

Strings e Arrays

Desenvolver Algoritmos

Msc. Lucas G. F. Alves
e-mail: lgfalves@senacrs.com.br

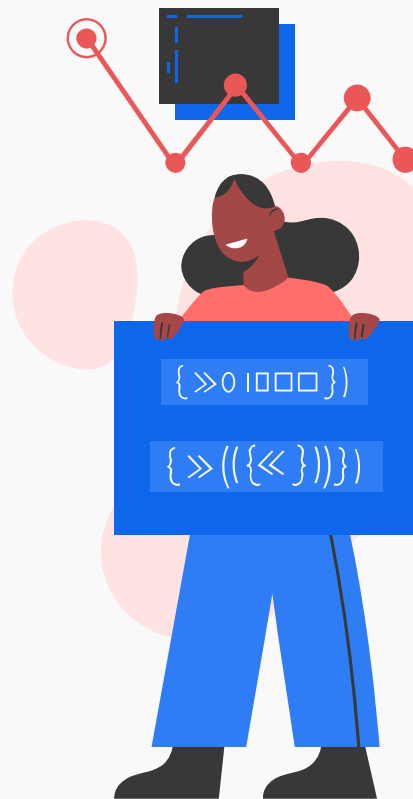


Planejamento de Aula

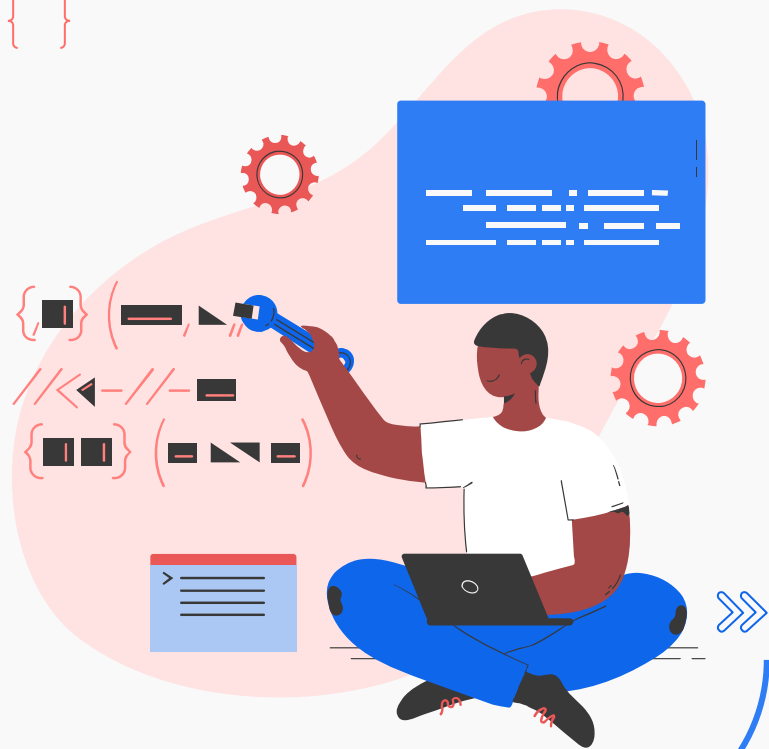
Revisão Strings

Revisão Arrays

Exercícios



Revisão Strings





Strings

[]

O que são Strings?

Como vimos anteriormente, Strings são os tipos referentes à **textos**.

Temos 3 maneiras de escrever uma string:

- Aspas Duplas: "Olá Mundo"
- Aspas Simples: 'Olá Mundo'
- Crase (Template String ou Template Literals):
`Olá Mundo`

{ }

{ ((({ >> }))) << }

- []



Strings

[]

Concatenação.

Também podemos juntar várias strings para formar uma nova.

Chamamos esse processo de **concatenação** e utilizamos o sinal de + para fazê-lo.

```
const nome = "Lucas"
const idade = 32
{ } const frase = "Meu nome é " + nome + " e tenho " + idade + " anos"
```

É necessário colocar o
espaço para separar palavras

{((({>>}))<<}

-{ }



Strings

[]

Template.

