

Iniciaremos em alguns instantes...

“As aulas disponibilizadas por meios digitais envolvem horas de estudo, pesquisa e organização dos(as) professores(as) responsável(eis) e destinam-se, única e exclusivamente, aos alunos regularmente matriculados em cursos da Universidade Cruzeiro do Sul, para fins acadêmicos

A disponibilização/ divulgação dos links de acesso às aulas síncronas e/ ou registros das aulas a colegas não matriculados e a indivíduos estranhos à comunidade acadêmica da universidade implicará responsabilização civil e criminal conforme Artigo 46, inciso IV da Lei 9610/98, combinado com o artigo 184 do Código Penal, que estabelece direito autoral ao docente.

*Assim, em atendimento à Lei e à valorização do trabalho docente, solicita-se que **não compartilhem as aulas com indivíduos que não estejam regularmente matriculados.**”*



A close-up photograph of a person's finger pointing at a computer screen. The screen displays several lines of JavaScript code, including `type="text/javascript">` and `</script>`. The background is a dark grey gradient with a white curved line separating it from the image on the left.

Técnicas de Programação



Técnicas de Programação. - Prof. Msc. Luiz C M Lozano

JavaScript



Técnicas de Programação. - Prof. Msc. Luiz C M Lozano

Javascript

Criar interação entre o Hmtl e CSS (DOM);

Manipular elementos e dados que existem na página;

Utilizada como linguagem server-side(Node.js);

Base de grandes frameworks (React, Angular, Vue).



Javascript

JavaScript (linguagem de programação para web)

Os principais navegadores vem com o interpretador p/ html, css e javascript;

Hoje em dia é comum programar javascript fora do navegador, exatamente da mesma maneira que programamos no navegador.



Onde colocar o código JavaScript

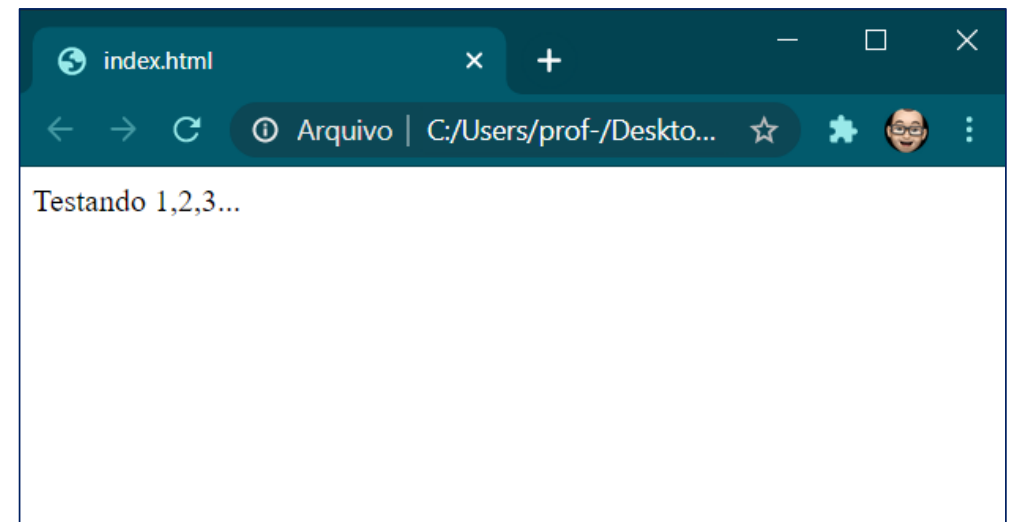
No navegador o javascript pode ser utilizado de duas formas diferentes:

- Primeira forma é utilizando uma tag chamada script.
- Segunda forma escrever o javascript em um arquivo externo.



