

Curso de JavaScript 2023



Aula 03 - Parte 1 - Estruturas de Controle e Repetição: if, else, switch, loops



Instrutor: *Prof. Me. Mário Carvalho*

E-mail para contato: mario.carvalho@ufms.br

Realização: *UFMS e Semadur*





1. Sobre o curso

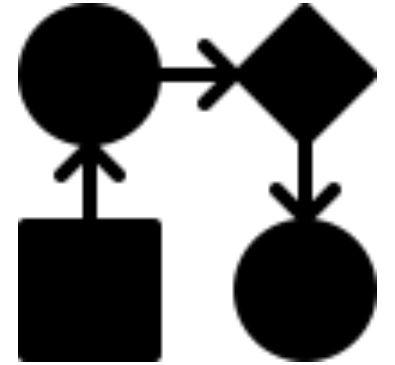
- Instrutor: **Prof. Me. Mário de Araújo Carvalho**
- E-mail para contato: mario.carvalho@ufms.br
- Sala Google Meet: <https://meet.google.com/fcq-djzs-dzd>
- Repositório oficial do curso:
- <https://github.com/MarioCarvalhoBr/semadur-curso-javascript-2023>
- **Ava:** <https://www.eadfapec.com.br/course/view.php?id=245§ion=4>
- Instituição: **UFMS e Semadur**
- Modalidade: **Online síncrono**
- Duração: 45 horas

Introdução

- - Definição de Estruturas de Controle

Fundamentos: Estruturas de controle são construções da linguagem de programação que gerenciam o fluxo de execução do código.

Propósito: Permitem a tomada de decisões, a execução repetitiva de blocos de código e o controle do fluxo do programa com base em condições.



Introdução

- - Definição de Estruturas de Controle

Tipos Principais de Estruturas de Controle

Estruturas Condicionais:

Tomam decisões com base em condições.

Exemplos: **if**, **else**, **switch**.

Estruturas de Loop:

Executam um bloco de código repetidamente.

Exemplos: **for**, **while**, **do-while**.



Introdução

- - Definição de Estruturas de Controle

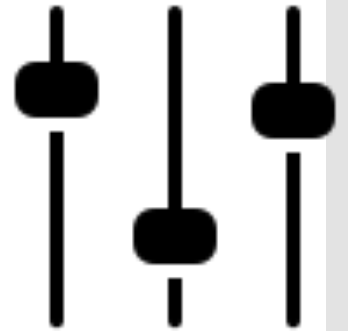
Importância em Programação

Flexibilidade: Permitem a escrita de códigos que podem lidar com diferentes situações e dados de entrada.

Controle: Oferecem ao programador controle sobre o comportamento do programa.

Papel Central: As estruturas de controle são fundamentais para a criação de programas dinâmicos e responsivos.

Variedade: Existem várias estruturas de controle, cada uma adequada para diferentes tipos de tarefas de programação.



Introdução

- - Importância no Controle de Fluxo de Programas

Dinamismo: Torna os programas adaptáveis a diferentes situações e dados de entrada.

Eficiência: Permite a execução de código somente quando necessário, economizando recursos.

Introdução

- - Importância no Controle de Fluxo de Programas
- **Controle de Fluxo em Ação**

Tomada de Decisões:

Adaptar a execução com base em condições.

Exemplo: Login bem-sucedido vs. falha de autenticação.

Repetição Controlada:

Executar tarefas repetitivas de forma eficiente.

Exemplo: Processamento de itens em uma lista.

Direcionamento do Programa:

Guiar o caminho que o programa segue.

Exemplo: Navegação em um menu de aplicativo.

Introdução

- - Importância no Controle de Fluxo de Programas

Robustez: Maior capacidade de lidar com cenários variados.

Legibilidade: Código mais claro e fácil de entender.

Manutenção: Facilita a atualização e correção de códigos.

Papel Fundamental: O controle de fluxo é essencial para a criação de programas eficazes e versáteis.

Habilidade Necessária: Entender e aplicar estruturas de controle é uma habilidade chave em programação.

Parte 1: Estruturas Condicionais (if, else)

- - Conceito e Sintaxe de if e else

Uso: Tomada de decisões binárias.

Funcionamento: Executa blocos de código com base em uma condição booleana.

Exemplo:

```
if (condicao) {  
    // Código se a condição for verdadeira  
} else {  
    // Código se a condição for falsa  
}
```

Parte 1: Estruturas Condicionais (if, else)

- - Exemplos Práticos

Exemplo 1: Uso do if

- **Propósito:** Executar código se uma condição for verdadeira.
- **Código:**

```
if (temperatura > 30) {  
    console.log("Está quente hoje!");  
}
```

Parte 1: Estruturas Condicionais (if, else)

- - Exemplos Práticos

Exemplo 2: Uso do else

- **Propósito:** Executar código se a condição do `if` anterior não for atendida.

