

JS



JAVASCRIPT NO MODO SÁBIO

Daniel Thomaz

01

Despertando o Chakra do JavaScript

Por que aprender JavaScript?



Despertando o Chakra do JavaScript

Por que aprender JavaScript?

JavaScript é uma linguagem de programação que dá movimento, lógica e inteligência às páginas da web. Sem ele, um site é como um pergaminho: bonito, mas imóvel. Com JavaScript, tudo ganha vida — botões funcionam, menus se abrem, mensagens aparecem, jogos rodam e formulários são validados em tempo real.

É uma linguagem **interpretada**, o que significa que ela roda diretamente no navegador, sem necessidade de instalação ou compilação. Basta abrir um site, e o JavaScript está ali, funcionando nos bastidores.

Além disso, ele se conecta com o **HTML** (estrutura da página) e o **CSS** (estilo visual), formando o trio essencial da web. Se o HTML é o corpo, o CSS é a roupa e o JavaScript é o cérebro e os reflexos.



Despertando o Chakra do JavaScript

...continuando.

Exemplo real e prático: Imagine que você preenche um campo de nome e clica em “Enviar”. Ao clicar, aparece a mensagem “Olá, [seu nome]!” sem que a página seja recarregada. Isso é feito com JavaScript.



JavaS - Ninja

```
<!-- Campo de texto para o usuário digitar o nome -->
<input id="nome" placeholder="Digite seu nome">

<!-- Botão que executa a função 'saudar' quando clicado -->
<button onclick="saudar()">Enviar</button>

<!-- Parágrafo onde será exibida a mensagem de saudação -->
<p id="mensagem"></p>

<script>
    // Função que será chamada ao clicar no botão
    function saudar() {
        // Pega o valor digitado no campo de texto
        let nome = document.getElementById("nome").value;

        // Altera o conteúdo do parágrafo para mostrar a saudação
        document.getElementById("mensagem").innerText = "Olá, " + nome + "!";
    }
</script>
```



02

Fundamentos Ninjas: As Armas Básicas

- Variáveis: Armazenando Informação
- Tipos de Dados
- Condições: Tomando Decisões
- Funções: Jutsus Reutilizáveis



Fundamentos Ninjas: As Armas Básicas

Variáveis: Armazenando Informação

Imagine que você é um ninja em missão. Para concluir tarefas com precisão, você precisa carregar informações: nomes, números, estados... No JavaScript, essas informações são armazenadas em variáveis. Variáveis são como compartimentos da sua bolsa ninja: você guarda um item com um nome, e pode usá-lo.

Elas são essenciais porque quase tudo em programação depende de armazenar e manipular dados. Se você quer mostrar o nome de um usuário, calcular um valor ou controlar um botão, precisa de uma variável. Como criar uma variável Existem duas formas principais no JavaScript moderno:

Como criar uma variável?

No JavaScript moderno, usamos duas palavras principais para declarar variáveis:

- `let` → quando o valor pode mudar ao longo do tempo.
- `const` → quando o valor é fixo e não deve ser alterado.



Fundamentos Ninjas: As Armas Básicas

... *continuação*



JavaS - Ninja

```
let nome = "Naruto";           // Pode mudar depois  
const aldeia = "Folha";       // Nunca muda  
  
console.log(nome);            // Exibe "Naruto"  
console.log(aldeia);          // Exibe "Folha"
```

- Usamos `let` para dados dinâmicos, como o nome de um usuário que pode ser trocado.
- Usamos `const` para dados que não devem mudar, como o nome da aldeia ou o número de dias de uma semana.

Mudando valores com `let`:



JavaS - Ninja

```
let chakra = 100;  
chakra = chakra - 20; // Usou um jutsu e gastou chakra  
  
console.log(chakra); // Resultado: 80
```



Fundamentos Ninjas: As Armas Básicas

Tipos de Dados

No mundo ninja do JavaScript, conhecer os tipos de dados é escolher a arma certa para cada missão. Texto (**string**) guarda nomes e mensagens; números servem para cálculos e contagens; booleanos (true/false) são sinais que ajudam a tomar decisões no código.