Dentro da Tag <script>

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html>
3    <head>
4
5    </head>
6    <body>
7      <script>
8        document.write('Testando 1,2,3...');
9      </script>
10   </body>
11 </html>
```



Em um arquivo .js externo

Obs.: Se colocar o link no head ele carrega antes do body (sempre que for necessário do javascript antes de carregar a pagina).
Melhor lugar é colocar o javascript no final do body, pois uma função dele é manipular os componentes visuais dentro da tela.

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4
5      </head>
6      <body>
7          <script src="script.js"></script>
8      </body>
9  </html>
```

```
JS script.js
1  document.write('Testando 1,2,3...');
```



Usando o Console do Chrome

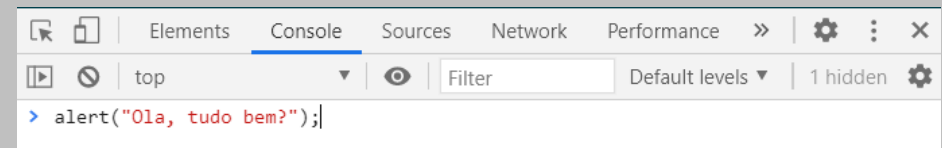
No navegador, clicar com o botão direito do mouse, ir em inspecionar.

Vai abrir varias abas (elements, console (focaremos nela)).

Na aba console podem aparecer avisos do javascript, alertas, etc. Podemos rodar nosso código nesse próprio local.

Essa página diz
Ola, tudo bem?

OK



Formas de mostrar (Output)

primeira
forma:
manipulando
algum
elemento do
html

- inserir um id "titulo" no h1;
- depois no js externo.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     <h1 id="titulo">Aula de Tecnicas de Programação</h1>
7     <script src="script.js"></script>
8   </body>
9 </html>
```

```
JS script.js
1 document.getElementById("titulo").innerHTML = "Bora, aprender???"
2
```



Formas de mostrar (Output)

Segunda forma:
utilizar o
document.write
(criando seu
proprio
elemento)

- no js externo:
document.write("digite
um texto");

```
JS script.js
1 document.write("Aprendendo JavaScript");
2
```

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     <script src="script.js"></script>
7   </body>
8 </html>
```

← → ↺ ⓘ Arquivo | C:/Users/prof-/Desktop/aula03/index.html

Aprendendo JavaScript



Formas de mostrar (Output)

Terceira forma,
utilizando a
janela do
navegador.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4      </head>
5      <body>
6          <script src="script.js"></script>
7      </body>
8  </html>
```

```
JS script.js
1  alert("Aprendendo JavaScript");
2
```

Essa página diz

Aprendendo JavaScript

OK



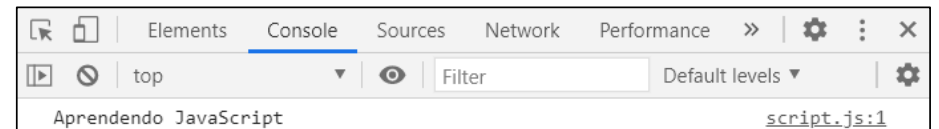
Formas de mostrar (Output)

Quarta forma: o Console

- no js externo (depois olhar o console). Utilizar o `console.log("A mensagem que eu inserir aqui aparecerá no console");`

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     <script src="script.js"></script>
7   </body>
8 </html>
```

```
JS script.js
1 console.log("Aprendendo JavaScript");
2
```



Javascript – Tipos de Dados

Number (Aritmético, Número);

String;

Boolean;

Valores vazios (null, undefined).



Tipos de Dados

JS aula01.js > ...

```
1  var nome = "Luiz";  
2  var idade = 90;  
3  var salvo = true;  
4  var cidade;  
5  var tipo = typeof nome;
```

```
> console.log(nome);  
Luiz
```

```
> typeof (nome);  
< "string"
```



Javascript – Numbers: aritméticas

Operações em JS

- `Console.log(2+1);`
- `Console.log(2*4-3)`
- `Console.log(5+(2*4));`

Operadores

- `+` (soma)
- `-` (subtração)
- `/` (divisão)
- `*` (multiplicação)
- `%` - (resto da divisão)
- `//` (parte inteira da divisão)



Exemplos:

```
JS script.js > ...
```

```
1    var x = 10;
```

```
2    var y = 15;
```

```
3    resultado = x + y;
```

```
4    alert(resultado);
```



