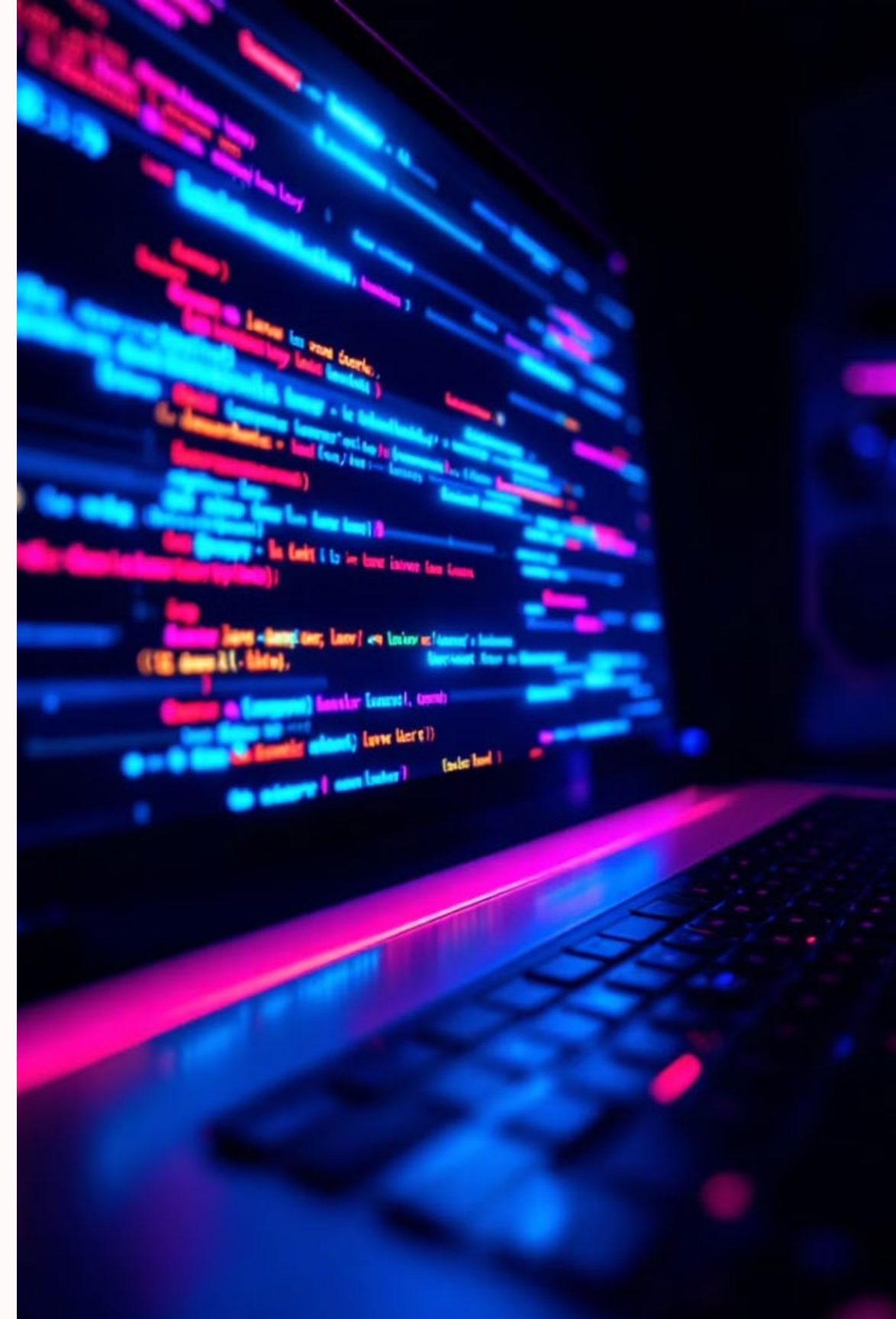




JavaScript: Uma Introdução Completa

Bem-vindos ao mundo do JavaScript! Esta apresentação guiará você pelos conceitos essenciais para começar a programar na web. Prepare-se para dominar a linguagem que dá vida aos sites.





O Que É JavaScript e Por Que É Essencial no Desenvolvimento Web?

Interatividade

JavaScript permite criar páginas web interativas. Os usuários podem interagir com elementos dinâmicos.

