

# Aula 3: JavaScript

Prof. Gustavo Clay



**Centro Universitário**  
A teoria e a prática juntas

## Objetivos da Aula

1. Entender o que é **JavaScript** e sua importância.
2. Aprender os conceitos básicos da linguagem.
3. Escrever e executar código JavaScript.
4. Praticar com exemplos e exercícios.

## O que é JavaScript?

- **JavaScript:** Linguagem de programação para desenvolvimento web.
- Permite adicionar interatividade e dinamismo às páginas.
- Executada no navegador do usuário (client-side).
- Também pode ser executada no servidor (Node.js).
- Criada por Brendan Eich em 1995.

## Por que aprender JavaScript?

1. Fundamental para desenvolvimento web.
2. Usado em front-end (React, Angular, Vue) e back-end (Node.js).
3. Usado em aplicações web, mobile e desktop.(React Native, Electron).
4. Grande comunidade e ecossistema.

# Configurando o Ambiente

1. **Navegador:** Use o Chrome ou Firefox (ferramentas de desenvolvedor).
2. **Node.js:** Para executar JavaScript fora do navegador.
  - Instale em [nodejs.org](https://nodejs.org).
3. **VS Code:** Editor de código recomendado.

# Sintaxe Básica



## Comentários

- Comentários de uma linha:

```
// Isso é um comentário de uma linha.
```

- Comentários de várias linhas:

```
/*  
Isso é um comentário  
de várias linhas.  
*/
```

# Variáveis

- Declarar variáveis com `var`, `let` e `const`.

```
var nome = "João José";  
let idade = 25;  
const cidade = "Brasília";
```



## Diferenças entre `var`, `let` e `const` :

- `var` : escopo global ou de função.
- `let` : escopo de bloco.
- `const` : constante (não pode ser reatribuída).

## Exemplos de Variáveis

```
var indefinido; // undefined
var _condicao = false; // Boolean
var x = 2019; // Número sem decimal
var juros = 2.56; // Número com decimal
let $mensagem = 'Olá Mundo!'; // String
let pessoa = { nome: "José", idade: 30 }; // Objeto
let carros = ["Audi", "Volvo", "BMW"]; // Array
const soma = (a, b) => a + b; // Função
const PI = 3.141592653589793; // Variável imutável
PI = 3.14; // Atribuição inválida
var 1variavel; // Variável inválida
```

## Tipos de Dados

- **Primitivos:** `string`, `number`, `boolean`, `null`, `undefined`, `symbol`.

```
let nome = "Gustavo"; // string
let idade = 25; // number
let ativo = true; // boolean
let vazio = null; // null
let indefinido; // undefined
```

## Manipulação de Strings

```
let str = "Gustavo"; // string
let tamanho = str.length; // 7
let maiusculas = str.toUpperCase(); // "GUSTAVO"
let minusculas = str.toLowerCase(); // "gustavo"
let primeiraLetra = str[0]; // "G"
let ultimaLetra = str[str.length - 1]; // "o"
let posicao = str.indexOf("t"); // 2
let substituir = str.replace("Gustavo", "Clay"); // "Clay"
```

# Operadores

- + soma ou concatenação
- subtração
- \* multiplicação
- / divisão
- % resto da divisão
- ++ incremento
- decremento
- = atribuição
- += soma ou concatenação e atribuição
- = subtração e atribuição
- \*= multiplicação e atribuição
- /= divisão e atribuição
- %= resto e atribuição

- **Aritméticos:** `+`, `-`, `*`, `/`, `%`.

```
let soma = 10 + 5;  
let subtracao = 10 - 5;  
let multiplicacao = 10 * 5;  
let divisao = 10 / 5;  
let resto = 10 % 3;
```

- **Comparação:** `==` , `===` , `!=` , `!==` , `>` , `<` , `>=` , `<=` .

```
let igual = 10 == "10"; // true
let estritamenteIgual = 10 === "10"; // false
let diferente = 10 != "10"; // false
let estritamenteDiferente = 10 !== "10"; // true
let maior = 10 > 5; // true
let menor = 10 < 5; // false
let maiorIgual = 10 >= 10; // true
let menorIgual = 10 <= 5; // false
```

# Estruturas de Controle

## Condicionais

- **if, else if, else:**

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



- **switch:**

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

## Laços de Repetição

- **for:**

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

- **for...of:**

```
let frutas = ["Maçã", "Banana", "Laranja"];  
for (let fruta of frutas) {  
  console.log(fruta);  
}
```

- **while:**

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

- **do...while:**

```
let i = 0;
do {
  console.log(i);
  i++;
} while (i < 5);
```