Não há diferença entre usar aspas simples ou duplas!

A única diferente é a Template String, pois ela nos permite colocar variáveis javascript no meio da string.

```
const nome = "Lucas"
```

```
const idade = 32
```

{ }

```
const frase = `Meu nome é ${nome} e tenho ${idade} anos`
```

```
// Meu nome é Lucas e tenho 32 anos
```

```
{((({>>}))<<)}
```

Para sinalizar que é uma
variável, usamos \$ e {}

[]



Exercícios

[]

1. Crie um programa que peça ao usuário para inserir o seu nome e sua cor favorita e imprima a mensagem:

"A cor favorita de FULANO é COR: "

2. Realize o exercícios duas vezes. Utilizando template strings e concatenação.

{ }

{((({>>}))<<}

-[]



Protótipo de Strings

[]

O javascript nos fornece algumas informações (**propriedades**) e ações (**métodos**) que podemos realizar sobre uma string.

Será falado algumas delas na aula, mas se quiser conhecer mais, pode visitar o site do W3Schools que é referência em desenvolvimento web.

https://www.w3schools.com/Jsref/jsref_obj_string.asp

{ }

{((({>>}))<<}

-[]



Protótipo de Strings

[]

Propriedade **length**

A propriedade **length** nos diz qual é o tamanho de uma string, incluindo espaços.

```
const nome = "Prof Lucas Alves"
```

```
console.log(nome.length) // 16
```

{ }

{ ((({ >> }))) << }

- []



Protótipo de Strings

[]

Método **toLowerCase()**

O método **toLowerCase()** transforma todas as letras da sua string em minúsculas

```
const frase = "OieEeEee!"  
const fraseMaiuscula = frase.toLowerCase()  
// fraseMaiuscula = oieeeeeee!
```

{ }

{((({>> }))) << }

- []



Protótipo de Strings

[]

Método **toUpperCase()**

O método **toUpperCase()** transforma todas as letras da sua string em maiúsculas.

```
const frase = "OieEeEee!"  
const fraseMinuscula = frase.toLowerCase()  
// fraseMinuscula = oieeeeeee!
```

{ }

{((({>>}))<<}

-[]



Protótipo de Strings

[]

Método **trim()**

O método **trim()** retira os espaços que existem antes e depois da sua string.

Útil em formulários como por exemplo de login!

{ }

```
const email = " LGFALVES@senacrs.com.br "  
console.log(email.trim())  
// "LGFALVES@senacrs.com.br"
```

{ ((({ >> }))) << }

- []



Protótipo de Strings

[]

Método **includes(caracteres)**

O método **includes(caracteres)** determina se um conjunto de caracteres pode ser encontrado dentro de outra string, retornando **true** ou **false**.

```
const frase = "Hoje comi cenoura"  
frase.includes("cenoura") // true  
frase.includes("batata") // false
```

{ }

{((({>>}))<<}

-[]



Protótipo de Strings

[]

Método **replaceAll(chars1, chars2)**

O método **replaceAll(chars1, chars2)** troca todas as ocorrências de um conjunto de caracteres (chars1) por alguma outra coisa (chars2)

```
const frase = "Hoje comi cenoura, adoro cenoura"  
const novaFrase = frase.replaceAll("cenoura", "batata")  
// novaFrase = Hoje comi batata, adoro batata
```

{ }

{ ((({ >> }))) << }

- []



Exercícios

[]

Peça para o usuário escrever uma frase e imprima no console a frase alterada, com:

- Todas as letras maiúsculas;
- Na língua do i (substituindo a vogal "o" por "i");
- O tamanho da frase.

{ }

{((({>>}))<<}

-[]



Resumo

[[]]

Existem 3 Maneiras de escrever Strings:

Aspas duplas, Aspas simples e Crase;

Formando novas Strings:

- Concatenação e Template Strings;

Protótipo de Strings:

length, toLowerCase(), toUpperCase(), trim(), includes(caracteres) e replaceAll(chars1, chars2).