Saber o tipo evita erros e garante que seu jutsu (programa) funcione com precisão. É a base para um código forte, rápido e organizado, pronto para qualquer batalha.

No mundo ninja do JavaScript, entender os tipos de dados é fundamental para manipular corretamente a informação. Cada tipo representa um formato diferente de valor que o código pode usar.

- Texto (String): são palavras ou frases, sempre entre aspas.

Exemplo:



JavaS - Ninja

```
let nome = "Kakashi";
```



Fundamentos Ninjas: As Armas Básicas

Tipos de Dados

- Número (Number): valores numéricos, usados para cálculos ou contagens.

Exemplo:



JavaS - Ninja

```
let idade = 42;
```



JavaS - Ninja

```
let estaAcordado = true;
```

Além desses, existem outros tipos como:

- **Undefined**: quando uma variável não tem valor definido ainda.
- **Null**: indica ausência intencional de valor.
- **Objeto e Array**: tipos mais complexos que guardam coleções ou estruturas de dados.



Fundamentos Ninjas: As Armas Básicas

Condições: Tomando Decisões

Todo ninja precisa decidir rápido: atacar ou recuar? Ativar o Modo Sábio ou poupar chakra? No JavaScript, quem faz esse papel é a estrutura condicional.

Com o `if`, você cria caminhos diferentes para o código seguir, dependendo da situação.



JavaS - Ninja

```
let chakra = 80; // Define o nível de chakra do ninja

if (chakra > 50) { // Se o chakra for maior que 50...
  console.log("Modo Sábio ativado!"); // ...ativa o Modo Sábio
}
```

Se a condição for verdadeira (`chakra > 50`), o código dentro do bloco `{}` será executado. Se não for, o JavaScript ignora esse trecho.



Fundamentos Ninjas: As Armas Básicas

Condições: Tomando Decisões

Você também pode usar `else` e `else if` para cobrir outros cenários:



JavaS - Ninja

```
if (chakra > 50) { // Se o chakra for maior que 50...
  console.log("Modo Sábio ativado!"); // ...ativa o Modo Sábio
} else if (chakra > 20) { // Senão, se for maior que 20...
  console.log("Use o Rasengan com cautela."); // ...permite usar jutsus leves
} else { // Senão, se o chakra for 20 ou menos...
  console.log("Você precisa descansar."); // ...recomenda repouso
}
```

Essas estruturas permitem que o programa reaja ao que está acontecendo — exatamente como um ninja que analisa a batalha antes de agir.

Fundamentos Ninjas: As Armas Básicas

Funções: Jutsus Reutilizáveis

Todo ninja tem seus jutsus preferidos — técnicas que ele pode usar várias vezes. No JavaScript, essas técnicas são chamadas de **funções**. Com elas, você escreve um bloco de código uma vez e pode reutilizá-lo sempre que quiser, apenas chamando pelo nome.



JavaS - Ninja

```
function cumprimentar(nome) {  
    // Esta função recebe um nome como parâmetro  
    console.log("Olá, " + nome); // Exibe uma saudação personalizada no console  
}  
  
cumprimentar("Hinata"); // Chama a função e envia "Hinata" como nome
```

A palavra-chave ***function*** cria a função. O nome da função (neste caso, `cumprimentar`) é como o nome do jutsu. Entre parênteses, você define o parâmetro — uma informação que a função precisa para funcionar. Dentro das chaves ***{}***, fica o que será executado quando o jutsu for ativado.

Fundamentos Ninjas: As Armas Básicas

Funções: Jutsus Reutilizáveis

Você pode criar funções para qualquer coisa:



JavaS - Ninja

```
function somar(a, b) {  
    // Esta função soma dois valores  
    return a + b; // Retorna o resultado da soma  
}  
  
let resultado = somar(5, 7); // Executa a função com os valores 5 e 7  
console.log(resultado); // Mostra 12 no console
```

Funções ajudam a organizar o código e evitam repetições. Em vez de escrever o mesmo comando várias vezes, você cria um jutsu único e chama quando precisar.

