

**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 #05

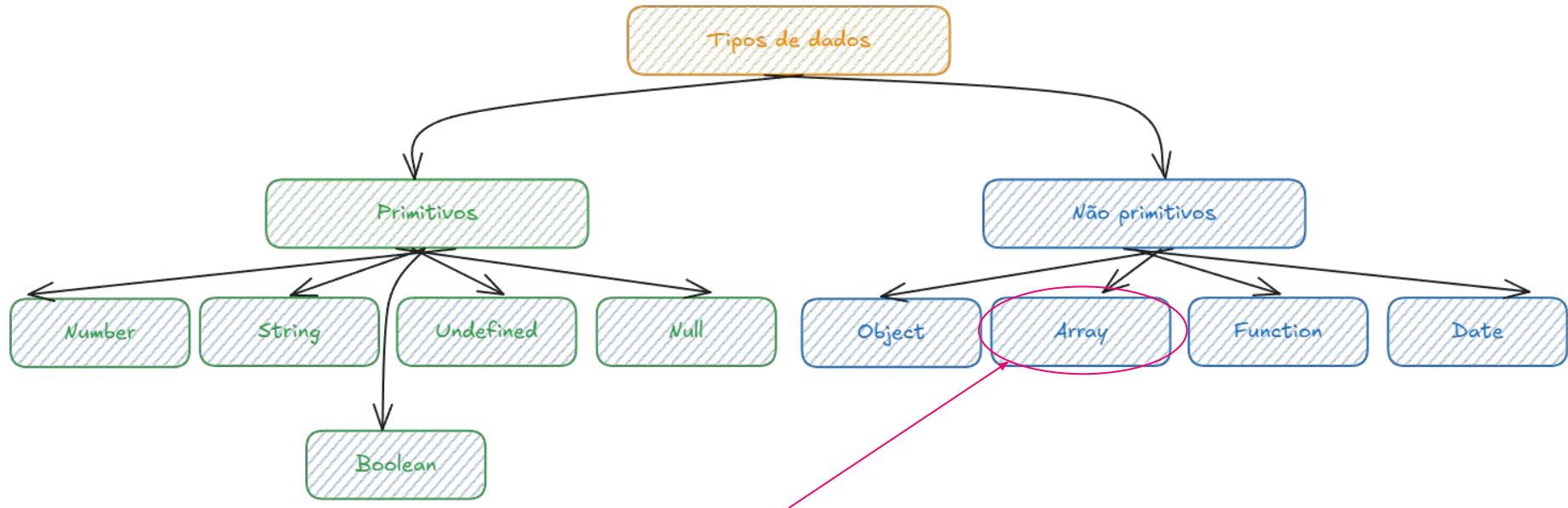
# Missão de hoje

- Introdução às Matrizes
- Acessando itens da matriz
- Comprimento da matriz
- Adicionando itens à matriz



## Tipos de dados

# TIPOS DE DADOS | Estruturas de dados



# Array

O objeto **Array** do JavaScript é um objeto global usado na construção de 'arrays': objetos de alto nível semelhantes a listas.

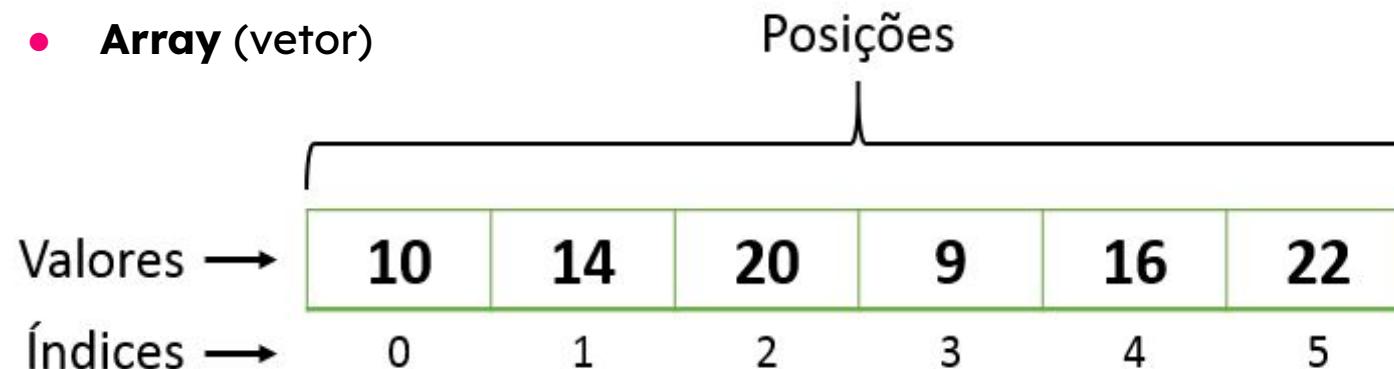
Arrays são objetos semelhantes a listas que vêm com uma série de métodos embutidos para realizar operações de travessia e mutação. Nem o tamanho de um array JavaScript nem os tipos de elementos são fixos.

# 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.

- **Array (vetor)**

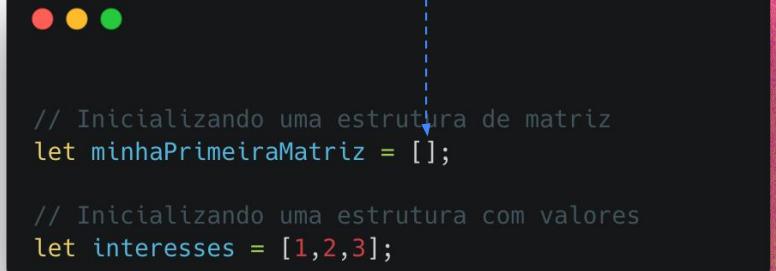




# Criando sua primeira matriz

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];
```

# Exemplos



```
let listaVazia = [];

let diasDaSemana = [1,2,3];

let lista = ["este é o primeiro elemento", "este é o segundo elemento"];

let listaComDiversosTipos = [1, "dois", false, undefined]
```

# 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".
```

# Comprimento da Matriz (Array)

## Comprimento (`.length`)

- Todo Array possui uma propriedade chamada `.length`.
- Essa propriedade **sempre** indica o número **total de elementos** no Array.

```
let itens = ["Maçã", "Pera", "Uva"];      // Array com 3 elementos.  
itens.length;   // Retorna o comprimento: 3.
```

# Como podemos acessar o último elemento ?



Para ter acesso ao último elemento, podemos executar a seguinte operação: `Array.length - 1`.

# Comprimento da Matriz (Array)

## Comprimento (`.length`)

- Todo Array possui uma propriedade chamada `.length`.
- Essa propriedade **sempre** indica o número **total de elementos** no Array.



```
let itens = ["Maçã", "Pera", "Uva"];      // Array com 3 elementos.  
itens.length;    // Retorna o comprimento: 3.
```

# 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")
```

# 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.

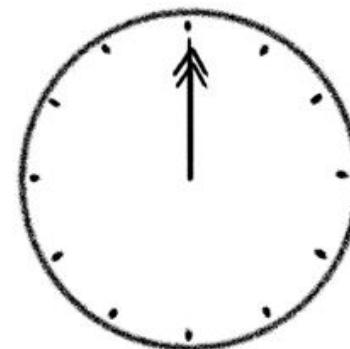
# Intervalo!

Finalizamos o nosso primeiro período de hoje.

Nos vemos em 15min.

**Início:** 20:00

**Retorno:** 20:15



# Dúvidas ?

<LAB365>

