

SEJAM BEM-VINDOS!



Jornada DEV AM. CE. RR

Formação gratuita
em Desenvolvimento
Front-End SENAI



Começamos em breve!

A aula vai iniciar em **5 minutos**.

Relaxe e aproveite a música enquanto aguardamos todos chegarem!



Lógica com JavaScript #06

Missão de hoje

- Métodos de matriz
 - Includes
 - Join
- Iteração em matrizes
 - forEach
- Soma com matrizes

Matrizes | Arrays

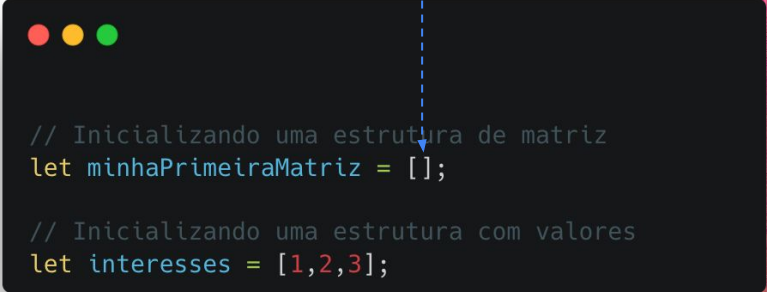
Você pode encontrar diferentes termos, como matrizes, vetores, arrays e listas. No contexto do JavaScript, todos esses nomes geralmente se referem ao mesmo conceito fundamental: o tipo de dado Array

- **Matrizes (*arrays*)**: são sequências de dados (itens), cada um destes itens da sequência está acessível por um índice numérico de 0 a N (0, 1, 2 ... N).
- **Índice (*index*)**: Chamamos os índices de um *array* de *indexes*. Quando inserimos um item em um *array*, este pode ser acessado através de seu *index* referente à sua posição ocupada no vetor.

Criando sua primeira matriz

Iniciaremos a criação dos nossos Arrays usando a sintaxe literal de **colchetes** (`[]`).

Por se tratar de um Objeto, existe também a inicialização via construtor (`new Array()`), mas o formato com colchetes é o mais **simples** iremos utilizar aqui.



```
// Inicializando uma estrutura de matriz
let minhaPrimeiraMatriz = [];

// Inicializando uma estrutura com valores
let interesses = [1,2,3];
```


Como Acessar um Item?

- Arrays são **zero-indexed**: a contagem começa em **0**.
- Usamos o **índice** dentro de **colchetes** (`[]`) para acessar um valor.



```
let lista = ["A", "B", "C"];    // Array com 3 itens.  
  
lista[0];    // Retorna o primeiro item: "A".  
lista[2];    // Retorna o último item: "C".
```

Adicionando item a lista

Arrays herdam de `Array.prototype`, que fornece a eles um punhado de métodos convenientes para manipular arrays.

Por exemplo, `.push()` é um método que permite adicionar um item no final da lista.



```
let lista = [];  
  
lista.push("Seu item")
```


Método `.includes()`

- Verifica se um **elemento** está **presente** no array.
- Retorna **true** ou **false**.

```
let notas = [7, 8, 9]
```

```
console.log(notas.includes(8)) // true
```

```
console.log(notas.includes(5)) // false
```

Exemplos



```
let listaNumerica = [1,2,3]

console.log(listaNumerica.includes(2)) // Saída: true

let pets = ["gato", "cachorro", "morcego"]

console.log(pets.includes("morcego")) // Saída: true

console.log(pets.includes("rro")) // Saída: false
```

Método `.join()`

- Esse método junta todos os elementos do array em uma string.
- Podemos definir o separador.



```
let frutas = ["maçã", "banana", "laranja"]  
  
console.log(frutas.join(", ")) // maçã, banana, laranja  
console.log(frutas.join(" & ")) // maçã & banana & laranja
```

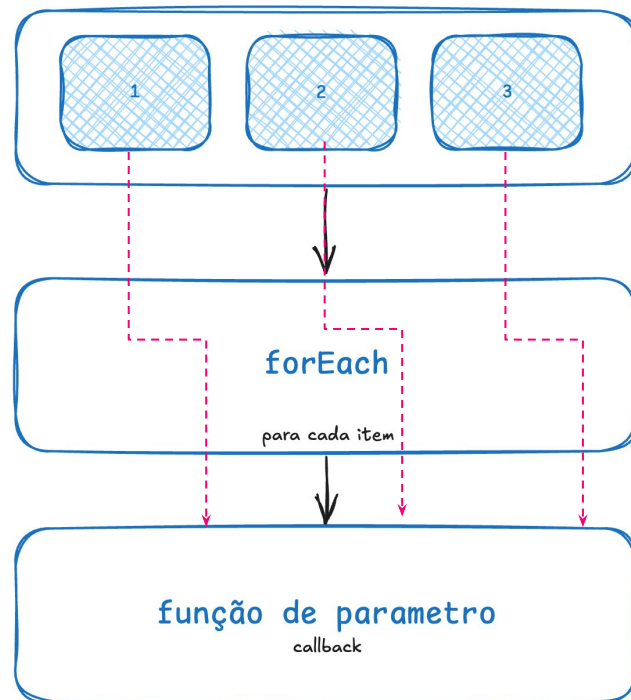
O que é iteração ?

- Processo de **percorrer todos os itens** de uma matriz.
- Vários métodos ajudam nisso.

Método `.forEach`

Executa uma **função (callback)** para **cada item** do array.

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



Array

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

```
numeros.forEach(function(n) {  
  console.log(n)  
})
```

Método forEach
ex.: forEach()

Valor do item

função de callback

Chamamos de callback uma função que é passada como parâmetro para outra função.

Esse termo existe porque essa função será "chamada de volta" (executada) dentro da função que a recebeu, geralmente para processar dados ou responder a um evento.

Vamos praticar

Acesse esse link

Caso você esteja revisando este material após a aula, o repositório pode conter **arquivos modificados ou adições de código**, pois ele foi atualizado e desenvolvido em tempo real durante a apresentação.

Vamos praticar

- **Métodos**
 - Converter CSV
 - Unir informações com "&" comercial
-
- Interação de matrizes
- Iterar temperaturas
- Dobrar valores de uma lista
- Calcular média do aluno
- Soma das notas iguais ou acima de 6

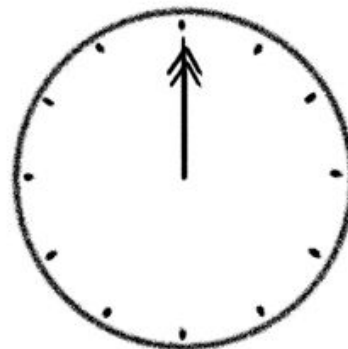
Intervalo!

Finalizamos o nosso primeiro período de hoje.

Nos vemos em 15min.

Início: 20:00

Retorno: 20:15



<LAB365>

SENAI