# Funções

- Declarar funções:

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

- Funções anônimas e arrow functions:

```
const soma = function(a, b) {  
    return a + b;  
};  
console.log(soma(5, 3));  
  
const multiplicacao = (a, b) => a * b;  
console.log(multiplicacao(5, 3));
```

# Arrays

- Declarar e manipular arrays:

```
let frutas = ["Maçã", "Banana", "Laranja"];  
console.log(frutas[0]); // Maçã  
frutas.push("Uva");  
console.log(frutas.length); // 4
```

## Principais Métodos de Arrays

### push()

Adiciona um elemento ao final do array.

```
frutas.push("Morango");  
console.log(frutas); // ["Maçã", "Banana", "Laranja", "Morango"]
```

### pop()

Remove o último elemento do array.

```
frutas.pop();  
console.log(frutas); // ["Maçã", "Banana", "Laranja"]
```

## shift()

Remove o primeiro elemento do array.

```
frutas.shift();  
console.log(frutas); // ["Banana", "Laranja"]
```

## unshift()

Adiciona um elemento no início do array.

```
frutas.unshift("Uva");  
console.log(frutas); // ["Uva", "Banana", "Laranja"]
```

## slice()

Retorna uma cópia de parte do array.

```
let citricas = frutas.slice(1, 3);  
console.log(citricas); // ["Banana", "Laranja"]
```

## splice()

Adiciona ou remove elementos em qualquer posição.

```
frutas.splice(1, 1); // Remove 1 elemento a partir do índice 1  
console.log(frutas); // ["Uva", "Laranja"]
```



## forEach()

Executa uma função para cada elemento do array.

```
frutas.forEach((fruta) => console.log(fruta));
```

## filter()

Cria um novo array com elementos que passam em um teste.

```
let frutasComA = frutas.filter((fruta) => fruta.includes("a"));  
console.log(frutasComA); // ["Laranja"]
```

## map()

Cria um novo array com os resultados de uma função aplicada a cada elemento.

```
let frutasMaiusculas = frutas.map((fruta) => fruta.toUpperCase());  
console.log(frutasMaiusculas); // ["UVA", "LARANJA"]
```

## reduce()

Reduz o array a um único valor (soma, média, etc.).

```
let numeros = [1, 2, 3, 4];  
let soma = numeros.reduce((total, num) => total + num, 0);  
console.log(soma); // 10
```

# Objetos

- Declarar e manipular objetos:

```
let pessoa = {  
  nome: "Jorge",  
  idade: 25,  
  cidade: "Brasília"  
};  
console.log(pessoa.nome); // Jorge  
pessoa.idade = 26;  
console.log(pessoa.idade); // 26
```

## Desestruturação de Objetos

- Atribuir valores de propriedades a variáveis:

```
let pessoa = {  
  nome: "Jorge",  
  idade: 25,  
  cidade: "Brasília"  
};  
  
let { nome, idade } = pessoa;  
  
console.log(nome); // Jorge  
console.log(idade); // 26
```

## Propagação de Objetos

- Combinar propriedades de objetos:

```
let pessoa = {  
  nome: "Jorge",  
  idade: 25  
};  
  
let endereco = {  
  cidade: "Brasília",  
  estado: "DF"  
};  
  
let pessoaCompleta = { ...pessoa, ...endereco };  
  
console.log(pessoaCompleta);
```

# Modulos

- Exportar funções, variáveis e objetos:

```
export const soma = (a, b) => a + b;  
export const PI = 3.141592653589793;
```

- Importar funções, variáveis e objetos:

```
import { soma, PI } from "../matematica.js";  
console.log(soma(5, 3));
```

## Exercícios

1. Crie uma função que receba dois números e retorne a soma deles.
2. Crie um array de frutas e adicione uma nova fruta ao final.
3. Crie um objeto representando uma pessoa com nome, idade e cidade, e altere a idade dessa pessoa.
4. Utilize o método `map` para criar um novo array com os nomes das frutas em maiúsculas.
5. Utilize o método `filter` para criar um novo array apenas com as frutas que contêm a letra "a".



6. Utilize o método `reduce` para calcular a soma de um array de números.
7. Crie um objeto representando um carro com propriedades como marca, modelo e ano, e utilize a desestruturação para extrair essas propriedades em variáveis.
8. Crie dois objetos, carro e proprietário e combine utilizando a propagação de objetos.
9. Crie um módulo que exporte uma função de soma e uma constante PI, e importe esses valores em outro arquivo.
10. Crie um programa que receba um número e imprima a tabuada desse número.