

## Aula 13

# Operadores Relacionais e Lógicos

## Slide 1/4: Revisão da Aula 3

### Revisão da Aula 3: Cálculos e Operadores

- **Operadores Aritméticos:**

- Vimos como usar `+`, `-`, `*`, `/` para as operações básicas.
- Exploramos operadores especiais como `%` (módulo/resto) e `**` (potência).

- **Ordem de Precedência:**

- Relembramos que `*` e `/` são resolvidos antes de `+` e `-`.
- Usamos parênteses `()` para controlar a ordem dos cálculos.

- **Atalhos e Ferramentas:**

- Aprendemos a usar operadores de atribuição (`+=`, `-=`) e incremento/decremento (`++`, `--`) para escrever código mais rápido.
- Conhecemos a "caixa de ferramentas" `Math` para funções como `Math.sqrt()` (raiz) e `Math.pow()` (potência).

## Slide 2/4: O Início da Tomada de Decisão

### Aula 4: Operadores Relacionais e Lógicos

- **Objetivo de Hoje:**
  - Sair do mundo dos cálculos e entrar no mundo das **comparações**.
  - Aprender a criar expressões que resultam em **verdadeiro** ( `true` ) ou **falso** ( `false` ).
- **O que são Operadores Relacionais?**
  - São símbolos que usamos para comparar dois valores.
  - Maior que: `>`
  - Menor que: `<`
  - Maior ou igual a: `>=`
  - Menor ou igual a: `<=`

## Slide 3/4: A Importância da Igualdade Estrita

### Comparando Igualdade e Diferença

- **Igualdade Solta ( `==` ):**
  - Compara **apenas o valor**, tentando converter os tipos.
  - Exemplo: `5 == "5"` resulta em `true`. **(Pode causar bugs!)**
- **Igualdade Estrita ( `===` ):**
  - Compara o **valor E o tipo**. Não faz conversão.
  - Exemplo: `5 === "5"` resulta em `false`.
  - **BOA PRÁTICA:** Use SEMPRE a comparação estrita ( `===` e `!==` )!
- **Diferença:**
  - Diferença Solta: `!=`
  - Diferença Estrita: `!==` (Recomendado)

## Slide 4/4: Combinando Ideias com Lógica


### Operadores Lógicos: Conectando Comparações

- **&& (E / AND):**
  - Retorna **true** **somente se AMBAS** as condições forem verdadeiras.
  - Ex: "Para dirigir, preciso ter CNH **E** ser maior de 18 anos."
- **|| (OU / OR):**
  - Retorna **true** se **PELO MENOS UMA** das condições for verdadeira.
  - Ex: "Para ter desconto, preciso ser estudante **OU** ter o cupom."
- **! (NÃO / NOT):**
  - **Inverte** o valor booleano. **!true** vira **false**, e **!false** vira **true**.
  - Ex: "Se **NÃO** estiver chovendo, vou ao parque."

Prática

<> index.html X

JS index.js

<> index.html >  html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <h1>Operadores Relacionais e Lógicos</h1>
10     <script src="index.js"></script>
11 </body>
12 </html>
```

```
11 // =====
12 // 1. OPERADORES RELACIONAIS ↔
13 // =====
14 // Comparam dois valores e retornam `true` ou `false`.
15
16 let num1 = 10;
17 let num2 = 15;
18 let num3 = 10;
19
20 console.log("10 > 15 é:", num1 > num2); // Maior que -> false
21 console.log("10 < 15 é:", num1 < num2); // Menor que -> true
22 console.log("10 >= 10 é:", num1 >= num3); // Maior ou igual a -> true
23 console.log("15 <= 10 é:", num2 <= num3); // Menor ou igual a -> false
24
```



```
25 // =====
26 // 2. IGUALDADE (== vs ===) e DIFERENÇA (!= vs !==) ⚠
27 // =====
28 // Este é um dos conceitos mais importantes em JavaScript!
29
30 let a = 20;
31 let b = "20"; // b é uma string!
32
33 // 2.1. Igualdade Solta (==): Tenta converter os tipos antes de comparar. EVITE USAR!
34 console.log("20 == '20' (solto):", a == b); // true, porque o JS converte a string "20" para um número.
35
36 // 2.2. Igualdade Estrita (===): Compara o valor E o tipo. NÃO faz conversão. USE SEMPRE!
37 console.log("20 === '20' (estrito):", a === b); // false, porque um é `number` e o outro é `string`.
38
39 // 2.3. Diferença Solta (!=): O oposto de `==`. EVITE USAR!
40 console.log("20 != '20' (solto):", a != b); // false
41
42 // 2.4. Diferença Estrita (!==): O oposto de `===`. USE SEMPRE!
43 console.log("20 !== '20' (estrito):", a !== b); // true
44
45 // REGRA DE OURO: Para evitar bugs e comportamentos inesperados, SEMPRE use `===` e `!==`.
46
```

