

# JavaScript

**Tecnologia em Análise e Desenvolvimento de Sistemas**

Senac - Portão - Curitiba/PR

Prof. Msc. Guilherme Matos Barbosa

## Operadores lógicos

- São valores booleanos que quando verificados resultam em verdadeiro ou falso, ou seja, outro valor booleano.
- Existem três tipos principais no JavaScript:

## •Operadores lógicos

### Operador AND ( && )

Retorna verdadeiro se ambos os operandos forem verdadeiros.

Exemplo:

```
( 4 > 2 ) && ( 5 < 10 ) // retorna verdadeiro
```

```
( 4 > 2 ) && ( 5 > 10 ) // retorna falso
```

## •Operadores lógicos

### Operador OR ( || )

Retorna verdadeiro se pelo menos um dos operandos for verdadeiro. Exemplo:

```
( 4 > 2 ) || ( 5 < 10 ) // retorna verdadeiro
```

```
( 4 < 2 ) || ( 5 > 10 ) // retorna falso
```



# JavaScript

## Operadores lógicos

### Operando NOT ( ! )

Inverte o valor booleano do operando. Exemplo:

```
!(4 > 2) // retorna falso
```

```
!(4 < 2) // retorna verdadeiro
```



# JavaScript

## Operador condicional ternário

É uma forma compacta de escrever uma estrutura condicional em uma única linha de código.

A sintaxe de um operador ternário é a seguinte:

`Condicao ? valor_se_verdadeiro : valor_se_falso`

Se condição for verdadeira retorna **valor\_se\_verdadeiro**,  
senão **valor\_se\_falso**.



# JavaScript

## Operador condicional ternário

EXEMPLO:

```
let idade = 16;
```

```
const podeDirigir = idade >= 18 ? "pode dirigir" : "não pode  
dirigir "
```

```
console.log(podeDirigir)
```

# JavaScript

## Operador condicional ternário

### EXEMPLO:

- Declaramos uma variável *idade* com o valor 16 e em seguida fazemos uma condicional que verifica se a variável é maior ou igual a 18.
- Nesse caso a condição é verdadeira? Não! Então o valor retornado é o valor que vem depois dos : (dois pontos), o *não pode dirigir*.
- Se a condição fosse verdadeira, o valor retornado seria o primeiro, o *pode dirigir*.





# JavaScript

## Qual a precedência dos operadores?

Quando falamos em precedência, estamos nos referindo a qual ordem que as operações serão realizadas quando existirem múltiplos operadores na mesma expressão.

# JavaScript

Lista dos operadores da mais alta precedência para a mais baixa:

- Expressões de agrupamento: ( )
- Negação ( ! ), Incremento ( ++ ) e Decremento ( -- )
- Multiplicação ( \* ), Divisão ( / ) e Módulo (%)
- Adição ( + ) e Subtração ( - )
- Relacionais ( < <= > >= )
- Igualdade ( == != === !== )
- Operador lógico AND ( && )
- Operador lógico OR ( || )
- Operador ternário ( ? : )
- Atribuição ( = )