Front-end e Back-end

JavaScript não se limita ao front-end. Com Node.js, também pode ser usado no backend.

Fundamentos Essenciais: Variáveis, Tipos de Dados e Operadores

1

Variáveis

Variáveis armazenam dados. Use **var**, **let** ou **const** para declarar.

2

Tipos de Dados

Tipos comuns incluem números, strings, booleanos e arrays.

3

Operadores

Operadores realizam ações. Exemplos: **+**, ***** e **/**.

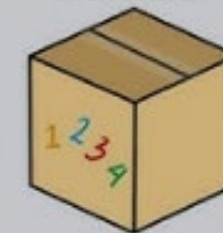
Tipos de dados



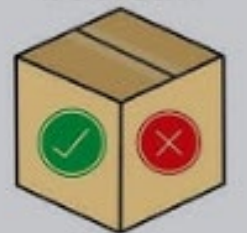
string



number



boolean



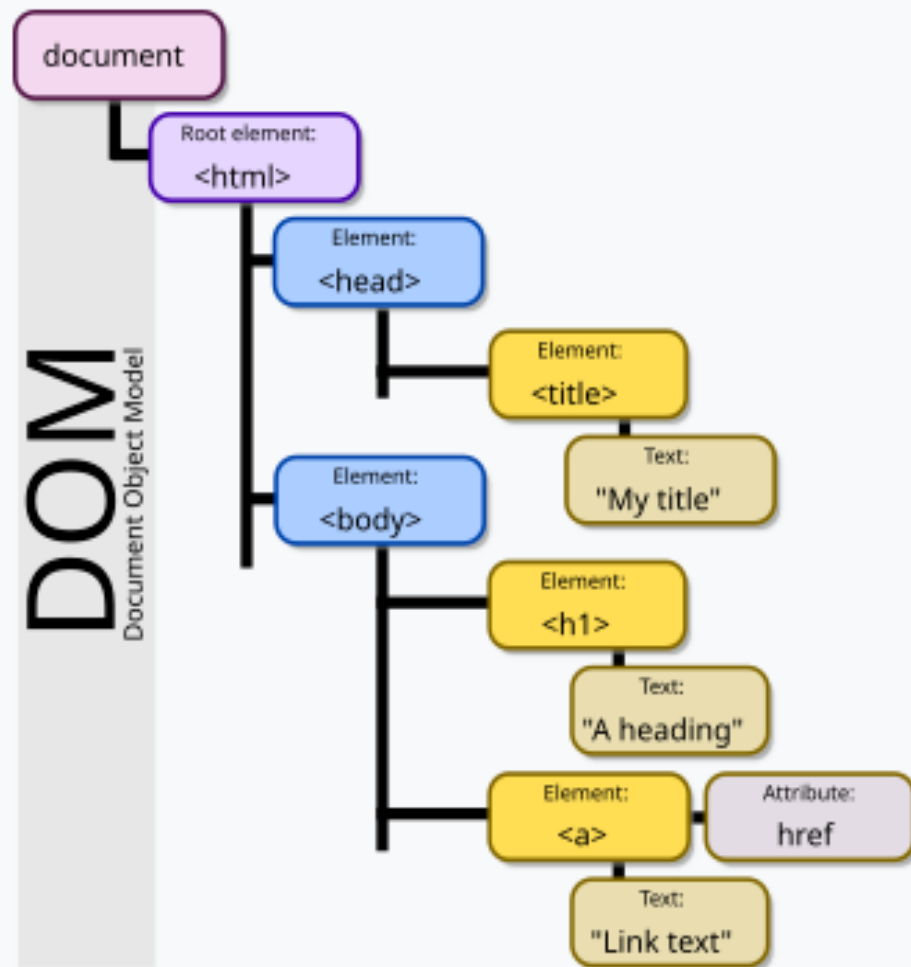
null



undefined



Document Object Model (DOM)



Example of DOM hierarchy in an HTML document

Manipulando o DOM: Dando Vida à Sua Página Web

O Que É o DOM?

O DOM é a estrutura da sua página web. JavaScript pode acessá-lo.

Acessando Elementos

Use métodos como `document.getElementById()` para selecionar elementos.

Modificando Elementos

Altere o conteúdo, atributos e estilos dos elementos.