Javascript – Boolean

Valor de comparação (verdadeiro ou falso)

- True
- False
 - `Console.log(2>2)`
 - `Console.log(3>10);`

- Maior que: `>`
- Menor que: `<`
- Maior ou igual: `>=`
- Menor ou igual: `<=`
- Igual: `==`
- Diferente: `!=`
- Idêntico: `===`



Javascript – Operadores Lógicos

- Valor de comparação
- && - and
- || - or
- ! – not
- ? (operador ternário)

Tabelas de verdade: AND

a	b	a && b
TRUE	TRUE	TRUE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
FALSE	FALSE	FALSE

Tabelas de verdade: NOT

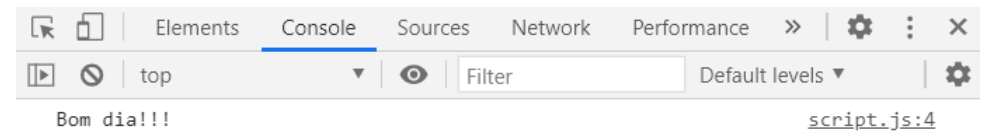
a	!a
TRUE	FALSE
FALSE	TRUE

Tabelas de verdade: OR

a	b	a b
TRUE	TRUE	TRUE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	FALSE	FALSE



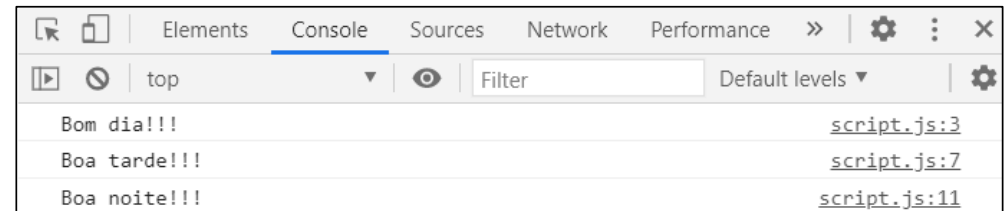
```
JS script.js > ...  
1  var hora = 11;  
2  if (hora < 12){  
3      console.log("Bom dia!!!")  
4  }
```



Condicionais



```
JS script.js > ...
1  var hora = 11;
2  if (hora < 12){
3      console.log("Bom dia!!!")
4  }
5
6  if (hora < 18){
7      console.log("Boa tarde!!!")
8  }
9
10 if (hora < 23){
11     console.log("Boa noite!!!")
12 }
```



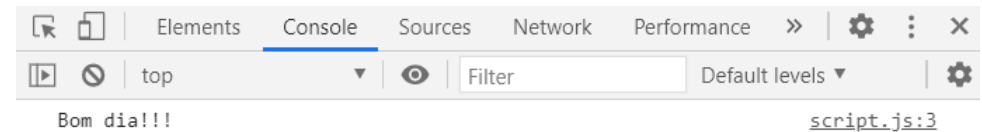
O 11 entrou em todas as condições...Porque???

Condicionais



Técnicas de Programação. - Prof. Msc. Luiz C M Lozano

```
JS script.js > ...
1  var hora = 11;
2  if (hora < 12){
3    console.log("Bom dia!!!")
4  }
5
6  else if (hora < 18){
7    console.log("Boa tarde!!!")
8  }
9
10 else if (hora < 23){
11   console.log("Boa noite!!!")
12 }
```

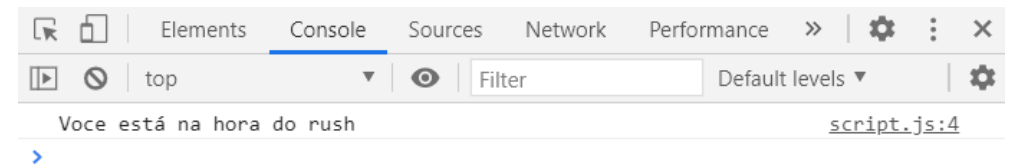


Condicionais



Condicionais

```
JS script.js > ...  
1  var hora = 12;  
2  
3  if (hora == 12 || hora == 18){  
4      console.log("Voce está na hora do rush");  
5  }
```





