



# INTRODUÇÃO AO JAVASCRIPT NO FRONTEND

Prof. Me. Hélio Esperidião

# O que é JavaScript?



Linguagem de script orientada a objetos e multiplataforma



Leve e pequena



É uma linguagem interpretada.

# JavaScript no client-side

JavaScript originalmente foi projetada como uma linguagem de programação do tipo Client Side, ou seja, é executada no computador do usuário.

Essa definição é crucial para compreender muitos aspectos dessa linguagem.

# Não cometa enganos!



É importante esclarecer que o termo JavaScript não tem relação com a linguagem de programação Java, apesar da confusão comum na comunidade de desenvolvimento web.



Essa confusão muitas vezes é feita por pessoas que nunca trabalharam com a tecnologia.



No entanto, deixando de lado os contextos históricos e discursos sobre como surgiu o nome, vamos focar no que realmente importa

# DOM - Document Object Model



O Modelo de Objeto de Documento, conhecido como DOM, é a representação em memória dos elementos que constituem a estrutura e o conteúdo de um documento da web, seja ele em HTML ou XML.



É uma interface de programação que se aplica aos documentos HTML e XML, permitindo que a página seja representada de tal forma que os programas tenham a capacidade de modificar sua estrutura, estilo e conteúdo.



O DOM representa o documento(HTML/XML) por meio de nós e objetos, o que possibilita que linguagens de programação se conectem à página.

Document

Root element:  
<html>

Element:  
<head>

Element:  
<body>

Element:  
<title>

Attribute:  
"href"

Element:  
<a>

Element:  
<h1>

Text:  
"My title"

Text:  
"My link"

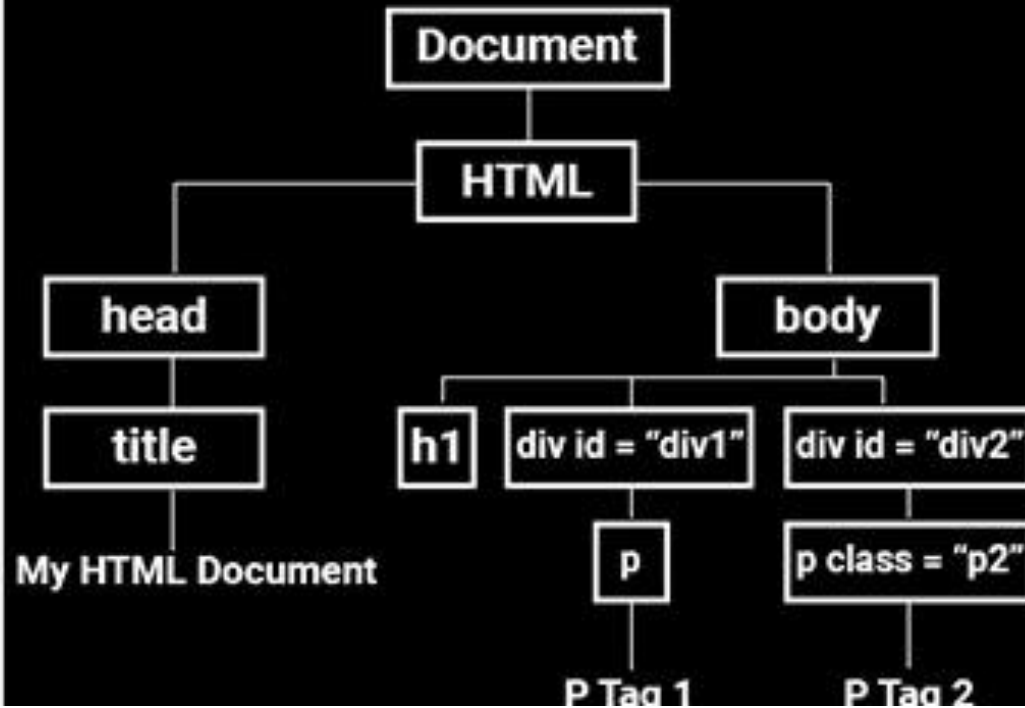
Text:  
"My header"

# What is Document Object Model ?

## HTML Document

```
index.html x
1 <html>
2   <head>
3     <title>My HTML Document</title>
4   </head>
5
6   <body>
7     <h1>Heading</h1>
8     <div id="div1">
9       <p>P Tag 1</p>
10    </div>
11    <div id="div2">
12      <p class="p2">P Tag 2</p>
13    </div>
14  </body>
15 </html>
```

## Document Object Model (DOM)



Exemplo prático

# Tipos Primitivos (Primitive Types)

Tipo	Exemplo	Descrição
string	"Olá mundo"	Texto (cadeias de caracteres)
number	42, 3.14	Números inteiros ou decimais
bigint	1234567890123456789n	Números inteiros grandes
boolean	true, false	Lógico verdadeiro/falso
undefined	undefined	Valor de uma variável <b>não inicializada</b>
null	null	Valor "nulo", usado para indicar ausência

# Tipos de Referência (Objetos)

Tipo	Exemplo	Descrição
Object	{ nome: "Ana", idade: 30 }	Estrutura chave-valor
Array	[1, 2, 3]	Lista ordenada de elementos
Function	function soma(a, b) {}	Blocos de código executáveis
Date	new Date()	Representa data e hora

# Declaração de variáveis

- As palavras-chave **let** e **var** em JavaScript são usadas para declarar variáveis, mas têm diferenças importantes em comportamento.

```
let x =1;
```

```
var y=1;
```

# Exemplos de variáveis

- `let x = "Ola mundo"; // x é string`
- `let y = 1; // x é número`
- `let z = true; // é booleano`
- `let w; // é indefinido`
- `let pessoa = {nome:"Hélio", cargo:"Professor"}; // é um objeto`
  
- `var y = "Ola mundo"; // y é string`

# Diferença entre let e var

Característica	var	let
Escopo	Função	Bloco
Redeclaração	Permitida no mesmo escopo	<b>Não permitida</b> no mesmo escopo
Declaração global	Adiciona variável ao objeto global (ex: window)	<b>Não adiciona</b> ao objeto global
Uso recomendado	<b>Evite usar em código moderno</b>	Recomendado

# Escopo

```
function teste() {  
  if (true) {  
    var x = 10;  
  }  
  console.log(x); // Funciona: x = 10  
}
```

```
function teste() {  
  if (true) {  
    let x = 10;  
  }  
  console.log(x); // Erro: x is not defined  
}
```

# Operadores Relacionais

Operador	Descrição
==	Igual a
===	Valor igual e tipo de dado igual
!=	diferente
!==	Valor não igual ou tipo não igual
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a
?	ternary operator

# Operadores aritméticos

Operator	Description
+	somar
-	subtrair
*	Multiplicar
**	Potência
/	divisão
%	Resto de divisão
++	incremento
--	decremento

# Operadores lógicos

<b>Operator</b>	<b>Description</b>
&&	E lógico
	Ou lógico
!	Inversor lógico

# Objeto Document



- Utilizando o objeto document, é possível acessar um extenso conjunto de propriedades.

