

# O KI DO DESENVOLVIMENTO WEB APRENDENDO JAVASCRIPT COM GOKU



FELIPE CESAR RODRIGUES

**Olá, guerreiros! Eu sou o Goku, e hoje vamos descobrir juntos o incrível poder do JavaScript. JavaScript é uma linguagem de programação super poderosa usada para tornar sites interativos e dinâmicos. Se você quer criar páginas web que brilham como uma Genki Dama, JavaScript é o seu aliado. Vamos começar essa jornada épica!**





**ADICIONANDO  
JAVASCRIPT AO HTML**



# DENTRO DA TAG <SCRIPT> NO CABEÇALHO OU CORPO

Você pode escrever JavaScript diretamente dentro da tag <script> no seu HTML. Vamos ver um exemplo simples:

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <title>Exemplo de JavaScript</title>
6   <script>
7     // Este comando mostra uma mensagem no console do navegador
8     console.log('Olá, mundo!');
9   </script>
10 </head>
11 <body>
12   <h1>Bem-vindo ao meu site!</h1>
13 </body>
14 </html>
```

# ADICIONANDO JAVASCRIPT EXTERNO

Uma maneira mais organizada é colocar seu código JavaScript em um arquivo separado e vinculá-lo ao seu HTML. Veja como:

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <title>Exemplo de JavaScript</title>
6  </head>
7  <body>
8      <h1>Bem-vindo ao meu site!</h1>
9      <!-- Importa o arquivo JavaScript -->
10     <script src="script.js"></script>
11 </body>
12 </html>
```

```
1  // Este comando mostra uma mensagem no console do navegador
2  console.log('Olá, mundo!');
```

Sempre inclua a tag **<script>** no final do **<body>** para garantir que o HTML seja carregado antes do JavaScript. Isso evita erros no seu código.



# **ENTENDO OPERADORES EM JAVASCRIPT**

---

# OPERADORES MATEMÁTICOS

Os operadores matemáticos ajudam você a fazer cálculos em JavaScript. Vamos ver alguns exemplos:

```
1  let a = 10;
2  let b = 5;
3
4  let soma = a + b; // Adição
5  let subtracao = a - b; // Subtração
6  let multiplicacao = a * b; // Multiplicação
7  let divisao = a / b; // Divisão
8  let resto = a % b; // Resto da divisão
9
10 console.log(soma); // 15
11 console.log(subtracao); // 5
12 console.log(multiplicacao); // 50
13 console.log(divisao); // 2
14 console.log(resto); // 0
```

---

# OPERADORES LÓGICOS

Os operadores lógicos são usados para comparar valores e retornar true ou false.



```
1  let x = 5;
2  let y = 10;
3
4  let and = (x < 10) && (y > 5); // AND lógico
5  let or = (x < 10) || (y < 5); // OR lógico
6  let not = !(x < 10); // NOT lógico
7
8  console.log(and); // true
9  console.log(or); // true
10 console.log(not); // false
11
```






**VARIÁVEIS - 'VAR', 'LET'  
E 'CONST'**

---

## **'VAR'**

Usada para declarar variáveis que podem ser redeclaradas e alteradas.




```
1  var nome = 'Goku';  
2  // Mostra o nome no console  
3  console.log(nome); // Goku
```

---

## 'LET'

Similar ao var, mas com escopo de bloco, ou seja, a variável só existe dentro do bloco onde foi declarada.



```
1  let poder = 9000;  
2  // Mostra o poder no console  
3  console.log(poder); // 9000
```

---

## 'CONST'

Usada para declarar constantes, ou seja, valores que não podem ser alterados depois de atribuídos.



```
1  const planeta = 'Terra';  
2  // Mostra o planeta no console  
3  console.log(planeta); // Terra
```



**ALERTANDO COM  
'ALERT'**

---

# USANDO 'ALERT'

O comando alert é usado para mostrar mensagens ao usuário. Vamos ver como ele funciona:

