

Aula 14

Estruturas Condicionais

Simples: if e else

Revisão da Aula 4: Comparações e Lógica

- **Operadores Relacionais:**

- Aprendemos a comparar valores com: `>`, `<`, `>=` e `<=`.
- Vimos a diferença crucial entre `==` (solto) e `===` (estrito), e porque **devemos sempre preferir o estrito** (`===` e `!==`).

- **Operadores Lógicos:**

- Combinamos expressões booleanas com:
 - `&&` (E): Ambas as partes devem ser `true`.
 - `||` (OU): Apenas uma das partes precisa ser `true`.
 - `!` (NÃO): Inverte o valor de `true` para `false` e vice-versa.

- **Resultado Final:**

- Toda expressão relacional ou lógica resulta em um único valor: `true` ou `false`.

Slide 2/4: Dando Vida às Comparações

Aula 5: Tomando Decisões com `if`

- **Objetivo de Hoje:**
 - Usar os resultados `true` e `false` da última aula para **controlar o fluxo** do nosso programa.
- **A Estrutura `if` (O "SE" do VisualG):**
 - É o bloco de decisão mais fundamental da programação.
 - A ideia é simples: **SE** uma condição for verdadeira, **FAÇA ALGO**.
- **Sintaxe Básica:**

JavaScript



```
if (condicao) {  
    // Este bloco de código só será  
    // executado se a 'condicao' for true.  
}
```

- A `condicao` vai dentro dos parênteses `()`.
- O código a ser executado vai dentro das chaves `{}`.

Slide 3/4: E Se a Condição for Falsa?

O Caminho Alternativo com `else`

- A Estrutura `else` (O "SENÃO" do VisualG):
 - O `else` nos dá um caminho alternativo para quando a condição do `if` é `false`.
- Sintaxe Completa:

JavaScript



```
if (condicao) {  
    // Bloco de código para o caso 'true'.  
} else {  
    // Bloco de código para o caso 'false'.  
}
```

- **Ponto Chave:** O programa sempre seguirá **apenas um** dos caminhos. Ou o bloco `if` é executado, ou o bloco `else` é executado. Nunca os dois.

Slide 4/4: Fluxo e Exemplos Práticos

Controlando o Fluxo do Programa

- **Como o Programa "Pensa":**

1. Avalia a `condicao` dentro do `if`.
2. O resultado é `true` ? Executa o bloco `if` e ignora o `else`.
3. O resultado é `false` ? Pula o bloco `if` e executa o bloco `else`.

- **Exemplos Práticos que Faremos Hoje:**

- Verificar se um aluno foi aprovado com base na nota.
- Checar se um usuário é maior de idade para acessar um conteúdo.
- Validar uma senha de acesso.

- **Meta da Aula:**

- Criar programas que não seguem mais um caminho único, mas que reagem e se adaptam de acordo com as condições que definirmos.

Prática

<> index.html X

JS script.js

<> index.html > html > body > script

```
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>Estruturas Condicionais Simples: if e else</h1>
10     <script src="script.js"></script>
11 </body>
12 </html>
```