```
48 // 3. OPERADORES LÓGICOS 🧠
49 // =====
50 // Usados para combinar múltiplas expressões booleanas.
51
52 // 3.1. && (E / AND): Retorna `true` apenas se AMBOS os lados forem `true`.
53 let idadeUsuario = 25;
54 let temCNH = true;
55 let podeDirigir = (idadeUsuario >= 18) && (temCNH === true);
56 //           (true)           &&           (true)
57 console.log("Pode dirigir?", podeDirigir); // Resultado: true
58
59 let media = 6.0;
60 let frequencia = 0.8;
61 let aprovado = (media >= 7.0) && (frequencia >= 0.75);
62 //           (false)           &&           (true)
63 console.log("Aluno aprovado?", aprovado); // Resultado: false
64
65 // 3.2. || (OU / OR): Retorna `true` se PELO MENOS UM dos lados for `true`.
66 let temIngresso = false;
67 let ehConvidadoVIP = true;
68 let podeEntrarNaFesta = temIngresso || ehConvidadoVIP;
69 //           (false)           ||           (true)
70 console.log("Pode entrar na festa?", podeEntrarNaFesta); // Resultado: true
71
72 let comprouProdutoA = true;
73 let comprouProdutoB = false;
74 let ganhouBrinde = comprouProdutoA || comprouProdutoB;
75 //           (true)           ||           (false)
76 console.log("Ganha brinde?", ganhouBrinde); // Resultado: true
77
78 // 3.3. ! (NÃO / NOT): Inverte um valor booleano.
79 let diaChuvoso = true;
80 let vouSair = !diaChuvoso; // se NÃO está chuvoso
81 console.log("Vou sair?", vouSair); // Resultado: false
82
83 let portaAberta = false;
84 console.log("A porta NÃO está aberta?", !portaAberta); // Resultado: true
85
```

```
86 // -----
87 // 4. COMBINANDO TUDO ✖
88 // =====
89 // Podemos criar expressões complexas para lógicas mais elaboradas.
90
91 let notaProva = 8;
92 let notaTrabalho = 5;
93 let presencaMinima = true;
94
95 // Regra: Aprovado se (nota da prova > 7 OU nota do trabalho > 6) E se tiver presença mínima.
96 let resultadoFinal = (notaProva > 7 || notaTrabalho > 6) && presencaMinima;
97 //           (true      ||   false)      &&   true
98 //           (true)                        &&   true
99 console.log("Resultado final do aluno:", resultadoFinal); // Resultado: true
100
101 console.log("Fim da Aula 4! Estamos prontos para fazer nosso código tomar decisões!");
102 // Próxima aula: `if`, `else if` e `else`!
```

# ATIVIDADE

**Faça cada exercício em um JS  
diferente**

### 1. Verificador de Idade para Votação:

Peça ao usuário para digitar sua idade. Crie uma variável booleana `podeVotar` que seja `true` se a idade for maior ou igual a 16. Exiba o valor dessa variável no console.

**Faça cada exercício em um JS diferente**

## 2. Comparador de Senhas:

Peça ao usuário para digitar uma senha e depois para confirmá-la. Use o operador de igualdade estrita ( `===` ) para verificar se as duas senhas digitadas são idênticas. Armazene o resultado ( `true` ou `false` ) em uma variável `senhasCoincidem` e exiba no console.

**Faça cada exercício em um JS diferente**

### 3. Acesso Permitido:

Crie duas variáveis: `loginCorreto = true` e `senhaCorreta = false`. Crie uma terceira variável `acessoPermitido` que seja `true` apenas se `loginCorreto` E `senhaCorreta` forem verdadeiros. Exiba o resultado no console.

**Faça cada exercício em um JS diferente**

#### 4. Entrada Gratuita:

Um evento tem entrada gratuita para menores de 12 anos ou maiores de 65 anos. Peça a idade do usuário e crie uma variável `entradaGratuita` que seja `true` se a idade atender a uma dessas condições. Use o operador OU (`||`). Exiba o resultado.

**Faça cada exercício em um JS diferente**



## Desafio 1: Validador de Cupom de Desconto

1. Defina duas constantes: `CODIGO_CUPOM = "PROMO15"` e `DATA_VALIDADE = 25`.
2. Peça ao usuário o código do cupom que ele possui.
3. Peça ao usuário o dia do mês atual (um número de 1 a 31).
4. Crie uma variável booleana `cupomValido` que será `true` **apenas se** o código digitado for **idêntico** ao `CODIGO_CUPOM` **E** o dia atual for **menor ou igual** à `DATA_VALIDADE`.
5. Exiba no `console.log()` e em um `alert` a mensagem: "Cupom válido? [true/false]".

**Faça cada exercício em um JS diferente**

## Desafio 2: Situação do Aluno

1. Peça ao usuário para inserir três notas de provas (de 0 a 10).
2. Peça ao usuário a porcentagem de frequência do aluno (de 0 a 1, ex: 0.85 para 85%).
3. Calcule a **média** das três notas.
4. Crie as seguintes variáveis booleanas:
  - `mediaAprovacao` : `true` se a média for maior ou igual a 7.
  - `frequenciaAprovacao` : `true` se a frequência for maior ou igual a 0.75.
  - `aprovadoComLouvor` : `true` se a média for maior ou igual a 9 **E** a frequência for de 1 (100%).
  - `alunoAprovado` : `true` se `mediaAprovacao` **E** `frequenciaAprovacao` forem `true`.
5. Exiba cada uma dessas variáveis booleanas no `console.log()` com uma mensagem explicativa.
  - Ex: `console.log("Aprovado por média?", mediaAprovacao);`

**Faça cada exercício em um JS diferente**