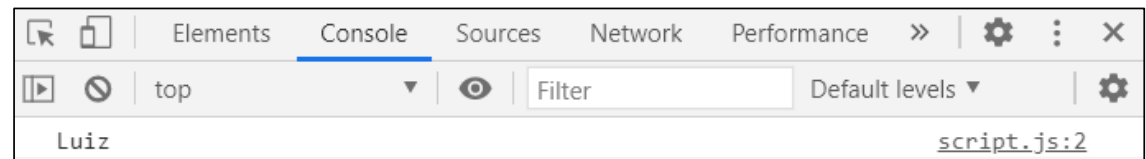
var, let e
const



var

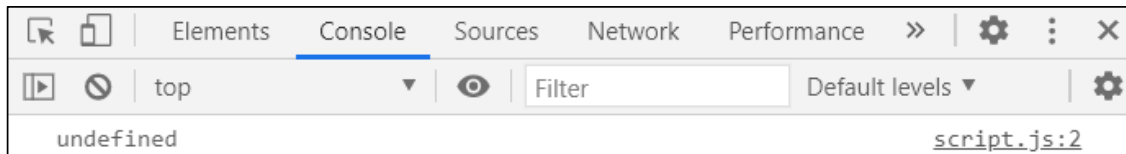
usando o var a variável utiliza um processo chamado hoisting, ela será gerada para o escopo geral da sua tela. Você consegue usar essa variável no window, pois ela é transferida para o escopo geral do site. Portanto você tem acesso a ele em qualquer lugar do seu sistema.

```
JS script.js > ...  
1   var nome = "Luiz";  
2   console.log(window.nome);  
3
```



let

```
JS script.js > ...  
1 let nome = "Luiz";  
2 console.log(window.nome);
```



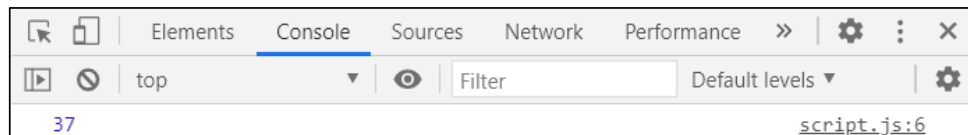
Não funciona com window por ser local.

Quando usamos o `let` a variável fica disponível apenas no escopo específico do código, portanto bloco de código, que pode ser um `if` por exemplo.

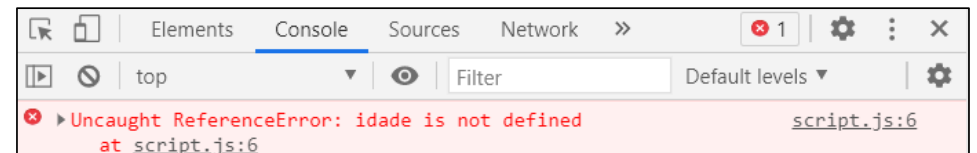


Técnicas de Programação. - Prof. Msc. Luiz C M Lozano

```
JS script.js > ...
1   var nome = "Luiz";
2
3   if (nome == "Luiz"){
4       var idade = 37;
5   }
6   console.log(idade);
```



```
JS script.js > ...
1   var nome = "Luiz";
2
3   if (nome == "Luiz"){
4       let idade = 37;
5   }
6   console.log(idade);
```

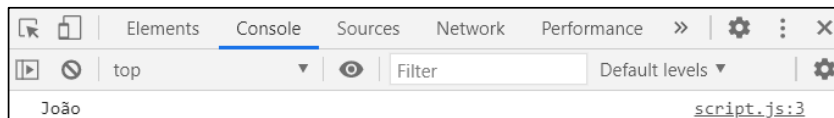


Trocando var por let (dentro de um if)