 $\{ \quad \}$
$$(\{((\{ \gg \})) \ll \}$$
$$\left[\begin{array}{c} \text{---} \end{array} \right]$$



Arrays



[]

O que são **arrays**?

Arrays nada mais são do que **listas de elementos**.

Será armazenado mais de uma coisa dentro de uma variável.

Ex: lista de compras, lista de alunos, lista de números da loteria, lista telefônica...

No javascript, usamos **colchetes []** para agrupar os itens de uma lista:

{ }

```
const listaDeCompras = ["batata", "alface", "queijo"]
```

```
const listaDeNumerosMega = [2, 13, 26, 35, 41, 60]
```

```
({{{({>>}}))<<}
```

- []



Arrays

[]

O que são arrays?

Podemos colocar elementos de **qualquer tipo** que vimos até agora dentro de um array!

Números, strings e booleanos.

Também podemos ter elementos de tipos diferentes dentro de um mesmo array

{ }

```
const meuArray = ["banana", 15, true]
```

{ ((({ >> })) <<) }

- []



Arrays

[]

O que são arrays?

VALORES



ÍNDICES



0

1

2

3

4

5

{ }

{((({>>}))<<}

[]



Arrays

[]

Acessando um elemento

Em um array, acessamos os elementos através da **posição**(índice) deles na lista!

Funciona como se fosse uma lista numerada:

Lista de Compras

1. Abacate
2. Banana
3. Tomate



Qual é o item na posição 2?

Resposta: Banana

{ }

{((({>>}))<<}

-[]



Arrays

[]

Acessando um elemento

Mas no caso dos arrays, a numeração não começa no 1, **mas sim no 0!**

Para acessar um item, colocamos a sua posição (**índice**) entre **colchetes** após o nome do array

Lista de Compras

- 0. Abacate
- 1. Banana
- 2. Tomate

{ }

```
const listaDeCompras = ["Abacate", "Banana", "Tomate"]  
const segundoItem = listaDeCompras[2] // "Tomate"
```

{ ((({ >> })) <<) }

{ }



Exercícios

[]

1. Crie um array com pelo menos 5 raças de cachorro.
2. Peça para o usuário inserir um número de 0 a 4 e Imprima no console a raça correspondente à posição escolhida pelo usuário.

{ }

({{{({>>}}))<<}

-[]



Resumo

[]

Arrays são listas que podem conter elementos de qualquer tipo (strings, números, etc).

Para acessar um elemento de um array, utilizamos a sua posição (ou **índice**).

{ }

({{{({>>}}))<<}

- []



Protótipo de Arrays

[]

O javascript nos fornece algumas informações (**propriedades**) e ações (**métodos**) que podemos realizar sobre uma lista (array)

Também será falado algumas delas na aula, mas se quiser conhecer mais, pode visitar o site do W3Schools que é referência em desenvolvimento web.

https://www.w3schools.com/Jsref/jsref_obj_array.asp

{ }

({ (({ >> })) } <<)

[]



Protótipo de Arrays

[]

Método **includes(elemento)**

O método **includes(elemento)** determina se um array contém um determinado elemento, retornando **true** ou **false**

```
const seriesBoas = ["Vikings", "The Big Bang Theory"]
```

```
seriesBoas.includes("Vikings") // true
```

```
seriesBoas.includes("Game of Thrones") // false
```

{ }

{ { { { { } } } } } }

{ }



Protótipo de Arrays

[]

Método **push(elemento)**

O método **push(elemento)** adiciona um ou mais elementos ao final de um array

```
const numeros = [1, 2, 3]
```

```
numeros.push(4)  
console.log(numeros) // [1, 2, 3, 4]
```

{ }

```
numeros.push(5, 6, 7)  
console.log(numeros) // [1, 2, 3, 4, 5, 6, 7]
```

```
({{{({>>}}))<<}
```

- []



Protótipo de Arrays

[]