# Métodos Básicos.

<b>Propriedade</b>	<b>Descrição</b>
getElementById	Busca um elemento da página Web com o uso do atributo id do elemento.
createElement	Cria um nodo elemento.
appendChild	Insere um novo elemento filho.
removeChild	Remove um elemento filho.

# Antes do olá mundo

*Quando todo o conteúdo HTML da página for carregado e analisado, execute o código dentro dessa função :*

```
document.addEventListener("DOMContentLoaded", () => {  
  
});
```

# Olá Mundo

```
<body>
  <button id="btn01">Clique Aqui</button>
  <div id="divResposta"></div>
  <script>
    // Aguarda o carregamento completo do conteúdo da página (DOM)
    document.addEventListener("DOMContentLoaded", () => {
      // Seleciona o botão pelo seu ID e armazena na variável 'btn'
      const btn = document.getElementById("btn01");
      // Seleciona a div onde vamos mostrar o texto e armazena na variável 'divResposta'
      const divResposta = document.getElementById("divResposta");
      // Adiciona um ouvinte de evento ao botão, que espera por um clique
      btn.addEventListener("click", () => {
        // Quando o botão for clicado, o conteúdo da div será alterado para "Olá, Mundo!"
        divResposta.textContent = "Olá, Mundo!";
      });
    });
  </script>
</body>
```

- Observe o evento onclick que foi definido no botão que possui o id “btnClick1”. Quando o evento click acontecer a função será executada.
- document.getElementById("divResposta") recupera o elemento html que possui o id **divResposta**.
- É inserido um novo filho na div.
  - O filho é o um nó de texto

← → ↻ ⓘ 127.0.0.1:5500/Exemplo0001.html

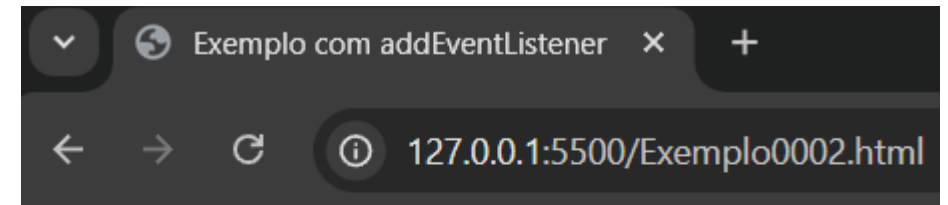
Clique Aqui

Olá, Mundo!

# Olá Mundo

```
<body>
  <button id="btn01">Clique Aqui</button>
  <div id="divResposta"></div>
  <script>
    // Aguarda o carregamento completo do conteúdo da página antes de executar o código
    document.addEventListener("DOMContentLoaded", () => {
      // Seleciona o botão com id "btn01" e armazena na variável btn
      const btn = document.getElementById("btn01");
      // Seleciona a div com id "divResposta" e armazena na variável divResposta
      const divResposta = document.getElementById("divResposta");
      // Adiciona ao botão um "ouvinte de evento" que espera pelo clique
      btn.addEventListener("click", () => {
        // Quando o botão é clicado, altera o conteúdo da div para "Olá, Mundo!" em negrito
        divResposta.innerHTML = "<b>Olá, Mundo!</b>";
      });
    });
  </script>
</body>
```

- Observe o evento onclick que foi definido no botão que possui o id "btnClick1". Quando o evento click acontecer a função será executada.
- document.getElementById("divResposta") recupera o elemento html que possui o id divResposta.
- É inserido um novo filho na div.
  - O filho é o um nó de texto

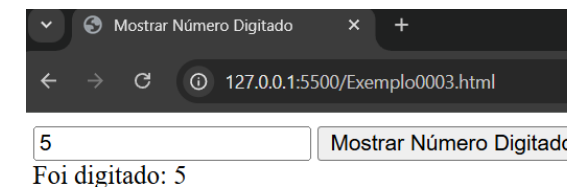


Clique Aqui  
**Olá, Mundo!**

```
<body>
  <input id="txtNumero" type="number" placeholder="Digite um número">
  <button id="btnClick1">Mostrar Número Digitado</button>
  <div id="divResposta"></div>
<script>
  // Adiciona um ouvinte para o evento DOMContentLoaded.
  // Esse evento garante que o código só será executado após toda a estrutura HTML da página ter sido carregada.
  document.addEventListener("DOMContentLoaded", () => {
    // Seleciona o campo de input (caixa de texto) com id "txtNumero" e armazena na variável txtNumero.
    const txtNumero = document.getElementById("txtNumero");
    // Seleciona o botão com id "btnClick1" e armazena na variável btn.
    const btn = document.getElementById("btnClick1");
    // Seleciona a div com id "divResposta" onde o resultado será exibido, e armazena na variável divResposta.
    const divResposta = document.getElementById("divResposta");
    // Adiciona ao botão um ouvinte para o evento de clique.
    // Ou seja, quando o usuário clicar no botão, será executada a função abaixo.
    btn.addEventListener("click", () => {
      // Lê o valor digitado na caixa de texto e armazena na variável numero.
      const numero = txtNumero.value;
      // Altera o texto da div para mostrar a mensagem com o número digitado.
      // textContent é usado para inserir texto puro (não interpreta tags HTML).
      divResposta.textContent = `Foi digitado: ${numero}`;
    });
  });
</script>

</body>
```

# Caixas de texto

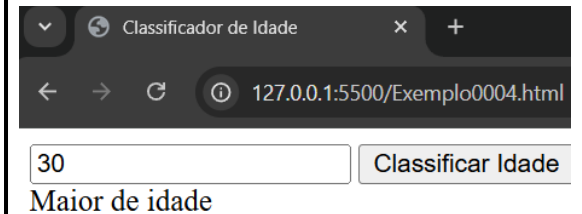


```

<body>
  <input id="txtNumero" type="number" placeholder="Digite a idade">
  <button id="btn01">Classificar Idade</button>
  <div id="divResposta"></div>
<script>
  document.addEventListener("DOMContentLoaded", () => {
    // Captura o elemento input com id "txtNumero" (onde o usuário digita um número).
    const txtNumero = document.getElementById("txtNumero");
    // Captura o botão com id "btn01" (que o usuário vai clicar para enviar o número).
    const btn = document.getElementById("btn01");
    // Captura a div com id "divResposta" (onde será exibida a resposta ao usuário).
    const divResposta = document.getElementById("divResposta");
    // Adiciona um ouvinte de evento ao botão, que espera pelo clique do usuário.
    btn.addEventListener("click", () => {
      // Lê o valor digitado no input, converte para número e armazena na variável numero.
      const numero = Number(txtNumero.value);
      // Cria uma variável para armazenar a resposta que será mostrada ao usuário.
      let resposta = "";
      if (numero >= 18 && numero < 65) {
        resposta = "Maior de idade";
      } else if (numero >= 65) {
        resposta = "Idoso";
      } else {
        resposta = "Menor de idade";
      }
      // Exibe a resposta dentro da div, substituindo o conteúdo anterior.
      divResposta.textContent = resposta;
    });
  });
</script>
</body>

