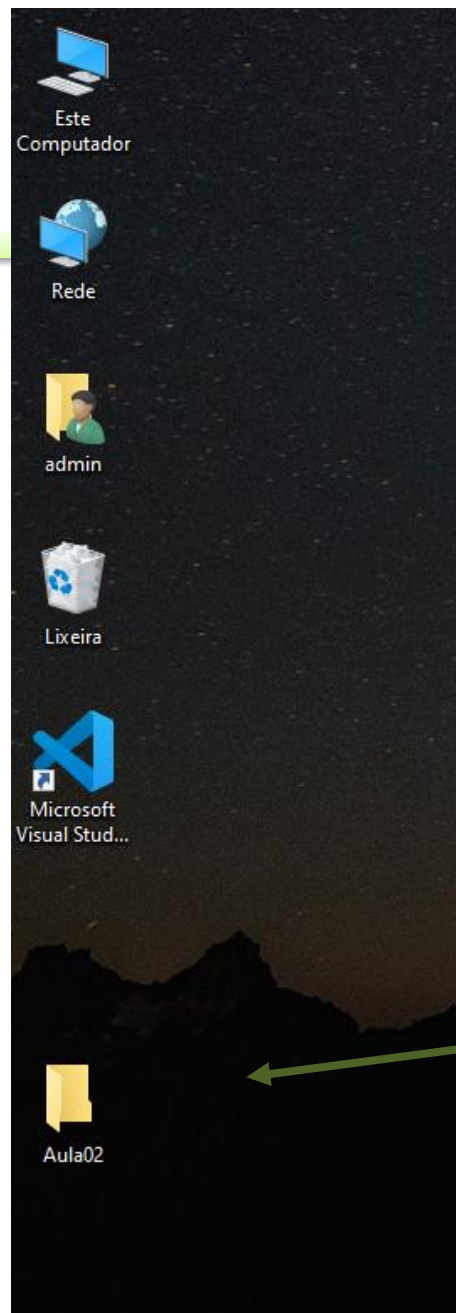


Estudo de Funções

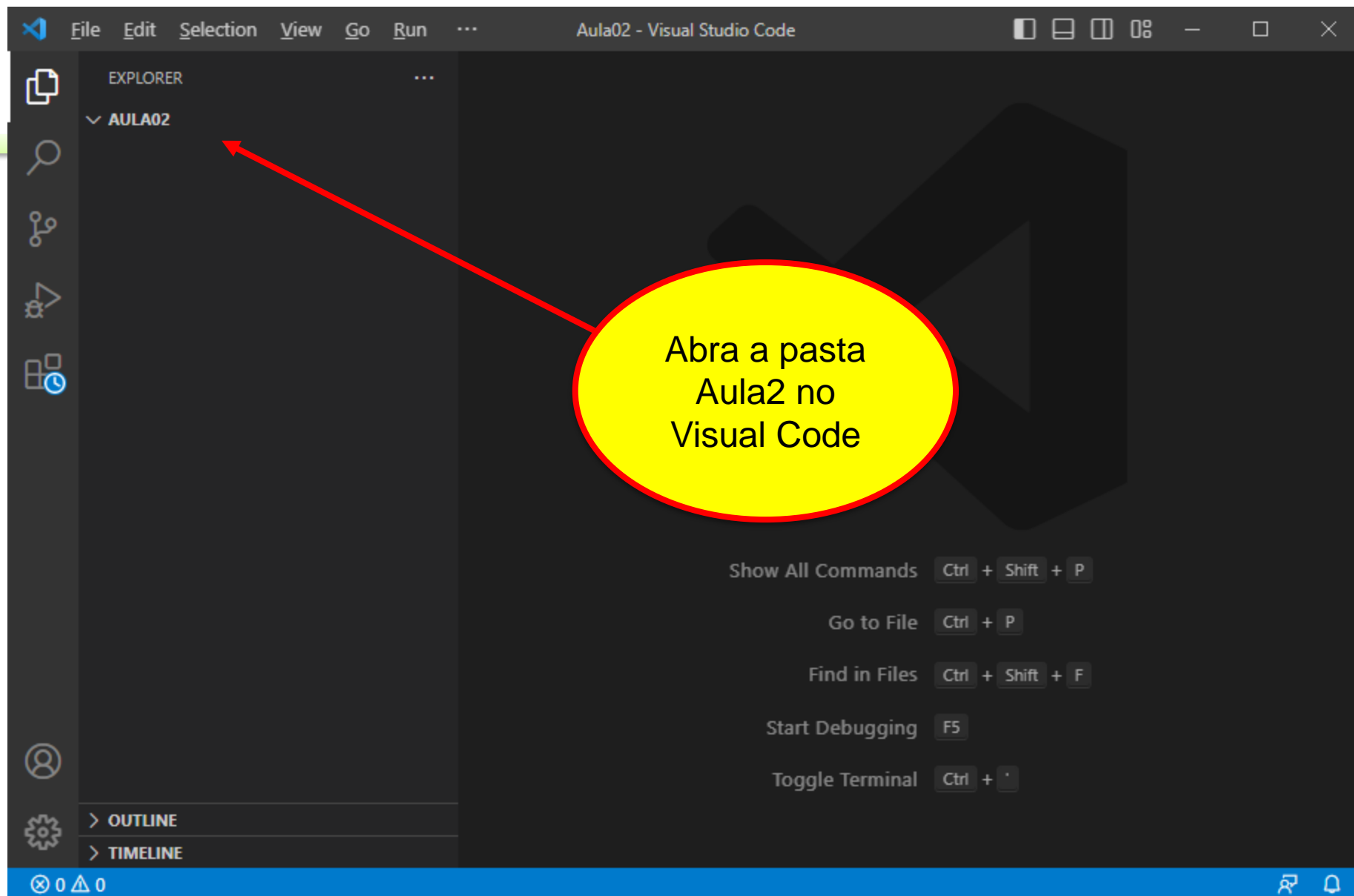
Guilherme Henrique de Souza

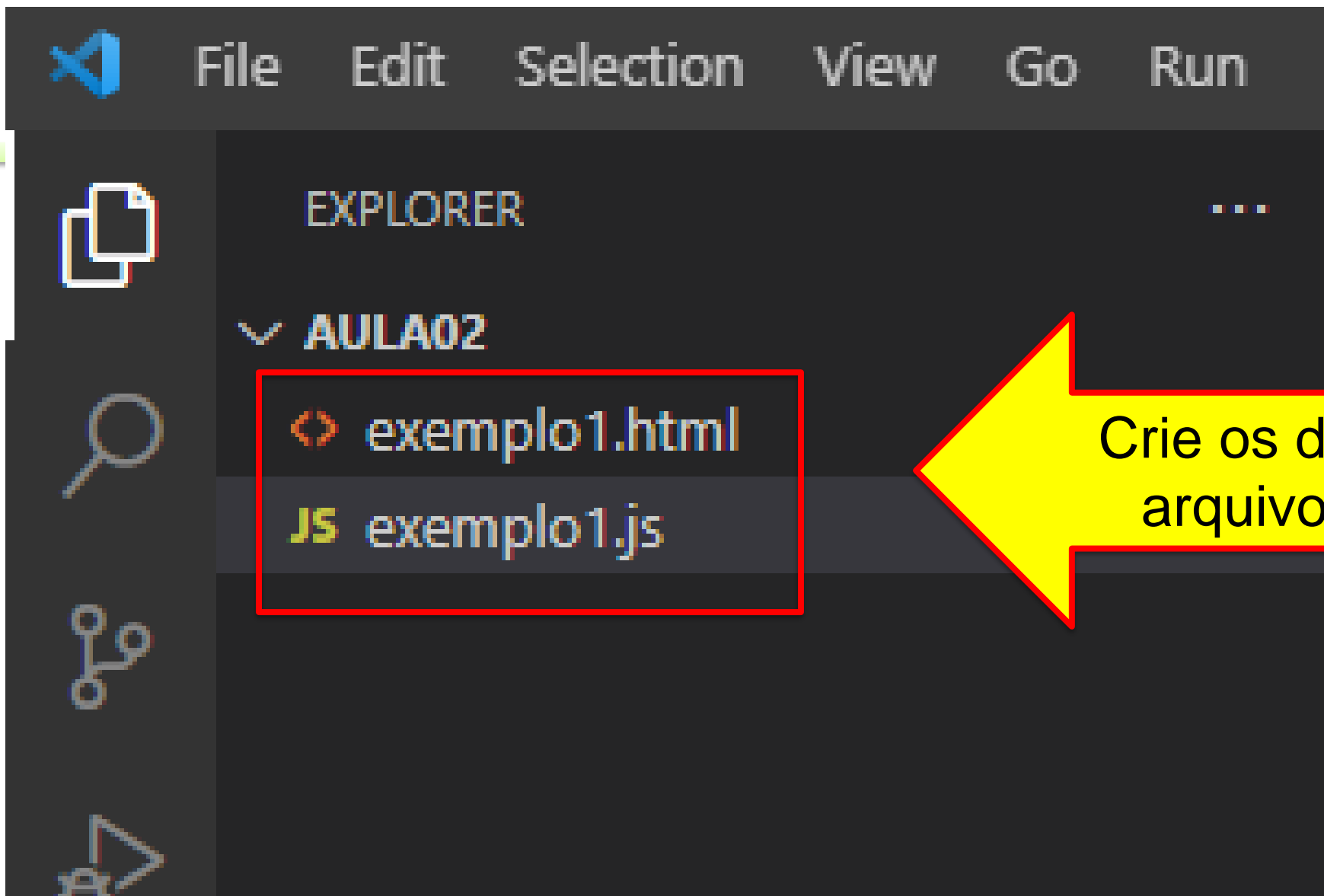
guilherme.souza@etec.sp.gov.br

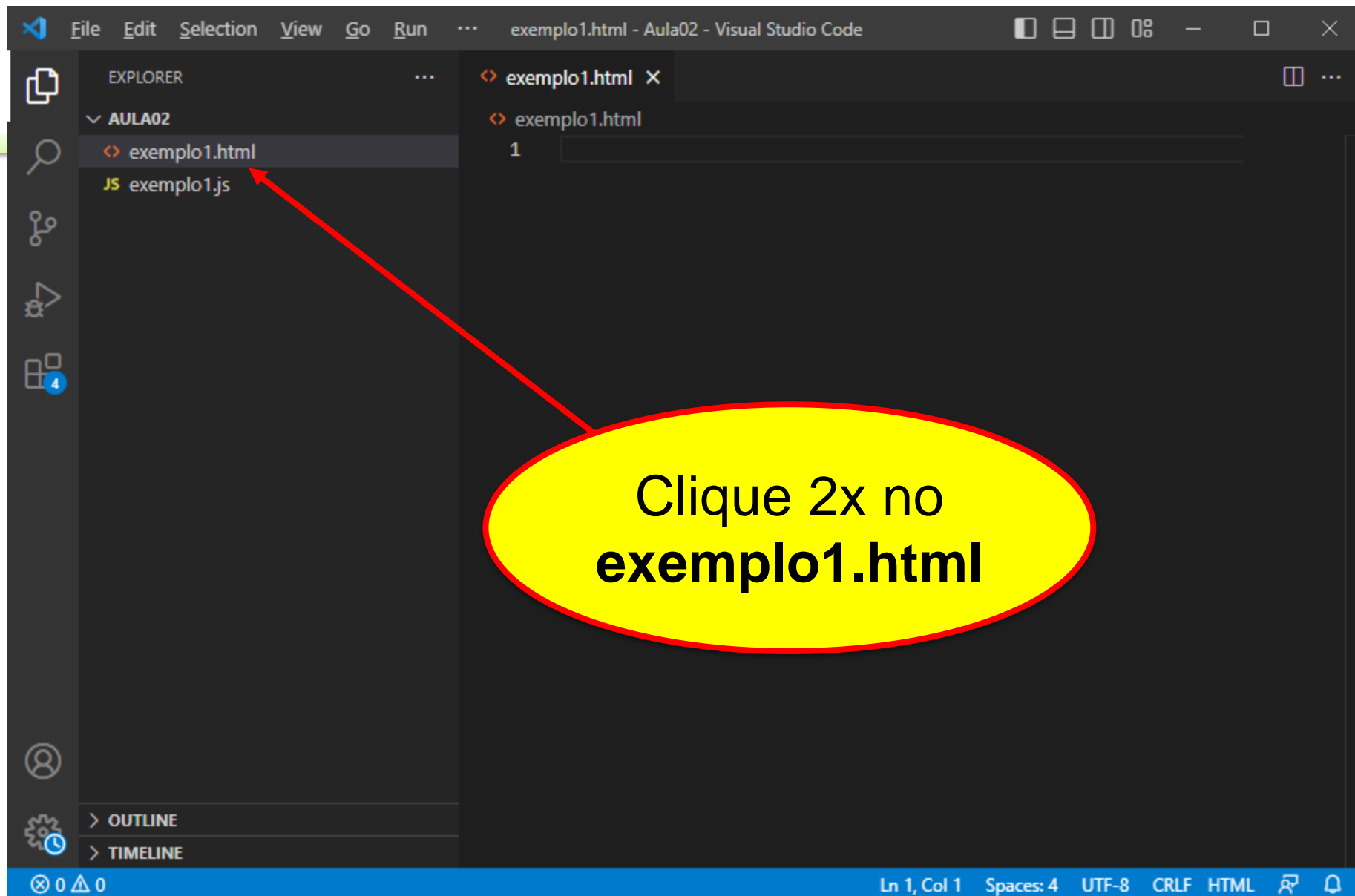
guilherme.souza183@fatec.sp.gov.br

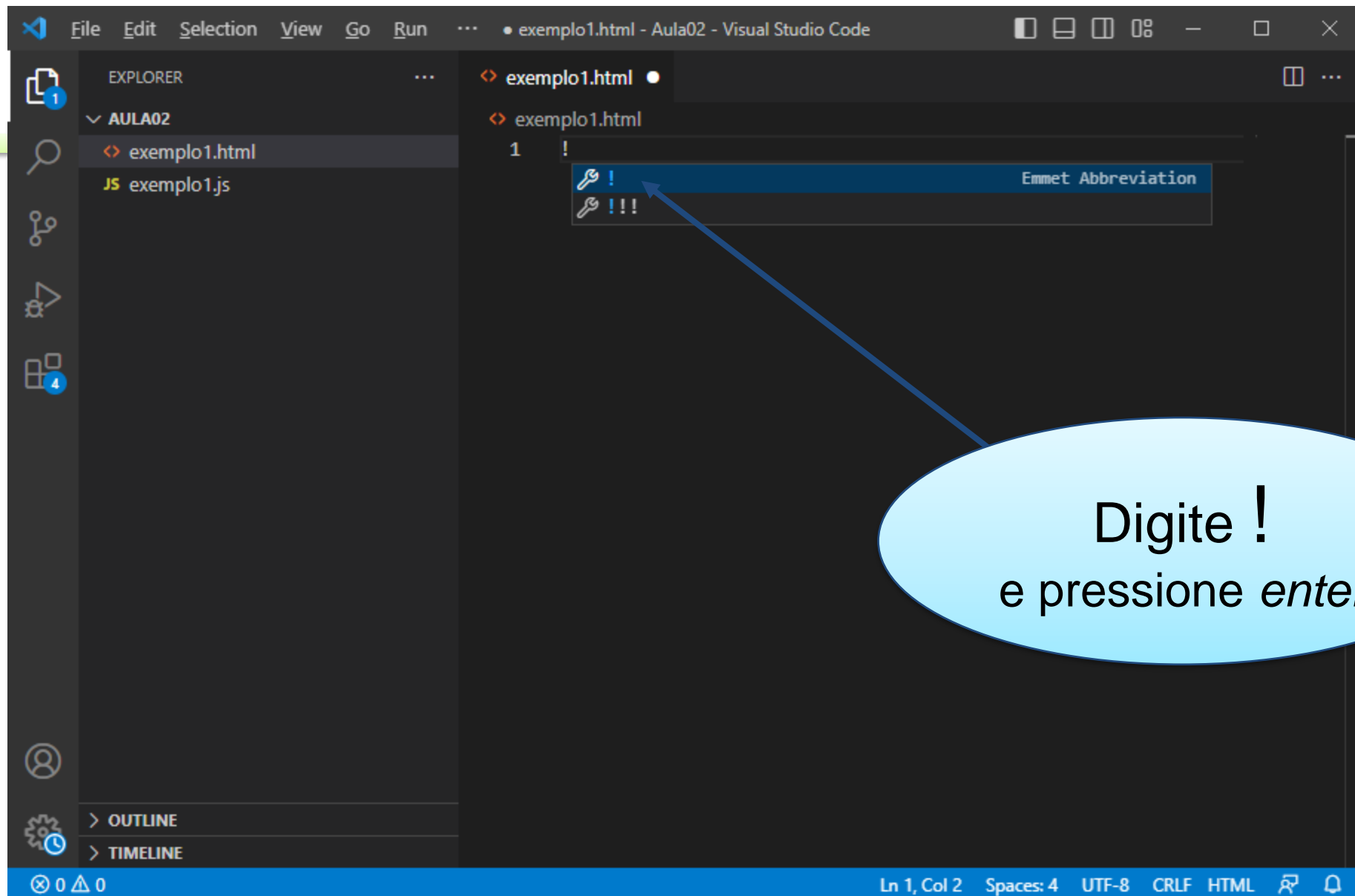


Criar a pasta
Aula2

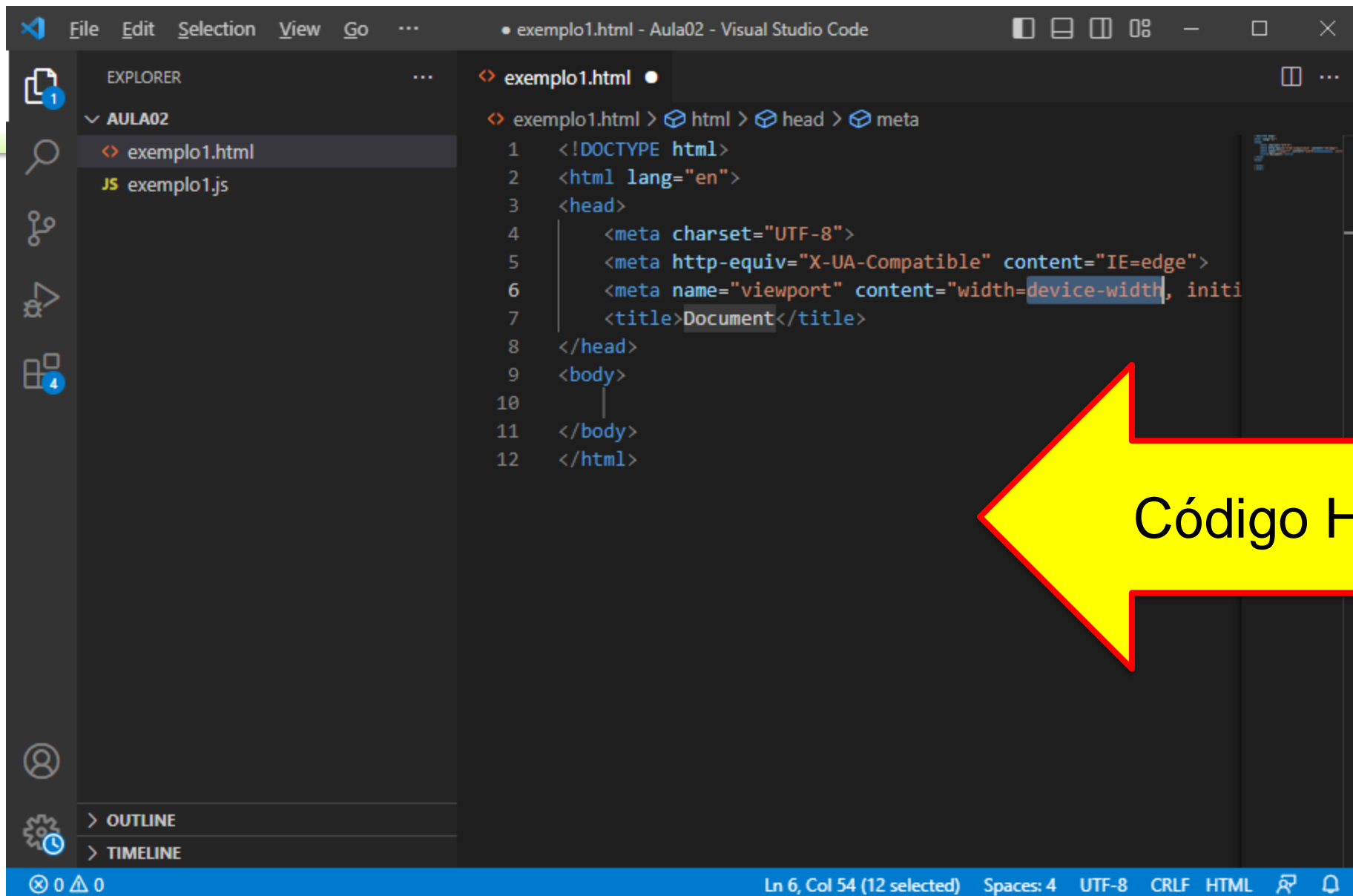








Digite !
e pressione *enter*



```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Document</title>
</head>
<body>
</body>
</html>
  
```

Ln 6, Col 54 (12 selected) Spaces: 4 UTF-8 CRLF HTML

Código HTML

```
</head>
```

```
<body>
```

```
<h1>Aula 02 - Funções</h1>
```

```
<p id="mensagem">
```

```
</p>
```

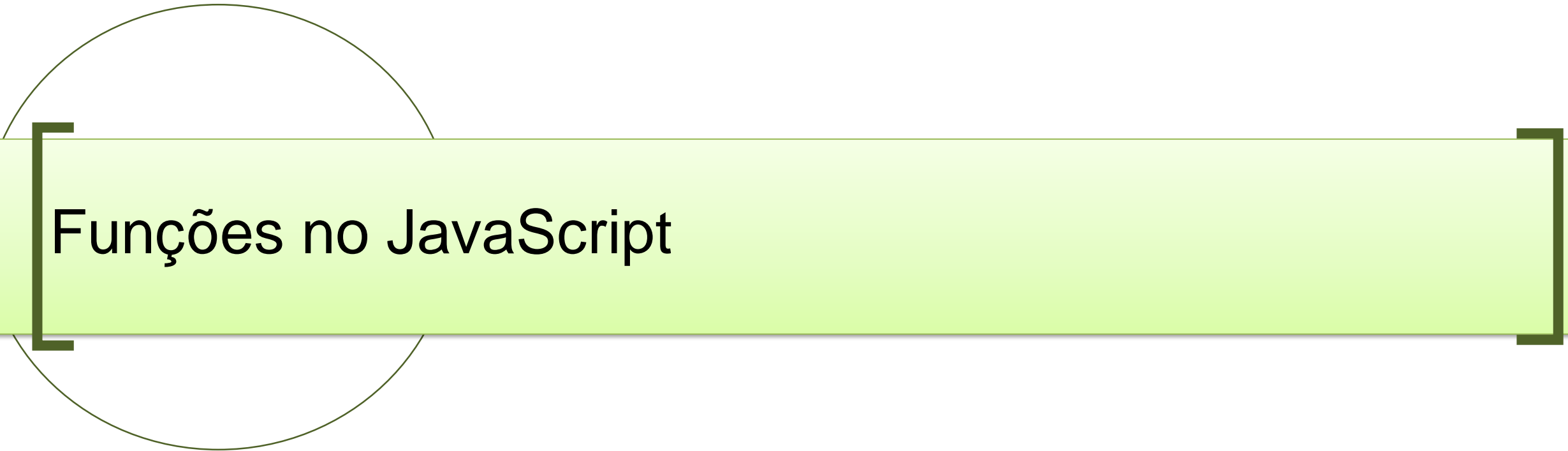
```
<script src="exemplo1.js"></script>
```

```
</body>
```

```
</html>
```



Implemente este código no HTML

A decorative graphic on the left side of the slide. It features a thin green circle that is partially obscured by a thick green bracket-like shape. The bracket starts from the left edge, goes up, then curves around the circle, and then goes down to the bottom edge. The top and bottom horizontal bars of the bracket extend across the width of the slide, framing the title.

Funções no JavaScript

Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

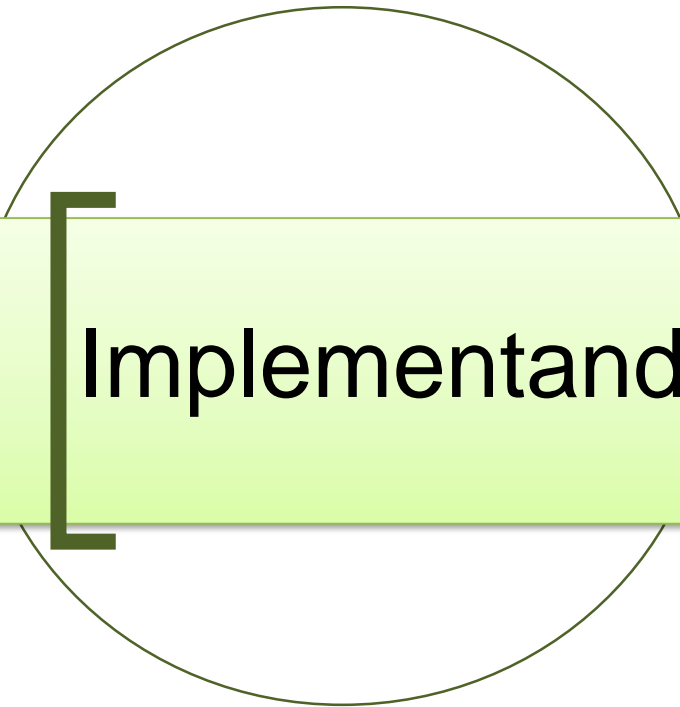
guilherme.souza183@fatec.sp.gov.br

Funções no JavaScript