Método **pop()**

O método **pop()** remove o último elemento de um array

```
const meusPeixes = ["palhaço", "mandarim", "esturjão"]
```

```
meusPeixes.pop()
```

```
console.log(meusPeixes) // ["palhaço", "mandarim"]
```

{ }

{ { { { { } } } } } }

[]



Protótipo de Arrays

[]

Método **splice(i, n)**

O método **splice(i, n)** remove **n** (n representa a quantidade) elementos à partir da posição **i** do array.

```
const letras = ["A", "B", "C", "D", "E", "F", "G", "H"]  
// índices (i)  0  1  2  3  4  5  6  7
```

```
letras.splice(2, 1)  
//  letras = ["A", "B", "D", "E", "F", "G", "H"]  
// índices (i)  0  1  2  3  4  5  6
```

{ }

```
{((( {>>}) )<<)} letras.splice(3, 2) // letras = ["A", "B", "D", "G", "H"]
```

- []



Exercícios

[]

Para este exercício, comece criando um array com os valores: 1, 2, 3, 4, 5 e 6.

1. Determine o tamanho do array.

2. Adicione o número 7.

3. Remova os números 4 e 5.

{ }

4. Determine o novo tamanho do array.

{((({>>}))<<}

- []



Resumo

[]

Arrays são listas de elementos que podem ter qualquer tipo.

Agrupamos esses itens usando **colchetes []**.

Acessamos um item pelo **índice** (ou seja, sua posição na lista)

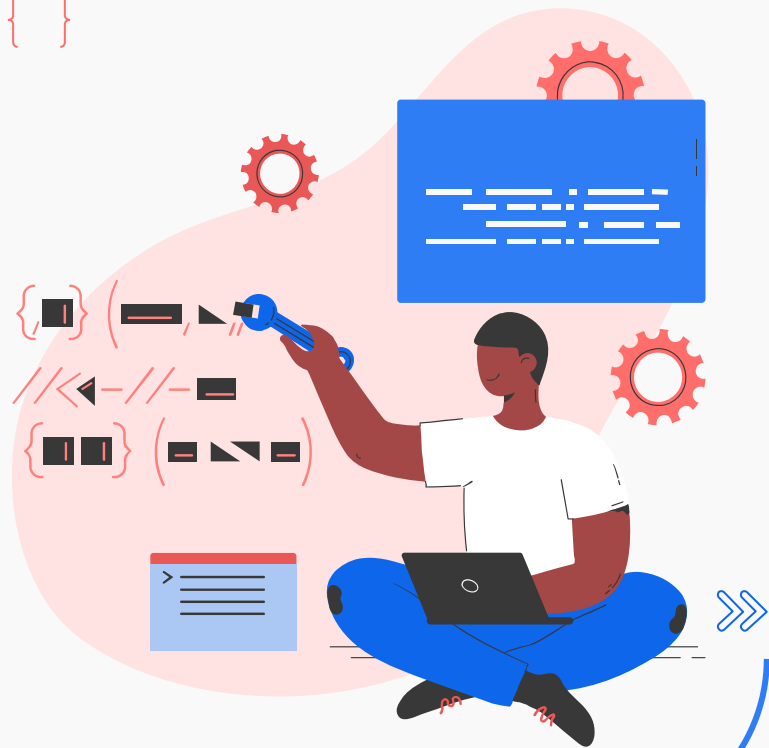
Protótipo de Arrays:

length, includes(), push(elemento), pop() e splice(i, n);

{ }

{ ((({ >> }))) << }

- []





Exercícios de Fixação

[]

Exercícios de interpretação de código

1. Indique todas as mensagens impressas no console, SEM EXECUTAR o programa.

```
let array
console.log('a. ', array)

array = null
console.log('b. ', array)

array = [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
console.log('c. ', array.length)

let i = 0
console.log('d. ', array[i])

array[i+1] = 19
console.log('e. ', array)

const valor = array[i+6]
console.log('f. ', valor)
```

{ }

{((({>>}))<<)}

-[]



Exercícios de Fixação

[]

Exercícios de interpretação de código

2. Leia o código abaixo com atenção

```
const frase = prompt("Digite uma frase")  
  
console.log(frase.toUpperCase().replaceAll("A", "I"), frase.length)
```

{ } Qual será o valor impresso no console se a entrada do usuário for: "Subi num
ônibus em Marrocos"?