- **Código:**

```
if (saldo > 0) {  
    console.log("Você tem saldo positivo!");  
} else {  
    console.log("Você não tem saldo!");  
}
```

Parte 1: Estruturas Condicionais (if, else)

● - Exemplos Práticos

Exemplo 3: Uso do if-else

- **Propósito:** Escolher entre vários blocos de código com base na condição.

- **Código:**

```
if (cadastro < 0) {  
    console.log("Usuário não completou o cadastro");  
} else if (cadastro == 0) {  
    console.log("Usuário pendente de ativação");  
} else {  
    console.log("Usuário ativado");  
}
```

Parte 1: Estruturas Condicionais (if, else)

- - Exercícios e Discussão

Exercício 1: Verificação de Maioridade

Peça ao usuário para inserir sua idade e use `if` e `else` para verificar se ele é maior de idade. Se a idade for 18 anos ou mais, imprima "Você é maior de idade". Caso contrário, imprima "Você é menor de idade".

Exercício 2: Classificador de Notas

Solicite ao usuário para inserir uma nota (por exemplo, de 0 a 100). Use uma série de `if` e `else` para classificar a nota em categorias (por exemplo, 90-100 é 'A', 80-89 é 'B', etc.) e imprimir a categoria correspondente.

Parte 1: Estruturas Condicionais (if, else)

- - Exercícios e Discussão

Exercício 3: Calculadora de Desconto

Crie um programa que peça ao usuário o valor de um produto e a idade do comprador. Se a idade do comprador for maior que 60 anos, aplique um desconto de 10% sobre o valor. Use `if` e `else` para aplicar o desconto e imprimir o valor final.

Exercício 4: Verificação de Senha

Peça ao usuário para digitar uma senha e, em seguida, a confirmação dessa senha. Use `if` e `else` para verificar se as duas senhas são iguais. Se forem, imprima "Senha confirmada". Se não, imprima "As senhas não coincidem".

Parte 2: Estrutura Condicional (switch)

- - Conceito e Sintaxe de switch

Definição: Uma alternativa à sequência de if e else if para controle de fluxo baseado em múltiplas condições.

Funcionamento: Compara uma variável com diferentes valores e executa o bloco de código correspondente.

Estrutura básica:

```
switch(expressao) {  
    case valor1:  
        // Código para valor1  
        break;  
    case valor2:  
        // Código para valor2  
        break;  
    default:  
        // Código se nenhum caso corresponder  
}
```

Parte 2: Estrutura Condicional (switch)

● - Conceito e Sintaxe de switch

Quando Usar switch?

Vários Casos: Ideal quando há muitas condições a serem verificadas.

Legibilidade: Melhora a clareza do código em comparação com múltiplos if-else.

Exemplo:

```
// Escolha de Menu
switch(menuOption) {
    case 'Início':
        // Executa código para 'Início'
        break;
    case 'Sobre':
        // Executa código para 'Sobre'
        break;
    // Outros casos
    default:
        // Executa se nenhuma opção corresponder
}
```


Parte 2: Estrutura Condicional (switch)

- - Exemplos Práticos
 - Cálculo do IMC
 - Escolher menus

Parte 2: Estrutura Condicional (switch)

- - Exercícios e Discussão

- 1. **Exercício 1: Calculadora Simples**

Desenvolva uma calculadora simples que aceite dois números e uma operação (por exemplo, '+', '-', '*', '/') do usuário. Use a estrutura switch para realizar a operação matemática escolhida e imprimir o resultado. Por exemplo:

- 2. **Exercício 2: Planejador de Atividades**

Crie um programa que sugira atividades baseadas no dia da semana. Por exemplo, "Segunda-feira: Estudo", "Terça-feira: Esporte", etc. Peça ao usuário para inserir um dia da semana e use switch para imprimir a atividade sugerida para aquele dia. Por exemplo:

Parte 3: Loops (For e While)

- - For Loop: Conceito, Sintaxe e Exemplos

Parte 3: Loops (For e While)

- - While Loop: Conceito, Sintaxe, Diferenças com For, Exemplos

Parte 3: Loops (For e While)

- - Loops Infinitos e Controle de Loop

Parte 3: Loops (For e While)

- - Exercícios e Discussão

Parte 4. Encerramento

- - Resumo dos tópicos abordados e orientações sobre recursos para continuar aprendendo JavaScript:
 - MDN Web Docs:
<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>
- - Próximos passos e recursos adicionais para aprendizado autônomo:
 - HTML: <https://www.w3schools.com/html/default.asp>
 - JavaScript: <https://www.w3schools.com/js/default.asp>

Dúvidas?



Obrigado!

Mário Carvalho

mario.carvalho@ufms.br