■ O que é ?

- Uma função é um bloco de código reutilizável que executa uma tarefa específica.
- As funções podem ou não receber parâmetros (entradas), podem retornar um valor direto ou executar uma ação
- As funções ajudam a organizar e reutilizar o código, tornando-o mais modular e legível.

```
function somar(n1, n2) ...  
}  
  
function solicitarEntradaDados() ...  
}  
  
function processar() ...  
}
```



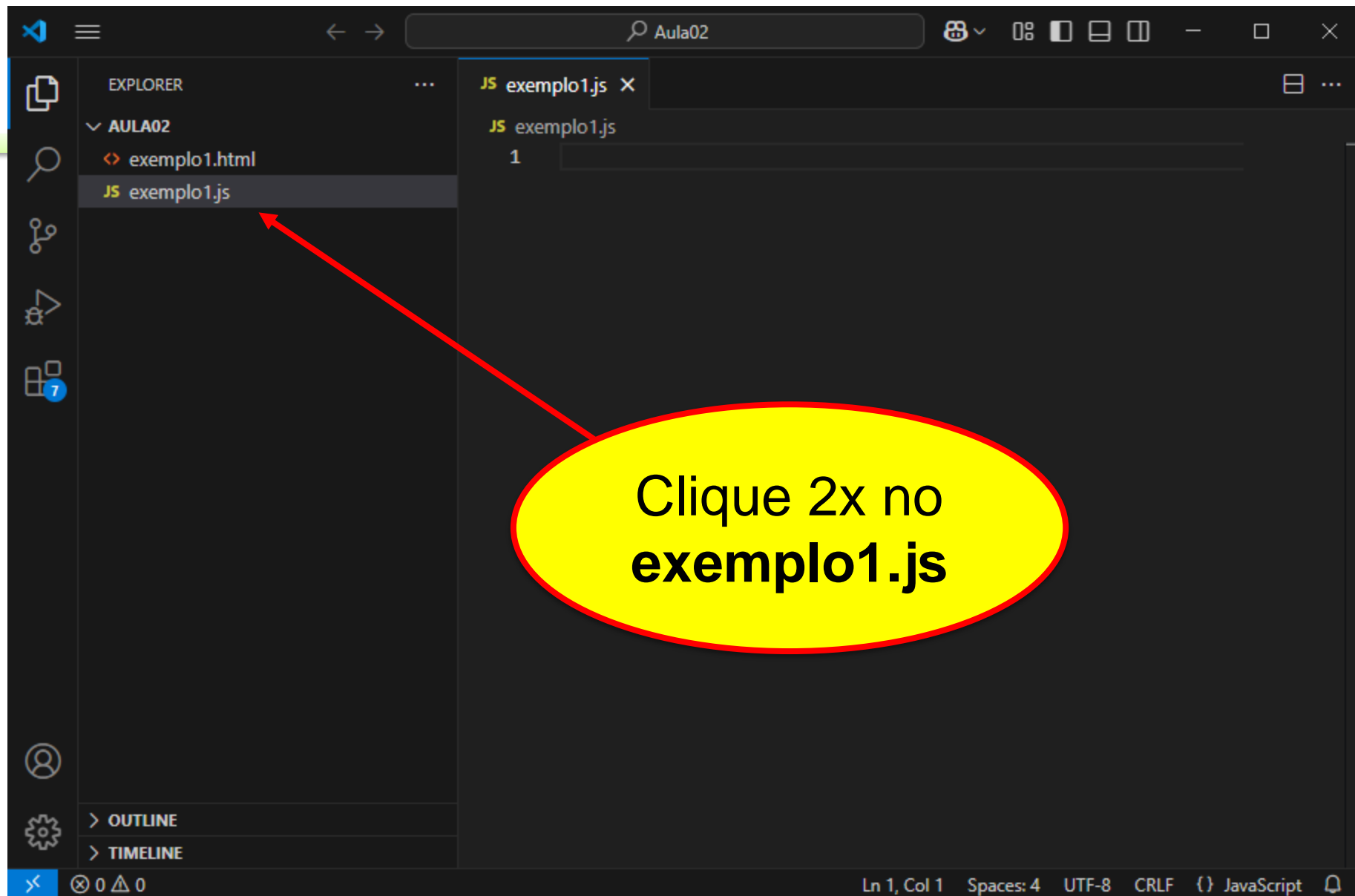
Implementando o Exemplo1.js



Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br



JS exemplo1.js X

JS exemplo1.js > ...

```
1 let numero1, numero2, resultado;
```

```
2
```

- Declaração das variáveis `numero1`, `numero2`, `resultado`
 - O `let` é uma palavra-chave usada para declarar variáveis com escopo de bloco (diferente de `var`, que tem escopo de função).

JS exemplo1.js X

JS exemplo1.js > somar

1 let numero1, numero2, resultado;

2

3 function somar(n1, n2)

4 {

5

6 }

■ Estrutura de uma Função em JavaScript

- A palavra-chave function é usada para definir uma função em JavaScript. Toda função em JavaScript começa com essa palavra-chave.
- Este é o nome da função. Você pode escolher qualquer nome válido para a função. O nome é como você irá referenciar e chamar a função em outras partes do código.
- (n1, n2):
- Entre os parênteses, são definidos os parâmetros da função. No caso, a função somar espera dois parâmetros: n1 e n2.