Encerrando o Treinamento

- Com as variáveis, os tipos de dados, as condições e as funções, você já tem as armas básicas de um verdadeiro ninja do código. Esses fundamentos são como os primeiros jutsus aprendidos na Academia Ninja: simples, mas poderosos. Dominar essa base vai te preparar para enfrentar missões mais complexas com confiança e precisão.
- Agora que você já controla essas técnicas, está pronto para combiná-las com o HTML e começar a criar interações reais com o usuário. No próximo capítulo, vamos unir forças com o HTML e partir para ações conjuntas — como em uma missão em equipe da Vila da Folha.



03

HTML + JavaScript: Missão em Dupla

- **Respondendo ao Usuário:
Primeiras Missões Interativas**
- **Alterando Elementos da Página:
Manipulando o DOM**



HTML + JavaScript: Missão em Dupla

Respondendo ao Usuário: Primeiras Missões Interativas

JavaScript sozinho já é forte, mas sua verdadeira força desperta quando ele se junta ao HTML. É como uma missão em dupla entre dois ninjas com habilidades complementares: o HTML constrói a estrutura da página, enquanto o JavaScript dá vida e interação a ela.

Exemplo: Mostrar nome na tela ao clicar num botão

Vamos ver um exemplo simples de como essa parceria funciona:

```
... JavaS - Ninja

<!-- Campo para digitar o nome -->
<input id="nome" placeholder="Digite seu nome">

<!-- Botão que ativa a função -->
<button onclick="mostrar()">Enviar</button>

<!-- Área onde o resultado será exibido -->
<p id="res"></p>

<script>
  function mostrar() {
    // Pega o valor digitado no campo com id "nome"
    let nome = document.getElementById("nome").value;

    // Altera o conteúdo do parágrafo com id "res"
    document.getElementById("res").innerText = "Olá, " + nome;
  }
</script>
```



HTML + JavaScript: Missão em Dupla

Respondendo ao Usuário: Primeiras Missões Interativas

Explicação ninja do código:

1. O HTML cria três elementos: um campo de texto, um botão e um parágrafo.
2. O botão usa o atributo `onclick` para chamar a função `mostrar()` quando clicado.
3. Dentro da função, o JavaScript acessa o valor digitado pelo usuário (`.value`) e muda o texto da `<p>` com o método `.innerText`.

Esse é o básico da interação entre HTML e JavaScript. Você coleta informação com o HTML e responde com ações usando o JS.



HTML + JavaScript: Missão em Dupla

Respondendo ao Usuário: Primeiras Missões Interativas

Quando um usuário interage com a página — clica em algo, digita um texto ou move o mouse — o JavaScript pode responder a esses movimentos como um ninja atento a cada passo do inimigo. Para isso, usamos **eventos**.

Exemplo 1: Detectando um clique.

```
● ● ● JavaS - Ninja

<button onclick="saudar()">Clique para ser saudado</button>

<script>
  function saudar() {
    // Executa esta ação quando o botão for clicado
    alert("Saudações, shinobi!");
  }
</script>
```

O que está acontecendo?

- O botão HTML chama a função `saudar()` quando é clicado (`onclick`).
- A função exibe uma mensagem com `alert()` — um jutsu direto e simples.



HTML + JavaScript: Missão em Dupla

Respondendo ao Usuário: Primeiras Missões Interativas

Exemplo 2: Lendo o que o usuário digitou.

```
JavaS - Ninja

<input id="clan" placeholder="Digite seu clã">
<button onclick="identificar()">Enviar</button>

<script>
  function identificar() {
    // Captura o valor digitado no input
    let clan = document.getElementById("clan").value;

    // Mostra uma saudação personalizada
    alert("Bem-vindo, ninja do clã " + clan + "!");
  }
</script>
```

O que esse jutsu faz?

