

STRINGS E CONVERSÕES DE TIPO

(AULA 16)

CURSO BÁSICO DE PROGRAMAÇÃO COM JAVASCRIPT

MAYARA MARQUES

mmrosatab@gmail.com

SUMÁRIO

- Strings
- Operações em strings
- Conversões de tipo

STRINGS

As strings em JavaScript **são sequências de caracteres** usadas para representar texto. **Elas são imutáveis**, o que significa que **uma vez criada, uma string não pode ser alterada**, embora novas strings possam ser criadas com base em operações nas existentes.

STRINGS

As strings podem ser vistas como arrays de caracteres. Pelo fato de serem **imutáveis** não é possível modificar o seu conteúdo diretamente por meio de atribuição de um novo valor a um respectivo índice.

```
const array = [1,2,3,4]
array[0] = 500
console.log(array) // (4) [500, 2, 3, 4]
```

```
let texto = "Olá"
texto[0] = "A"
console.log(texto) // Olá
```



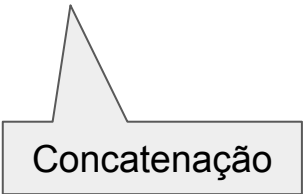
Isso não funciona!

STRINGS

Operações como concatenação **são custosas em loops!**

Como **cada operação que altera uma string cria uma nova cópia**, a concatenação excessiva dentro de loops pode ser ineficiente.

```
let resultado = ""  
  
for (let i = 0; i < 1000; i++) {  
  resultado += "a"  
}
```



Concatenação

STRINGS

Operações como concatenação **são custosas em loops!**

Como **cada operação que altera uma string cria uma nova cópia**, a concatenação excessiva dentro de loops pode ser ineficiente.

Opção mais eficiente

```
let resultado = ""  
  
for (let i = 0; i < 1000; i++) {  
  resultado += "a"  
}
```

Concatenação

```
let resultado = new Array(1001).join("a")
```

Cria um array de 1001 posições vazias (mas sem valores atribuídos).

O método **join()** concatena os elementos do array, inserindo "a" entre eles.

STRINGS

A seguir, veremos uma série de métodos que auxiliam no tratamento de strings no JavaScript.

CONCATENAR STRINGS

Duas ou mais strings podem ser concatenadas usando o operador `+` ou o método **`concat()`**.

CONCATENAR STRINGS

```
const str1 = "Hello"  
const str2 = "World"  
const result = str1 + " " + str2 // "Hello World"
```

INTERPOLAR STRINGS

A interpolação de strings no JavaScript é feita utilizando template literals (template strings), que foram introduzidos no ECMAScript 6 (ES6). Template literals permitem incorporar expressões dentro de strings, usando a sintaxe de crase (``) ao invés de aspas simples (') ou duplas (").

ECMAScript é a especificação padronizada da linguagem de programação que serve como base para o **JavaScript**, **JScript** (da Microsoft) e **ActionScript** (do Flash). Ele define a sintaxe, recursos e funcionalidades essenciais da linguagem, garantindo que diferentes implementações sigam um mesmo conjunto de regras.

INTERPOLAR STRINGS

```
const name = "Alice"  
const age = 30  
const greeting = `Hello, my name is ${name} and I am ${age} years old.`  
  
console.log(greeting) // "Hello, my name is Alice and I am 30 years old."
```

INTERPOLAR STRINGS

Template literals também permitem criar strings com múltiplas linhas sem a necessidade de caracteres de escape.

INTERPOLAR STRINGS

```
const multiline = `This is a string  
that spans across  
multiple lines.`  
console.log(multiline)  
// Output:  
// This is a string  
// that spans across  
// multiple lines.
```

ACESSO A STRINGS

O acesso a caracteres em uma string é feito com a notação de colchetes [] ou com o método **charAt()**

ACESSO A STRINGS

```
const str = "Hello"  
console.log(str[1]) // "e"  
console.log(str.charAt(1)) // "e"
```

COMPRIMENTO DE UMA STRING

A propriedade **length** é utilizada para obter o número de caracteres em uma string.

COMPRIMENTO DE UMA STRING

```
const str = "Hello"  
console.log(str.length) // 5
```

EXTRAIR SUBSTRING

Os métodos como **slice()**, **substring()** e **substr()** permitem extrair partes de uma string.

EXTRAIR SUBSTRING

```
const str = "Hello World"  
console.log(str.slice(0, 5)) // "Hello"  
console.log(str.substring(0, 5)) // "Hello"  
console.log(str.substr(0, 5)) // "Hello"
```

PROCURAR SUBSTRING

Os métodos como **indexOf()**, **lastIndexOf()**, **includes()**, **startsWith()** e **endsWith()** permitem procurar substrings.

PROCURAR SUBSTRING

```
const str = "Hello World"
console.log(str.indexOf("World")) // 6
console.log(str.includes("World")) // true
console.log(str.lastIndexOf("l")) // 9
console.log(str.lastIndexOf("o")) // 7
console.log(str.startsWith("Hello")) // true
console.log(str.startsWith("World")) // false
console.log(str.endsWith("World")) // true
console.log(str.endsWith("Hello")) // false
```

MODIFICAR SUBSTRING

Os métodos como **replace()**, **toUpperCase()**, **toLowerCase()**, **trim()**, **repeat()**, e **split()** permitem modificar strings.