Parâmetros são como variáveis dentro da função, que recebem valores quando a função é

JS exemplo1.js X

JS exemplo1.js > somar

```
1  let numero1, numero2, resultado;
2
3  function somar(n1, n2)
4  {
5      let soma = Number(n1) + Number(n2);
6      document.getElementById("mensagem").innerHTML = "A soma dos números é: " + soma;
7  }
```

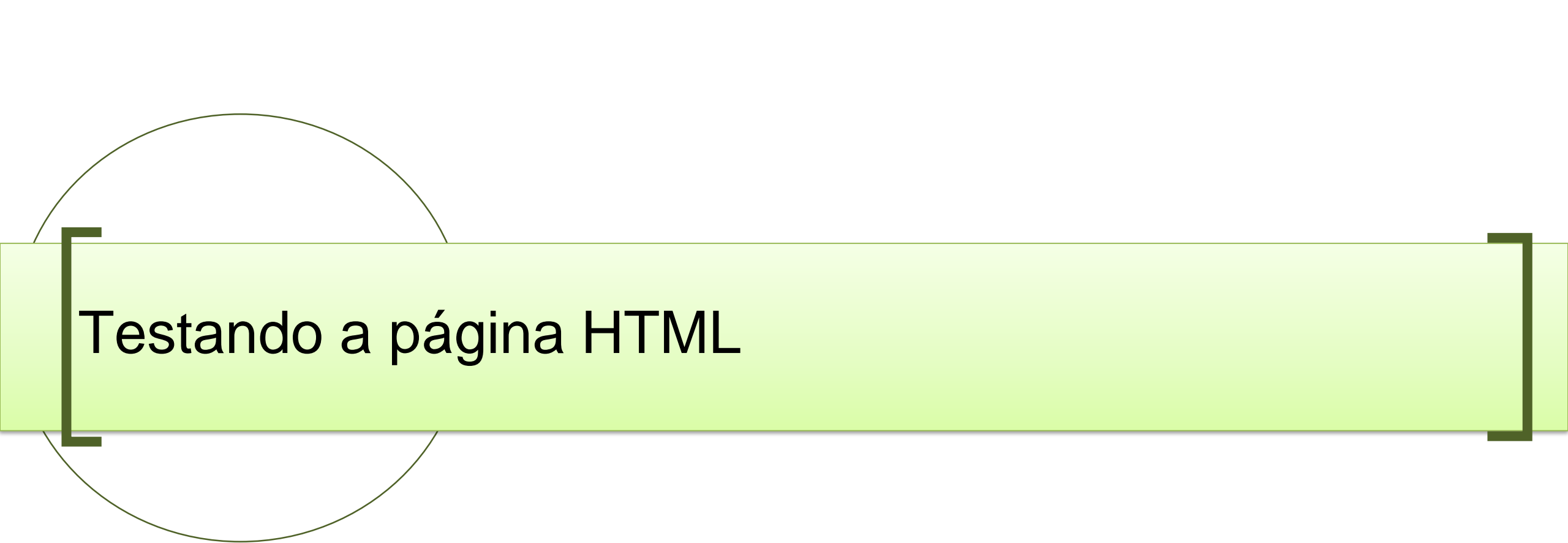
- Declaração das variáveis `numero1`, `numero2`, `resultado`
- O código acima define uma função `somar(n1, n2)` que recebe dois números como parâmetros.
 - Os parâmetros serão convertidos para números usando `Number()` e a soma será calculada.
 - Exibe o resultado na página HTML, inserindo o texto "A soma dos números é: " seguido da soma na tag com o ID `"mensagem"`

JS exemplo1.js X

JS exemplo1.js > ...

```
1  let numero1, numero2, resultado;  
2  
3  function somar(n1, n2)  
4  {  
5      let soma = Number(n1) + Number(n2);  
6      document.getElementById("mensagem").innerHTML = "A soma dos números é: " + soma;  
7  }  
8  
9  somar();
```

Chamada da Função

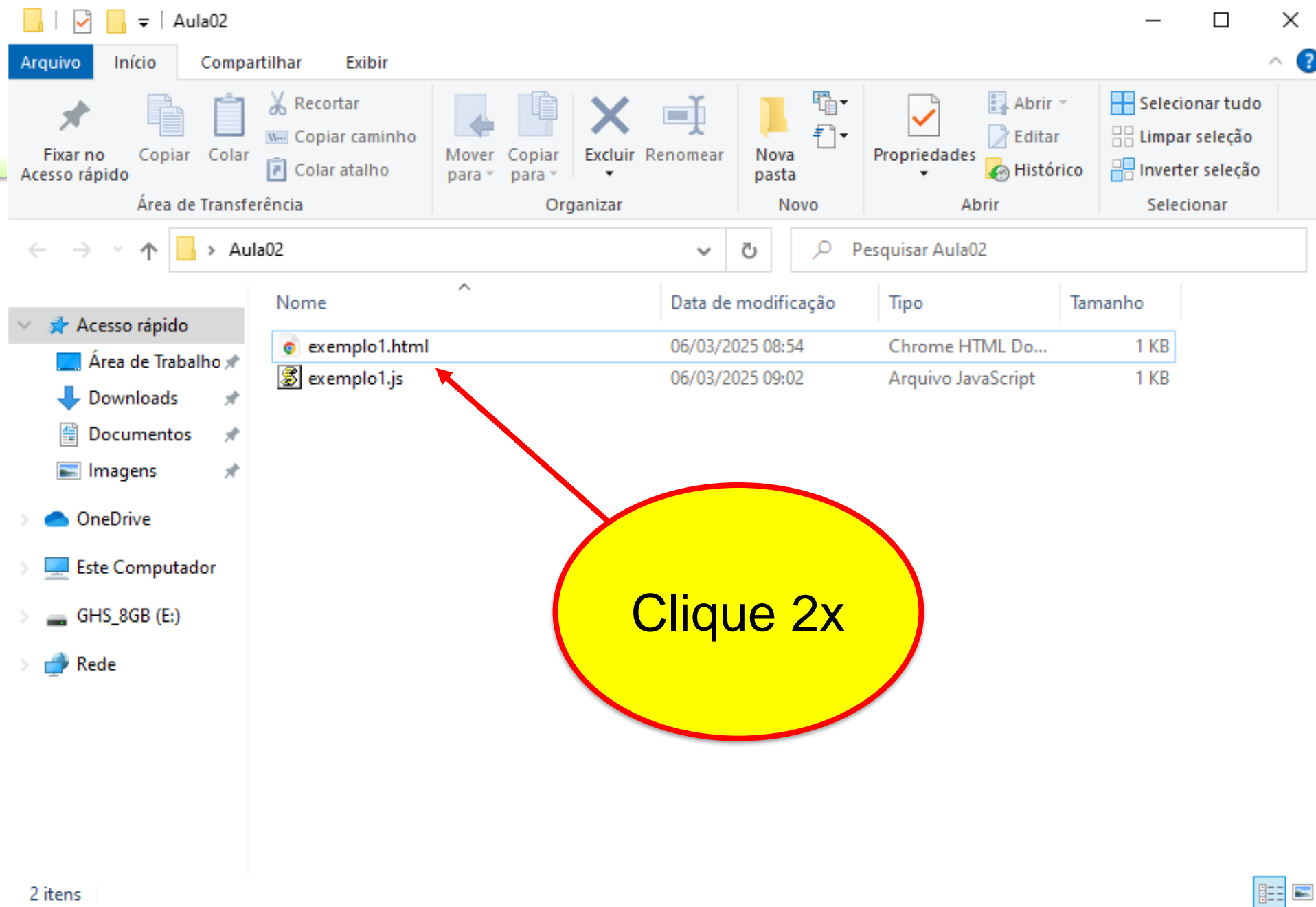


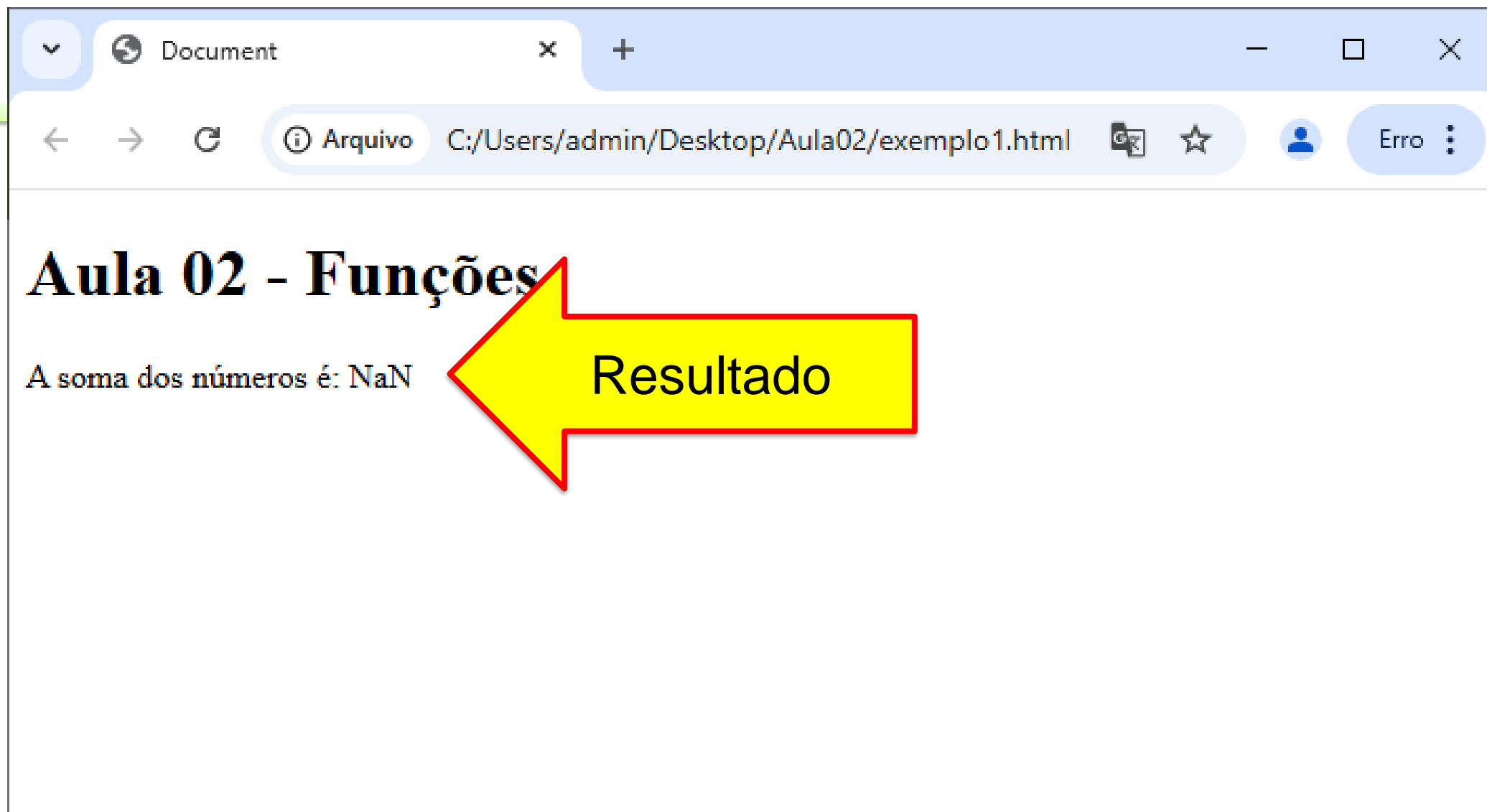
Testando a página HTML

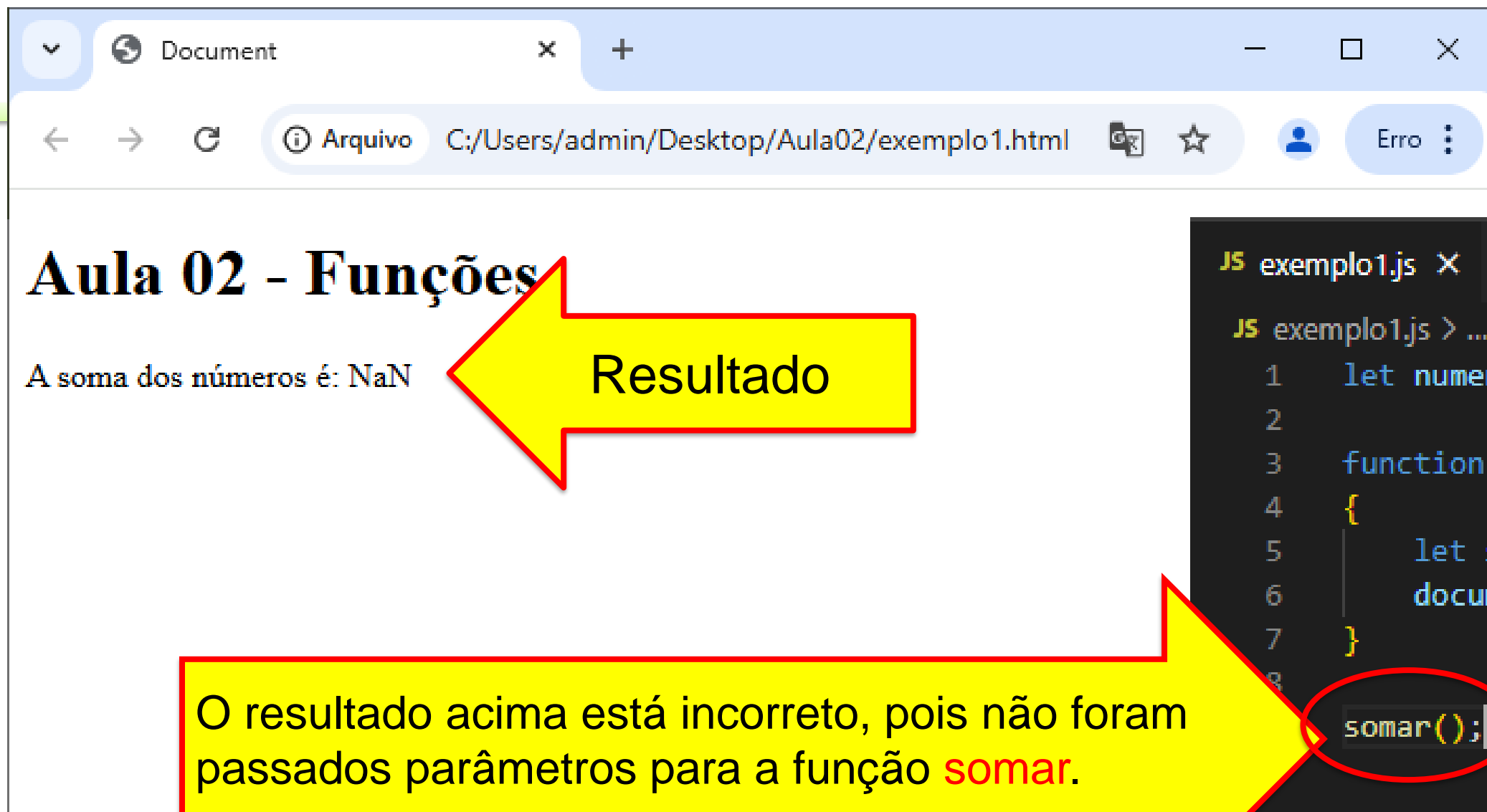
Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br







The image shows a web browser window with a single tab titled 'Document'. The address bar displays the file path 'C:/Users/admin/Desktop/Aula02/exemplo1.html'. Below the address bar, the page content reads 'Aula 02 - Funções' followed by 'A soma dos números é: NaN'. A large yellow arrow with a red border points from the text 'Resultado' to the 'NaN' value. In the bottom right corner, a code editor window titled 'JS exemplo1.js' is open, showing the following JavaScript code:

```
JS exemplo1.js > ...  
1 let numero1, numero2;  
2  
3 function somar(a, b) {  
4  
5     let soma = a + b;  
6     document.getElementById('resultado').innerHTML = soma;  
7 }  
8  
9 somar();
```

The last line of code, `somar();`, is circled in red. A second large yellow arrow with a red border points from the text 'O resultado acima está incorreto, pois não foram passados parâmetros para a função **somar**.' to the circled `somar();` line.