```

# Exemplo if



# document.createElement(“elemento”)

- A função `document.createElement("elemento")` é usada no JavaScript para criar um novo elemento HTML no DOM (Document Object Model).
- `let elemento = document.createElement("tagName");`

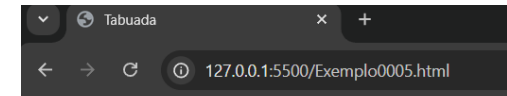
- A função `document.createElement(“tagName”)` cria um novo elemento HTML especificado pelo `tagName` fornecido. Esse elemento ainda não é parte do documento, mas pode ser adicionado posteriormente usando métodos como `appendChild`.
- `let novoParagrafo = document.createElement("p");`

```

<body>
  <input id="txtNumero" type="number" placeholder="Digite um número">
  <button id="btn01">Gerar Tabuada</button>
  <div id="divResposta"></div>
<script>
  // Espera o carregamento completo do DOM (estrutura HTML da página) antes de rodar o código
  document.addEventListener("DOMContentLoaded", () => {
    // Captura o input onde o usuário vai digitar o número (id="txtNumero")
    const input = document.getElementById("txtNumero");
    // Captura o botão que o usuário vai clicar para gerar a tabuada (id="btn01")
    const button = document.getElementById("btn01");
    // Captura a div onde a tabuada será exibida (id="divResposta")
    const resposta = document.getElementById("divResposta");
    // Adiciona um ouvinte ao botão: quando o botão for clicado, executa a função abaixo
    button.addEventListener("click", () => {
      // Lê o valor digitado no input, converte para número, e armazena na variável numero
      const numero = Number(input.value);
      // Limpa qualquer conteúdo anterior da div resposta (apaga tabuada anterior)
      resposta.innerHTML = "";
      // Laço de repetição que vai de 0 até 10 (inclusive)
      for (let i = 0; i <= 10; i++) {
        // Cria um novo parágrafo (elemento <p>) para cada linha da tabuada
        const linha = document.createElement("p");
        // Define o conteúdo do parágrafo com o cálculo da multiplicação
        linha.textContent = `${numero} x ${i} = ${numero * i}`;
        // Adiciona o parágrafo dentro da div resposta (exibe na tela)
        resposta.appendChild(linha);
      }
    });
  });
</script>
</body>

```

## For



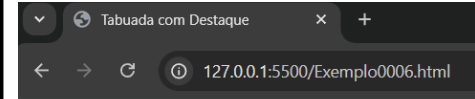
```

2 x 0 = 0
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

```

# For

```
<body>
  <!-- Campo de entrada para o usuário digitar um número -->
  <input id="txtNumero" type="number" placeholder="Digite um número">
  <!-- Botão que, ao ser clicado, gera a tabuada -->
  <button id="btn01">Gerar Tabuada</button>
  <!-- Div onde a tabuada será exibida -->
  <div id="divResposta"></div>
  <script>
    // Aguarda o carregamento completo do conteúdo da página antes de executar o código
    document.addEventListener("DOMContentLoaded", () => {
      // Obtém as referências dos elementos da página
      const input = document.getElementById("txtNumero"); // campo de entrada
      const button = document.getElementById("btn01"); // botão
      const resposta = document.getElementById("divResposta"); // div para mostrar a resposta
      // Adiciona um evento de clique no botão
      button.addEventListener("click", () => {
        // Converte o valor do input para número
        const n = Number(input.value);
        // Limpa o conteúdo anterior da divResposta
        resposta.innerHTML = "";
        // Laço que gera a tabuada de 0 até 10
        for (let i = 0; i <= 10; i++) {
          // Cria um novo parágrafo para cada linha da tabuada
          const paragrafo = document.createElement("p");
          // Define o conteúdo do parágrafo com a multiplicação e o resultado em negrito
          paragrafo.innerHTML = `${n} x ${i} = <b>${n * i}</b>`;
          // Adiciona o parágrafo na divResposta
          resposta.appendChild(paragrafo);
        }
      });
    });
  </script>
</body>
```



3

3 x 0 = 0

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

3 x 4 = 12

3 x 5 = 15

3 x 6 = 18

3 x 7 = 21

3 x 8 = 24

3 x 9 = 27

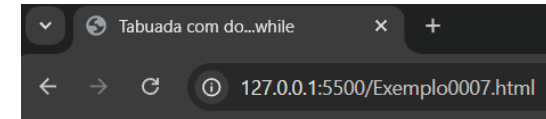
3 x 10 = 30

```

<body>
  <input id="txtNumero" type="number" placeholder="Digite um número" />
  <button id="btn01">Gerar Tabuada</button>
  <div id="divResposta"></div>
  <script>
    document.addEventListener("DOMContentLoaded", () => {
      const input = document.getElementById("txtNumero"); // campo onde digita o número
      const button = document.getElementById("btn01"); // botão que gera a tabuada
      const resposta = document.getElementById("divResposta"); // div que vai exibir a tabuada
      // Adiciona um evento ao botão: quando clicar, executa a função
      button.addEventListener("click", () => {
        // Converte o valor digitado no input para número
        const n = Number(input.value);
        // Limpa a div antes de exibir a nova tabuada (para não acumular)
        resposta.innerHTML = "";
        // Inicializa o contador i em 0
        let i = 0;
        // Estrutura do...while: primeiro executa, depois verifica a condição
        do {
          // Cria um parágrafo para cada linha da tabuada
          const paragrafo = document.createElement("p");
          // Monta o texto do parágrafo, mostrando a multiplicação e o resultado em negrito
          paragrafo.innerHTML = `${n} x ${i} = <b>${n} * i</b>`;
          // Adiciona o parágrafo na div de resposta
          resposta.appendChild(paragrafo);
          i++; // Incrementa o contador i
        } while (i <= 10); // Continua até i chegar a 10 (tabuada de 0 a 10)
      });
    });
  </script>
</body>

```

# do - while



3      Gerar Tabuada

3 x 0 = 0

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

3 x 4 = 12

3 x 5 = 15

3 x 6 = 18

3 x 7 = 21

3 x 8 = 24

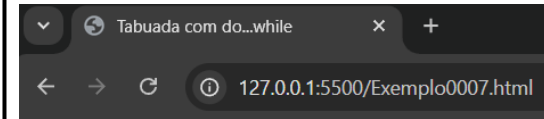
3 x 9 = 27

3 x 10 = 30

```
<script>
```

```
document.addEventListener("DOMContentLoaded", () => {  
  // Captura os elementos da página (input, botão e div de resposta)  
  const input = document.getElementById("txtNumero"); // campo onde digita o número  
  const button = document.getElementById("btn01"); // botão que gera a tabuada  
  const resposta = document.getElementById("divResposta"); // div que vai exibir a tabuada  
  // Adiciona um evento ao botão: quando clicar, executa a função  
  button.addEventListener("click", () => {  
    // Converte o valor digitado no input para número  
    const n = Number(input.value);  
    // Limpa a div antes de exibir a nova tabuada (para não acumular)  
    resposta.innerHTML = "";  
    // Inicializa o contador i em 0  
    let i = 0;  
    // Estrutura while: verifica a condição antes de executar o bloco  
    while (i <= 10) {  
      // Cria um parágrafo para cada linha da tabuada  
      const paragrafo = document.createElement("p");  
      // Monta o texto do parágrafo, mostrando a multiplicação e o resultado em negrito  
      paragrafo.innerHTML = `${n} x ${i} = <b>${n} * i</b>`;  
      // Adiciona o parágrafo na div de resposta  
      resposta.appendChild(paragrafo);  
      i++; // Incrementa o contador i  
    }  
  });  
});  
</script>
```

