

Curso de JavaScript 2023



Módulo 2 - Aula 05: Funções e Estruturas de Dados (Arrays, Objetos e JSON)



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

- Por que Usar Funções?

Organização: Torna o código mais organizado e legível.

Reutilização: Permite reutilizar código, evitando repetições.

Manutenção: Facilita a manutenção e atualização do código.



Funções

Introdução

- Organização do Código

Divisão em Blocos Lógicos: Funções ajudam a dividir o código em blocos lógicos, cada um com uma tarefa específica.

Facilidade de Leitura: Códigos bem organizados são mais fáceis de ler e entender.



Funções

Introdução

- Reutilização Eficiente

Evitar Redundância: Com funções, você escreve um bloco de código uma vez e o reutiliza onde necessário.

Exemplo: Função de cálculo pode ser usada em diferentes partes do programa.

Facilidade de Manutenção

Atualizações Simplificadas: Alterações em uma função se aplicam a todas as suas instâncias.

Detecção de Erros: Erros em uma função centralizada são mais fáceis de encontrar e corrigir.



Funções



Funções

- - O que são Funções?

Definição: Blocos de código reutilizáveis projetados para executar uma tarefa específica.

Vantagem: Simplificam o código e promovem a reutilização.

Estrutura Básica de uma Função:

Definição de Função:

```
function nomeDaFuncao(parametro1, parametro2) {  
    // Corpo da função  
    return resultado;  
}
```

Chamada de Função:

```
nomeDaFuncao(valor1, valor2);
```



Funções

- Parâmetros e Argumentos

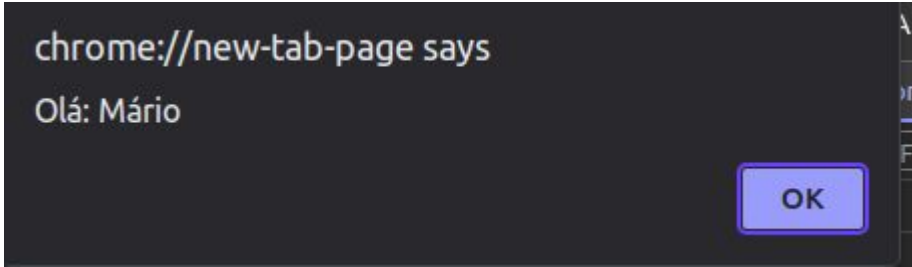
Parâmetros: Variáveis listadas como parte da definição da função.

Argumentos: Valores passados para a função quando ela é chamada.

Exemplo 1:

```
function hello(nome) {  
  alert("Olá: " + nome);  
}
```

```
hello("Mário");
```

A screenshot of a Chrome alert dialog box. The title bar reads "chrome://new-tab-page says". The main text area contains "Olá: Mário". At the bottom right, there is a blue button with the text "OK".

chrome://new-tab-page says

Olá: Mário

OK

Funções

- Parâmetros e Argumentos

Parâmetros: Variáveis listadas como parte da definição da função.

Argumentos: Valores passados para a função quando ela é chamada.

Exemplo 2:

```
function calcularArea(largura, altura) {  
    return largura * altura;  
}
```

// Usada para calcular áreas em diferentes contextos do programa



Funções

- Escopo em Funções

Escopo Local: Variáveis declaradas dentro de uma função não são acessíveis fora dela.

Escopo Global: Variáveis declaradas fora de todas as funções são globais.

Exemplos:

Função soma	Função saudação
<pre>function soma(a, b) { return a + b; } console.log(soma(5, 3)); // 8</pre>	<pre>function saudacao() { let mensagem = "Olá!"; return mensagem; } console.log(saudacao()); // "Olá!" // console.log(mensagem); // Erro: // mensagem não definida</pre>



Funções

Conclusões

Flexibilidade e Reutilização: Funções são fundamentais para criar códigos eficientes e organizados.

Entendimento de Escopo: O escopo é crucial para controlar onde e como as variáveis de uma função são acessíveis.

Eficiência e Clareza: O uso de funções é uma prática essencial para escrever códigos eficientes, claros e facilmente mantidos.

Habilidade Fundamental: Dominar funções é crucial para qualquer desenvolvedor.

Funções

Exercícios e Discussão

1. Criar uma Função de Multiplicação: Escreva uma função que multiplica dois números.
2. Função de Verificação: Crie uma função que aceite um número e verifique se é par ou ímpar.
3. Função de Saudação: Escreva uma função que aceite um nome como parâmetro e retorne uma saudação personalizada.
4. Controle de Escopo: Experimente com variáveis de escopo local e global e observe seu comportamento.



Arrays e Objetos

- O que são Arrays?

Definição: Uma coleção de elementos, geralmente do mesmo tipo, acessíveis por índices.

Uso: Armazenar e manipular conjuntos de dados.