Colocando os parâmetros na função “somar”

Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br

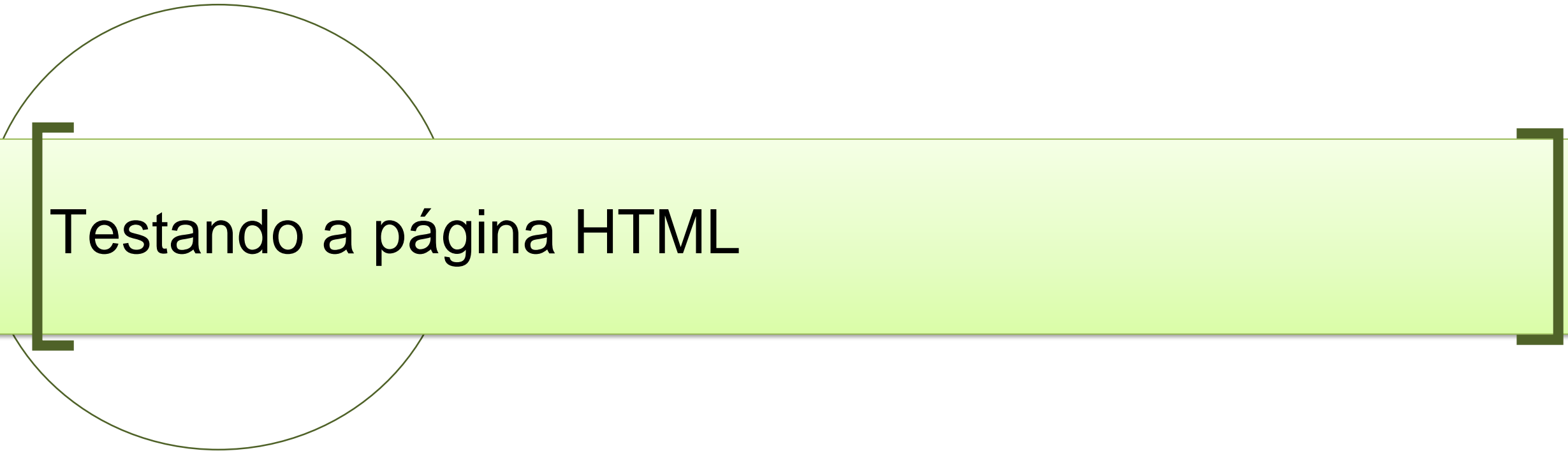
JS exemplo1.js X

JS exemplo1.js > ...

```
1  let numero1, numero2, resultado;
2
3  function somar(n1, n2)
4  {
5      let soma = Number(n1) + Number(n2);
6      document.getElementById("mensagem").innerHTML = "A soma dos números é: " + soma;
7  }
8
9  somar(10,20);
```

Passagem de parâmetros
para a função

- Aqui fazemos a chamada da função “somar”
 - Note que foram passados dois valores inteiros: 10 e 20

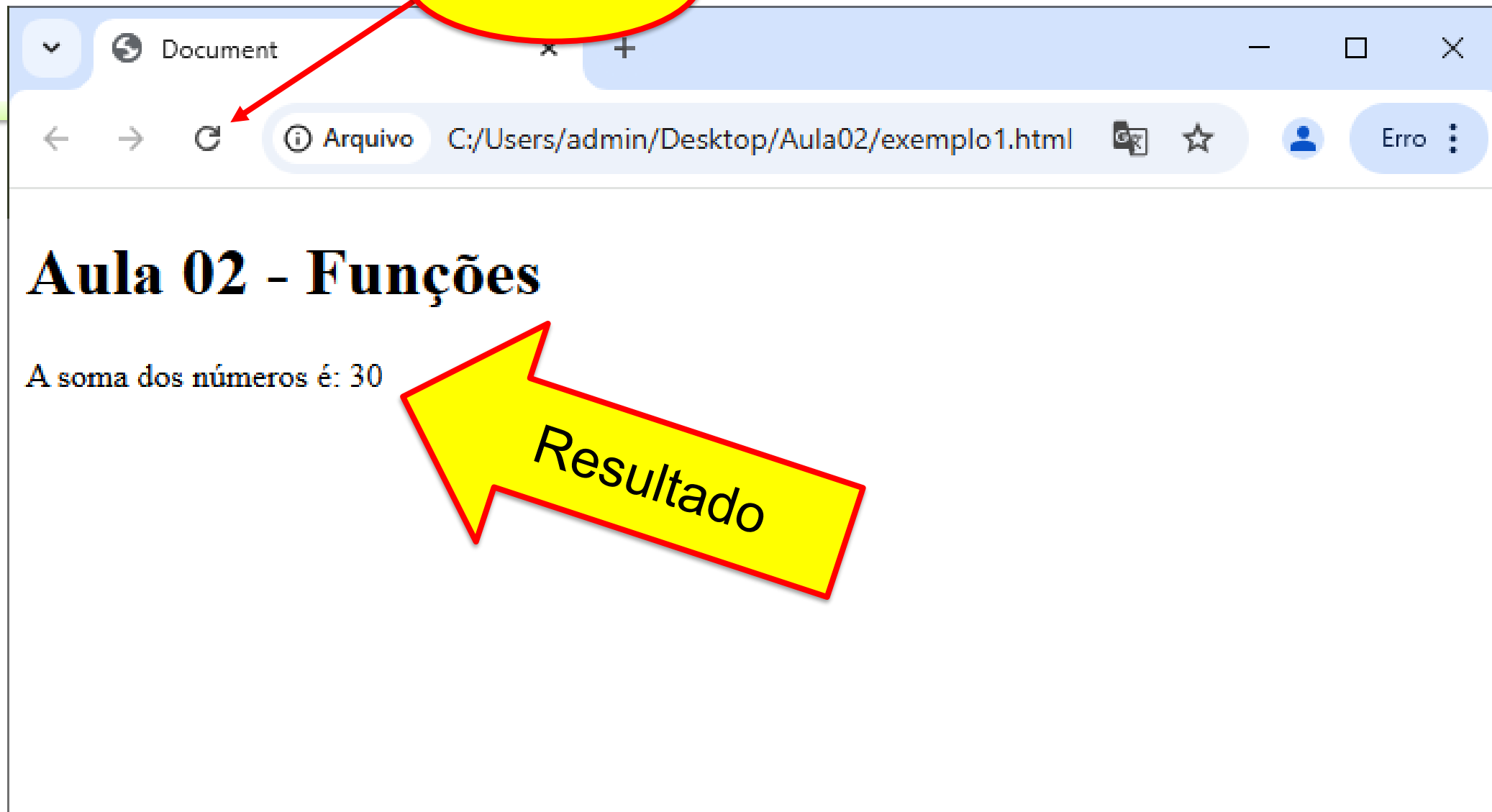
A decorative graphic on the left side of the slide, featuring a thin green circle and a thick green bracket-like shape that frames the title area.

Testando a página HTML

Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br



JS exemplo1.js X

JS exemplo1.js > ...

```
1  let numero1, numero2, resultado;
2
3  function somar(n1, n2)
4  {
5      let soma = Number(n1) + Number(n2);
6      document.getElementById("mensagem").innerHTML = "A soma dos números é: " + soma;
7  }
8
9  function solicitarEntradaDados()
10 {
```



Função: solicitarEntradaDados()

JS exemplo1.js X

JS exemplo1.js > ...

```
1  let numero1, numero2, resultado;
2
3  function somar(n1, n2)
4  {
5      let soma = Number(n1) + Number(n2);
6      document.getElementById("mensagem").innerHTML = "A soma dos números é: " + soma;
7  }
8
9  function solicitarEntradaDados()
10 {
11     numero1 = Number(prompt("digite o número 1:"));
12     numero2 = Number(prompt("digite o número 2:"));
13
14     somar(numero1, numero2);
15
16 }
17
18 solicitarEntradaDados();
```

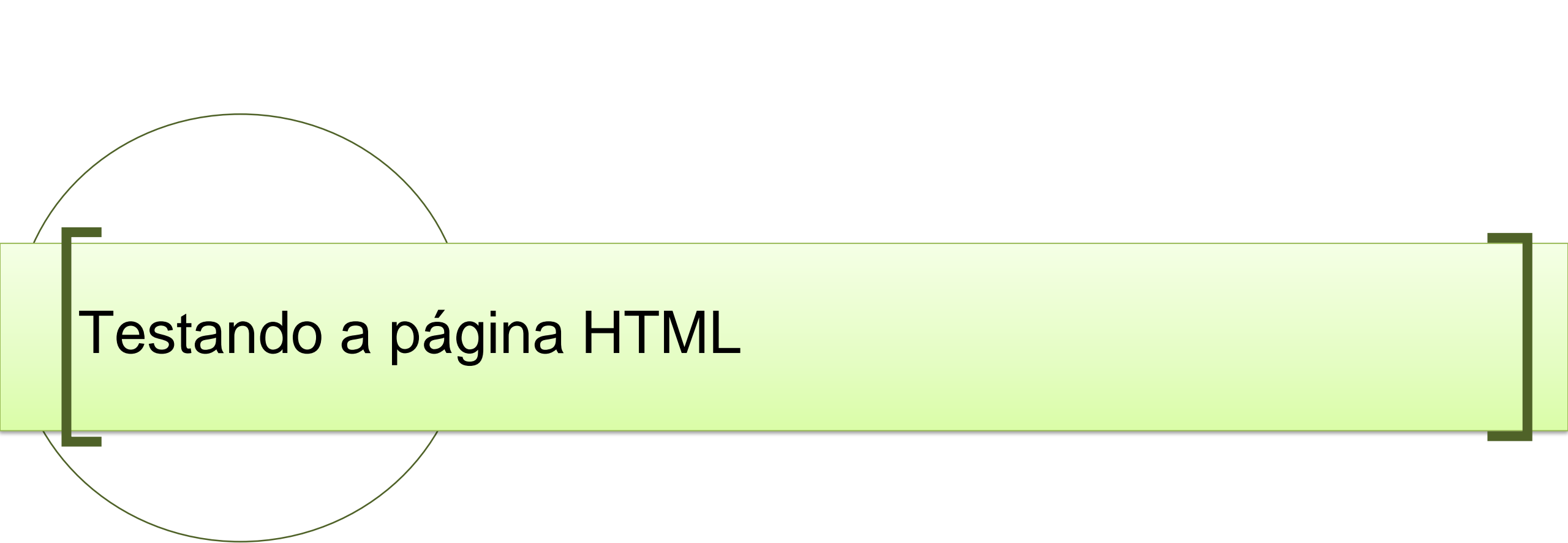
 Função: solicitarEntradaDados() Chamada da função

```
7 }  
8  
9 function solicitarEntradaDados()  
10 {  
11     numero1 = Number(prompt("digite o número 1:"));  
12     numero2 = Number(prompt("digite o número 2:"));  
13  
14     somar(numero1, numero2);  
15  
16 }  
17  
18 solicitarEntradaDados();
```

Função: solicitarEntradaDados()

Chamada da função

- O código acima define uma função `solicitarEntradaDados()` que não tem parâmetros de entrada.
 - Os valores são solicitados ao usuário a partir de uma caixa de alerta (função `prompt`)
 - Os valores recebidos são convertidos utilizando a função `Number()` e a soma será calculada.
 - Para realizar a soma, os valores recebidos são passados como parâmetros para a função `somar()`. Note que são passado a variável `numero1` e a `numero2`