# while



3

3 x 0 = 0

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

3 x 4 = 12

3 x 5 = 15

3 x 6 = 18

3 x 7 = 21

3 x 8 = 24

3 x 9 = 27

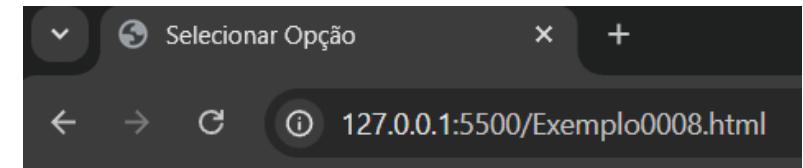
3 x 10 = 30

```

<body>
  <select id="cboLista"> <!-- Combobox (select) com 3 opções -->
    <!-- Cada opção tem um valor e um texto que será exibido -->
    <option value="1">Programação 1</option>
    <option value="2">Programação 2</option>
    <option value="3">Programação 3</option>
  </select>
  <!-- Botão que, ao ser clicado, exibe informações da opção selecionada -->
  <button id="btnClick1">Click Aqui</button>
  <div id="divResposta"></div> <!-- Div onde a resposta será exibida -->
  <script>
    // Aguarda o carregamento completo do conteúdo da página antes de executar o código
    document.addEventListener("DOMContentLoaded", () => {
      // Captura o select (combobox) pelo id
      const cboLista = document.getElementById("cboLista");
      // Captura a div onde será exibida a resposta
      const divResposta = document.getElementById("divResposta");
      // Captura o botão pelo id
      const btnClick = document.getElementById("btnClick1");
      // Adiciona um evento ao botão: quando clicado, executa a função
      btnClick.addEventListener("click", () => {
        // Limpa o conteúdo anterior da divResposta antes de mostrar o novo
        divResposta.textContent = "";
        // Pega o índice da opção selecionada no select (começa em 0)
        const indice = cboLista.selectedIndex;
        // Pega o valor (value) da opção selecionada
        const value = cboLista.value;
        // Pega o texto visível da opção selecionada
        const texto = cboLista.options[indice].text;
        // Cria um novo parágrafo
        const p = document.createElement("p");
        // Define o texto do parágrafo com as informações da seleção
        p.textContent = `Índice: ${indice}, Valor: ${value}, Texto: ${texto}`;
        // Adiciona o parágrafo na div de resposta
        divResposta.appendChild(p);
      });
    });
  </script>
</body>

```

# Recuperar dados de uma caixa de seleção



Índice: 1, Valor: 2, Texto: Programação 2

```

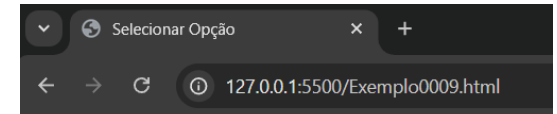
<body>
  <input type="radio" name="rdoNomeGrupo" id="rdoItem1" value="a">
  <input type="radio" name="rdoNomeGrupo" id="rdoItem2" value="b">
  <input type="radio" name="rdoNomeGrupo" id="rdoItem3" value="c">
  <button id="btnClick1">Click Aqui</button>
  <div id="divResposta"></div> <!-- Div onde o resultado será exibido -->
  <script>
    document.addEventListener("DOMContentLoaded", () => {
      // Captura os inputs de radio pelo id
      const objRdoItem1 = document.getElementById("rdoItem1");
      const objRdoItem2 = document.getElementById("rdoItem2");
      const objRdoItem3 = document.getElementById("rdoItem3");
      // Captura a div onde será mostrada a resposta
      const objDivResposta = document.getElementById("divResposta");
      // Captura o botão
      const objBtn = document.getElementById("btnClick1");
      // Adiciona um evento ao botão: quando clicado, executa a função
      objBtn.addEventListener("click", () => {
        // Limpa o conteúdo anterior da div antes de exibir novo resultado
        objDivResposta.textContent = "";
        // Verifica se cada radio está selecionado (checked = true ou false)
        const valor1 = objRdoItem1.checked; // true se o primeiro radio está selecionado
        const valor2 = objRdoItem2.checked; // true se o segundo radio está selecionado
        const valor3 = objRdoItem3.checked; // true se o terceiro radio está selecionado
        // Cria um novo parágrafo
        const p = document.createElement("p");

        // Define o texto do parágrafo com o estado de cada radio
        // Exemplo: true - false - false
        p.textContent = `${valor1} - ${valor2} - ${valor3}`;

        // Adiciona o parágrafo na div de resposta
        objDivResposta.appendChild(p);
      });
    });
  </script>
</body>

```

# Selecionar caixa radio



Click Aqui

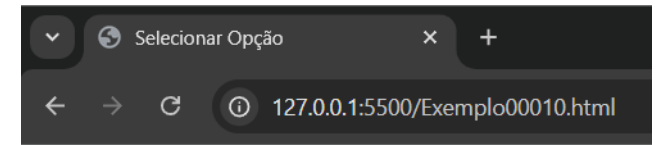
true - false - false

```

<body>
  <input type="checkbox" id="chkItem1">
  <input type="checkbox" id="chkItem2">
  <input type="checkbox" id="chkItem3">
  <button id="btnClick1">Click Aqui</button>
  <div id="divResposta"></div>
  <script>
    document.addEventListener("DOMContentLoaded", () => {
      // Captura os checkboxes pelos seus respectivos ids
      const objChkItem1 = document.getElementById("chkItem1"); // checkbox 1
      const objChkItem2 = document.getElementById("chkItem2"); // checkbox 2
      const objChkItem3 = document.getElementById("chkItem3"); // checkbox 3
      // Captura a div onde será mostrada a resposta
      const objDivResposta = document.getElementById("divResposta");
      // Captura o botão
      const objBtn = document.getElementById("btnClick1");
      // Adiciona um evento ao botão: quando clicado, executa a função
      objBtn.addEventListener("click", () => {
        // Limpa o conteúdo anterior da div antes de exibir o novo resultado
        objDivResposta.textContent = "";
        // Verifica se cada checkbox está marcado (checked = true ou false)
        const valor1 = objChkItem1.checked; // true se o checkbox 1 está marcado
        const valor2 = objChkItem2.checked; // true se o checkbox 2 está marcado
        const valor3 = objChkItem3.checked; // true se o checkbox 3 está marcado
        // Cria um novo parágrafo
        const p = document.createElement("p");
        // Define o texto do parágrafo com o estado de cada checkbox
        // Exemplo: true - false - true
        p.textContent = `${valor1} - ${valor2} - ${valor3}`;
        // Adiciona o parágrafo na div de resposta
        objDivResposta.appendChild(p);
      });
    });
  </script>
</body>

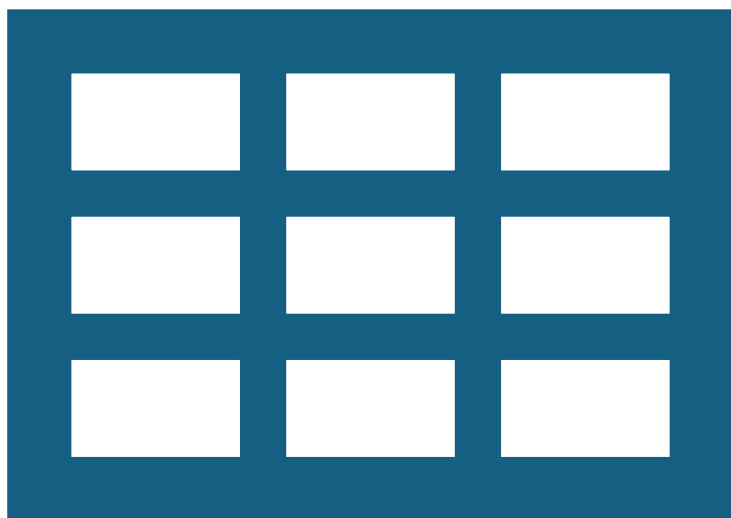
```