MODIFICAR SUBSTRING

```
const str = "Hello World"
console.log(str.replace("World", "JavaScript")) // "Hello JavaScript"
console.log(str.toUpperCase()) // "HELLO WORLD"
console.log(str.toLowerCase()) // "hello world"
console.log(str.trim()) // "Hello World" (remove espaços no início e no final)
console.log(str.repeat(2)) // "Hello WorldHello World"
console.log(str.split(" ")) // ["Hello", "World"]
```

CONVERSÕES DE TIPO

JavaScript é uma linguagem de **tipagem dinâmica**, o que significa que as variáveis podem mudar de tipo. Existem várias maneiras de converter tipos em JavaScript, tanto de forma explícita quanto implícita.

CONVERSÕES DE TIPO

JavaScript é uma linguagem de **tipagem dinâmica**, o que significa que as variáveis podem mudar de tipo. Existem várias maneiras de converter tipos em JavaScript, tanto de forma explícita quanto implícita.

CONVERSÕES DE TIPO

Conversões Explícitas

- Conversão para String
 - `String()`: Converte um valor para string.

```
let num = 123
let str = String(num) // "123"
```

CONVERSÕES DE TIPO

Conversões Explícitas

- Conversão para String
 - `toString()`: Método usado em números e objetos para convertê-los para string.

```
let num = 123
let str = num.toString() // "123"
```

CONVERSÕES DE TIPO

Conversões Explícitas

- Conversão para Número
 - Number(): Converte um valor para número.

```
let str = "123"  
let num = Number(str) // 123
```

CONVERSÕES DE TIPO

Conversões Explícitas

- Conversão para Número
 - `parseInt()`: Converte uma string para um número inteiro.

```
let str = "123"  
let num = parseInt(str) // 123
```

CONVERSÕES DE TIPO

Conversões Explícitas

- Conversão para Número
 - `parseFloat()`: Converte uma string para um número de ponto flutuante.

```
let str = "123.45"  
let num = parseFloat(str) // 123.45
```

CONVERSÕES DE TIPO

Conversões Explícitas

- Conversão para Booleano
 - Boolean(): Converte um valor para booleano

```
let str = ""  
let bool = Boolean(str) // false
```

CONVERSÕES DE TIPO

Conversões Implícitas

- Concatenar Strings e Números

```
let num = 123  
let str = "Hello " + num // "Hello 123"
```


CONVERSÕES DE TIPO

Conversões Implícitas

- Operações Matemáticas

```
let str = "123"  
let sum = str * 2 // 246
```

CONVERSÕES DE TIPO

Conversões Implícitas

- Conversão para Booleano:
 - Em contextos booleanos, valores como 0, "", null, undefined, e NaN são convertidos para false, enquanto todos os outros valores são true.

```
if ("" ) {  
    // não executa, pois "" é false  
}
```



MÃO NA MASSA

MÃO NA MASSA



1. Você tem duas variáveis, `firstName` e `lastName`, que armazenam o primeiro e o último nome de uma pessoa. Crie uma variável `fullName` que combine ambos os nomes com um espaço entre eles e a imprima.
2. Dada a string `greeting` com o valor `"Hello, World!"`, acesse e imprima o primeiro e o último caractere da string.
3. Dada a string `text` com o valor `"JavaScript"`, imprima o número de caracteres desta string.
4. Dada a string `str` com o valor `"Hello, World!"`, extraia a palavra `"World"`. Dada a string `str` com o valor `"JavaScript"`, extraia a palavra `"Java"`.

MÃO NA MASSA



5. Dada a string `str` com o valor "Hello, World!"

- Encontre a posição da primeira ocorrência da letra "o".
- Na mesma string, encontre a posição da última ocorrência da letra "l".
- Verifique se a string "JavaScript" contém a palavra "Script".
- Verifique se a string "JavaScript" começa com a palavra "Java".
- Verifique se a string "JavaScript" termina com a palavra "Script".

MÃO NA MASSA



6. Dada a string **str** com o valor "Hello, World!"

- Substitua a palavra "World" por "JavaScript".
- Converta a string "hello" para letras maiúsculas.
- Converta a string "HELLO" para letras minúsculas.
- Remova os espaços em branco no início e no fim da string " Hello ".
- Repita a string "Ha!" três vezes.
- Divida a string "one, two, three" em um array de palavras.

MÃO NA MASSA



7.

- Converta o número 123 para uma string.
- Converta o valor booleano true para uma string.
- Converta a string "123" para um número.
- Converta a string "123.45" para um número com ponto flutuante.
- Converta a string "123abc" para um número inteiro. Observe o resultado.
- Converta a string vazia "" para um valor booleano.
- Converta o número 0 para um valor booleano.
- Converta a string "Hello" para um valor booleano.

REFERÊNCIAS

- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/String
- [Strings com JavaScript: o que são e como manipulá-las | Alura](#)
- [Template strings - JavaScript | MDN](#)
- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/join