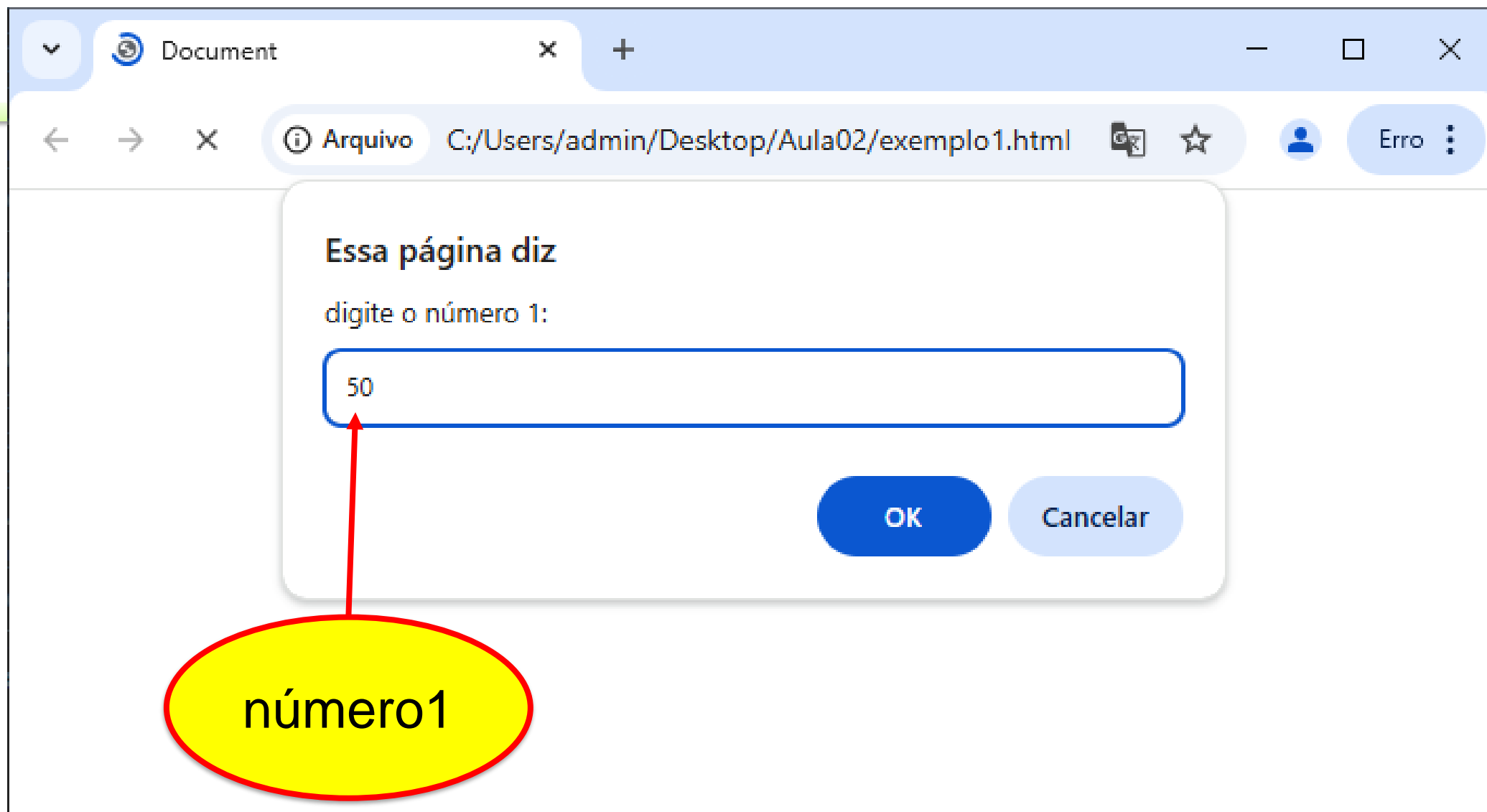


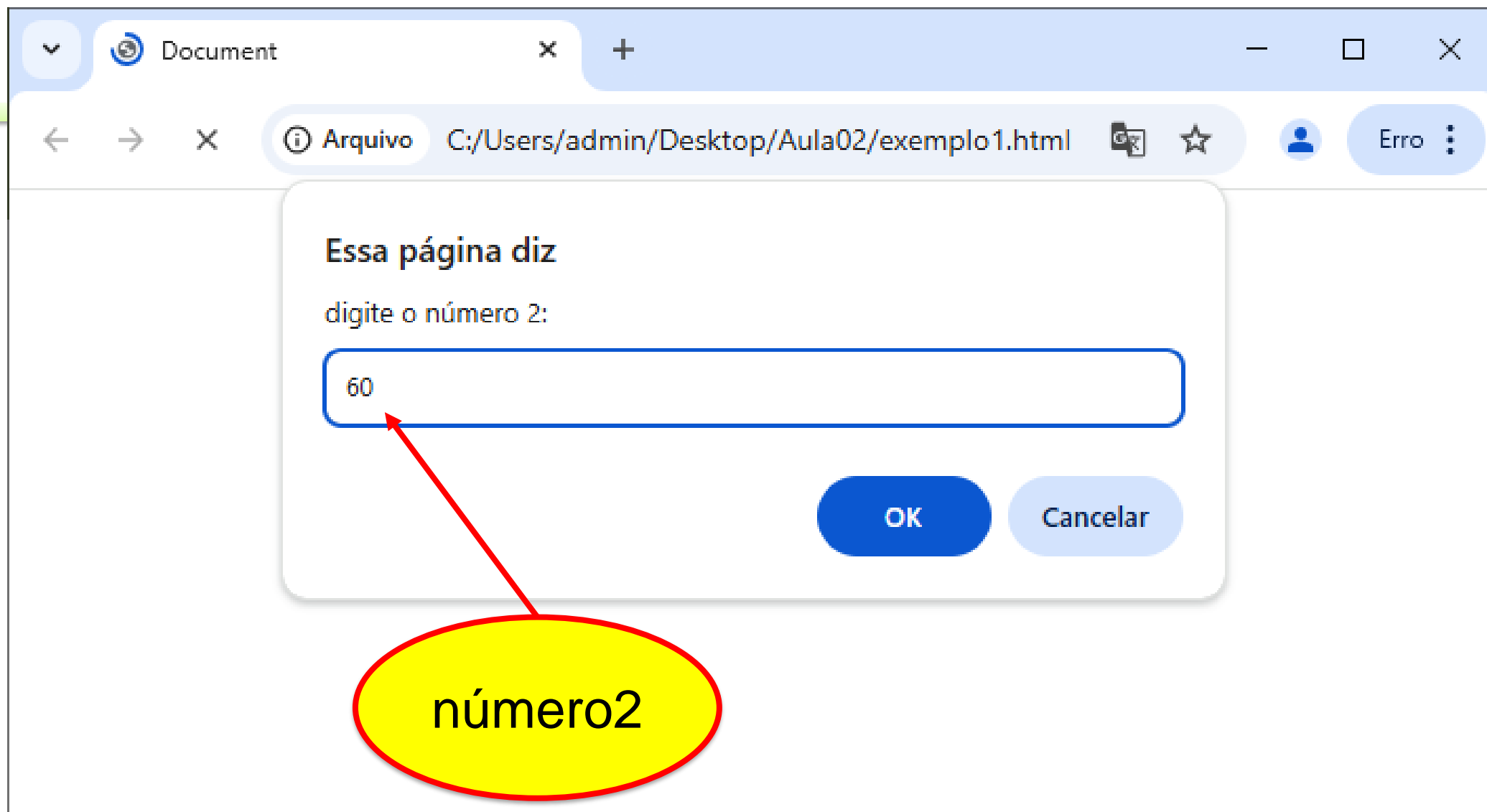
Testando a página HTML

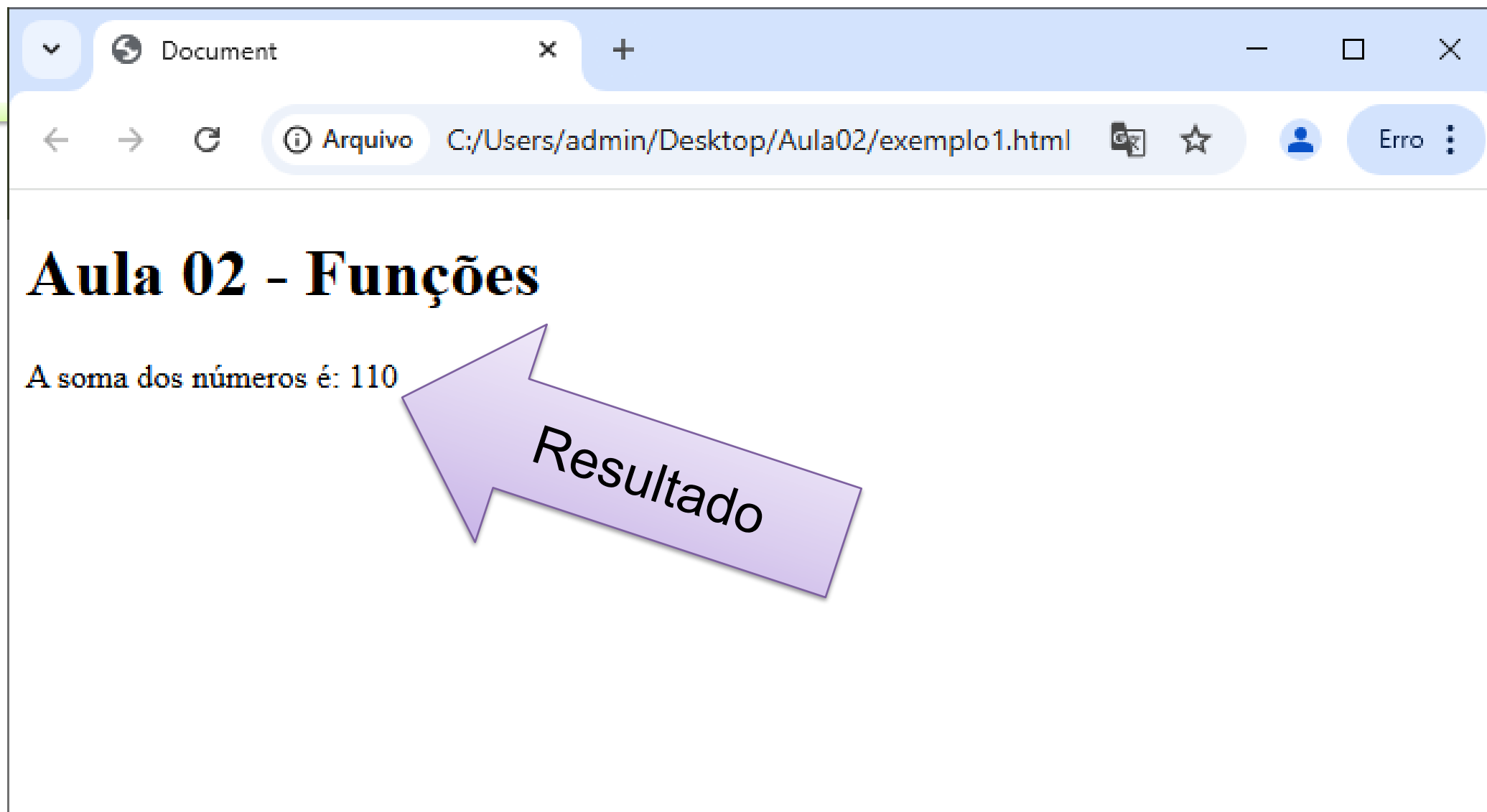
Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br







A decorative graphic on the left side of the slide, consisting of a thin green circle and a thick green bracket-like shape that frames the main title.

Excluindo a chamada da ultima função

Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br

JS exemplo1.js X

JS exemplo1.js > ...

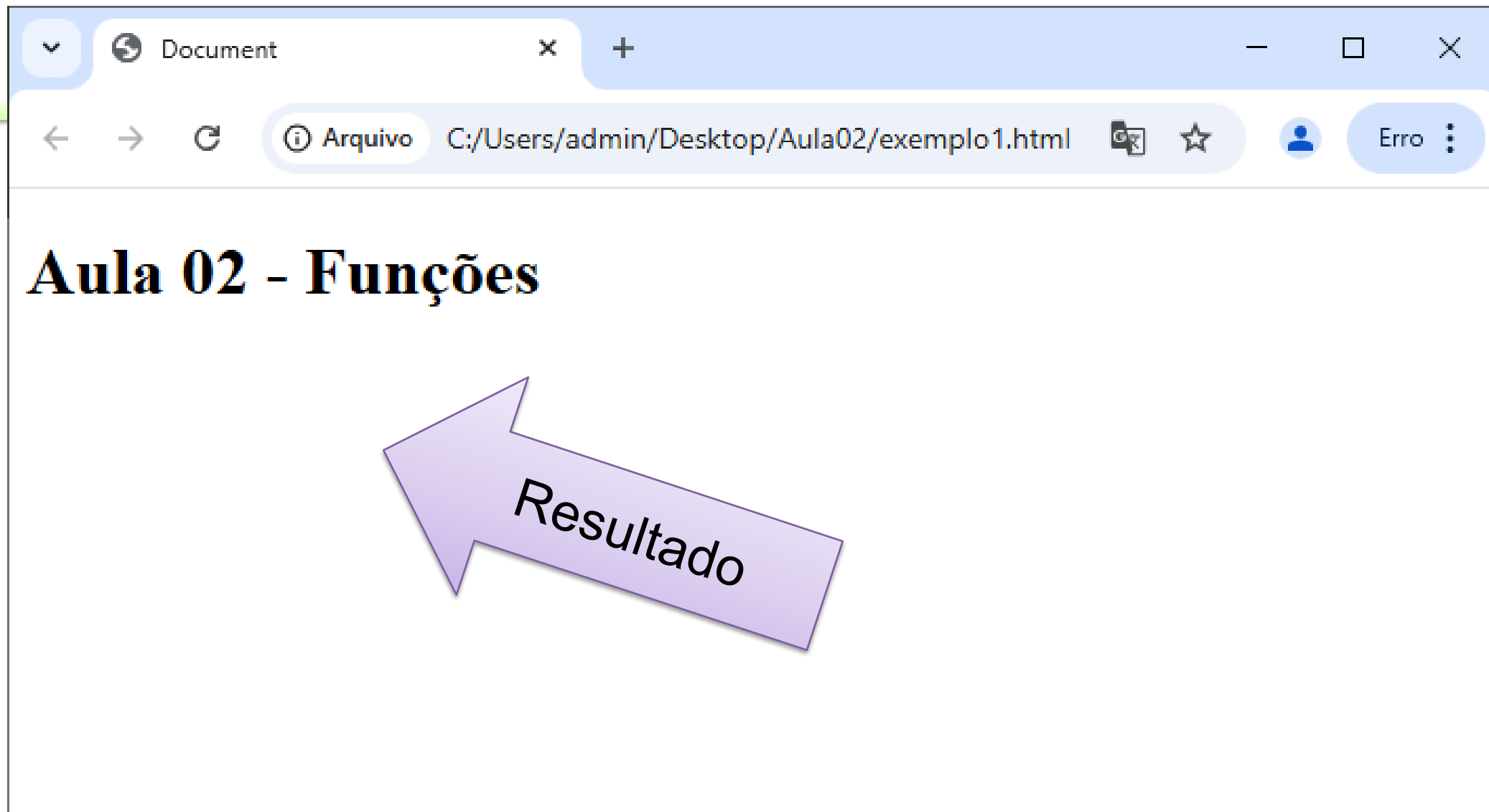
```
1  let numero1, numero2, resultado;
2
3  function somar(n1, n2)
4  {
5      let soma = Number(n1) + Number(n2);
6      document.getElementById("mensagem").innerHTML = "A soma dos números é: " + soma;
7  }
8
9  function solicitarEntradaDados()
10 {
11     numero1 = Number(prompt("digite o número 1:"));
12     numero2 = Number(prompt("digite o número 2:"));
13
14     somar(numero1, numero2);
15
16 }
17
18 solicitarEntradaDados();
```

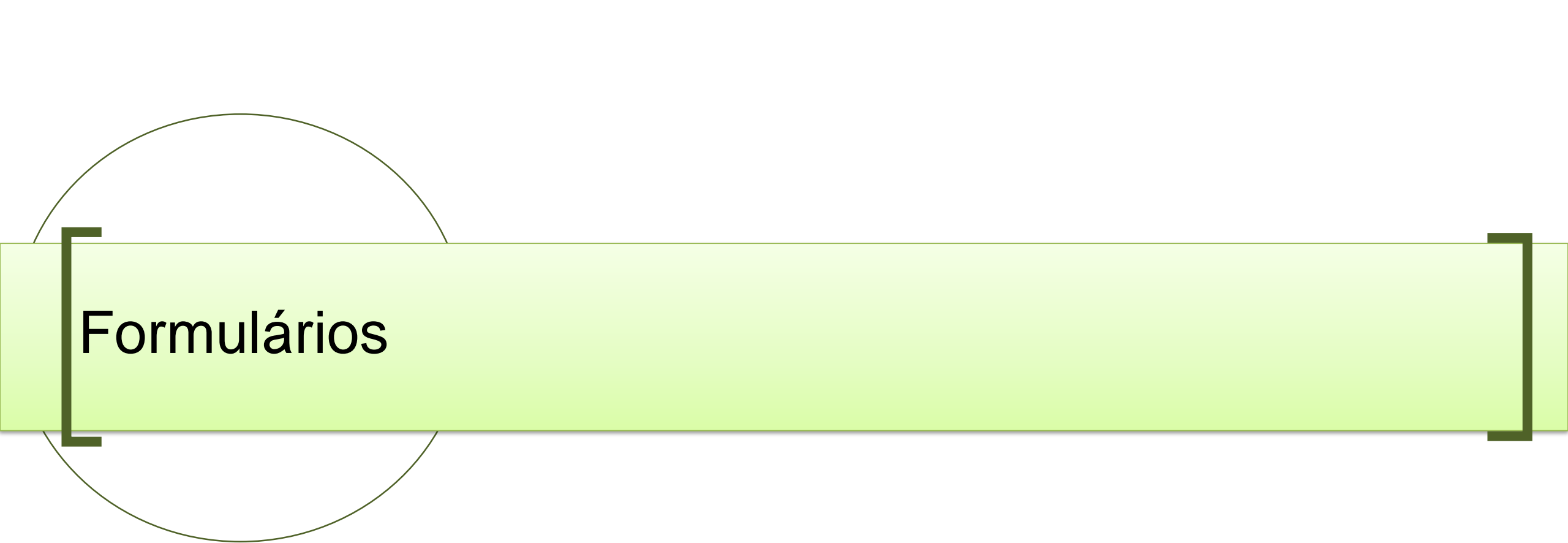
Exclua a
chamada da
função

```
JS exemplo1.js X
JS exemplo1.js > ...
1  let numero1, numero2, resultado;
2
3  function somar(n1, n2)
4  {
5      let soma = Number(n1) + Number(n2);
6      document.getElementById("mensagem").innerHTML = "A soma dos números é: " + soma;
7  }
8
9  function solicitarEntradaDados()
10 {
11     numero1 = Number(prompt("digite o número 1:"));
12     numero2 = Number(prompt("digite o número 2:"));
13
14     somar(numero1, numero2);
15
16 }
17
18
```

Não haverá chamada por aqui

Ln 18, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript



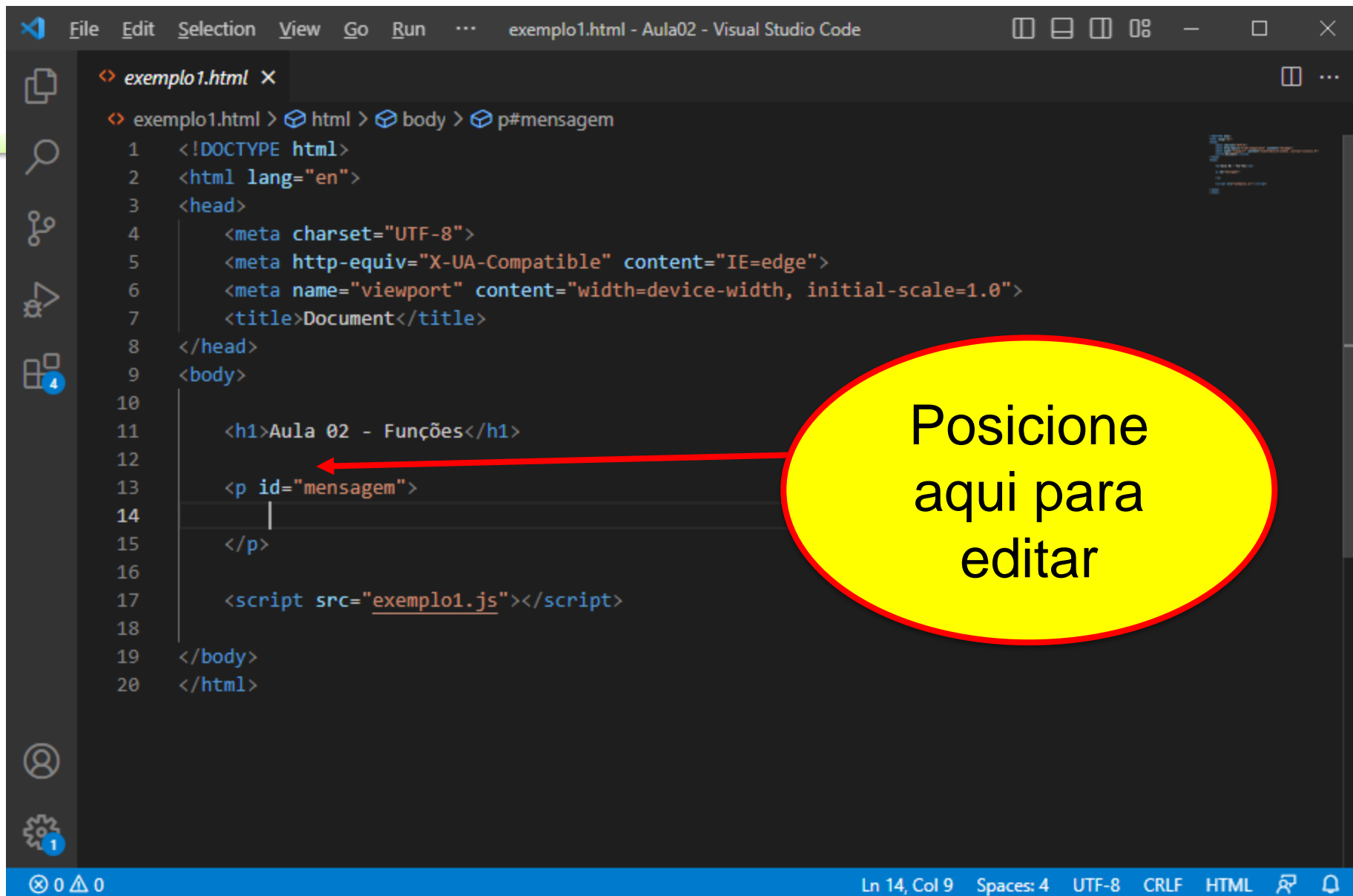


Formulários

Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br



```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10
11      <h1>Aula 02 - Funções</h1>
12
13      <p id="mensagem">
14
15      </p>
16
17      <script src="exemplo1.js"></script>
18
19  </body>
20  </html>
    
```

Posicione aqui para editar

```
</head>
<body>

  <h1>Aula 02 - Funções</h1>

  <form>
    |
  </form>

  <p id="mensagem">

  </p>

  <script src="exemplo1.js"></script>

</body>
</html>
```



