

WebMAC

Com FastAPI e HTMX

Aula 3

Javascript - Manipulação da DOM e Eventos

Semanas do Curso

1. HTML, CSS e Responsividade

2. Javascript

3. FastAPI

4. SQL

5. HTMX e UI

6. Docker

Objetivos das semanas

1. Criar páginas web
- 2. Tornar páginas dinâmicas**
3. Criar um back-end
4. Gerenciar dados
5. Criar interfaces modernas
6. Rodar com containers

Canal do Codelab no Youtube



CodeLab

@CodeLabBR · 1,47 mil inscritos · 138 vídeos

O CodeLab é um grupo de extensão inter-universitário que tem como objetivo estimular a ...mais

[uclab.xyz/site](#) e mais 5 links

Inscrever-se

Início

Vídeos

Shorts

Ao vivo

Playlists

Q

Mais recentes

Em alta

Mais antigos

WebDev

Com Django e HTMX

Aula 5

Deployment

 CodeLab



1:04:31

Deployment - Aula5 (Webdev 2025)

56 visualizações · há 3 meses

WebDev

Com Django e HTMX

Aula 4

HTMX e UI

 CodeLab



1:30:34

HTMX - Aula4 (Webdev 2025)

65 visualizações · há 3 meses

WebDev

Com Django e HTMX

Aula 3

SQL e Django Models

 CodeLab



1:27:53

SQL e Django Models - Aula3 (Webdev 2025)

78 visualizações · há 4 meses

WebDev

Com Django e HTMX

Aula 2

Django - Parte 1

 CodeLab



1:18:30

Django - Aula2 (Webdev 2025)

242 visualizações · há 5 meses

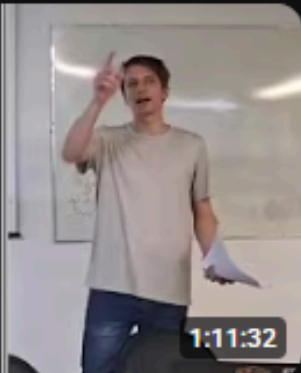
WebDev

Com Django e HTMX

Aula 1

Javascript - Manipulação da DOM, Eventos, etc.

 CodeLab



1:11:32

JavaScript - Aula1 (Webdev 2025)

234 visualizações · há 5 meses

WebDev

Com Django e HTMX

Aula 0

HTML, CSS e Responsividade

 CodeLab



1:33:48

HTML e CSS - Aula0 (Webdev 2025)

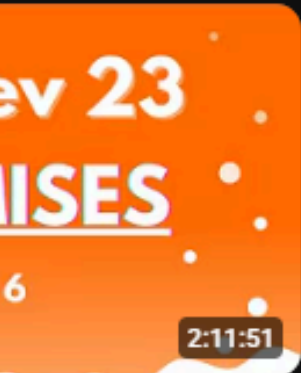
156 visualizações · há 5 meses

Webdev 23

PROMISES

Aula 6

 CodeLab



2:11:51

Promises - Aula 6 - WEBDEV 23

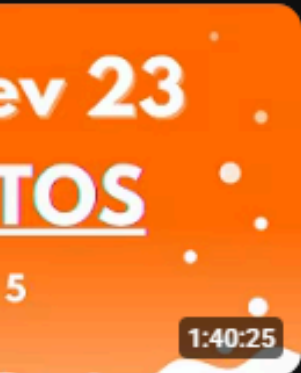
103 visualizações · há 2 anos

Webdev 23

OBJETOS

Aula 5

 CodeLab



1:40:25

Objetos e Arrays - Aula 5 - WEBDEV 23

89 visualizações · há 2 anos

Sobre Javascript:

Roda no web browser:



Dinamicamente tipado

Não tem problema esquecer do ;