## Selecionar caixa checagem



Click Aqui

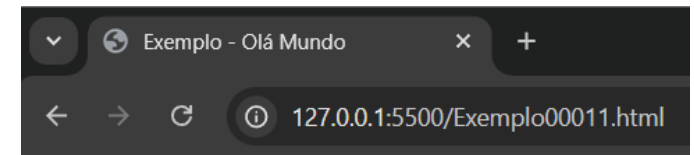
false - true - false



Inserir dados em  
elementos de formulário

## Escrever em uma caixa de texto

```
<!DOCTYPE html>
<html lang="pt-BR" > <!-- Define o idioma da página como português do Brasil -->
<head>
  <meta charset="UTF-8" > <!-- Define a codificação dos caracteres como UTF-8 -->
  <title>Exemplo - Olá Mundo</title> <!-- Título da aba do navegador -->
</head>
<body>
  <!-- Div reservada, neste exemplo não está sendo usada, mas poderia exibir informações -->
  <div id="divResposta"></div>
  <!-- Campo de texto onde será exibida a mensagem "Olá mundo :)" -->
  <input type="text" id="txtDado">
  <!-- Botão que, ao ser clicado, altera o valor do input de texto -->
  <button id="btnClick1">Click Aqui</button>
  <script>
    // Aguarda o carregamento de todo o conteúdo da página antes de executar o código
    document.addEventListener("DOMContentLoaded", () => {
      // Inicializa uma variável chamada x com valor 0
      // Essa variável será usada como contador para mostrar quantas vezes o botão foi clicado
      let x = 0;
      // Captura o campo de texto (input) pelo id e armazena na constante txtDados
      const txtDados = document.getElementById("txtDado");
      // Captura o botão pelo id e armazena na constante btn
      const btn = document.getElementById("btnClick1");
      // Adiciona um evento ao botão: quando o botão for clicado, executa a função abaixo
      btn.addEventListener("click", () => {
        // Incrementa o valor da variável x em 1 (conta o clique)
        x++;
        // Altera o valor do input de texto para exibir a mensagem "Olá mundo :)" seguida do número de cliques
        txtDados.value = `Olá mundo :) ${x}`;
      });
    });
  </script>
</body>
</html>
```



Olá mundo :) 6

Click Aqui

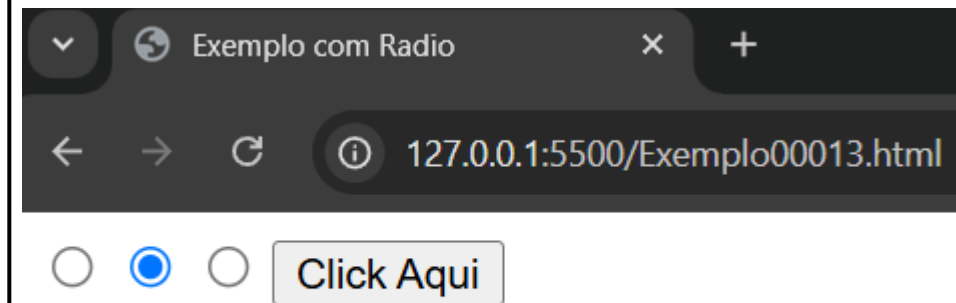
```
<body>
  <select id="cboLista">
    <option value="a">Programação 1</option>
    <option value="b">Programação 2</option>
    <option value="c">Programação 3</option>
  </select>
  <button id="btnClick1">Click Aqui</button>
  <script>
    // Aguarda o carregamento completo do conteúdo da página antes de executar o código
    document.addEventListener("DOMContentLoaded", () => {
      // Captura o elemento select pelo id
      const cboLista = document.getElementById("cboLista");
      // Captura o botão pelo id
      const btn = document.getElementById("btnClick1");
      // Adiciona um evento ao botão: quando clicado, executa a função
      btn.addEventListener("click", () => {
        // Altera o valor selecionado do select para "c" (Programação 3)
        cboLista.value = "c";
      });
    });
  </script>
</body>
```

# Selecionar caixa de combo

## Ao click do botão será selecionado o value “c”

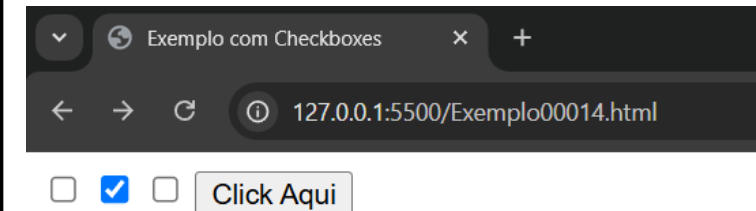
```
<body>
  <!-- Três radios com mesmo name formando grupo -->
  <input type="radio" name="rdoNomeGrupo" id="rdoItem1" value="a">
  <input type="radio" name="rdoNomeGrupo" id="rdoItem2" value="b">
  <input type="radio" name="rdoNomeGrupo" id="rdoItem3" value="c">
  <!-- Botão para alterar seleção -->
  <button id="btnClick1">Click Aqui</button>
  <script>
    // Aguarda o carregamento completo do conteúdo da página
    document.addEventListener("DOMContentLoaded", () => {
      // Captura os radios pelos ids
      const objRdoItem1 = document.getElementById("rdoItem1");
      const objRdoItem2 = document.getElementById("rdoItem2");
      const objRdoItem3 = document.getElementById("rdoItem3");
      // Captura o botão
      const objBtn = document.getElementById("btnClick1");
      // Evento de clique no botão para alterar seleção dos radios
      objBtn.addEventListener("click", () => {
        // Marca o segundo radio
        objRdoItem2.checked = true;
      });
    });
  </script>
</body>
```

