

JavaScript: Fundamentos para Desenvolvimento Web Interativo - Turma 2025B

2.9.1 Condicionais

As estruturas condicionais permitem executar partes do código com base em uma condição ou expressão booleana.

if

A estrutura de controle **if** é usada para tomar decisões baseadas em condições. Se a condição especificada for verdadeira, um bloco de código é executado; se for falsa, o bloco é ignorado.

Veja no quadro abaixo um exemplo de uso do **if**:

```
let idade = 20;  
if (idade >= 18) {  
    console.log("Você é maior de idade.");  
}
```

No exemplo acima, a condição **idade >= 18** é verdadeira, então a mensagem "**Você é maior de idade.**" é impressa. Se a condição fosse falsa, nada aconteceria.

if-else

A estrutura de controle **if-else** é usada para tomar decisões baseadas em condições. Ela permite que você execute um bloco de código se uma condição for verdadeira e um bloco diferente se for falsa.

Veja no quadro abaixo um exemplo:

```
let idade = 20;
if (idade >= 18) {
  console.log("Você é maior de idade."); // Executado se a condição for verdadeira
} else {
  console.log("Você é menor de idade."); // Executado se a condição for falsa
}
```

Essa estrutura é uma das mais comuns para controlar o fluxo do programa e lidar com situações de ramificação.

else if

A estrutura de controle **else if** adiciona condições intermediárias entre um **if** e um **else**, permitindo várias verificações.

Veja no quadro abaixo um exemplo:

```
let temperatura = 25;
if (temperatura > 30) {
  console.log("Está quente.");
} else if (temperatura < 15) {
  console.log("Está frio.");
} else {
  console.log("A temperatura está agradável.");
}
```

switch

O **switch** é uma estrutura de controle que permite decidir entre múltiplas opções com base em um valor ou expressão. Ele é uma alternativa ao uso repetitivo de **if-else**, especialmente quando há várias condições relacionadas a um único valor.

Veja no quadro abaixo um exemplo:

```
let dia = 2;
switch (dia) {
  case 1:
    console.log("Segunda-feira");
    break;
  case 2:
    console.log("Terça-feira");
    break;
  case 3:
    console.log("Quarta-feira");
    break;
  default:
    console.log("Dia desconhecido");
}
```

Neste exemplo, como **dia** é **2**, o **case 2** é executado, exibindo "**Terça-feira**". Se não houver correspondência com nenhum **case**, o bloco **default** é executado, fornecendo um caminho alternativo.

OBS: O uso do **break** é crucial para evitar a execução contínua de blocos subsequentes, um fenômeno conhecido como "fall-through". Se o **break** não for usado, o controle passa para o próximo **case**, independentemente do resultado da expressão. Veja no quadro abaixo um exemplo:

```
let fruta = "banana";
switch (fruta) {
  case "maçã":
    console.log("Você escolheu uma maçã.");
    break;
  case "banana":
    console.log("Você escolheu uma banana.");
    break;
  default:
    console.log("Fruta desconhecida.");
}
```

Se o **break** fosse omitido após "**banana**", a execução continuaria até o próximo **break** ou **default**, causando comportamento inesperado.

◀ 2.9 Estruturas de Controle

Seguir para...

2.9.2 Laços de repetição (loops) ▶

[Baixar o aplicativo móvel.](#)