Estilos Dinâmicos: Alterando a Aparência com JavaScript



Inline Styles

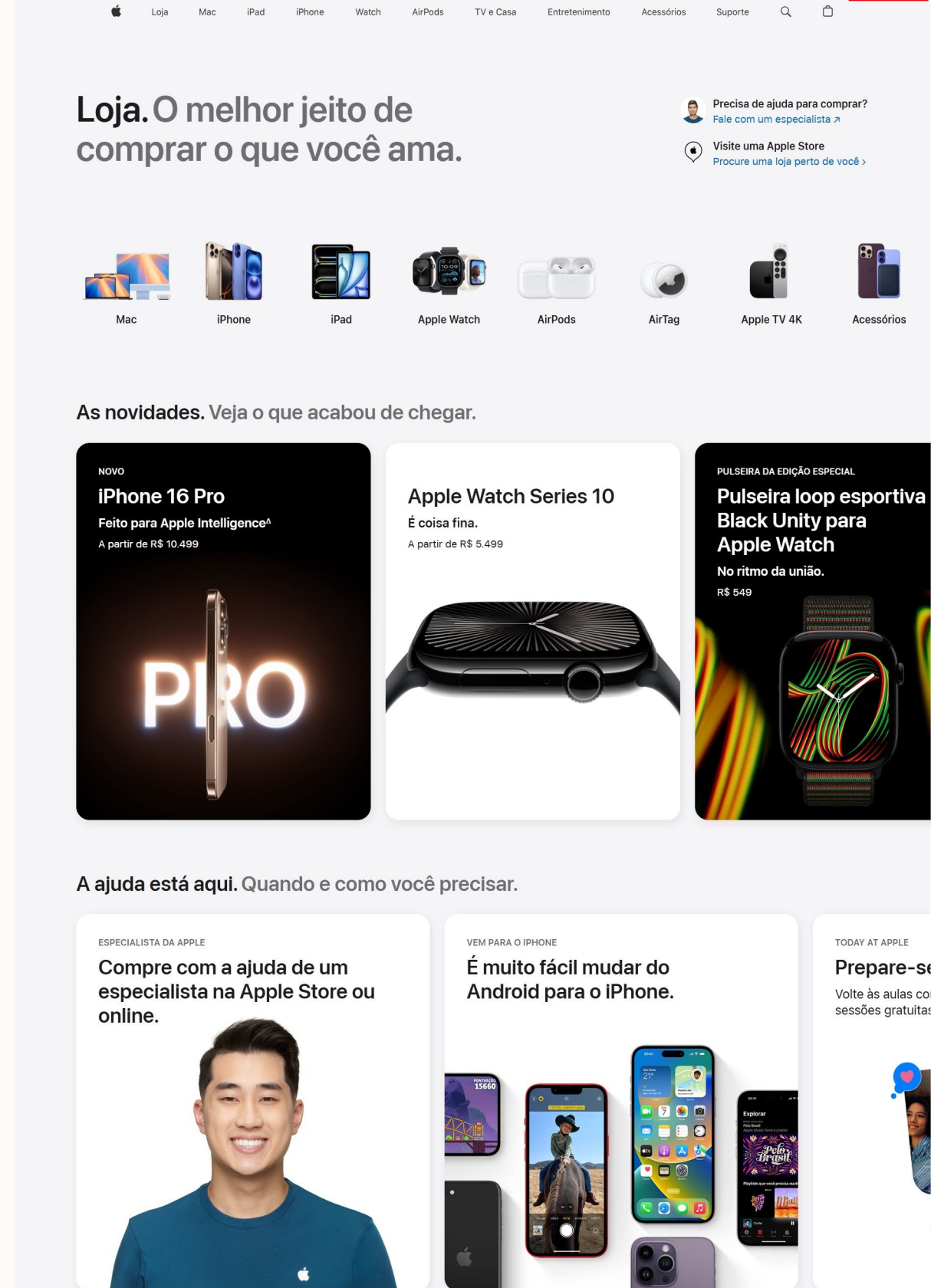
Modifique estilos diretamente nos elementos. Ex:

```
element.style.color = "red";
```



CSS Classes

Adicione ou remova classes CSS. Controle a aparência com classes.



