

Programação Front End – AULA 14

Matheus Moresco

Engenharia de Software – 3 Período

2025/01

Introdução

- Introdução ao JavaScript
- O que é?
- Como funciona?
- Como usar na página web?
- Variáveis, operadores, loops e funções

Introdução ao JavaScript

- JavaScript é uma linguagem de programação amplamente utilizada para desenvolvimento web. Ela permite criar páginas dinâmicas e interativas, sendo executada principalmente no navegador do usuário.
- Com JavaScript, é possível manipular elementos HTML e CSS, responder a eventos (como cliques e teclas pressionadas) e até mesmo consumir APIs para comunicação com servidores.
- Além do uso no front-end, JavaScript também pode ser executado no back-end com plataformas como **Node.js**, permitindo o desenvolvimento de aplicações completas (full stack) com a mesma linguagem.



Introdução ao JavaScript

Principais características do JavaScript:

- **Linguagem interpretada:** Não precisa ser compilada antes da execução.
- **Baseada em eventos:** Pode reagir a interações do usuário.
- **Tipagem dinâmica:** As variáveis podem armazenar diferentes tipos de dados sem precisar de uma declaração explícita.
- **Multiparadigma:** Suporta programação funcional, orientada a objetos e baseada em eventos.

Introdução ao JavaScript

- JavaScript é a linguagem de programação da web.
- Executado no navegador, permite interatividade nas páginas.
- Usado para desenvolvimento frontend e backend (Node.js).

Exemplo: Criando um "Hello, World!"

```
console.log("Hello, World!");
```

Tipos de Variáveis em JavaScript

No JavaScript, as variáveis podem ser declaradas usando três palavras-chave principais. Cada uma tem um comportamento diferente em relação ao escopo e mutabilidade. Além disso, JavaScript possui uma tipagem dinâmica, ou seja, uma variável pode armazenar diferentes tipos de valores ao longo do código.

- **var**: Declara uma variável globalmente ou dentro de uma função.
- **let**: Declara uma variável dentro do escopo do bloco.
- **const**: Define uma constante, não pode ser reatribuída.



Declaração de Variáveis - var

- Escopo de função (não respeita escopo de bloco).
- Pode ser redeclarada e reatribuída.
- Sofre hoisting (é movida para o topo do escopo antes da execução).
- Evitar a utilização.

```
var nome = "João";  
var nome = "Maria"; // ✓ Permitido  
console.log(nome); // Maria
```

Declaração de Variáveis - let

- Escopo de bloco ({ }).
- Pode ser reatribuída, mas não redeclarada no mesmo escopo.
- Também sofre hoisting, mas não pode ser usada antes da declaração.
- Preferível usar esta em vez da var.

```
let idade = 25;  
idade = 30; //  Permitido  
console.log(idade); // 30  
  
let idade = 35; //  Erro! Não pode redeclarar no mesmo escopo.
```


Declaração de Variáveis - const

- Escopo de bloco ({ }).
- Não pode ser reatribuída nem redeclarada.
- Deve ser inicializada na declaração.

```
const PI = 3.1415;  
PI = 3.14; // ✗ Erro! Não pode ser reatribuída.
```

Tipos de Dados em JavaScript

- JavaScript possui **tipagem dinâmica**, o que significa que uma variável pode armazenar diferentes tipos de valores ao longo do código. Os principais tipos são:
- Primitivos (Imutáveis)
 - **String** → `let nome = "Alice";`
 - **Number** → `let idade = 30;`
 - **Boolean** → `let ativo = true;`
 - **Undefined** → `let x;`
 - **Null** → `let y = null;`
 - **BigInt** → `let big = 12345678901234567890n;`
 - **Symbol** → `let id = Symbol("id");`
- Referência (Objetos)
 - **Array** → `let lista = [1, 2, 3];`
 - **Objeto** → `let pessoa = { nome: "Alice", idade: 30 };`
 - **Função** → `function dizerOla() { console.log("Olá!"); }`

Operadores em JavaScript

- Aritméticos: +, -, *, /, %
- Atribuição: =, +=, -=, *=, /=
- Comparativos: ==, ===, !=, !==, >, <
- Lógicos: &&, ||, !

Operadores aritméticos

- Usados para realizar cálculos matemáticos

Operador	Descrição	Exemplo	Resultado
+	Adição	$5 + 3$	8
-	Subtração	$05 - 3$	2
*	Multiplicação	$5 * 3$	15
/	Divisão	$10 / 2$	5
%	Módulo (Resto)	$10 \% 3$	1
**	Exponenciação	$2 ** 3$	8

Operadores de Atribuição

- Usados para armazenar valores em variáveis.

Operador	Exemplo	Equivalente a
=	$x = 5$	$x = 5$
+=	$x += 2$	$x = x + 2$
-=	$x -= 2$	$x = x - 2$
*=	$x *= 3$	$x = x * 3$
/=	$x /= 2$	$x = x / 2$
%=	$x \% = 3$	$x = x \% 3$

Operadores de Comparação

- Usados para comparar valores e retornam um booleano (true ou false).

Operador	Descrição	Exemplo	Resultado
==	Igualdade (com conversão)	"5" == 5	true
===	Igualdade estrita (sem conversão)	"5" === 5	false
!=	Diferença (com conversão)	"5" != 5	false
!==	Diferença estrita (sem conversão)	"5" !== 5	true
>	Maior que	10 > 5	true
<	Menor que	10 < 5	false
>=	Maior ou igual	10 >= 10	true
<=	Menor ou igual	5 <= 10	true

Operadores Lógicos

- Usados para combinar expressões booleanas.

Operador	Descrição	Exemplo	Resultado
&&	AND (E)	true && false	false
	OR(OU)	true false	true
!	NOT (NÃO)	!true	false

Estruturas Condicionais

If/Else:

```
let idade = 18;
if (idade > 18) {
    console.log("Mais de 18 anos");
} else if (idade == 18) {
    console.log("Tem 18 anos");
} else {
    console.log("Menor de idade");
}
```

Switch

```
let dia = "segunda";
switch (dia) {
    case "segunda":
        console.log("Início da semana");
        break;
    case "sexta":
        console.log("Final de semana chegando");
        break;
    default:
        console.log("Dia normal");
}
```


Loops (Estruturas de Repetição)

For

```
for (let i = 0; i < 5; i++) {  
    console.log("Repetição " + i);  
}
```

While

```
let contador = 0;  
while (contador < 5) {  
    console.log("Contagem: " + contador);  
    contador++;  
}
```

Funções em JavaScript

Função tradicional:

```
function saudacao(nome) {  
    return "Olá, " + nome + "!";  
}  
console.log(saudacao("Lucas"));
```

Função anônima:

```
let soma = function(a, b) {  
    return a + b;  
};
```

Arrow function:

```
const multiplicar = (a, b) => a * b;
```

Exemplo Prático

- Crie um script interativo:
 1. Fazer uma função que recebe o nome e idade de usuário.
 2. O programa verifica se ele é maior de idade.
 3. Exibe uma mensagem personalizada.

Conclusão e Recursos Extras

- Revisamos variáveis, operadores, loops e funções.
- JavaScript é a base do frontend e essencial para desenvolvimento web.
- Materiais para estudo:
 - [MDN JavaScript](#)
 - [W3Schools JavaScript](#)