- O campo `<input>` permite que o usuário digite o nome do clã.
- O botão ativa a função `identificar()`.
- A função pega o valor com `.value` e mostra uma mensagem com o nome digitado.

Esses são os primeiros passos para criar interfaces interativas. Com eles, você começa a sentir o poder de um sistema que reage ao usuário — como um shinobi que lê os movimentos do oponente.



HTML + JavaScript: Missão em Dupla

Alterando Elementos da Página: Manipulando o DOM

DOM significa Document Object Model — é como se fosse o mapa da aldeia HTML que o JavaScript pode explorar, modificar e controlar. Sempre que você muda o conteúdo de uma página com JS, está manipulando o DOM.

Exemplo: Mudando o texto de um parágrafo.

```
JavaS - Ninja

<p id="mensagem">Aguardando chakra...</p>
<button onclick="ativar()">Ativar Chakra</button>

<script>
  function ativar() {
    // Altera o conteúdo do parágrafo com id "mensagem"
    document.getElementById("mensagem").innerText = "Chakra ativado!";
  }
</script>
```

Explicando a técnica:

- O `getElementById()` localiza o elemento HTML pelo ID.
- O `innerText` troca o conteúdo de texto.
- Ao clicar no botão, o parágrafo muda — sinal de que o chakra fluiu!



HTML + JavaScript: Missão em Dupla

Alterando Elementos da Página: Manipulando o DOM

Exemplo: Alterando a aparência de um elemento.

```
JavaS - Ninja

<p id="status">Status: Desativado</p>
<button onclick="mudarStatus()">Ativar</button>

<script>
  function mudarStatus() {
    let elemento = document.getElementById("status");
    elemento.innerText = "Status: Ativado";
    elemento.style.color = "green"; // Muda a cor do texto
    elemento.style.fontWeight = "bold"; // Deixa o texto em negrito
  }
</script>
```

O que aprendemos aqui:

- Além de alterar o conteúdo, também podemos mudar o estilo visual do elemento via JavaScript.
- O `.style` permite aplicar estilos diretamente, como mudar cores, tamanhos, fontes etc.

Manipular o DOM é como controlar o campo de batalha com precisão: você pode mudar o cenário, o comportamento dos elementos e guiar o fluxo da experiência do usuário.



04

Seus Primeiros Passos como Shinobi do Código

- **Interagindo com o Usuário**
- **Repetições: Dominando os Clones das Sombras**



Seus Primeiros Passos como Shinobi do Código

Interagindo com o Usuário

Para iniciar uma conversa com quem usa seu site, o JavaScript tem comandos que capturam informações e mostram mensagens. Um dos mais diretos é o `prompt()`.

Exemplo: Receber o nome do clã do usuário.

```
...  
JavaS - Ninja  
  
let resposta = prompt("Qual seu clã?"); // Pergunta ao usuário e guarda a resposta na variável  
alert("Bem-vindo, ninja do clã " + resposta + "!"); // Exibe uma saudação usando a resposta do  
usuário
```

Explicação do jutsu:

- `prompt()` exibe uma caixa de pergunta e guarda o que o usuário digitar.
- O valor digitado é armazenado na variável `resposta`.
- Em seguida, `alert()` mostra uma saudação personalizada.

Esse tipo de interação é útil para formulários, cadastros e qualquer sistema onde você precisa receber uma informação do usuário de forma rápida.