Armazenamento Local: Persistindo Dados no Navegador com localStorage



1

O Que É localStorage?

Um espaço para armazenar dados no navegador do usuário.

2

setItem()

Use `localStorage.setItem("chave", "valor")` para salvar dados.

3

getItem()

Use `localStorage.getItem("chave")` para recuperar dados.

4

removeItem()

Use `localStorage.removeItem("chave")` para remove data.

Requisições Assíncronas: Buscando Dados com Fetch



1

O Que São Requisições?

Solicitações para um servidor. Buscam dados ou enviam informações.

2

Fetch API

Uma maneira moderna de fazer requisições. Substitui o XMLHttpRequest

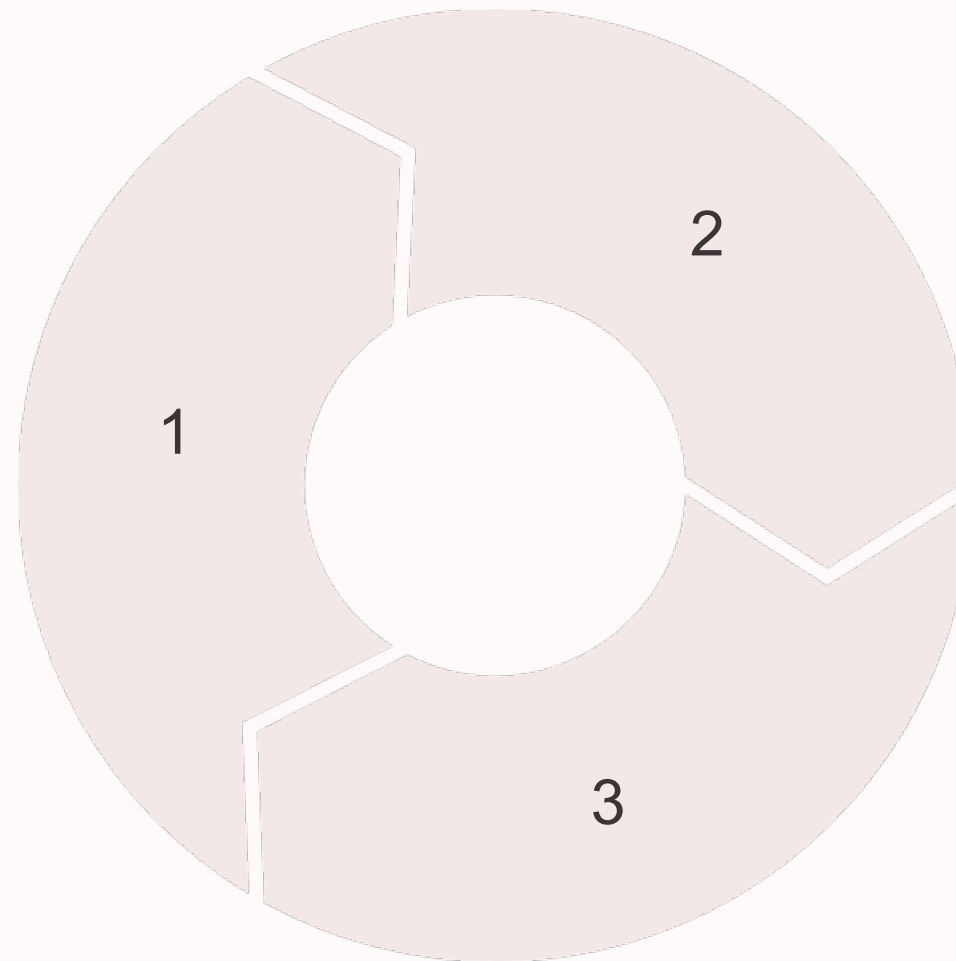
3

Promises

Fetch retorna uma Promise. Use `then()` para lidar com a resposta.

Interatividade Aprimorada: Eventos e Animações

Eventos
Ações do usuário. Clique, mouseover, submit, etc.