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <title>Exemplo de Alert</title>
6      <script>
7          // Mostra uma mensagem quando a página é carregada
8          alert('Olá, guerreiro! Bem-vindo ao site!');
9      </script>
10 </head>
11 <body>
12     <h1>Bem-vindo ao meu site!</h1>
13 </body>
14 </html>
```



**EVENTOS DE CLIQUE**

---

# ➡ ADICIONANDO EVENTOS DE CLIQUE

Os eventos de clique permitem que você execute funções quando o usuário clica em um elemento.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <title>Exemplo de Evento de Clique</title>
6      <script>
7          // Função que mostra uma mensagem de alerta
8          function saudacao() {
9              alert('Kamehameha!');
10         }
11     </script>
12 </head>
13 <body>
14     <h1>Bem-vindo ao meu site!</h1>
15     <!-- Botão que chama a função saudacao quando clicado -->
16     <button onclick="saudacao()">Clique em mim</button>
17 </body>
18 </html>
```

---



---

# USANDO 'ADDEVENTLISTENER

Outra maneira de adicionar eventos é usando o método `addEventListener`.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <title>Exemplo de addEventListener</title>
6  </head>
7  <body>
8      <h1>Bem-vindo ao meu site!</h1>
9      <!-- Botão com ID para ser referenciado no JavaScript -->
10     <button id="meuBotao">Clique em mim</button>
11     <script>
12         // Adiciona um evento de clique ao botão
13         document.getElementById('meuBotao').addEventListener('click', function() {
14             alert('Kamehameha!');
15         });
16     </script>
17 </body>
18 </html>
```

---



**MANIPULANDO O DOM**

---

# O QUE É DOM?

DOM significa Document Object Model. É a estrutura que o navegador usa para representar o HTML do seu site. Com JavaScript, podemos manipular o DOM para alterar dinamicamente o conteúdo e o estilo das nossas páginas.

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <title>Exemplo de Manipulação do DOM</title>
6   <script>
7     function mudarTexto() {
8       // Seleciona o elemento com o ID 'meuParagrafo' e altera seu texto
9       document.getElementById('meuParagrafo').innerText = 'Texto alterado!';
10    }
11  </script>
12 </head>
13 <body>
14   <h1>Bem-vindo ao meu site!</h1>
15   <p id="meuParagrafo">Texto original</p>
16   <!-- Botão que chama a função mudarTexto quando clicado -->
17   <button onclick="mudarTexto()">Mudar Texto</button>
18 </body>
19 </html>
```

---



# **FUNÇÕES EM JAVASCRIPT**

---

# O QUE SÃO FUNÇÕES?

Funções são blocos de código que realizam uma tarefa específica. Elas ajudam a organizar e reutilizar o código.

```
1 // Definindo uma função chamada saudacao
2 function saudacao(nome) {
3     return `Olá, ${nome}!`;
4 }
5
6 // Chamando a função saudacao e mostrando o resultado no console
7 console.log(saudacao('Goku')); // Olá, Goku!
```

---

# FUNÇÕES ANÔNIMAS E ARROW FUNCTIONS

Funções anônimas não têm nome, e arrow functions são uma maneira mais concisa de escrever funções.

```
1 // Função anônima atribuída a uma variável
2 const minhaFuncao = function() {
3     console.log('Esta é uma função anônima.');

---


```

# **AGRADECIMENTOS**

---



# OBRIGADO POR LER ATÉ AQUI

O conteúdo desse Ebook foi gerado utilizando inteligência artificial, e diagramado por um humano.

O Ebook foi feito para fins didáticos, afim de testar a ferramenta de inteligência artificial, Chat GPT, não foi realizado uma revisão sobre o conteúdo, portanto talvez haja erros gerados pela IA.



<https://github.com/Kiy0p0N>

<https://www.linkedin.com/in/felipe-cesar-rodrigues>

[https://www.instagram.com/\\_feliippe\\_/](https://www.instagram.com/_feliippe_/)