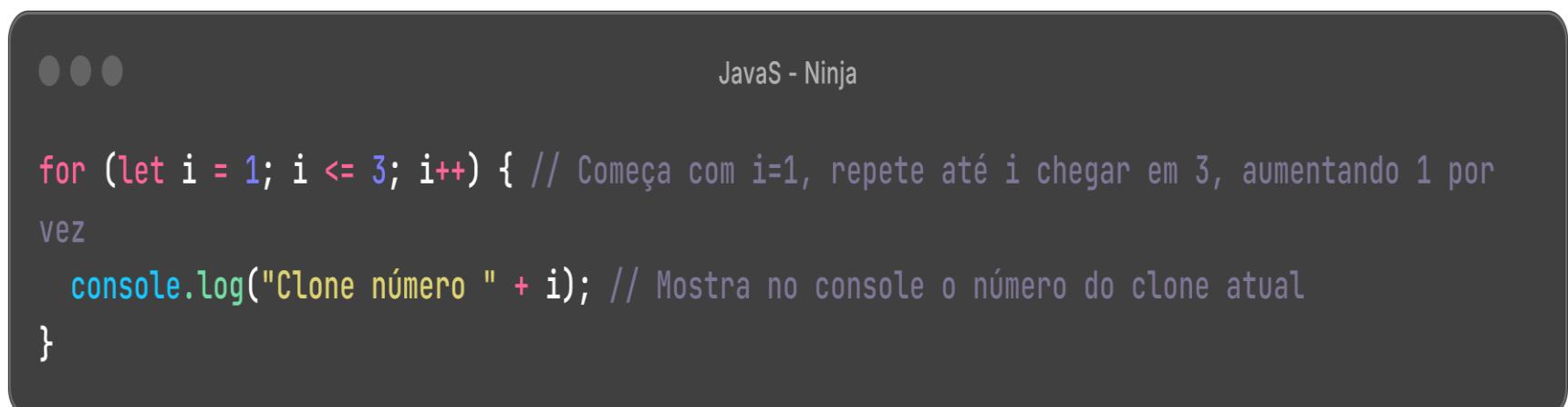


Seus Primeiros Passos como Shinobi do Código

Repetições: Dominando os Clones das Sombras

Às vezes, um único ninja não é suficiente. É hora de invocar os Clones das Sombras — ou seja, repetir ações várias vezes com um laço (*loop*). Em JavaScript, usamos estruturas como `for` para isso.

Exemplo: Mostrar três clones no console.



The screenshot shows a dark-themed code editor window. At the top, there are three small circular icons on the left and the text "JavaS - Ninja" on the right. The main area contains the following code:

```
for (let i = 1; i <= 3; i++) { // Começa com i=1, repete até i chegar em 3, aumentando 1 por vez
  console.log("Clone número " + i); // Mostra no console o número do clone atual
}
```

O que acontece aqui:

- `let i = 1` → o contador começa em 1.
- `i <= 3` → o laço continua enquanto i for menor ou igual a 3.
- `i++` → a cada volta, `i` aumenta 1.
- Dentro do laço, o `console.log()` mostra a ação repetida.

Isso é útil quando você quer repetir comandos automaticamente, como exibir listas, criar animações ou processar dados.



Seus Primeiros Passos como Shinobi do Código

Repetições: Dominando os Clones das Sombras

Exemplo: Criando um contador simples na tela.

```
... JavaS - Ninja

<p id="contador">0</p> <!-- Área onde o número será exibido -->
<button onclick="aumentar()">+1</button> <!-- Botão que ativa a função -->

<script>
let cont = 0; // Variável que guarda o número atual

function aumentar() {
    cont++; // Aumenta o valor da variável em 1
    document.getElementById("contador").innerText = cont; // Atualiza o conteúdo do parágrafo
}
</script>
```

Explicação da técnica:

- *cont* começa em 0.
- Toda vez que o botão é clicado, a função *aumentar()* é chamada.
- O valor aumenta e é exibido na tela usando *innerText*.



05

AGRADECIMENTO

JS

OBRIGADO POR LER ATÉ AQUI!

Esse Ebook foi gerado por IA, e diagramado por um humano.

Este eBook foi criado como uma forma prática e direta de ajudar iniciantes a entenderem e colocarem em prática os principais conceitos do JavaScript. Pense nele como uma **cola ninja** — sempre à mão quando precisar lembrar como usar uma variável, criar uma função ou fazer a página reagir com sabedoria.



<https://github.com/dan-th1>

Continue praticando, testando e melhorando seu código. Com treino constante, você logo será um verdadeiro mestre do JavaScript no Modo Sábio.

👉 Que o chakra do código esteja com você!