Abre a tag do formulário

```
<body>

  <h1>Aula 02 - Funções</h1>

  <form>

    <p>
      digite o número 1:<br>
      <input type="text" id="numero1">
    </p>

  </form>

  <p id="mensagem">

  </p>

  <script src="exemplo1.js"></script>

</body>
</html>
```

Vamos incluir este código

```
<body>

  <h1>Aula 02 - Funções</h1>

  <form>

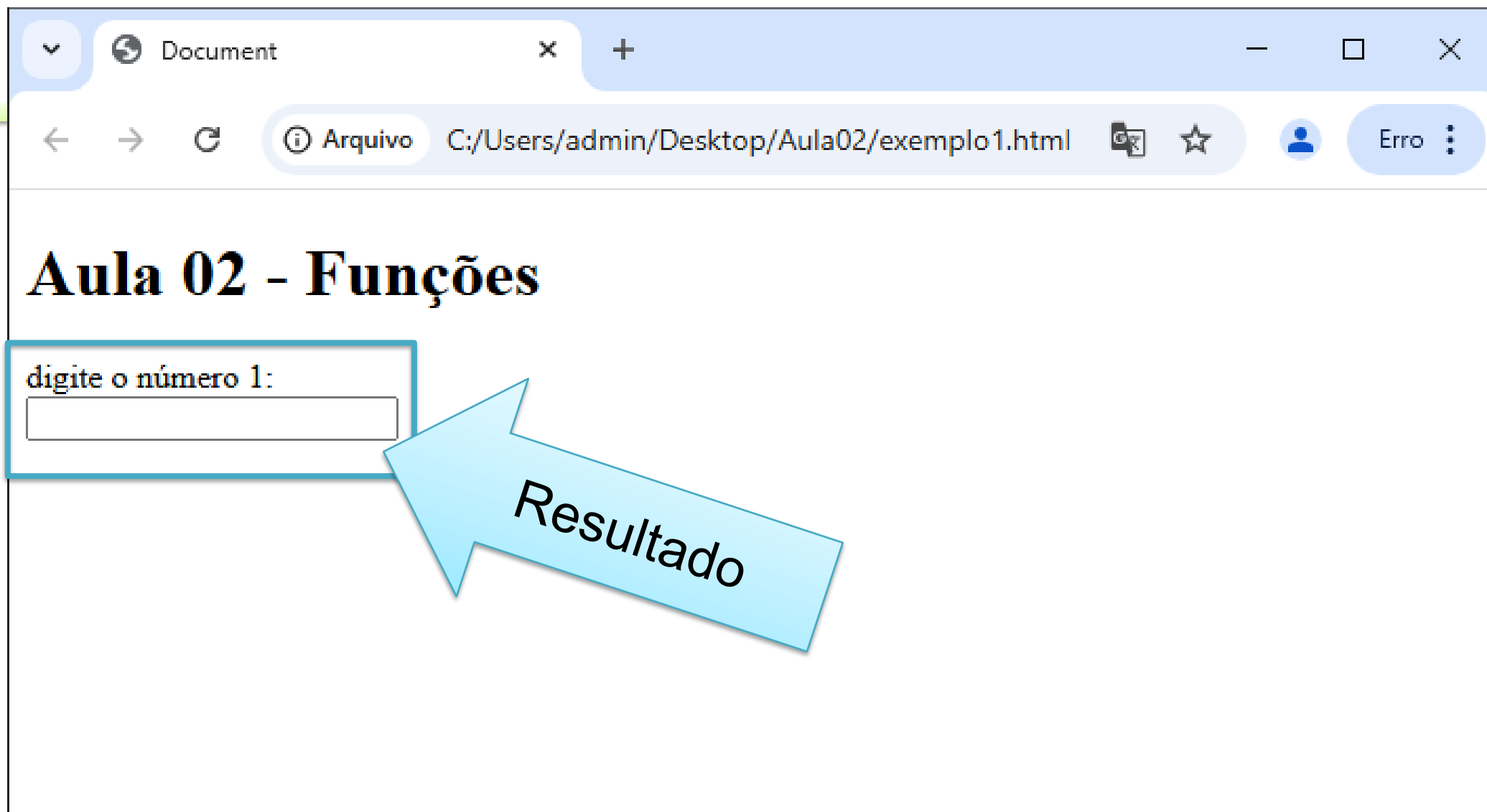
    <p>
      digite o número 1:<br>
      <input type="text" id="numero1">
    </p>

  </form>
```

Vamos incluir este código

■ Dentro do formulário <form>

- Vamos adicionar um campo de entrada de texto utilizando `<input type="text">`
- O `<input>` possui o `id="numero1"`, permitindo que o JavaScript acesse o valor digitado.
- O texto `"digite o número 1:"` aparece antes do campo, e o `
` adiciona uma quebra de linha.



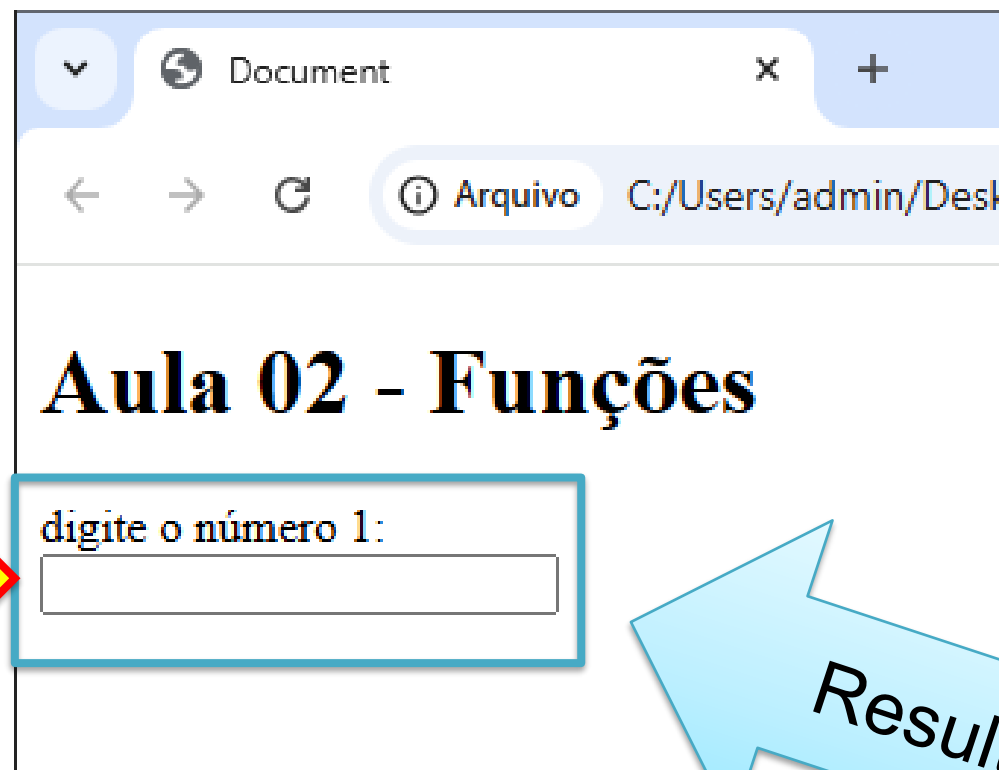
```
<body>

  <h1>Aula 02 - Funções</h1>

  <form>

    <p>
      digite o número 1:<br>
      <input type="text" id="numero1">
    </p>

  </form>
```



<h1>Aula 02 - Funções</h1>

<form>

<p>

digite o número 1:

<input type="text" id="numero1">

</p>

<p>

digite o número 2:

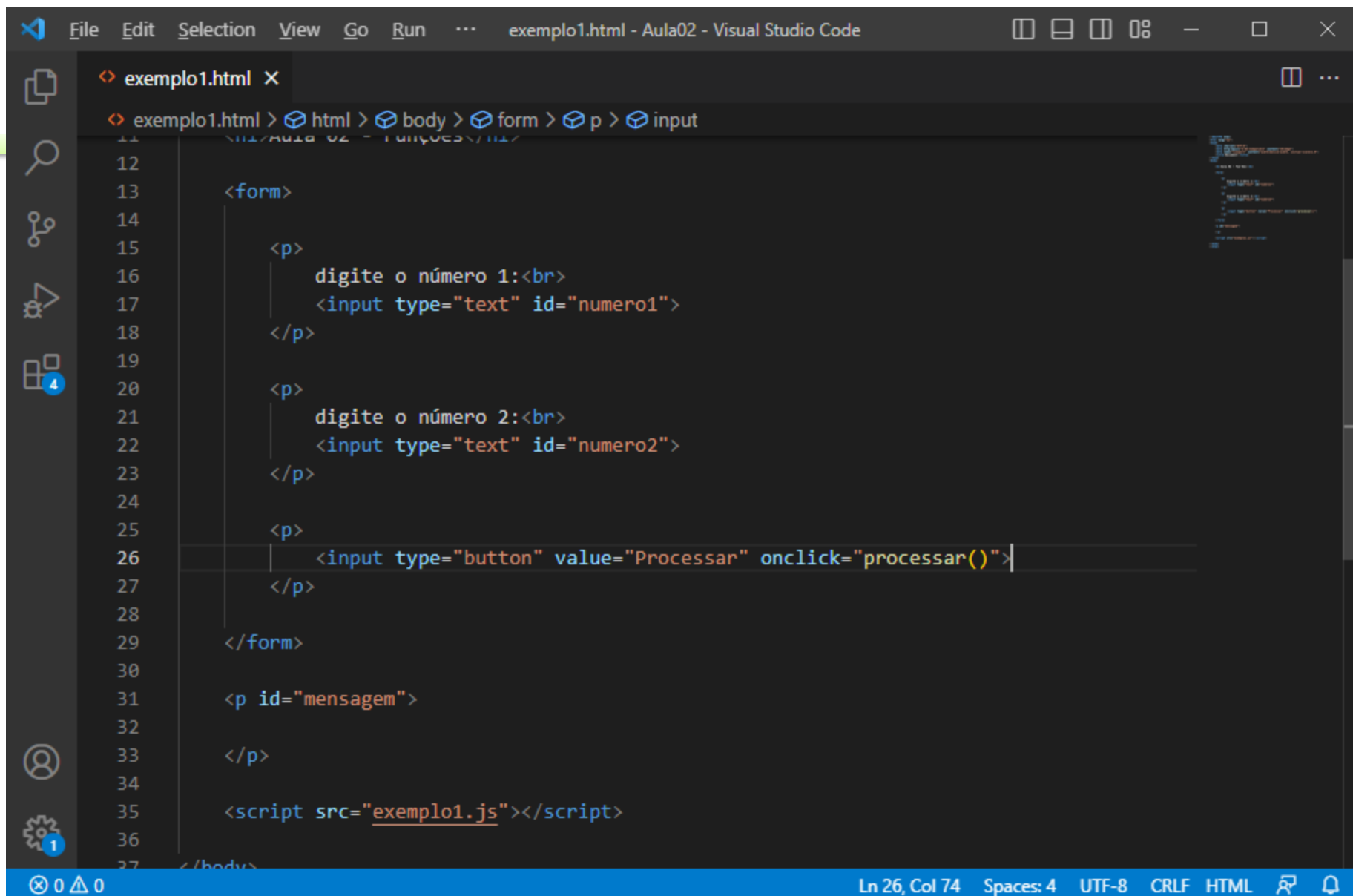
<input type="text" id="numero2">

</p>

</form>

<p id="mensagem">

Vamos incluir um segundo campo de texto



```
File Edit Selection View Go Run ... exemplo1.html - Aula02 - Visual Studio Code
<!-- Aula 02 - Funções -->
12
13 <form>
14
15     <p>
16         digite o número 1:<br>
17         <input type="text" id="numero1">
18     </p>
19
20     <p>
21         digite o número 2:<br>
22         <input type="text" id="numero2">
23     </p>
24
25     <p>
26         <input type="button" value="Processar" onclick="processar()">
27     </p>
28
29 </form>
30
31 <p id="mensagem">
32
33 </p>
34
35 <script src="exemplo1.js"></script>
36
37 </body>
```

```
<form>

  <p>
    digite o número 1:<br>
    <input type="text" id="numero1">
  </p>

  <p>
    digite o número 2:<br>
    <input type="text" id="numero2">
  </p>

  <p>
    <input type="button" value="Processar" onclick="processar()">
  </p>

</form>

<p id="mensagem">

</p>

<script src="exemplo1.js"></script>
```



Vamos incluir este código

```
<form>

  <p>
    digite o número 1:<br>
    <input type="text" id="numero1">
  </p>

  <p>
    digite o número 2:<br>
    <input type="text" id="numero2">
  </p>

  <p>
    <input type="button" value="Processar" onclick="processar()">
  </p>

</form>
```

Vamos incluir este código

- Criar um botão com o texto "**Processar**" usando `<input type="button">`.
- Quando o botão é clicado será chamado a função no JavaScript `processar()` isto acontece pois incluímos o atributo `onclick="processar()"`
- O **botão do tipo button** é ideal para executar ações no formulário sem recarregar a página.

Document

Arquivo C:/Users/admin/Desktop/Aula02/exemplo1.html

Aula 02 - Funções

digite o número 1:

digite o número 2:

Processar

Resultado

EXPLORER

- AULA02
 - exemplo1.html
 - JS exemplo1.js

JS exemplo1.js

```

1  let numero1, numero2, resultado;
2
3  function somar(n1, n2)
4  {
5      let soma = Number(n1) + Number(n2);
6      document.getElementById("mensagem").innerHTML = "A soma
7  }
8
9  function solicitarEntradaDados()
10 {
11     numero1 = Number(prompt("digite o número 1:"));
12     numero2 = Number(prompt("digite o número 2:"));
13
14     somar(numero1, numero2);
15 }
16
17 function processar()
18 {
19 }
20
21
22

```

Abra o exemplo1.js

Vamos definir a função processar

Ln 19, Col 5 Spaces: 4 UTF-8 CRLF {} JavaScript

JS exemplo1.js X

JS exemplo1.js > processar

```
1  let numero1, numero2, resultado;
2
3  function somar(n1, n2)
4  {
5      let soma = Number(n1) + Number(n2);
6      document.getElementById("mensagem").innerHTML = "A soma
7  }
8
9  function solicitarEntradaDados()
10 {
11     numero1 = Number(prompt("digite o número 1:"));
12     numero2 = Number(prompt("digite o número 2:"));
13
14     somar(numero1, numero2);
15 }
16
17 function processar()
18 {
19
20 }
21
22
```

Vamos definir a função processar

```
function processar()
```

```
{
```

```
    numero1 = document.getElementById("numero1").value;
```