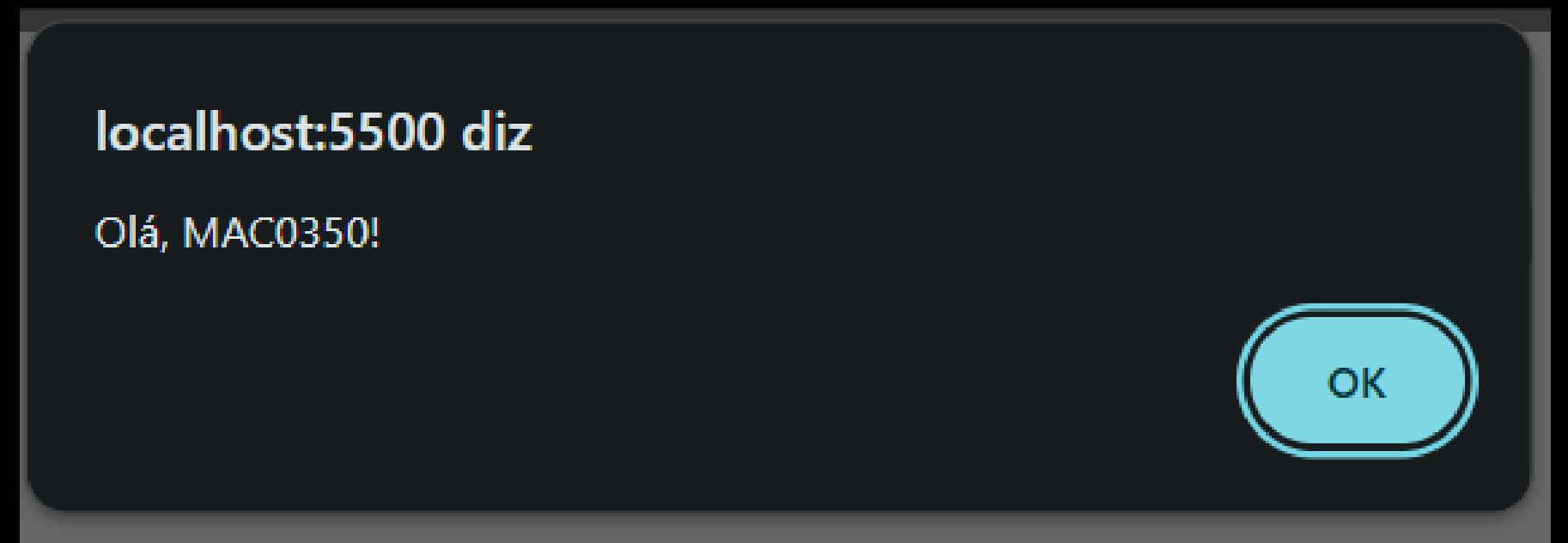
Tem suporte para

- variáveis,
- condicionais,
- laços,
- operadores,
- funções,
- objetos,
- Math.random(),
- Arrays,
- Estruturas de Dados
- Etc etc.

- Onde escrever javascript?
 - tag `<script></script>`
 - Documento separado x comum
 - `<head>` vs `<body>`

Exemplo - *Hello, World!* no console

```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <title>Olá</title>
    <script>
      alert('Olá, MAC0350!');
      console.log('olá');
    </script>
  </head>
  <body>
    <h1>Olá!</h1>
  </body>
</html>
```



Sintaxe Básica

// comentário

Declaração de variáveis:

- `let x = 10` *//definir uma variável com escopo limitado*
- `const y = 20` *//valor que não mudará*
- `var idade = 20;` *//definir uma variável globalmente*

Declaração de funções:

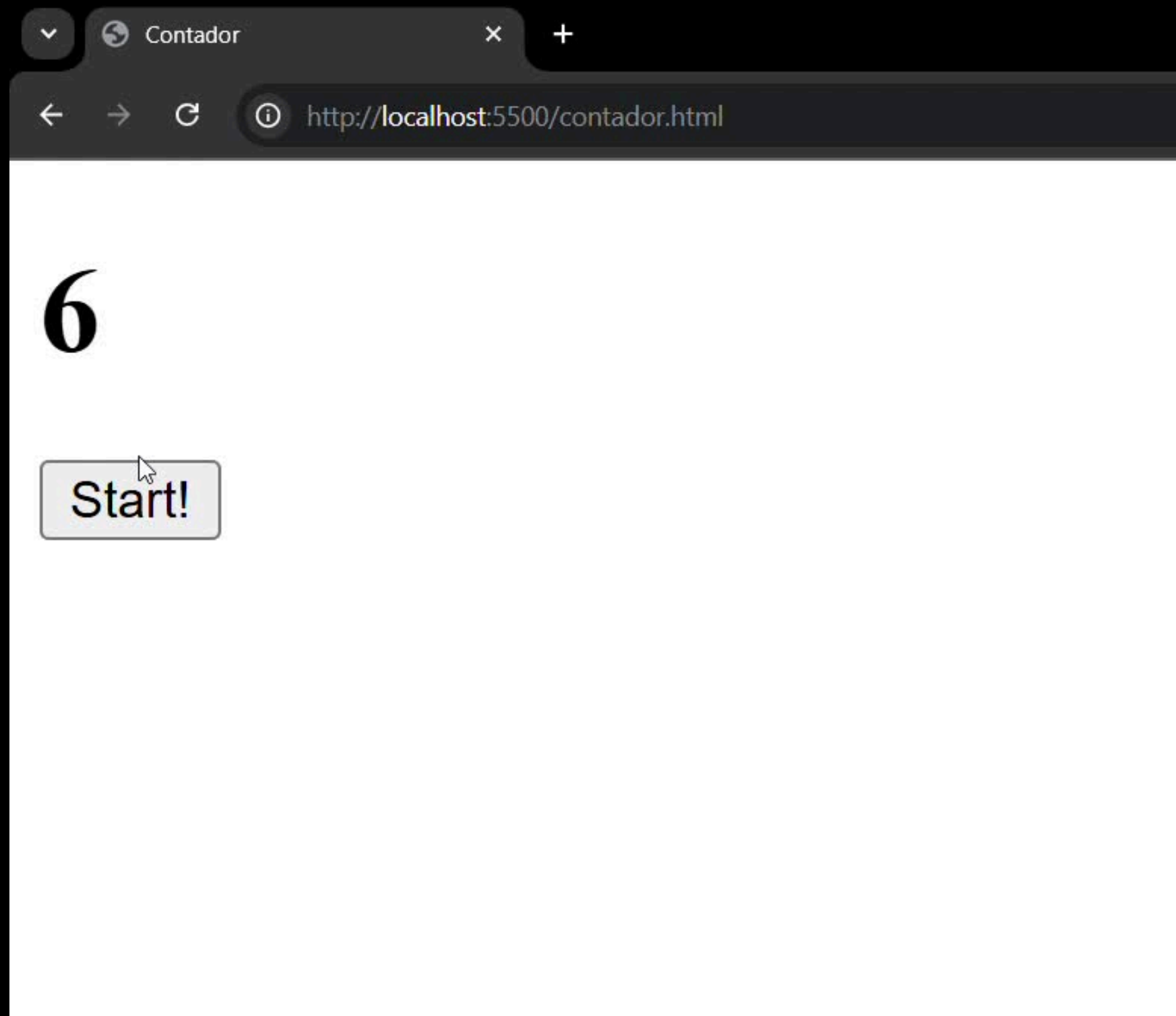
- `function isPrime(N)`
 `{for (let i = 2; i < N; i++)`
 `if (N % i == 0)`
 `return false`
 `return N > 1}`

`}`

Condicionais e Coerção de Tipos

- == (Igualdade solta): Converte os tipos antes de comparar (ex: "5" == 5 resulta em verdadeiro)
- === (Igualdade estrita): Compara tanto o valor quanto o tipo primitivo. A boa prática é sempre utilizar este operador para evitar bugs silenciosos.

Exemplo - Contador



```
<!DOCTYPE html>
<html lang="pt">
  <head>
    <title>Contador</title>
    <script>
      let contador = 0;
      function contar() {
        contador++;
        alert(contador);
      }
    </script>
  </head>
  <body>
    <h1>Olá!</h1>
    <button
onclick="contar()">Contar</button>
  </body>
</html>
```