JS script.js > ...

```
1  // Bem-vindos à Aula 5!
2  // Nas aulas anteriores, aprendemos a criar expressões que resultam em `true` ou `false`.
3  // Hoje, vamos USAR esses resultados para dar inteligência aos nossos programas.
4  // Com o `if` e `else`, nosso código pode tomar decisões e seguir caminhos diferentes.
5  //
6  // Tópicos de hoje:
7  // 1. A estrutura `if`: Executando código sob uma condição.
8  // 2. A estrutura `else`: O caminho alternativo.
9  // 3. Combinando com operadores lógicos para criar condições complexas.
10
11 // =====
12 // 1. A ESTRUTURA `if` (SE)
13 // =====
14 // O bloco de código dentro de um `if` só é executado se a condição
15 // dentro dos parênteses for `true`.
16
17 console.log("--- Exemplo 1: Estrutura if ---");
18 let temSol = true;
19
20 if (temSol === true) {
21 |   console.log("O dia está ensolarado! Vou à praia.");
22 }
```



```
24 // Se `temSol` fosse `false`, a mensagem acima nunca seria exibida.
25 // O programa simplesmente pularia o bloco de código.
26
27 // =====
28 // 2. ADICIONANDO O `else` (SENÃO)
29 // =====
30 // O `else` fornece um bloco de código alternativo que é executado
31 // se a condição do `if` for `false`.
32
33 console.log("\n--- Exemplo 2: if / else ---");
34 let idade = prompt("Por favor, digite sua idade:");
35 idade = Number(idade); // Não se esqueça de converter!
36
37 if (idade >= 18) {
38     alert("Você é maior de idade. Acesso permitido!");
39     console.log("Usuário maior de idade. Acesso concedido.");
40 } else {
41     alert("Você é menor de idade. Acesso negado.");
42     console.log("Usuário menor de idade. Acesso bloqueado.");
43 }
44
```

```
47 // =====
48 // 3. CONDIÇÕES COMPLEXAS NO `if`
49 // =====
50 // Podemos usar os operadores lógicos (&&, ||) que aprendemos na Aula 4
51 // para criar condições mais poderosas.
52
53 console.log("\n--- Exemplo 3: if com condição complexa ---");
54 let notaFinal = 8.5;
55 let frequenciaMinima = 0.8; // 80%
56
57 // Regra: Aprovado se nota >= 7 E frequência >= 0.75
58 if (notaFinal >= 7 && frequenciaMinima >= 0.75) {
59 |   console.log("Parabéns! Aluno(a) aprovado(a) com sucesso!");
60 } else {
61 |   console.log("Aluno(a) reprovado(a). É necessário verificar as notas ou a frequência.");
62 }
63
```

```
63
64 // =====
65 // 4. `if` ANINHADO (UM `if` DENTRO DE OUTRO)
66 // =====
67 // É possível colocar um `if` dentro de outro para criar verificações em etapas.
68
69 console.log("\n--- Exemplo 4: if aninhado ---");
70 let usuarioLogado = true;
71 let ehAdmin = false;
72
73 if (usuarioLogado === true) {
74     console.log("Usuário está logado. Verificando permissões...");
75
76     if (ehAdmin === true) {
77         console.log("Permissão de Administrador concedida. Acesso total ao painel.");
78     } else {
79         console.log("Permissão de usuário comum. Acesso limitado.");
80     }
81
82 } else {
83     console.log("Erro: Você precisa estar logado para acessar o sistema.");
84 }
85
86 console.log("Fim da Aula 5! Seus programas agora podem tomar decisões!");
87 // Próxima aula: `else if`, para criar múltiplos caminhos alternativos!
```

ATIVIDADE

Faça cada exercício em um JS diferente

1. **Aprovado ou Reprovado?**

Peça ao usuário para digitar a nota final de um aluno (de 0 a 10). Se a nota for maior ou igual a 7, mostre um `alert` com a mensagem "Parabéns, você foi aprovado!". Caso contrário, mostre "Você precisa estudar mais. Reprovado.".

Faça cada exercício em um JS diferente

2. Par ou Ímpar?

Peça ao usuário para digitar um número inteiro. Use o operador de módulo (`%`) para verificar se o número é par (`numero % 2 === 0`). Se for par, exiba "O número [número] é par.". Senão, exiba "O número [número] é ímpar.".

Faça cada exercício em um JS diferente

3. Verificador de Estoque

Peça ao usuário a quantidade de um produto em estoque. Se a quantidade for maior que 0, exiba no console "Produto disponível para venda.". Senão, exiba "Produto esgotado!".

Faça cada exercício em um JS diferente

4. Temperatura do Dia

Peça ao usuário a temperatura atual em graus Celsius. Se a temperatura for maior que 25, mostre um `alert` dizendo "O dia está quente! Ideal para um sorvete.". Senão, mostre "O clima está ameno.".

Faça cada exercício em um JS diferente

Desafio 1: Calculadora com Proteção contra Divisão por Zero

1. Peça ao usuário para inserir dois números.
2. Verifique se o **segundo número** é igual a `0`.
3. Se o segundo número for `0`, mostre um `alert` de erro com a mensagem: "Erro: Não é possível dividir um número por zero."
4. **Caso contrário** (`else`), realize a divisão do primeiro número pelo segundo e mostre o resultado em um `alert`, como: "O resultado da divisão é [resultado]."

Faça cada exercício em um JS diferente

Desafio 2: Verificador de Empréstimo Bancário

1. Um banco só aprova empréstimos se o cliente atender a **ambas** as condições:
 - Ter um salário mensal de R\$ 2.000,00 ou mais.
 - Não possuir pendências no Serasa (representado por uma variável booleana).
2. Peça ao usuário seu salário mensal (use `parseFloat`).
3. Crie uma variável `let temPendencias = false;` (para simular que o cliente não tem pendências). Você pode mudar para `true` para testar o outro caminho.
4. Crie uma condição `if` que verifique se `salario >= 2000` **E** se `temPendencias === false` .
5. Se a condição for verdadeira, mostre um `alert` parabenizando: "Parabéns! Seu empréstimo foi aprovado."
6. Senão, mostre um `alert` informando: "Desculpe, não foi possível aprovar seu empréstimo no momento."

Faça cada exercício em um JS diferente