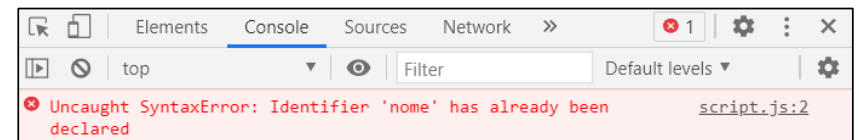


Redefinição de variável

```
JS script.js > ...  
1  var nome = "Luiz";  
2  var nome = "João";  
3  console.log(nome);
```



```
JS script.js > ...  
1  let nome = "Luiz";  
2  let nome = "João";  
3  console.log(nome);
```



```
JS script.js > ...  
1  let nome = "Luiz";  
2  nome = "João";  
3  console.log(nome);
```

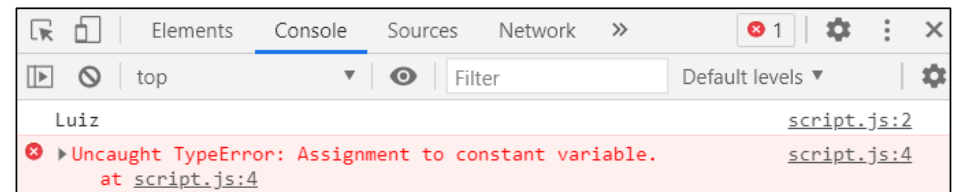
let não permite redefinição.

Apenas atualização.



Constante

```
1  const nome = "Luiz";  
2  console.log(nome);  
3  
4  nome = "João";
```



Comentarios

Comentário de
linha //

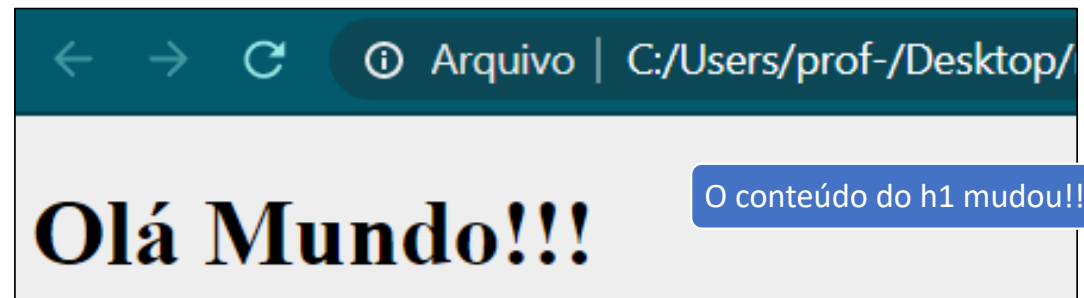
Comentário de
trecho /* trecho
*/



getElementById innerHTML

```
<> aula03.html > ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <link rel="stylesheet" href="style.css"/>
6      </head>
7      <body>
8          <h1 id="titulo">Curso de JavaScript</h1>
9
10         <script src="script.js"></script>
11     </body>
12 </html>
```

```
JS script.js
1  document.getElementById("titulo").innerHTML = "Olá Mundo!!!";
```



getElementById innerHTML

JS script.js > ...

```
1 var titulo = document.getElementById("titulo");//armazena o valor na variável titulo  
2 titulo.innerHTML = "Agora mudamos através da variável";
```

← → ↻ ⓘ Arquivo | C:/Users/prof-/Desktop/modulo2/aula03.html

Agora mudamos através da variável



getElementById innerHTML

As tags possuem conteúdo, por exemplo, o valor que fica entre `<>` e `</>`, já nos campos texto eles possuem o atributo `value`.

