

# PROGRAMAÇÃO FRONT END

João Choma Neto

[joao.choma@unicesumar.edu.br](mailto:joao.choma@unicesumar.edu.br)

<https://github.com/JoaoChoma/frontend>

Unicesumar – Maringá



## EVENTOS E FUNÇÕES

A linguagem de programação JavaScript proporciona a integração de conceitos lógicos, funções e orientações a objetivos aplicados com os elementos do HTML, assim, podemos utilizar formulário para criar eventos e funções

## EVENTOS E FUNÇÕES

- Para utilizar um evento, deve utilizar uma função
- A FUNÇÃO já é interpretada pelo compilador do JavaScript
- *(on)click, (on)change, (on)submit, (on)load,* entre outros

# FUNÇÃO

1. `function nomeDaFuncao(){`
2.       código
3. `}`

# O QUE É PARÂMETRO

Um parâmetro de uma função é um valor que pode ser passado para a função quando ela é invocada

## O QUE É PARÂMETRO

É uma forma de fornecer informações à função para que ela possa executar uma ação específica ou retornar um resultado baseado nos valores recebidos

## O QUE É PARÂMETRO

```
function nomeDaFuncao(parametro1,  
    parametro2, parametro3) {  
    // Corpo da função que usa os parâmetros  
}
```

## O QUE É PARÂMETRO

```
function somar(a, b) {  
    return a + b;  
}  
  
let resultado = somar(3, 5);
```



## O QUE É RETURN

Estas funções se tornam mais úteis se retornam um valor específico

Para fazer um programa retornar um valor específico, utilizamos o comando “***return***”, seguido do conteúdo que desejamos retornar

## RETURN

```
function somar(a, b) {  
    return a + b;  
}
```

```
let resultado = somar(3, 5);
```

# FUNÇÃO ANÔNIMA

Nas funções anônimas, é definida a programação de um evento sem atribuir um nome para a função

Utilizamos o comando "***function()***" e inserimos a lógica da programação que necessitamos resolver

# FUNÇÃO ANÔNIMA

```
1. var dobro = function(a) {  
3.   return a * 2;  
4. }  
5. var num = Number(prompt("Número: "));  
6. alert("O dobro é: " + dobro(num));
```

## OUTRA FORMA DE ESCREVER A FUNÇÃO

```
1. function situacaoAluno(nota, media) {  
2.     var situacao = (nota >= media) ? "Aprovado" : "Reprovado";  
3.     return situacao;  
4. }
```

O sinal de interrogação "?" é a primeira parte da estrutura condicional e o sinal de ":" seria a parte do "***else***" (se não)

EVENTOS

## REAÇÃO

Quando JavaScript é usado em páginas HTML, o JavaScript pode “reagir” a esses eventos.

Um evento HTML pode ser algo que o navegador faz ou algo que um usuário faz.