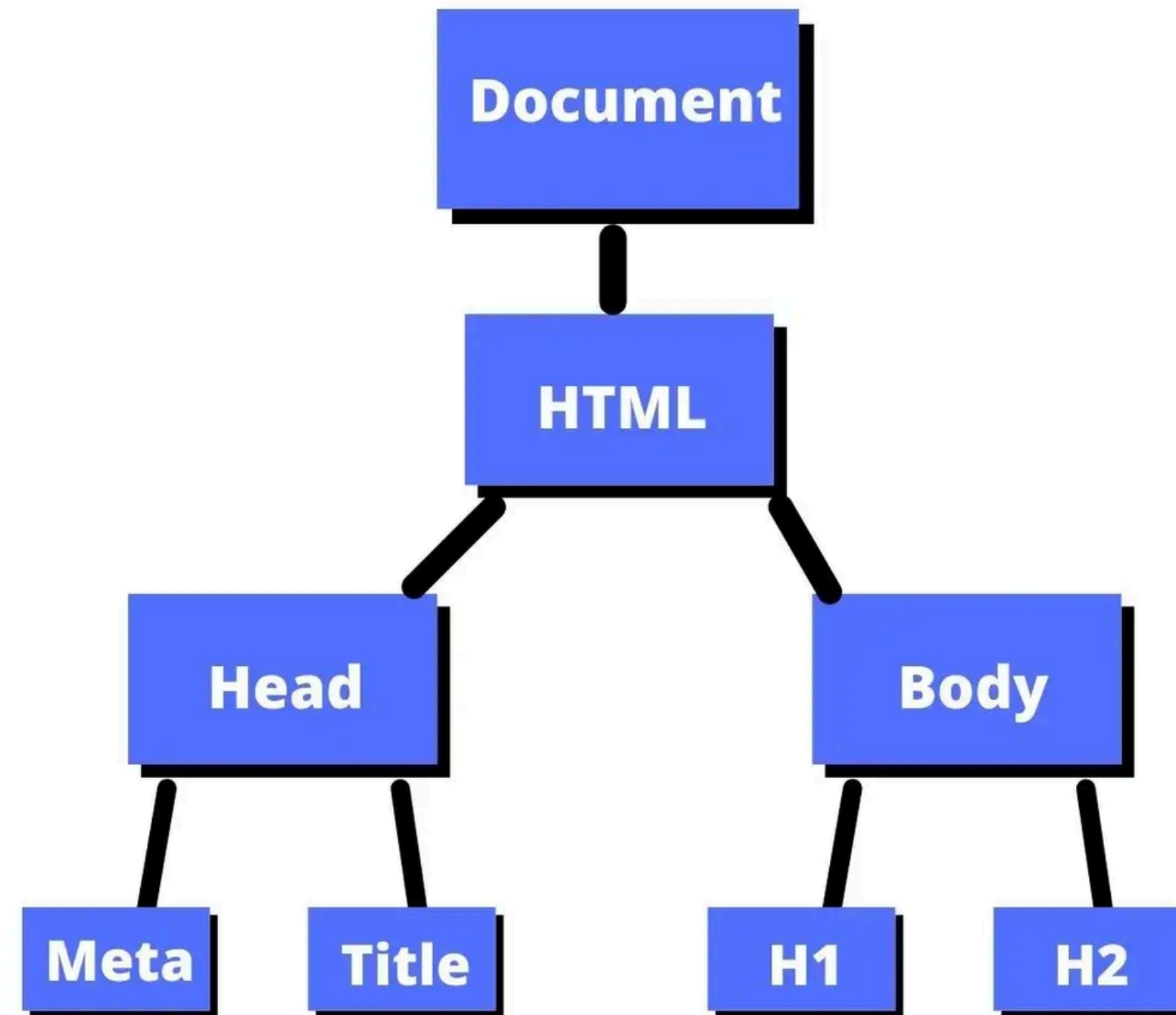
Eventos

- *onclick*
- *onchange*
- *onmouseover*
- *mousedown/mouseup*
- *submit*
- DOMContentLoaded
- etc.

Event Handlers:

- `<tag evento="código em javascript">`
- `<tag evento="f()">`
- `elemento.addEventListener("click", function f(){})`

DOM - Document Object Model



Manipulação da DOM- Comandos

Selecionadores de Nós

- `document.querySelector()`
- `document.querySelectorAll()`
- `document.getElementById()`

Métodos dos Nós

- `document.createElement()`
- `.appendChild()`
- `.insertBefore(new, reference)`
- `.replaceChild(new, old)`
- `.getRootNode()`
- `.remove()`

Atributos dos Nós

- `.childNodes`
- `.innerHTML`
- `.innerText`
- `.style.color`
- `.firstChild`
- `.lastChild`

Arrow Functions

- `let f = (a,b) => a+b+1`
- `let foo = () => console.log("hello")`
- `let bar = x => 2*x`

Por que é útil?

```
document.addEventListener('DOMContentLoaded', () => {  
    // comandos  
});
```

Removendo JS Inline e Event Listeners

```
document.querySelector('button').onclick = contar;
```

```
document.querySelector('button').addEventListener('click', contar);
```

Arquivos Separados

```
<script src="/static/contador.js"></script>
```

Exemplo: Teste de Primalidade

Local Storage

- `localStorage.getItem(chave)` `//Retorna o valor salvo.`
- `localStorage.setItem(chave, valor)` `//Salva um novo valor.`

setInterval

- `setInterval(foo, time)` `//tempo em ms`

Exemplo: gatinho dançante