# Selecionar caixa radio



```
<body>
  <!-- Três checkboxes com valores distintos -->
  <input type="checkbox" id="chkItem1" value="a" />
  <input type="checkbox" id="chkItem2" value="b" />
  <input type="checkbox" id="chkItem3" value="c" />
  <!-- Botão para alterar o estado dos checkboxes -->
  <button id="btnClick1">Click Aqui</button>
  <script>
    // Aguarda o carregamento completo do conteúdo da página
    document.addEventListener("DOMContentLoaded", () => {
      // Captura os checkboxes pelos ids
      const objChkItem1 = document.getElementById("chkItem1");
      const objChkItem2 = document.getElementById("chkItem2");
      const objChkItem3 = document.getElementById("chkItem3");
      // Captura o botão
      const objBtn = document.getElementById("btnClick1");
      // Evento de clique no botão para alterar os checkboxes
      objBtn.addEventListener("click", () => {
        // Desmarca o primeiro checkbox
        objChkItem1.checked = false;
        // Marca o segundo checkbox
        objChkItem2.checked = true;
        // Desmarca o terceiro checkbox
        objChkItem3.checked = false;
      });
    });
  </script>
</body>
```

# Selecionar caixa checagem



# Classes em js

- As **classes em JavaScript** são uma forma moderna (introduzida no ES6 – ECMAScript 2015) de criar objetos e estruturar código orientado a objetos, facilitando a organização e reutilização de código.
- Principais componentes:
  - constructor():
    - É o método chamado automaticamente ao criar uma instância da classe.
  - Métodos:
    - Funções dentro da classe. São adicionadas ao prototype do objeto.
  - Getters e Setters:
    - Permitem controlar como propriedades são lidas e modificadas:

# Exemplo Base

```
class Conta {
    #saldo = 0;

    depositar(valor) {
        if (valor > 0) {
            this.#saldo += valor;
        }
    }

    verSaldo() {
        return this.#saldo;
    }
}
```

- Atributos com # só podem ser acessados dentro da classe.
- # representa o mediador de acesso privado

```
export default class Quadrado {
  #lado = 0; // atributo privado que guarda o tamanho do lado do quadrado
  // construtor que inicializa o lado com o valor passado ou 0 por padrão
  constructor(lado = 0) {
    this.#lado = lado;
  }
  // Getter para acessar o valor do lado de fora da classe
  get lado() {
    return this.#lado;
  }
  // Setter para modificar o valor do lado de fora da classe
  set lado(valor) {
    this.#lado = valor;
  }
  // Método para calcular e retornar a área do quadrado
  calcularArea() {
    return this.#lado ** 2;
  }
  // Método para calcular e retornar o perímetro do quadrado
  calcularPerimetro() {
    return 4 * this.#lado;
  }
  // Método para calcular e retornar o comprimento da diagonal do quadrado
  calcularDiagonal() {
    return this.#lado * Math.sqrt(2);
  }
}
```

## Classe simples: Quadrado.js

# Utilizar a classe

```
<script type="module">
  // Importa a classe Quadrado do arquivo externo Quadrado.js
  import Quadrado from './Quadrado.js';
  // Aguarda o carregamento completo da página antes de executar o script
  document.addEventListener('DOMContentLoaded', () => {
    // Seleciona o input onde o usuário digita o lado
    const inputLado = document.getElementById('lado');
    // Seleciona o botão para calcular
    const btnCalcular = document.getElementById('btnCalcular');
    // Seleciona a div onde será mostrado o resultado
    const resultado = document.getElementById('resultado');
    // Adiciona um evento ao botão para executar o cálculo quando clicado
    btnCalcular.addEventListener('click', () => {
      // Converte o valor do input para número decimal
      const n = parseFloat(inputLado.value);
      // Cria uma nova instância da classe Quadrado
      const quadrado = new Quadrado();
      // Define o lado do quadrado com o valor digitado pelo usuário
      quadrado.lado = n;
      // Atualiza o conteúdo da div resultado com os cálculos formatados
      resultado.innerHTML = `
        Lado: ${quadrado.lado} cm<br />
        Área: ${quadrado.calcularArea().toFixed(2)} cm²<br />
        Perímetro: ${quadrado.calcularPerimetro().toFixed(2)} cm<br />
        Diagonal: ${quadrado.calcularDiagonal().toFixed(2)} cm
      `;
    });
  });
</script>
```

```
export default class Retangulo {
  #largura = 0; // campo privado para largura
  #altura = 0; // campo privado para altura
  constructor(largura = 0, altura = 0) {
    this.#largura = largura;
    this.#altura = altura;
  }
  // Método para calcular a área
  calcularArea() {
    return this.#largura * this.#altura;
  }
  // Método para calcular o perímetro
  calcularPerimetro() {
    return 2 * (this.#largura + this.#altura);
  }
  // Método para calcular a diagonal do retângulo
  calcularDiagonal() {
    return Math.sqrt(this.#largura ** 2 + this.#altura ** 2);
  }
  // Getter para largura
  get largura() {
    return this.#largura;
  }
  // Setter para largura
  set largura(valor) {
    this.#largura = valor;
  }
  // Getter para altura
  get altura() {
    return this.#altura;
  }
  // Setter para altura
  set altura(valor) {
    this.#altura = valor;
  }
}
```

# Retangulo.js

```

<script type="module">
import Retangulo from './Retangulo.js';
document.addEventListener('DOMContentLoaded', () => {
  const inputLargura = document.getElementById('largura'); // captura input da largura
  const inputAltura = document.getElementById('altura'); // captura input da altura
  const btnCalcular = document.getElementById('btnCalcular'); // captura botão calcular
  const resultado = document.getElementById('resultado'); // captura div onde resultado será mostrado
  btnCalcular.addEventListener('click', () => {
    // Converte os valores digitados para número decimal
    const largura = parseFloat(inputLargura.value);
    const altura = parseFloat(inputAltura.value);
    // Validação: verifica se largura é número e positivo
    if (isNaN(largura) || largura <= 0) {
      resultado.textContent = 'Por favor, insira um valor válido e positivo para a largura.';
      return; // interrompe execução se inválido
    }
    // Validação: verifica se altura é número e positivo
    if (isNaN(altura) || altura <= 0) {
      resultado.textContent = 'Por favor, insira um valor válido e positivo para a altura.';
      return; // interrompe execução se inválido
    }
    // Cria nova instância da classe Retangulo
    const retangulo = new Retangulo();
    retangulo.largura = largura; // seta largura usando setter
    retangulo.altura = altura; // seta altura usando setter
    // Exibe os resultados formatados na div resultado
    resultado.innerHTML = `
      Largura: ${retangulo.largura} cm<br />
      Altura: ${retangulo.altura} cm<br />
      Área: ${retangulo.calcularArea().toFixed(2)} cm²<br />
      Perímetro: ${retangulo.calcularPerimetro().toFixed(2)} cm
    `;
  });
});
</script>