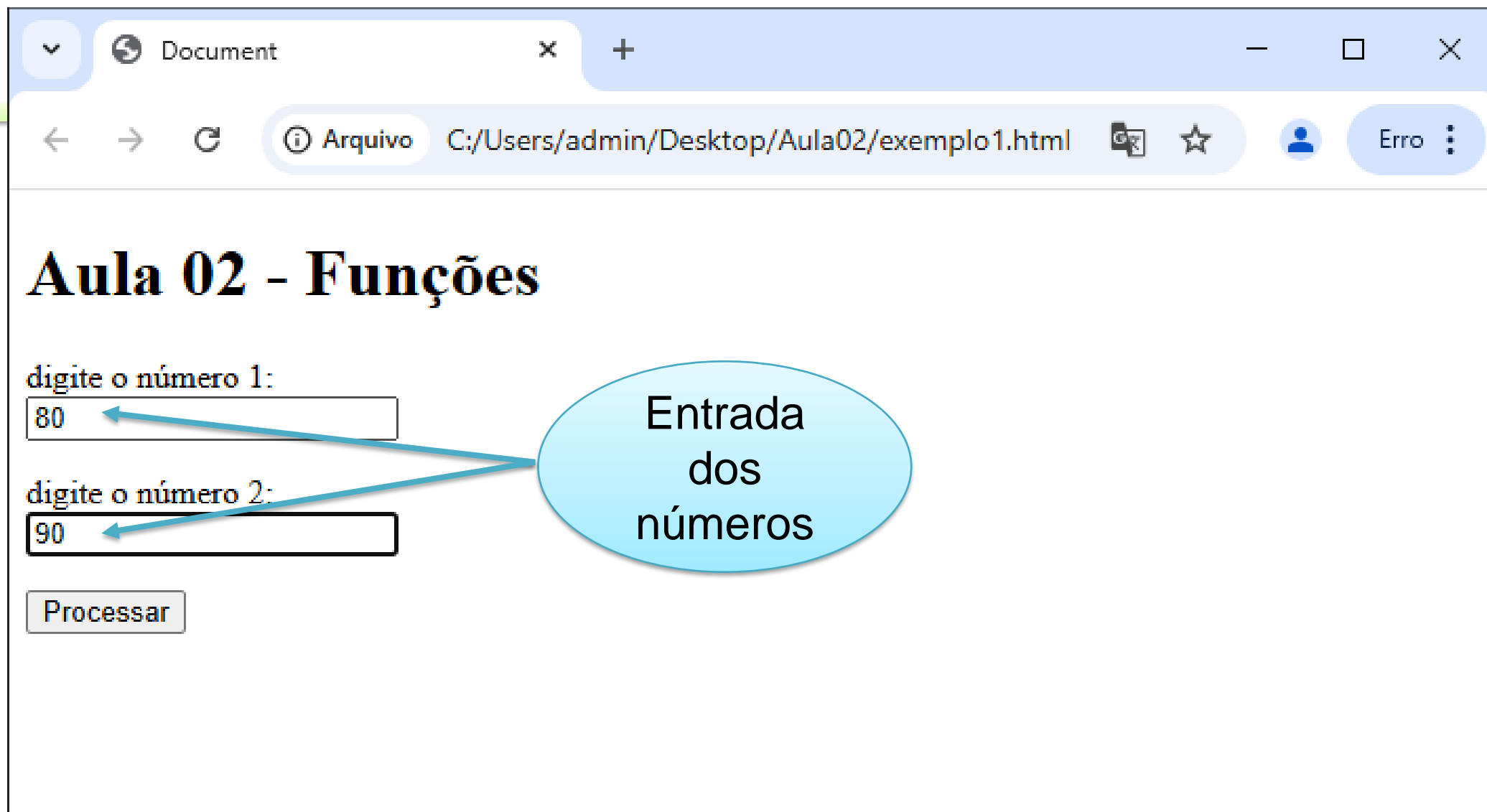
```
    numero2 = document.getElementById("numero2").value;
```

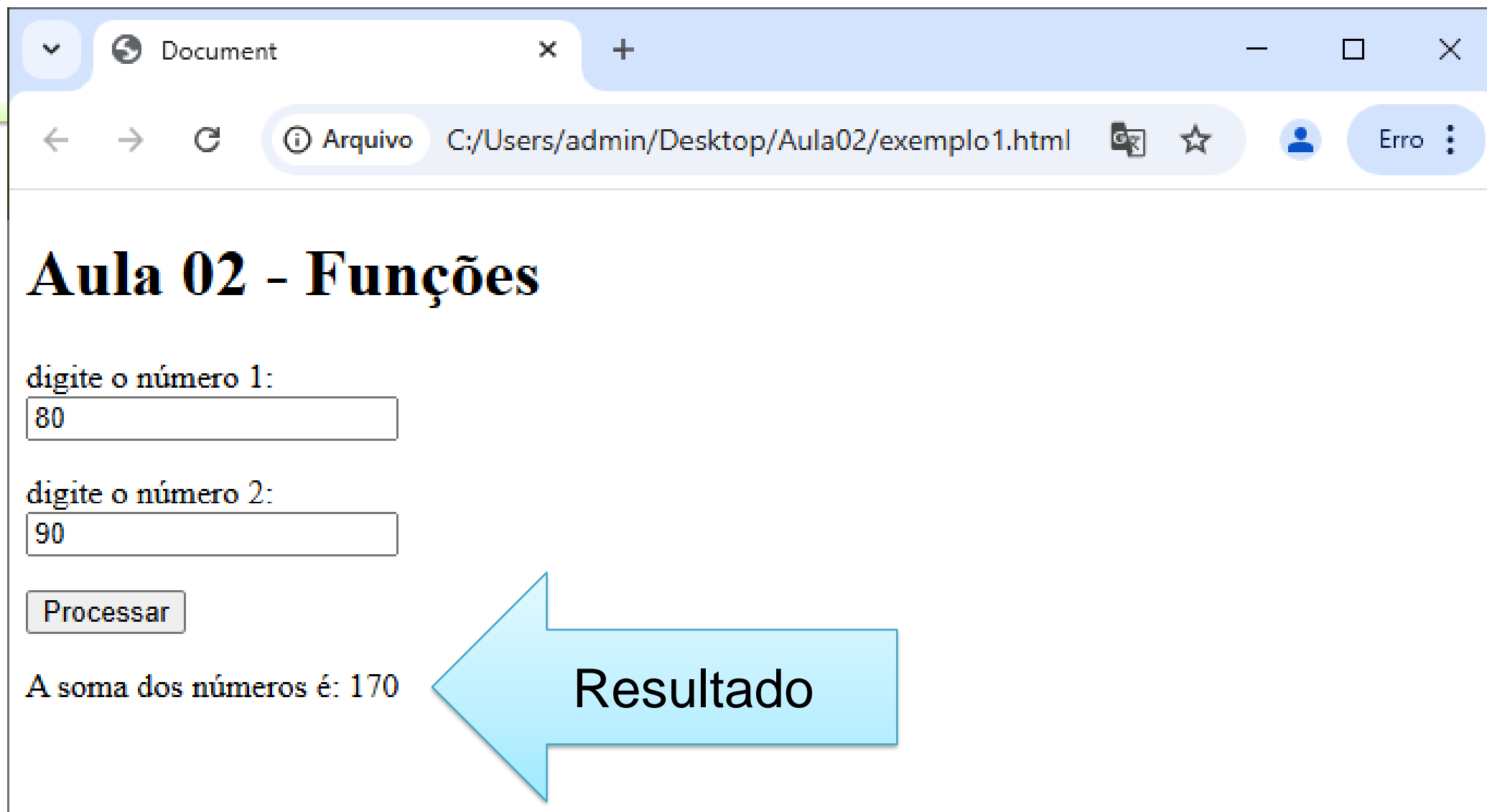
```
    somar(numero1, numero2);
```

```
}
```

Digite o código

- A função `processar()` obtém os valores dos campos de input com IDs `numero1` e `numero2` usando `document.getElementById().value`.
- Os valores são armazenados respectivamente nas variáveis `numero1` e `numero2`.
- Chama a função `somar(numero1, numero2)`, que realiza a soma e exibe o resultado na página.





Document

Arquivo C:/Users/admin/Desktop/Aula02/exemplo1.html

Aula 02 - Funções

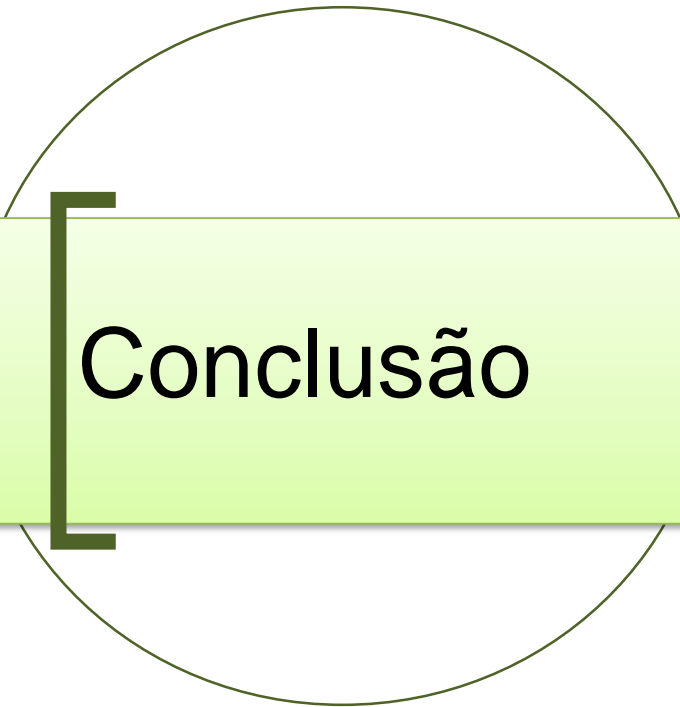
digite o número 1:

digite o número 2:


Processar

A soma dos números é: 170

Resultado



Conclusão



Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br

Funções no JavaScript

■ O que é ?

- Uma função é um bloco de código reutilizável que executa uma tarefa específica.
- As funções podem ou não receber parâmetros (entradas), podem retornar um valor direto ou executar uma ação
- As funções ajudam a organizar e reutilizar o código, tornando-o mais modular e legível.

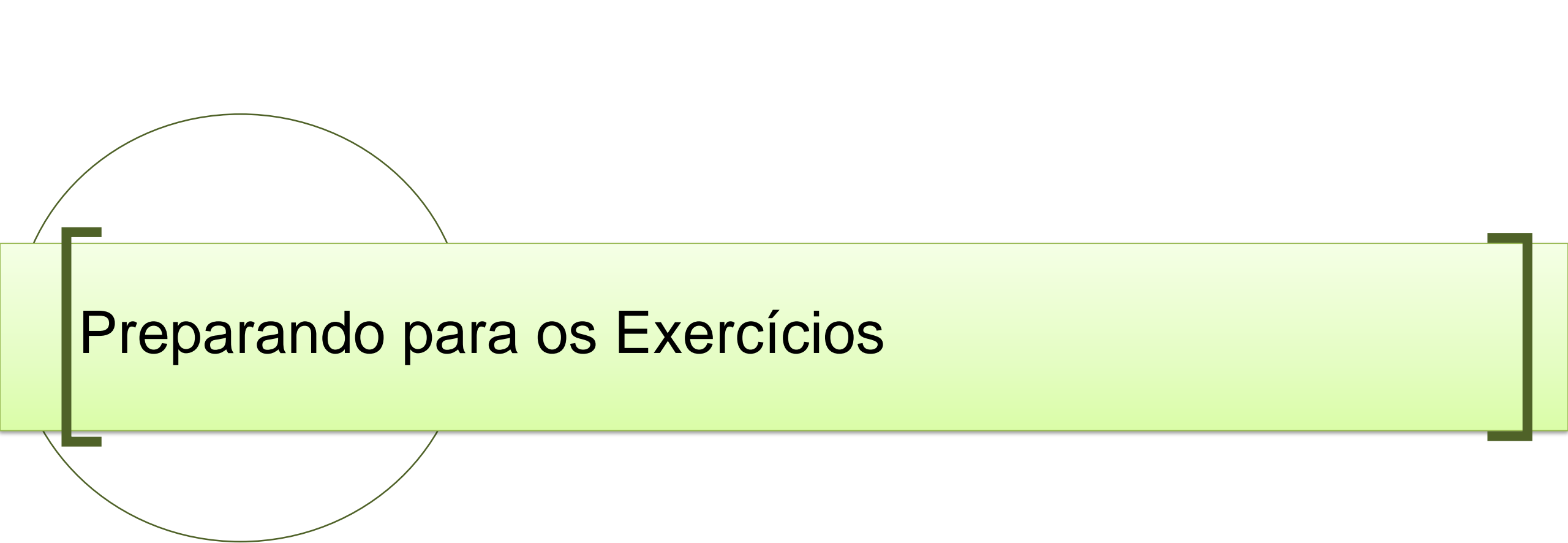
```
function somar(n1, n2) ...  
}  
  
function solicitarEntradaDados() ...  
}  
  
function processar() ...  
}
```

Funções

■ Eventos

- São ações ou ocorrências detectadas pelo navegador, como cliques de mouse, pressionamento de teclas ou carregamento da página.
- Podem ser associadas a funções para executar ações específicas quando o evento ocorre.
- Nesta aula, vimos que ao clicar no botão o evento click foi disparado, e a função **processar()** foi chamada.

```
<p>  
  <input type="button" value="Processar" onclick="processar()">  
</p>  
</form>
```