Ouvintes de Evento

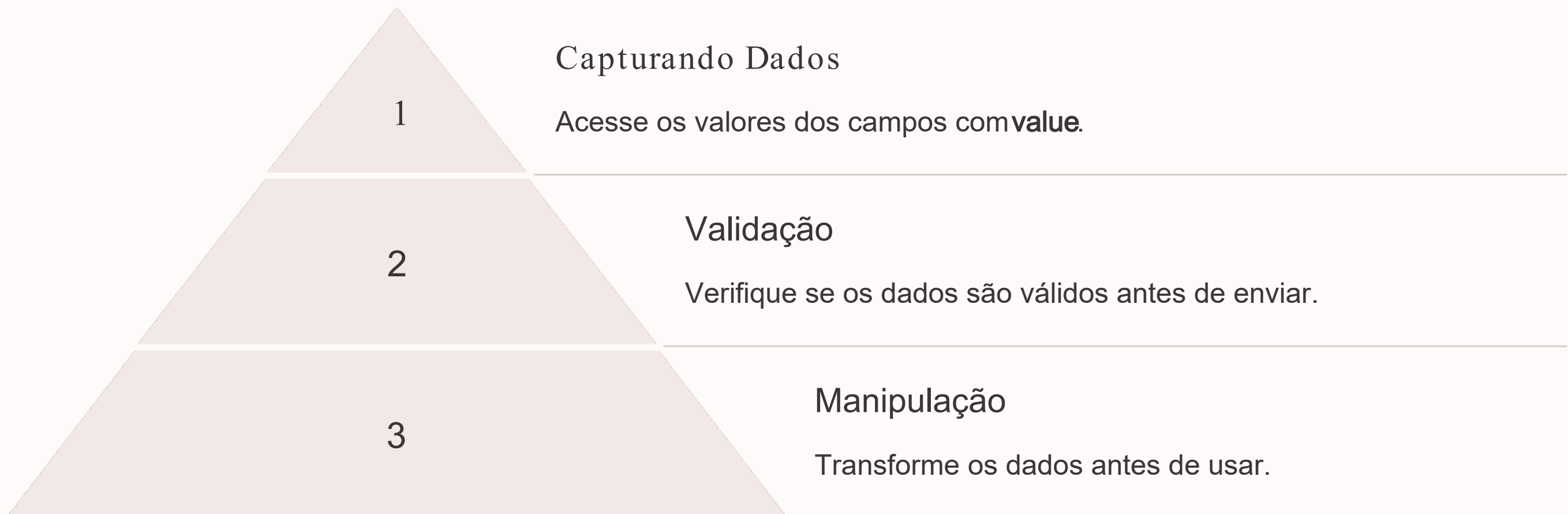
Funções que respondem aos eventos. Use `addEventListener()`

Animações

Crie animações com CSS ou JavaScript.



Formulários Sob Controle: Validação e Manipulação de Dados

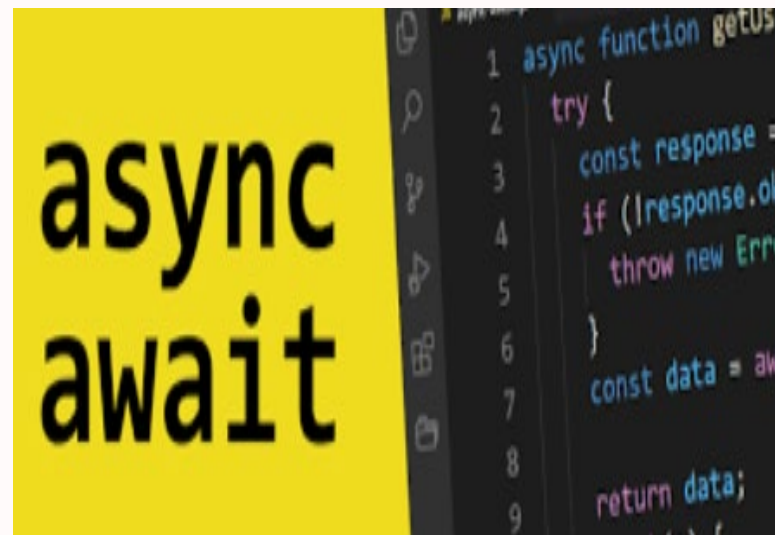


Tratamento de Erros e Async/Await: Código Robusto e Elegante



Try...Catch

Use **try...catch** para capturar erros.
Evite que o programa quebre.



Async/Await

Simplifica o código assíncrono. Torna-o
mais fácil de ler.



Código Robusto

Escreva um código que lida com os
erros corretamente.