```

# Utilizar a classe Retangulo.js

```
<body>
```

```
<button id="btnClick1">Click Aqui</button>
```

```
<div id="divResposta"></div>
```

```
<script>
```

```
document.addEventListener("DOMContentLoaded", () => {  
  const objDivResposta = document.getElementById("divResposta");  
  const objBtn = document.getElementById("btnClick1");  
  objBtn.addEventListener("click", () => {  
    // Array com as disciplinas  
    const vetorDisciplinas = ["Programação 1", "Programação 2", "Programação 3"];  
    // Limpa a div para não acumular conteúdo  
    objDivResposta.textContent = "";  
    // Percorre o array e cria um parágrafo para cada disciplina  
    vetorDisciplinas.forEach((disciplina, i) => {  
      const p = document.createElement("p");  
      p.textContent = `${i} - ${disciplina}`; // mostra índice e nome  
      objDivResposta.appendChild(p); // adiciona à divResposta  
    });  
  });  
});
```

```
</script>
```

```
</body>
```

# Vetores simples

← → ↻ ⓘ 127.0.0.1:5500/Exemplo00017.html

Click Aqui

0 - Programação 1

1 - Programação 2

2 - Programação 3

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8" />
  <title></title>
</head>
<body>
  <div id="divResposta"></div>
  <button id="btnClick1">Click Aqui</button>
  <script>
    document.addEventListener("DOMContentLoaded", () => {
      const divResposta = document.getElementById("divResposta");
      const btn = document.getElementById("btnClick1");
      btn.addEventListener("click", () => {
        // String JSON com dados a serem convertidos em objeto
        const textoJson = '{"nome":"h lio", "email":"helioesperidiao@gmail.com"}';
        // Converte a string JSON em objeto JavaScript
        const objJson = JSON.parse(textoJson);
        // Cria um novo par grafo para mostrar os dados
        const p = document.createElement("p");
        p.textContent = `${objJson.nome} - ${objJson.email}`;
        // Adiciona o par grafo   divResposta
        divResposta.appendChild(p);
      });
    });
  </script>
</body>
</html>
```

# JSON

---

Click Aqui

h lio - helioesperidiao@gmail.com

# VETOR JSON

```
<script>
document.addEventListener("DOMContentLoaded", () => {
  const divResposta = document.getElementById("divResposta");
  const btn = document.getElementById("btnClick1");
  btn.addEventListener("click", () => {
    // Define uma string JSON contendo um array de objetos com informações de nome e idade
    const textoJson = '[{"nome":"hélío", "idade":"30"}, {"nome":"maria", "idade":"7"}, {"nome":"Joana", "idade":"17"}]';
    const vetorObjetosJson = JSON.parse(textoJson); // Converte a string JSON para um array de objetos JavaScript
    // Limpa o conteúdo anterior da div para não acumular os dados a cada clique
    divResposta.textContent = "";
    // Percorre o array de objetos para processar cada item individualmente
    for (const objeto of vetorObjetosJson) {
      const p = document.createElement("p"); // Cria um novo elemento <p> para agrupar as informações formatadas
      // Cria um elemento <b> para o label "NOME:" com estilo negrito
      const bNome = document.createElement("b");
      bNome.textContent = "NOME: ";
      // Cria um elemento <b> para o label "IDADE:" com estilo negrito
      const bIdade = document.createElement("b");
      bIdade.textContent = " IDADE: ";
      p.appendChild(bNome); // Adiciona o label "NOME:" ao parágrafo
      p.appendChild(objeto.nome); // Adiciona o valor do nome logo após o label, como texto simples
      p.appendChild(bIdade); // Adiciona o label "IDADE:" ao parágrafo, após o nome
      p.appendChild(objeto.idade); // Adiciona o valor da idade logo após o label, como texto simples
      divResposta.appendChild(p); // Insere o parágrafo completo dentro da divResposta para exibir na página
    }
  });
});
</script>
```

Click Aqui

**NOME: hélío IDADE: 30**

**NOME: maria IDADE: 7**

**NOME: Joana IDADE: 17**

# Criar tabela dinamicamente.

```
document.addEventListener("DOMContentLoaded", () => {
  const txtNome = document.getElementById("txtNome");
  const txtEmail = document.getElementById("txtEmail");
  const divResposta = document.getElementById("divResposta");
  const btnClick = document.getElementById("btnClick1");
  const tabela = document.createElement("table"); // Cria a tabela
  tabela.setAttribute("border", "1"); // e define sua borda
  divResposta.appendChild(tabela); // Adiciona a tabela dentro da div de resposta
  const thead = document.createElement("thead"); // Cria o cabeçalho da tabela (thead)
  const trHead = document.createElement("tr");
  ["Nome", "Email"].forEach(texto => { // Cria os títulos das colunas
    const th = document.createElement("th");
    th.textContent = texto;
    trHead.appendChild(th);
  });
  thead.appendChild(trHead); // Adiciona o cabeçalho na tabela
  tabela.appendChild(thead);
  const tbody = document.createElement("tbody"); // Cria o corpo da tabela (tbody) onde as linhas serão adicionadas
  tabela.appendChild(tbody);
  btnClick.addEventListener("click", () => {
    const nome = txtNome.value.trim(); // Lê e limpa os valores digitados nos inputs
    const email = txtEmail.value.trim();
    const tr = document.createElement("tr"); // Cria uma nova linha
    const tdNome = document.createElement("td"); // Cria célula para o nome e adiciona na linha
    tdNome.textContent = nome;
    tr.appendChild(tdNome);
    const tdEmail = document.createElement("td"); // Cria célula para o e-mail e adiciona na linha
    tdEmail.textContent = email;
    tr.appendChild(tdEmail);
    tbody.appendChild(tr); // Adiciona a linha completa no corpo da tabela
  });
});
```

maria	maria@gmail.com	Click Aqui
Nome	Email	
helio	helioesperidiao@gmail.com	
maria	maria@gmail.com	

# Alterar css via javascript

- Para alterar o CSS de um elemento HTML via JavaScript, você pode usar a propriedade style do elemento.
- `let elemento = document.getElementById("meuElemento");`
- `elemento.style.color = "red"; // Muda a cor do texto`  
`elemento.style.backgroundColor = "yellow"; // Muda a cor de fundo`  
`elemento.style.fontSize = "20px"; // Muda o tamanho da fonte`  
`elemento.style.padding = "10px"; // Adiciona padding`  
`elemento.style.border = "2px solid black"; // Adiciona uma borda`

```
<body>
  <input type="color" id="txt1" />
  <button id="btnClick1">OK</button>
  <div id="divCor"></div>
  <div id="divLog"></div>