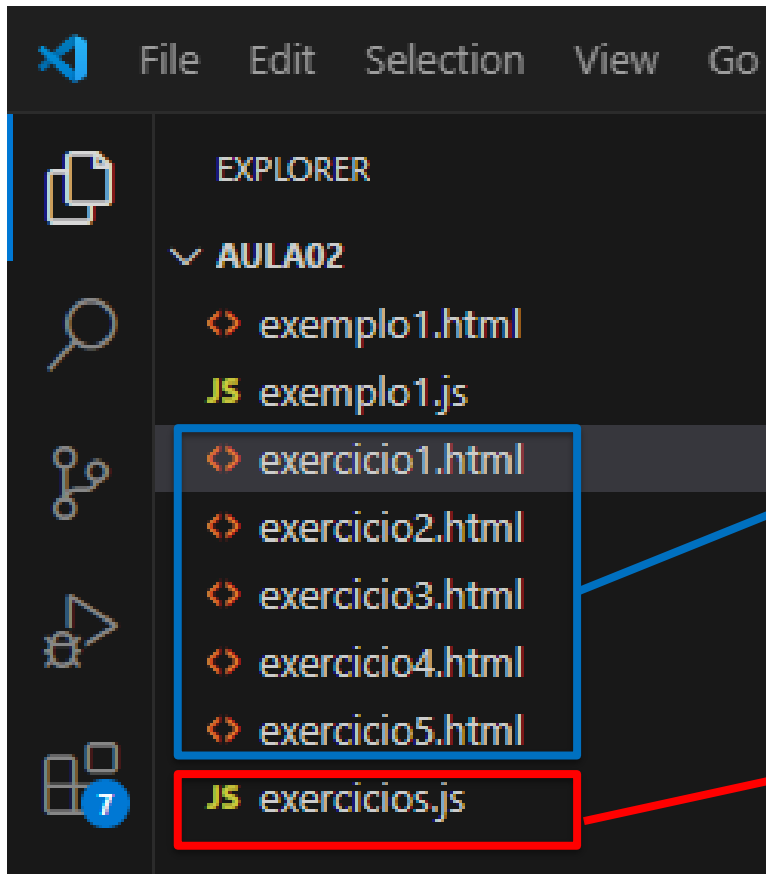
Preparando para os Exercícios

Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br

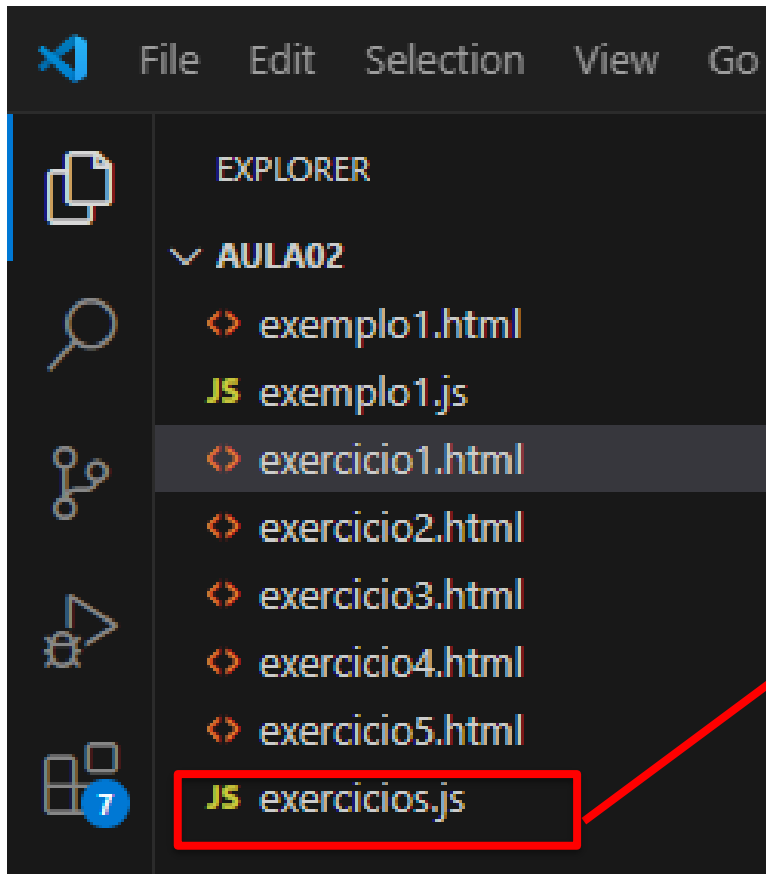
Preparando para os Exercícios



■ Crie os arquivos

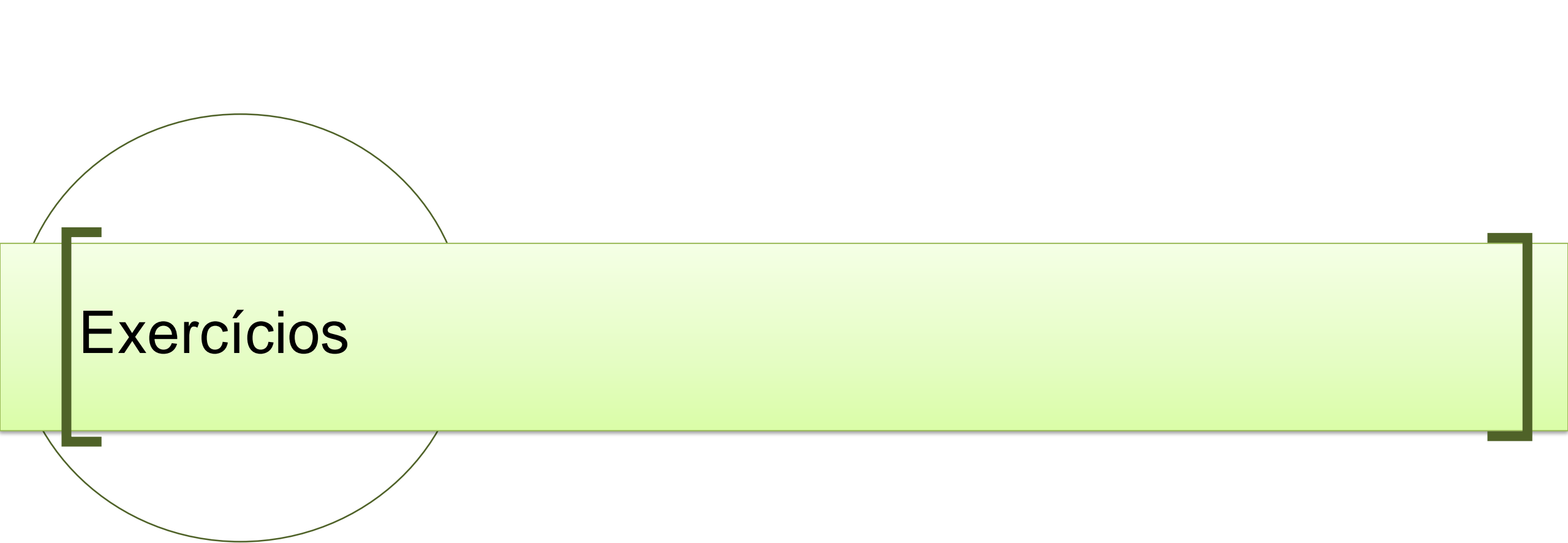
- `exercicio1.html`
- `exercicio2.html`
- `exercicio3.html`
- `exercicio4.html`
- `exercicio5.html`
- `exercicios.js`

Preparando para os Exercícios



Criar as
funções
dentro do
exercicios.js

```
JS exercicios.js X
JS exercicios.js > ...
1  function exercicio1()
2  {
3
4  }
5
6  function exercicio2()
7  {
8
9  }
10
11 function exercicio3()
12 {
13
14 }
15
16 function exercicio4()
17 {
18
19 }
20
21 function exercicio5()
22 {
23
24 }
25
```



Exercícios

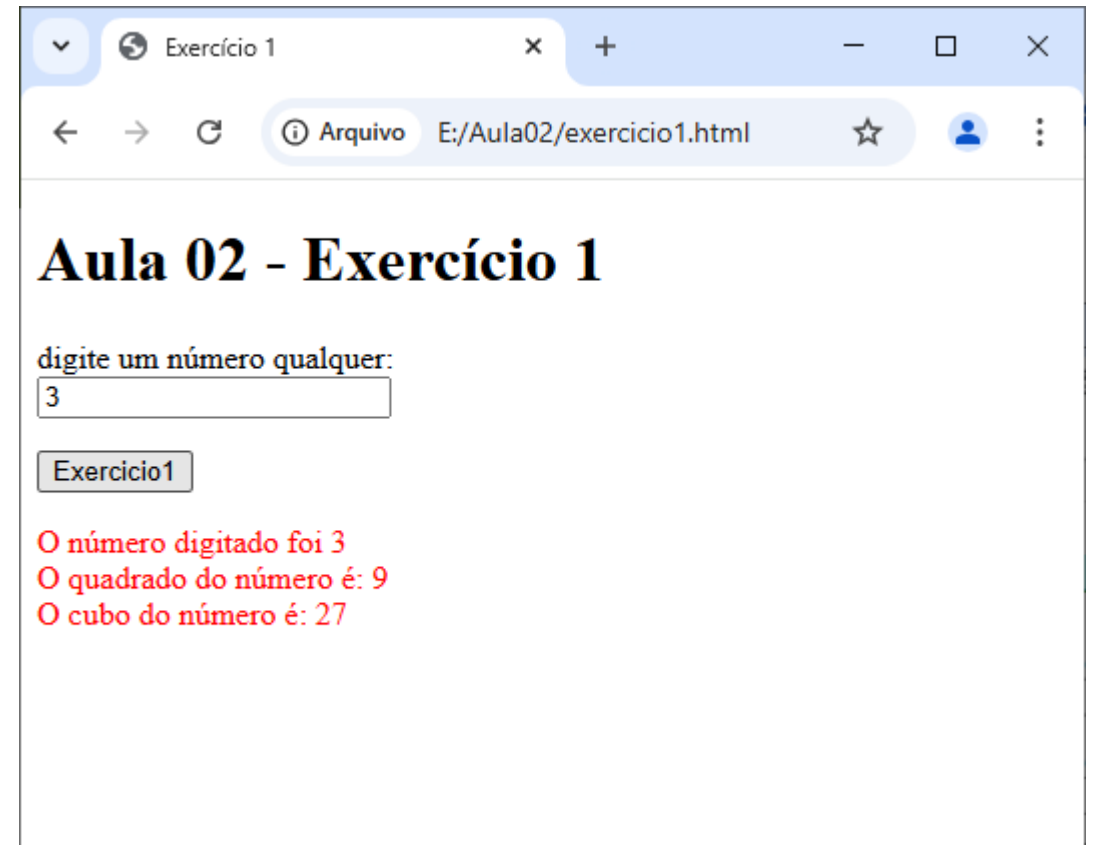
Guilherme Henrique de Souza

guilherme.souza@etec.sp.gov.br

guilherme.souza183@fatec.sp.gov.br

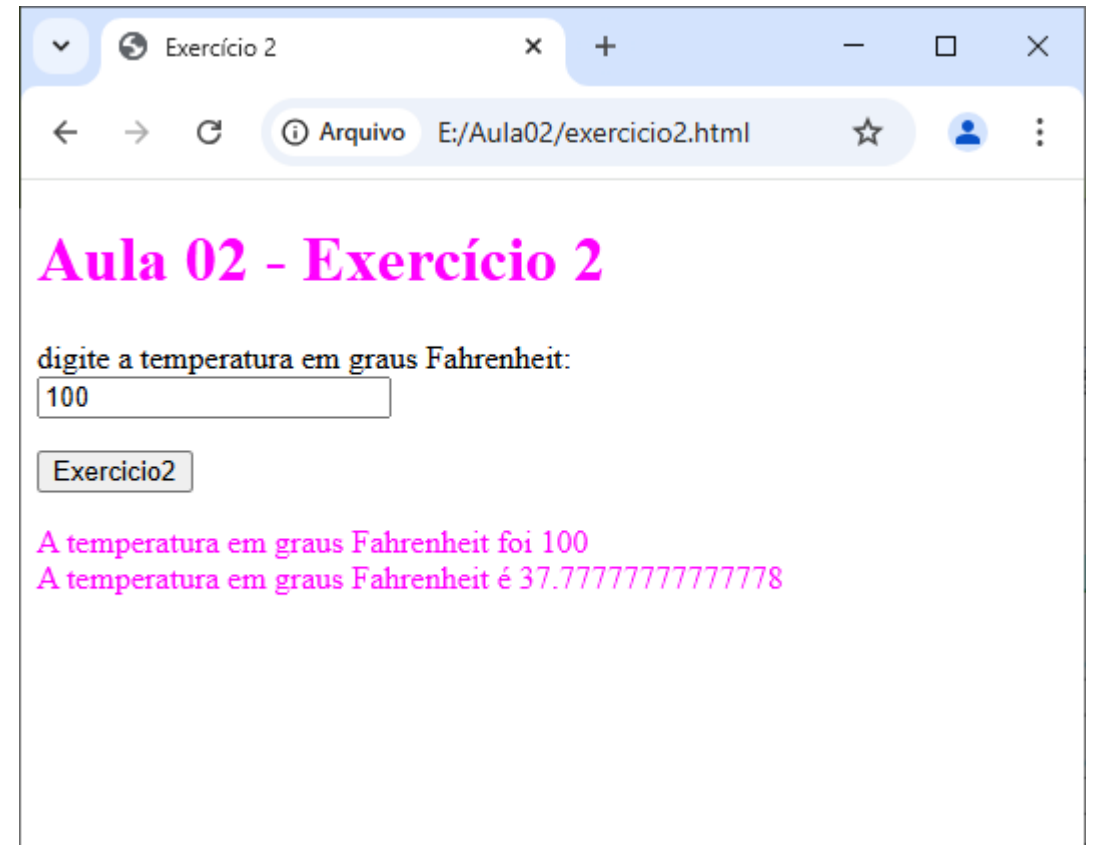
Exercício 1

- Faça uma página HTML com JS que através de uma caixa de texto (*input type=text*) receba:
 - Um numero qualquer
- A pagina deverá exibir:
 - O quadrado do numero digitado
 - O cubo do numero digitado



Exercício 2

- Faça uma página HTML com JS que através de uma caixa de texto (*input type=text*) receba:
 - A temperatura em graus Fahrenheit
- A página deverá calcular e exibir:
 - A temperatura em Celsius
 - ***Temperatura Celsius*** = $\frac{(Fah - 32) \times 5}{9}$



Exercício 3

- Faça uma página HTML com JS que através de uma caixa de texto (*input type=text*) receba os valores:
 - O **valor** da prestação em atraso
 - A **taxa** de juros (em porcentagem)
 - O **tempo** de dias de atraso
- Para realizar o cálculo do valor da parcela utilize a formula a seguir:

$$\text{valor parcela} = \text{valor} + \left(\text{valor} \times \left(\frac{\text{taxa}^2}{100} \right) \times \text{tempo} \right)$$

- A pagina deverá exibir o valor parcela atualizado

Exercício 3

Arquivo E:/Aula02/exercicio3.html

Aula 02 - Exercício 3

digite o valor da prestação:

digite a taxa de juros:

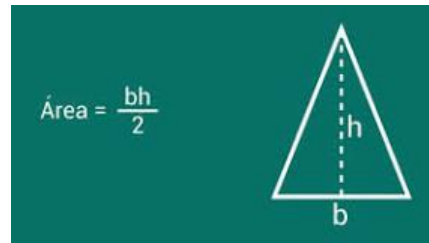
digite o tempo (em dias) de atraso:

Valores informados
Valor = 100
Taxa = 10
Tempo = 3

O valor atualizado é: 400

Exercício 4

- Faça uma página HTML com JS que através de uma caixa de texto (*input type=text*) receba:
 - O valor da base de um triângulo
 - O valor da altura de um triângulo
- A pagina deverá exibir:
 - A área do triângulo



Exercício 4

Arquivo E:/Aula02/exercicio4.html

Aula 02 - Exercício 4

digite o valor da base:

digite o valor da altura:

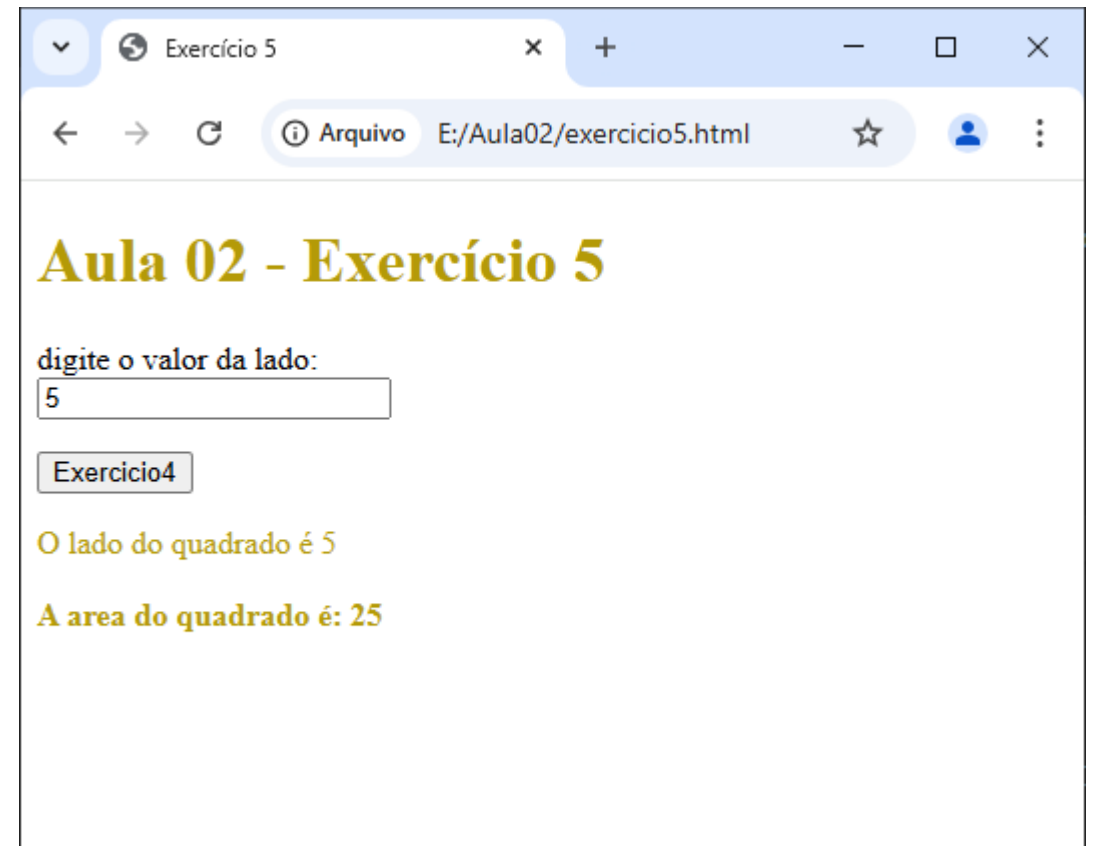
Exercicio4

Base = 20
Altura = 10

A area do triângulo é: 100

Exercício 5

- Faça uma página HTML com JS que através de uma caixa de texto (*input type=text*) receba:
 - O valor do lado de um quadrado
- A pagina deverá exibir:
 - A área do quadrado



Exercício 6

- Faça uma página HTML com JS que através de uma caixa de texto (*input type=text*) receba :
 - A quantidade de um produto
 - O preço de um produto
- A pagina deverá calcular e exibir:
 - O subtotal (quantidade x preço)
 - O desconto de 10% sobre o subtotal
 - O valor final a pagar (subtotal – desconto)



Exercício 7

- Faça uma página HTML com JS que através de uma caixa de texto (*input type=text*) receba :
 - 3 notas de um aluno

- A pagina deverá calcular e exibir:
 - A média aritmética

Exercício 8

- Faça uma página HTML com JS para calcular o IMC (Índice de Massa Corporal) de uma pessoa. Deverá ser solicitado ao usuário a seguinte entrada de dados através de uma caixa de texto (*input type=text*):
 - O peso (em kg)
 - A altura (em metros)
- A pagina deverá então calcular o IMC usando a fórmula:
 - $IMC = peso / (altura * altura)$
 - A pagina deverá exibir o valor do IMC

Exercício 9

- Faça uma página HTML com JS para calcular o montante de juros simples. Deverá ser solicitado ao usuário a seguinte entrada de dados através de uma caixa de texto (*input type=text*):
 - O principal (valor inicial),
 - A taxa de juros (em porcentagem)
 - O tempo (em anos)
- A pagina deverá calcular e exibir o montante usando a fórmula:
 - $Montante = Principal \times (1 + (taxa\ juros \times tempo))$
 - A pagina deverá exibir o valor do montante

Exercício 10

- Faça uma página HTML com JS para calcular o salario ao final do mes. Deverá ser solicitado ao usuário a seguinte entrada de dados através de uma caixa de texto (*input type=text*):
 - O valor da hora
 - A quantidade de horas trabalhadas por dia
- A pagina deverá calcular o valor do salário ao final do mês, lembrando que o mês possui 30 dias:
 - $Salario = Valor * Qtde\ Horas * 30$
 - A pagina deverá exibir o valor do salario