise 5: Search and Replace

```
const sentence = prompt("Enter a sentence:");
const searchWord = prompt("Enter the word to search:");
const replaceWord = prompt("Enter the replacement
word:");
const replacedSentence = sentence.replace(searchWord,
replaceWord);
```

```
console.log(`Modified sentence: ${replacedSentence}`);
```

Exercise 6: Find Index of Character

```
const userInput = prompt("Enter a string:");
const charToFind = prompt("Enter a character:");
const index = userInput.indexOf(charToFind);
console.log(`Index of ${charToFind}: ${index}`);
```

Exercise 7: Reverse a String

```
const userInput = prompt("Enter a string:");
const reversedString =
userInput.split("").reverse().join("");
console.log(`Reversed string: ${reversedString}`);
```

Exercise 8: Remove Whitespace

```
const sentence = prompt("Enter a sentence with extra
spaces:");
const cleanedSentence = sentence.split(" ").filter(word
=> word !== "").join(" ");
console.log(`Cleaned sentence: ${cleanedSentence}`);
```

Exercise 9: Check Palindrome

```
const userInput = prompt("Enter a string:");
```

```
const reversedString =
userInput.split("").reverse().join("");
const isPalindrome = userInput === reversedString;
console.log(`Is it a palindrome? ${isPalindrome}`);
```

Exercise 10: Title Case Conversion

```
const sentence = prompt("Enter a sentence:");
const words = sentence.split(" ");
const titleCaseWords = words.map(word =>
word.charAt(0).toUpperCase() +
word.slice(1).toLowerCase());
const titleCaseSentence = titleCaseWords.join(" ");
console.log(`Title case sentence:
${titleCaseSentence}`);
```

Feel free to try these exercises to practice your string manipulation skills in JavaScript!

String methods Commonly used

Strings are fundamental data types in JavaScript, representing sequences of characters. JavaScript provides a variety of methods to manipulate and work with

strings. In this guide, we'll explore some common string methods along with coding examples.

length: The length property returns the number of characters in a string.

```
const greeting = "Hello, World!";
const length = greeting.length; // 13
```

charAt(index): The charAt() method returns the character at a specified index in the string.

```
const str = "JavaScript";
const char = str.charAt(2); // "v"
```

substring(start, end): The substring() method extracts a portion of a string between two specified indices.

```
const text = "Hello, World!";
const result = text.substring(0, 5); // "Hello"
```

slice(start, end): The slice() method is similar to substring() but allows negative indices to count from the end of the string.

```
const phrase = "I love JavaScript!";
const sliced = phrase.slice(2, -3); // "love JavaScript"
```

split(separator): The split() method divides a string into an array of substrings based on a specified separator.

```
const fruits = "apple,banana,cherry";
```

```
const fruitArray = fruits.split(","); // ["apple", "banana", "cherry"]
```

indexOf(searchValue): The indexOf() method returns the index of the first occurrence of a specified value within the string.

```
const sentence = "JavaScript is awesome!";
const index = sentence.indexOf("awesome"); // 13
```

replace(searchValue, newValue): The replace() method replaces the first occurrence of a specified value with another value.

```
const message = "Hello, John!";
const newMessage = message.replace("John", "Alice"); // "Hello, Alice!"
```

toUpperCase() and toLowerCase(): These methods change the case of characters in a string.

```
const mixedCase = "ThIs Is MiXeD CaSe";
const upperCase = mixedCase.toUpperCase(); // "THIS IS MIXED CASE"
```

trim(): The trim() method removes whitespace from both ends of a string.

```
const padded = "   Hello, World!   ";
const trimmed = padded.trim(); // "Hello, World!"
```

concat(): The concat() method combines two or more strings into a new string.

```
const firstName = "John";
const lastName = "Doe";
```

```
const fullName = firstName.concat(" ", lastName); // "John Doe"
```

These are just a few of the many string methods available in JavaScript. They enable you to manipulate and transform strings to suit your specific needs, making them a crucial part of any JavaScript programmer's toolkit.