  <script>
    document.addEventListener("DOMContentLoaded", () => {
      // Captura dos elementos do DOM
      const txtColorInput = document.getElementById("txt1");
      const divCor = document.getElementById("divCor");
      const divLog = document.getElementById("divLog");
      const btn = document.getElementById("btnClick1");
      // Define o comportamento ao clicar no botão
      btn.addEventListener("click", () => {
        // Lê a cor escolhida pelo usuário
        const corSelecionada = txtColorInput.value;
        // Atualiza o estilo da divCor
        divCor.style.backgroundColor = corSelecionada;
        divCor.style.width = "100px";
        divCor.style.height = "100px";
        // Mostra o código da cor selecionada no divLog
        divLog.textContent = corSelecionada;
      });
    });
  </script>
</body>
```



# Carregar imagens dinamicamente

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>Imagem Dinâmica</title>
</head>
<body>
  <!-- Imagem que será atualizada dinamicamente -->
  <img src="" id="image" alt="Imagem" />
  <!-- Botão que, ao ser clicado, altera a imagem -->
  <button id="btn1">OK</button>
  <script>
    document.addEventListener("DOMContentLoaded", () => {
      // Captura o botão e a imagem pelo ID
      const btn = document.getElementById("btn1");
      const img = document.getElementById("image");
      // Adiciona o comportamento ao botão
      btn.addEventListener("click", () => {
        // Define o novo src e alt da imagem
        img.src = "http://helioesperidiao.com/imgs/eu.jpg";
        img.alt = "Exemplo";
      });
    });
  </script>
</body>
</html>
```

# Classes para manipular UI

- Você pode encapsular toda a lógica de manipulação do DOM em uma **classe separada**, tornando o código mais organizado, reutilizável e testável.

# Arquivos do exemplo

- RetanguloUi.js
- Retangulo.js
- front.html

```

import Retangulo from './Retangulo.js';
export default class RetanguloUI {
  // Atributos privados
  #inputLargura;
  #inputAltura;
  #btnCalcular;
  #divResultado;
  constructor(inputLargura, inputAltura, btnCalcular, divResultado) {
    this.#inputLargura = inputLargura;
    this.#inputAltura = inputAltura;
    this.#btnCalcular = btnCalcular;
    this.#divResultado = divResultado;
    this.#registrarEventos();
  }
  #registrarEventos() {
    this.#btnCalcular.addEventListener('click', () => this.calcular());
  }
  calcular() {
    const largura = this.inputLargura.value;
    const altura = this.inputAltura.value;
    if (isNaN(largura) || largura <= 0) {
      this.#divResultado.textContent = 'Largura inválida.';
      return;
    }
    if (isNaN(altura) || altura <= 0) {
      this.#divResultado.textContent = 'Altura inválida.';
      return;
    }
    const retangulo = new Retangulo(largura, altura);
    this.mostrarResultado(retangulo);
  }
}

```

```

mostrarResultado(retangulo) {
  this.#divResultado.innerHTML = `
    Largura: ${retangulo.largura} cm<br />
    Altura: ${retangulo.altura} cm<br />
    Área: ${retangulo.calcularArea().toFixed(2)} cm²<br />
    Perímetro: ${retangulo.calcularPerimetro().toFixed(2)} cm<br />
    Diagonal: ${retangulo.calcularDiagonal().toFixed(2)} cm
  `;
}
get inputLargura() {
  return this.#inputLargura;
}
set inputLargura(elemento) {
  this.#inputLargura = elemento;
}
get inputAltura() {
  return this.#inputAltura;
}
set inputAltura(elemento) {
  this.#inputAltura = elemento;
}
get btnCalcular() {
  return this.#btnCalcular;
}
set btnCalcular(elemento) {
  this.#btnCalcular = elemento;
  this.#registrarEventos(); // re-registra evento no novo botão
}
get divResultado() {
  return this.#divResultado;
}
set divResultado(elemento) {
  this.#divResultado = elemento;
}
}

```

```
<body>
  <input type="number" id="largura" min="0" step="any" /><br /><br />
  <input type="number" id="altura" min="0" step="any" /><br /><br />
  <button id="btnCalcular">Calcular</button>
  <div id="resultado"></div>

  <script type="module">
    import RetanguloUI from './retanguloUI.js';

    document.addEventListener('DOMContentLoaded', () => {
      const inputLargura = document.getElementById('largura');
      const inputAltura = document.getElementById('altura');
      const btnCalcular = document.getElementById('btnCalcular');
      const resultado = document.getElementById('resultado');

      // Passando elementos diretamente
      new RetanguloUI(inputLargura, inputAltura, btnCalcular, resultado);
    });
  </script>
</body>
```

# Retangulo.js

```
export default class Retangulo {
  #largura = 0; // campo privado para largura
  #altura = 0; // campo privado para altura
  constructor(largura = 0, altura = 0) {
    this.#largura = largura;
    this.#altura = altura;
  }
  // Método para calcular a área
  calcularArea() {
    return this.#largura * this.#altura;
  }
  // Método para calcular o perímetro
  calcularPerimetro() {
    return 2 * (this.#largura + this.#altura);
  }
  // Método para calcular a diagonal do retângulo
  calcularDiagonal() {
    return Math.sqrt(this.#largura ** 2 + this.#altura ** 2);
  }
  get largura() { // Getter para largura
    return this.#largura;
  }
  set largura(valor) { // Setter para largura
    this.#largura = valor;
  }
  get altura() { // Getter para altura
    return this.#altura;
  }
  set altura(valor) { // Setter para altura
    this.#altura = valor;
  }
}
```

# O que é LocalStorage?



O LocalStorage é uma das formas de armazenamento no navegador fornecidas pela Web Storage API.



Ele permite que você armazene dados no navegador do usuário de forma persistente, ou seja, os dados ficam salvos mesmo após o navegador ser fechado e reaberto.



Ele é útil para guardar informações que você deseja manter entre sessões, como configurações do usuário, temas, ou até dados de um carrinho de compras em um e-commerce.

# Principais características

- Armazena pares chave-valor como strings.
- Persistência ilimitada (até o usuário limpar os dados ou o navegador fazer isso).
- Disponível em quase todos os navegadores modernos.
- Capacidade típica: ~5MB por domínio.
- Dados são específicos para o protocolo + domínio (não são compartilhados entre sites).

# Como usar o LocalStorage

- Salvar dados:
  - `localStorage.setItem('nome', 'João');`
- Recuperar dados
  - `const nome = localStorage.getItem('nome');`
- Remover um item:
  - `localStorage.removeItem('nome');`
- Limpar tudo:
  - `localStorage.clear();`
- Descobrir quantos itens há no LocalStorage
  - `console.log(localStorage.length);`

# Observações importantes

- O LocalStorage só armazena strings. Se você quiser guardar objetos ou arrays, precisa converter para JSON:

// Para armazenar:

```
const user = { nome: 'João', idade: 30 };  
localStorage.setItem('user', JSON.stringify(user));
```

// Para recuperar:

```
const userSalvo = JSON.parse(localStorage.getItem('user'));  
console.log(userSalvo.nome); // João
```

# Diferença entre LocalStorage e SessionStorage

LocalStorage	SessionStorage
Dados persistem após fechar o navegador	Dados são apagados ao fechar o navegador
Compartilhado entre abas/janelas do mesmo domínio	Não é compartilhado entre abas
Mesmo limite de ~5MB	Limite semelhante