



# Fundamentos de JavaScript



## O que é JavaScript?

JavaScript é a linguagem que adiciona comportamento e interatividade a páginas web. Ele roda no navegador e permite responder a cliques, preencher dados, alterar o conteúdo da tela etc.



## Conceitos Básicos - Declaração de variáveis

```
let nome = "João";  
const idade = 25;
```

```
// comentário, vamos abrir em https://runjs.app/play  
// diferença de let pra const  
// ambas são limitadas ao bloco { ... } onde são definidas  
  
// const: uma vez atribuído, valor não pode ser mudado (referência não muda).  
  
// let: Permite reatribuição do valor. Ele pode mudar ao longo do tempo.
```



# Const

objetos e arrays declarados com **const** podem ter seus conteúdos modificados, pois o que é constante é a referência, não o conteúdo.

```
const nome = "Ana";  
  
// nome = "João"; // Erro!  
  
const usuario = { nome: "Ana" };  
usuario.nome = "João"; // OK (modificou o conteúdo do objeto)
```

# Comparativo

Característica	let	const
Pode ser reatribuída?	Sim	Não
Escopo	De bloco	De bloco
Pode declarar sem valor inicial?	Sim	Não (obrigatório inicializar)
Pode alterar conteúdo interno (objetos/arrays)?	Sim	Sim

## Boas Práticas

>Use const por padrão; >Use let apenas quando precisar reatribuir o valor depois.

# Tipos de Dados - Number



Number, String, Boolean, Array, Object, null, undefined

```
// NUMBER  
let idade = 30;  
  
let preco = 19.99;  
  
let temperatura = -5;  
  
//console.log() é importante pra ver um valor no console  
  
//typeof (vê o tipo de uma variável)  
  
console.log("idade", typeof idade); // resultado: "idade" "number"  
console.log("preco", typeof preco); // resultado: "preco" "number"  
console.log("temperatura", typeof temperatura); // resultado: "temperatura" "number"
```

# Tipos de Dados - String

 Number, **String**, Boolean, Array, Object, null, undefined

```
// STRING  
let nome = "Ana";  
  
let saudacao = 'Olá, mundo!';  
  
let frase = `Meu nome é ${nome}`; // template string  
  
console.log(typeof nome); // "string"  
  
console.log(frase); // " Meu nome é Ana"
```

# Tipos de Dados - Boolean

 Number, String, **Boolean**, Array, Object, null, undefined

```
// BOOLEAN  
let ligado = true;  
  
let temSaldo = false;  
  
console.log(typeof ligado); // "boolean"
```

# Tipos de Dados - Array


 Number, String, Boolean, **Array**, Object, null, undefined

É um tipo especial de objeto que armazena coleções ordenadas.

```
// ARRAY  
let frutas = ["maçã", "banana", "laranja"];  
  
let numeros = [10, 20, 30, 40];  
  
console.log(frutas[1]); // "banana"  
  
console.log(typeof frutas); // "object"
```



# Tipos de Dados - Object

 Number, String, Boolean, Array, **Object**, null, undefined

Estrutura de chave-valor.

```
// OBJECT
let pessoa = {
  nome: "Carlos",
  idade: 28,
  ativo: true
};

console.log(pessoa.nome); // "Carlos"
console.log(typeof pessoa); // "object"
```

# Tipos de Dados - Null

 Number, String, Boolean, Array, Object, **null**, undefined

Representa ausência intencional de valor.

```
// NULL  
let resposta = null;  
  
console.log(typeof resposta); // "object"  
  
// (isso é um bug histórico do JavaScript)
```

# Condicionais

Use `if` para especificar um bloco de código a ser executado se uma condição específica for verdadeira.

Use `else` para especificar um bloco de código a ser executado se essa mesma condição for falsa.

Use `else if` para especificar uma nova condição a ser testada, caso a primeira condição seja falsa.

