

DOMINANDO STRINGS EM JAVASCRIPT



OS PRINCIPAIS MÉTODOS

MATEUS FERNANDES

Introdução às Strings em JavaScript:

As strings são sequências de caracteres, e no JavaScript, elas são tratadas como objetos. Vamos explorar os principais métodos para manipular strings de forma eficiente e poderosa.



1. length - Obtendo o Comprimento da String:

O método length retorna o número de caracteres em uma string.

```
const frase = "Olá, mundo!";  
console.log(frase.length); // Saída: 11
```



2. toLowerCase e toUpperCase - Transformando Maiúsculas e Minúsculas:

Os métodos `toLowerCase()` e `toUpperCase()` convertem uma string para minúsculas e maiúsculas, respectivamente.

```
const texto = "Hello World";  
console.log(texto.toLowerCase()); // Saída: "hello world"  
console.log(texto.toUpperCase()); // Saída: "HELLO WORLD"
```



3. charAt - Acessando Caracteres em uma Posição Específica:

O método charAt() retorna o caractere em uma posição específica em uma string.

```
const palavra = "JavaScript";  
console.log(palavra.charAt(2)); // Saída: "v"
```



4. indexOf e lastIndexOf - Encontrando a Posição de Substrings:

indexOf() retorna a posição da primeira ocorrência de uma substring, enquanto lastIndexOf() retorna a posição da última ocorrência.

```
const texto = "A raposa rápida saltou sobre o cachorro.";
console.log(texto.indexOf("rápida")); // Saída: 9
console.log(texto.lastIndexOf("sobre")); // Saída: 27
```



5. substring - Extraíndo Partes de uma String:

substring() retorna uma parte da string entre dois índices, ou do índice especificado até o final da string.

```
const frase = "Eu amo programação!";  
console.log(frase.substring(3, 6)); // Saída: "amo"  
console.log(frase.substring(7)); // Saída: "programação!"
```



6. split - Dividindo uma String em um Array:

split() divide uma string em substrings com base em um separador e retorna um array.

```
const lista = "maçã,banana,uva,morango";  
const frutas = lista.split(",");  
console.log(frutas); // Saída: ["maçã", "banana", "uva", "morango"]
```



7. trim - Removendo Espaços em Branco:

trim() remove espaços em branco do início e do final de uma string.

```
const texto = "    Olá, mundo!    ";  
console.log(texto.trim()); // Saída: "Olá, mundo!"
```



8. replace - Substituindo Substrings:

replace() substitui uma substring por outra em uma string.

```
const frase = "Eu gosto de bananas.";
console.log(frase.replace("bananas", "maçãs"));
// Saída: "Eu gosto de maçãs."
```



9. concat - Concatenando Strings:

concat() junta duas ou mais strings e retorna uma nova string.

```
const nome = "John";  
const sobrenome = "Doe";  
console.log(nome.concat(" ", sobrenome)); // Saída: "John Doe"
```



10. startsWith e endsWith - Verificando o Início e o Fim da String:

startsWith() verifica se uma string começa com a substring especificada, enquanto endsWith() verifica se uma string termina com a substring especificada.

```
const texto = "Hello World";  
console.log(texto.startsWith("Hello")); // Saída: true  
console.log(texto.endsWith("World")); // Saída: true
```



11. includes - Verificando a Presença de uma Substring:

`includes()` verifica se uma string contém a substring especificada e retorna `true` ou `false`.

```
const texto = "JavaScript é incrível!";  
console.log(texto.includes("incrível")); // Saída: true
```



12. padStart e padEnd - Preenchendo Strings:

padStart() e padEnd() preenchem a string com outra string até atingir o comprimento especificado.

```
const numero = "42";  
console.log(numero.padStart(5, "0")); // Saída: "00042"
```



13. slice - Extraíndo uma Porção da String:

`slice()` extrai uma seção de uma string e retorna uma nova string.

```
const frase = "A vida é bela";  
console.log(frase.slice(2, 6)); // Saída: "vida"
```



14. match - Encontrando Correspondências com Expressões Regulares:

match() encontrar correspondências entre uma string e uma expressão regular.

```
const texto = "A casa é azul e branca";  
console.log(texto.match(/azul|verde/)); // Saída: ["azul"]
```



15. charCodeAt e fromCharCode - Trabalhando com Códigos de Caracteres:

charCodeAt() retorna o valor Unicode do caractere na posição especificada, enquanto fromCharCode() retorna a string criada a partir dos valores Unicode especificados.

```
const caractere = "A";  
console.log(caractere.charCodeAt(0)); // Saída: 65  
console.log(String.fromCharCode(65)); // Saída: "A"
```



16. toString - Convertendo Outros Tipos de Dados em Strings:

toString() converte outros tipos de dados em strings.

```
const numero = 42;  
console.log(numero.toString()); // Saída: "42"
```



17. String Literals - Usando Template Literals para Construir Strings:

Template literals são strings delimitadas por crases que permitem interpolação de expressões e quebras de linha.

```
const nome = "João";  
const idade = 30;  
console.log(`Meu nome é ${nome} e tenho ${idade} anos.`);  
// Saída: "Meu nome é João e tenho 30 anos."
```



OBRIGADO POR LER ATÉ AQUI

Esse Ebook foi gerado por IA, e diagramado por humano.

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.



www.linkedin.com/in/mateus-fernandesjs