

# 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 #08

# Missão de hoje

- Resolvendo atividades
- Strings avançadas
- Lógica booleana avançada
- Condicional `If` avançada
- Ponto e vírgula

# Vamos praticar

## Mapas

- <https://app.devstart.tech/learn/logica-de-programacao/mapa-de-matriz/ex1-mapa-de-matriz-ii>
- <https://app.devstart.tech/learn/logica-de-programacao/mapa-de-matriz/ex2-mapa-de-matriz-ii>
- <https://app.devstart.tech/learn/logica-de-programacao/mapa-de-matriz/ex3-mapa-de-matriz-ii>

## Redução

- <https://app.devstart.tech/learn/logica-de-programacao/reducao-de-matriz/ex1-reducao-por-soma>
- <https://app.devstart.tech/learn/logica-de-programacao/reducao-de-matriz/ex1-reducao-por-multiplicacao>
- <https://app.devstart.tech/learn/logica-de-programacao/reducao-de-matriz/ex1-projeto-14-aplicativo-sala-de-aula>
- <https://app.devstart.tech/learn/logica-de-programacao/reducao-de-matriz/ex3-reducao-por-multiplicacao>

# Strings avançadas: Múltiplas linhas

Existem três formas de declarar strings: **aspas duplas**, **aspas simples** e **crases**.

- Apenas template strings (entre crases) permitem quebrar linhas naturalmente.
- Use aspas duplas ou simples para textos curtos e crases para mensagens mais complexas.



```
const mensagem = `Bem-vindo ao  
Nosso site!`;  
  
console.log(mensagem);
```

# Escapes & Interpolação

- Para inserir aspas simples ou duplas dentro de uma string, use o escape \ (barra invertida).
- As template strings permitem misturar aspas sem escapes e também interpolar valores.
- Na interpolação, é possível usar variáveis, expressões e chamadas de função.


```
function obterMensagem(idade) {  
  return `Você tem atualmente ${idade}  
  anos.  
  No ano que vem, você terá ${idade + 1}  
  anos.`;  
}  
  
function darBoasVindas(nome) {  
  return `Bem-vindo,  
  ${nome.toLowerCase()}`;  
}
```

# Lógica booleana: Negação

- O operador `!(NOT)` inverte o valor lógico:
  - `!true` → `false`;
  - `!false` → `true`.
- Use parênteses para deixar clara a ordem de avaliação:
  - `!(idade >= 18)`.
- Dupla negação (`!! valor`) converte qualquer expressão em verdadeiro ou falso.



# Negação



```
const idade = 20;  
const maiorDeIdade = idade ≥ 18;  
console.log(!maiorDeIdade); // inverte o valor  
console.log(!!idade); // converte para boolean
```

# Lógica booleana: && e ||

- O operador && (E) retorna true somente quando **todas as condições são verdadeiras**.
- O operador || (OU) retorna true quando **pelo menos uma condição é verdadeira**.
- Ambos usam avaliação de curto-circuito: se o resultado já está determinado, a segunda expressão não é avaliada.

# Exemplo de implementação

```
// Exemplo com &&
let idade = 20;
if (idade >= 18 && idade <= 25) {
  console.log("Acesso permitido");
} else {
  console.log("Acesso negado");
}

// Exemplo com ||
let descontoIdade = false;
if (idade < 18 || idade > 60) {
  descontoIdade = true;
}
```

<LAB365>