{((({>>}))<<}

- []



Exercícios de Fixação

[]

Exercícios de escrita de código

3. Faça um programa que pergunte ao usuário seu nome e seu e-mail. Em seguida, Imprima no console a seguinte mensagem:

O e-mail ``emailDoUsuario`` foi cadastrado com sucesso. Seja bem-vinda(o), ``nomeDoUsuario``!

{ }

{((({>> }))) << }

- []



Exercícios de Fixação

[]

Exercícios de escrita de código

4. Faça um programa que contenha um array com 5 das suas comidas preferidas, armazenado em uma variável. Em seguida, siga os passos:

a) Imprima no console o array completo

b) Imprima no console a mensagem "Essas são as minhas comidas preferidas: ", seguida por cada uma das comidas, **uma embaixo da outra**.

{ } c) Aqui vai um desafio: pergunte ao usuário uma comida preferida. Troque a segunda comida da sua lista pela inserida pelo usuário. Imprima no console a nova lista

{ ((({ >> }))) << }

- []



Exercícios de Fixação

[]

Exercícios de escrita de código

5. Faça um programa, seguindo os passos:

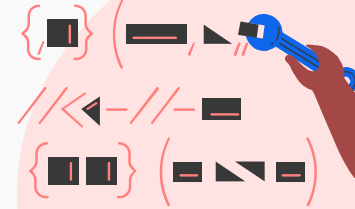
- a) Crie um array vazio e guarde-o em uma variável, chamada `'listaDeTarefas'`.
- b) Pergunte ao usuário 3 tarefas que ele precise realizar no dia e armazene-as, uma por uma, no array.
- c) Imprima o array no console.
- d) Peça ao usuário que digite o **índice** da tarefa que ele já realizou: 0, 1 ou 2.
- e) Remova da lista o item de índice que o usuário escolheu.
- f) Imprima o array no console.

{ }

```
({{{({>>}}))<<}
```

- []

Desafios





Desafios

[]

1. Receba uma frase e retorne um array onde cada elemento é uma das palavras da frase, ignorando os espaços
2. Dado o array `["Banana", "Morango", "Abacaxi", "Laranja", "Ameixa"]`, faça um programa que acha o índice da palavra Abacaxi e imprime tanto o índice quanto o tamanho do array

{ }

{((({>>}))<<}

-[]

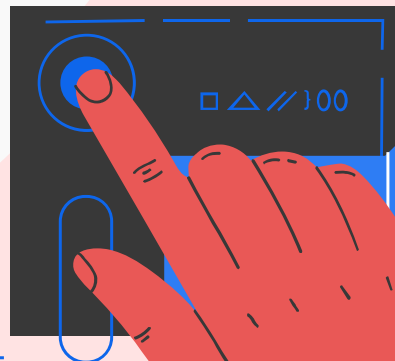
Obrigado!



E-mail: lgfalves@senacrs.com.br



{{({{>>}})}<<}

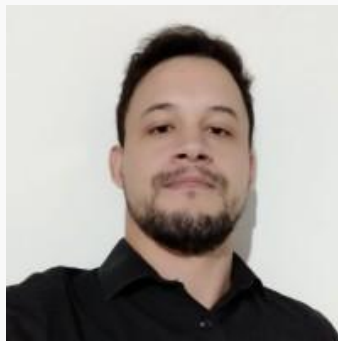


(({{>>0i□□□}}))

```
((: 00 - =>>})  
{ (<1 00 1 000 >>)}  
((: 0)>"< )  
<01 001} +100 0}>  
((: 0)>"< )  
{ (<1 00 1 000 >>)}
```



Professor



Lucas G. F. Alves