## REAÇÃO

Uma página da web HTML terminou de  
carregar

Um campo de entrada HTML foi alterado

Um botão HTML foi clicado



## EVENTOS

**onchange** - Um elemento HTML foi alterado

**onclick** - O usuário clica em um elemento HTML

**onmouseover** O usuário move o mouse sobre um elemento HTML

## EVENTOS

**onmouseout** - O usuário afasta o mouse de um elemento HTML

**onkeydown** - O usuário pressiona uma tecla do teclado

**onload** - O navegador terminou de carregar a página

DOM



## DOM

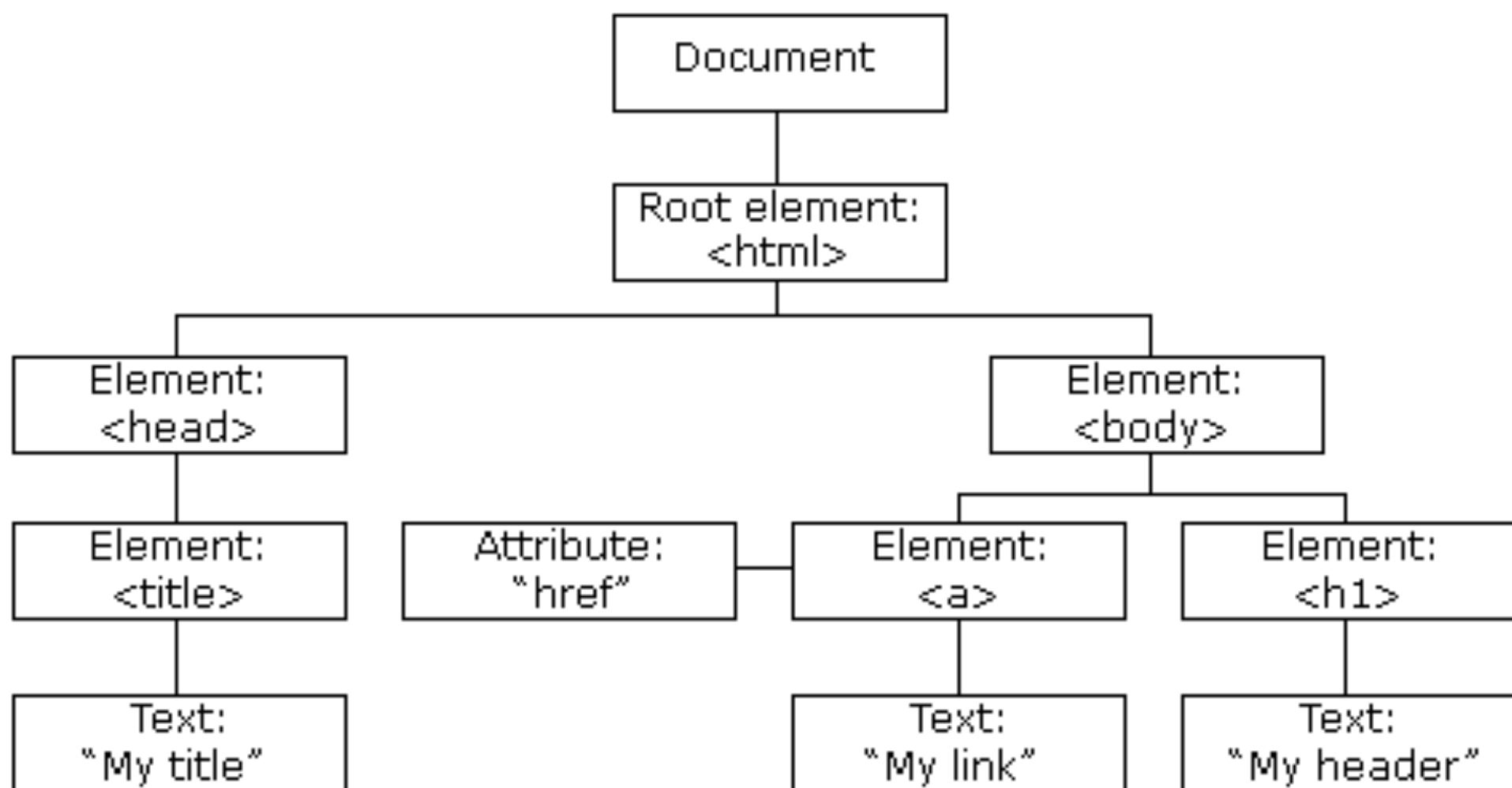
O DOM (Document Object Model) do HTML é uma interface de programação para documentos HTML e XML. Ele representa a página para que programas possam mudar a estrutura do documento, seu estilo e conteúdo.

O DOM representa o documento como uma árvore de objetos; cada objeto corresponde a uma parte do documento, como um elemento, atributo ou texto.

## DOM

Quando uma página é carregada no navegador, ela é convertida em um DOM. Isso permite que scripts, como JavaScript, interajam com a estrutura da página, acessando e modificando elementos HTML dinamicamente.

Por exemplo, você pode usar JavaScript para adicionar, remover ou alterar elementos HTML, modificar atributos de elementos e responder a eventos na página, como cliques ou movimentos do mouse, tudo através do DOM.



## **Métodos disponíveis**

[https://www.w3schools.com/jsref/dom\\_obj\\_document.asp](https://www.w3schools.com/jsref/dom_obj_document.asp)

## EXEMPLO

O método **addEventListener()** anexa um manipulador de eventos a um documento



## EXEMPLO

O método **createElement** é usado para criar um novo elemento no documento. Ele é um método do objeto `document`

Quando você cria um elemento, ele não aparece automaticamente na página; você precisa inseri-lo no documento usando métodos como **appendChild** ou **insertBefore**

## EXEMPLO

Enquanto `createElement` é usado para criar um novo elemento, **`removeChild`** é um método usado para remover um elemento existente da árvore DOM

Este método é chamado no elemento pai do elemento que você deseja remover



# JSON

---

## JSON

- JSON significa JavaScript Object Notation.
- É um formato leve e fácil de ler para troca de dados.
- Baseado na sintaxe de objetos JavaScript.
- Independente de linguagem.

# JSON

- Consiste em pares chave-valor.
- Os valores podem ser strings, números, objetos, arrays, booleanos ou nulos.
- Exemplo:

```
{ "nome": "Maria", "idade": 25, "cidade": "Rio de Janeiro", "ativos":  
  ["caminhada", "leitura", "cozinhar"], "casada": false, "endereco": null }
```

## **Por que JSON é Importante?**

- Ampla adoção na web devido à sua simplicidade.
- Facilita a troca de dados entre servidores e clientes.
- Usado em APIs, armazenamento de dados, etc.

## Como Usar JSON em JavaScript

- `JSON.parse()`: Converte uma string JSON em um objeto JavaScript.
- `JSON.stringify()`: Converte um objeto JavaScript em uma string JSON.

# PROGRAMAÇÃO FRONT END

João Choma Neto

[joao.choma@unicesumar.edu.br](mailto:joao.choma@unicesumar.edu.br)

<https://github.com/JoaoChoma/frontend>

Unicesumar – Maringá