Estrutura de um Array

Criação de um Array:

```
let numeros = [1, 2, 3, 4, 5];
```

Acesso a Elementos:

```
console.log(numeros[0]); //1
```

Arrays e Objetos

Exemplos práticos de Array

Iterando sobre Arrays:

```
let numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9];  
for (let i = 0; i < numeros.length;  
i++) {  
    console.log(numeros[i]);  
}
```

// Saída:

//1

//2

//3

...

//9

Arrays e Objetos

Manipulação de Arrays

Métodos Comuns de Arrays

Definição: Arrays em JavaScript são objetos que representam **listas ordenadas e oferecem métodos** para manipulação de seus elementos.

Métodos de Exemplo

push(): Adiciona elementos ao final do array.

pop(): Remove o último elemento do array.

map(): Cria um novo array com os resultados da chamada de uma função para cada elemento do array.

filter(): Cria um novo array com todos os elementos que passam no teste implementado pela função fornecida.

Arrays e Objetos

Manipulação de Arrays: Push

Exemplo 1: Uso de push()

Objetivo: Adicionar um novo elemento ao final de um array.

```
let frutas = ['Maçã', 'Banana', 'Laranja'];  
frutas.push('Uva');  
console.log(frutas); // Saída: ['Maçã',  
  'Banana', 'Laranja', 'Uva']
```

Manipulação de Arrays: Pop

Exemplo 2: Uso de pop()

Objetivo: Remover o último elemento de um array.

```
let numeros = [1, 2, 3, 4, 5];  
let ultimoNumero = numeros.pop();  
console.log(ultimoNumero); // Saída: 5  
console.log(numeros); // Saída: [1, 2,  
3, 4]
```

Arrays e Objetos

Manipulação de Arrays: Map

Exemplo 3: Uso de map()

Objetivo: Criar um novo array com os quadrados de cada número de um array original.

```
let numeros = [1, 2, 3, 4, 5];  
let quadrados = numeros.map(num => num * num);  
console.log(quadrados); // Saída: [1, 4, 9,  
16, 25]
```

Arrays e Objetos

Manipulação de Arrays: Filter

Exemplo 4: Uso de filter()

Objetivo: Filtrar um array para incluir apenas números pares.

```
let numeros = [1, 2, 3, 4, 5, 6];  
let pares = numeros.filter(num => num % 2 === 0);  
console.log(pares); // Saída: [2, 4, 6]
```



Arrays e Objetos

Exemplos práticos de Objetos

Manipulação de Objetos:

```
pessoa.idade = 31; // Atualizar idade  
console.log(pessoa.idade); // 31
```

Arrays e Objetos: Ferramentas poderosas para armazenamento e manipulação de dados.

Fundamentos da Programação: Essenciais para a organização de dados em muitas linguagens de programação.

Arrays e Objetos

Manipulação de Objetos

Métodos Comuns de Objetos

Definição: Objetos em JavaScript são coleções de **propriedades**, e podem ser manipulados de várias maneiras.

Métodos de Exemplo

Object.keys(): Retorna um array de chaves de propriedade de um dado objeto.

Object.values(): Retorna um array de valores de propriedade de um dado objeto.

Object.entries(): Retorna um array de pares de chave/valor do objeto.



Arrays e Objetos

Manipulação de Objetos: keys e values

```
let pessoa = { nome: "Ana", idade: 28 };  
let chaves = Object.keys(pessoa); // ["nome",  
  "idade"]  
let valores = Object.values(pessoa); // ["Ana",  
  28]
```



Arrays e Objetos

Manipulação de Objetos: entries

Objetivo: Converter um objeto em um array de pares [chave, valor].

```
let pessoa = {  
  nome: "João",  
  idade: 30,  
  profissao: "Engenheiro"  
};
```

```
let entradas = Object.entries(pessoa);  
console.log(entradas);
```

Saída Esperada:

```
[  
  [ "nome", "João" ],  
  [ "idade", 30 ],  
  [ "profissao", "Engenheiro" ]  
]
```



Arrays e Objetos

- O que são Objetos?

Definição: Uma coleção de dados e/ou funcionalidades relacionadas, geralmente representados como chave e valor.

Uso: Modelar entidades do mundo real e organizar dados de forma estruturada.

Estrutura de um Objeto

- Criação de um Objeto:

```
let pessoa = {  
  nome: "Maria",  
  idade: 30,  
  profissao: "Engenheira"  
};
```

- Acesso a Propriedades:

```
console.log(pessoa.nome); // Maria
```




Arrays e objetos

- **Exercícios e Discussão**

1. Inversão de Array: Crie um array e escreva um código para inverter sua ordem.
2. Média de Elementos: Calcule a média dos elementos de um array numérico.
3. Busca de Elemento: Escreva uma função que busca um elemento em um array e retorna sua posição.
4. Adicionar Propriedades: Adicione novas propriedades a um objeto existente.
5. Listar Chaves e Valores: Escreva uma função que lista todas as chaves e valores de um objeto.
6. Objeto para Array: Converta um objeto em um array de suas propriedades.

JSON

• ...



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