Use `switch` para especificar vários blocos alternativos de código a serem executados.

```
if (idade > 18) {  
    console.log("Maior de idade");  
} else if (idade > 12) {  
    console.log("Adolescente");  
} else {  
    console.log("Criança");  
}
```

```
switch(sexo) {  
    case "M":  
        console.log("masculino");  
        break;  
    case "F":  
        console.log("feminino");  
        break;  
    default:  
        console.log("indefinido");  
}
```

# Laços

**for** – percorre um bloco de código um número específico de vezes.

**for/in** – percorre as propriedades de um objeto.

**for/of** – percorre os valores de um objeto iterável.

**while** – percorre um bloco de código enquanto uma condição especificada for verdadeira.

**do/while** – também percorre um bloco de código enquanto uma condição especificada for verdadeira, mas garante que o bloco seja executado pelo menos uma vez.

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

# Laços

For..in = itera sobre as propriedade de um objeto.

```
const person = {name_1:"John", name_2:"Doe", age:25};  
let text = "";
```

```
for (let x in person) {  
  text += person[x];  
}
```

```
console.log(text)
```

```
// "JonDoe25"
```

For..of = itera valores de um iterável.

```
const cars = ["BMW", "Volvo", "Mini"];
```

```
let text = "";
```

```
for (let x of cars) {
```

```
  text += x;
```

```
}
```

For..each = consegue rodar uma função pra cada elemento

```
const numbers = [45, 4, 9, 16, 25];  
  
numbers.forEach(myFunction);  
  
function myFunction(value, index, array) {  
    console.log(value, index, array);  
  
};
```

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

O que será que está acontecendo aqui?

```
document.getElementById("titulo").innerText = "Novo título";
```

Seleciona o elemento de id Titulo,  
Muda o texto dele para "Novo Titulo"



O que será que está acontecendo aqui?

```
document.querySelector("button").addEventListener("click", function() { alert("Clicou!");});
```

Elemento de tipo Button,

Cria um escutador de eventos,

Quando clicar, roda função que alerta Clicou

# Exercício pra fixação – nosso site

// comentário

```
const coresNeon = ['#00ffe7', '#00bcd4', '#00ffc3', '#ff00cc', '#39ff14', '#f72585',  
'#3f51b5'];
```

// Referência ao título

```
const titulo = document.getElementById('titulo');
```

// Muda para uma cor aleatória ao clicar no título

```
titulo.addEventListener('click', function () {  
  const corAleatoria = coresNeon[Math.floor(Math.random() * coresNeon.length)];  
  titulo.style.color = corAleatoria;  
});
```

# Exercício pra fixação – nosso site

// Evento de envio do formulário

```
document.getElementById('formulario-contato').addEventListener('submit',  
function (event) {  
    event.preventDefault();  
  
    const termosAceitos = document.getElementById('termos').checked;  
  
    if (!termosAceitos) {  
        alert('Você precisa aceitar os termos de uso para enviar o formulário.');        return;  
    }  
}
```

# Exercício pra fixação – nosso site

```
// Coleta os dados do formulário
```

```
const nome = document.getElementById('nome').value;
```

```
const email = document.getElementById('email').value;
```

```
const assunto = document.getElementById('assunto').value;
```

```
const mensagem = document.getElementById('mensagem').value;
```

```
// Exibe no console os dados antes do reset
```

```
// console.log('📩 Dados enviados:');
```

```
// console.log('Nome:', nome);
```

```
// console.log('Email:', email);
```

```
// console.log('Assunto:', assunto);
```

```
// console.log('Mensagem:', mensagem);
```

```
// console.log('Aceitou os termos:', termosAceitos);
```

# Exercício pra fixação – nosso site

```
// Mostra também em um alert
```

```
alert(
```

```
"✅ Formulário enviado com sucesso!\n\n" +
```

```
"✉ Nome: " + nome + "\n" +
```

```
"📧 Email: " + email + "\n" +
```

```
"📌 Assunto: " + assunto + "\n" +
```

```
"📄 Mensagem: " + mensagem + "\n" +
```

```
"📋 Termos aceitos: " + (termosAceitos ? "Sim" : "Não")
```

```
);
```

```
this.reset(); // Limpa o formulário após envio
```

```
));
```

```
// Exibe o ano atual no rodapé
```

```
document.getElementById('ano').textContent = new  
Date().getFullYear();
```

# Obrigado!



    @fpftech.educacional