`<input id="campo" type="text" name="usuario" value="luiz"/>` o `innerHTML` altera o conteúdo de uma tag, porém para alterar o atributo utilizamos a propriedade `value`.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <link rel="stylesheet" href="style.css"/>
6   </head>
7   <body>
8     <h1 id="titulo">Curso de JavaScript</h1>
9
10    <input id="campo" type="text" name="usuario" value="digite o nome"/>
11
12    <script src="script.js"></script>
13  </body>
14 </html>
```

```
JS script.js > ...
1 var campo = document.getElementById("campo");
2 campo.value = "Luiz";
```

Alterou o texto digite o nome para Luiz



getElementById innerHTML



Técnicas de Programação. - Prof. Msc. Luiz C M Lozano

Javascript – Funções

Estruturas de códigos que serão reaproveitadas.

Também chamadas de subprogramas

Sempre com a palavra function

Devemos dar um nome

Argumentos/parâmetros definidos entre os ()

O corpo da função é definido entre as {}

Geralmente retorna um valor

Podemos declarar funções em variáveis

```
function multiplicarTresNumeros(x,y,z) {  
    return x * y * z;  
}  
  
console.log(multiplicarTresNumeros(2,3,4));  
  
const mult = multiplicarTresNumeros(5,4,8);  
  
console.log("O valor de mult é " + mult);
```

```
function podeDirigir(idade, cnh) {  
    if(idade >= 18 && cnh == true) {  
        console.log("Pode dirigir");  
    } else {  
        console.log("Não pode dirigir");  
    }  
}
```

```
function escreverNoConsole() {  
    console.log("Escrevendo no console!");  
}  
  
escreverNoConsole();  
  
const textoNoConsole = function() {  
    console.log("Texto no console!");  
}  
  
textoNoConsole();  
  
const textPorParametro = function(texto) {  
    console.log(texto);  
}  
  
textPorParametro("Testando por parâmetro!");
```



Javascript – Funções

Executar, chamar e invocar uma função (tudo a mesma coisa)

A função é representada pela () : nomeFuncao()

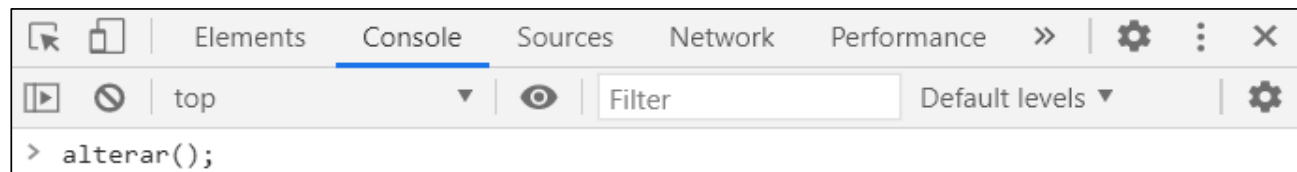
Passagem de parâmetros: nomeFuncao(param1, param2)



Funções

```
<> aula04.html > ...
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <link rel="stylesheet" href="style.css"/>
6    </head>
7    <body>
8      <h1 id="titulo">Seja bem vindo(a)</h1>
9      <input id="campo" type="text" name="usuario" value="digite o nome"/>
10     <script src="script.js"></script>
11   </body>
12 </html>
```

```
1  function alterar(){
2    document.getElementById("titulo").innerHTML = "Trocou o título";
3    document.getElementById("campo").value = "Trocou o campo";
4  }
```



Funções

Sempre que vemos um nome acompanhado de (), mesmo que tenha valores dentro, estamos vendo uma função. Por exemplo, `alert("funcionou")`, é uma função para imprimir valores e o valor esta sendo passado.

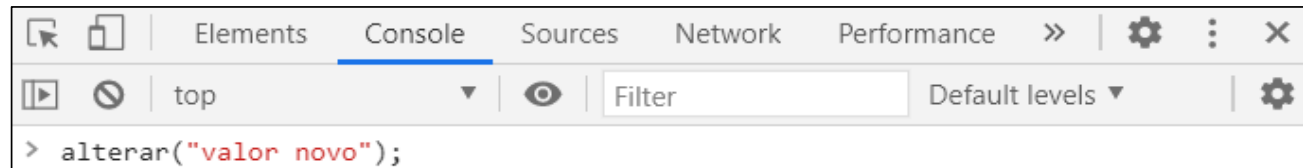
Qual diferença entre criar função e usar a função?



Funções

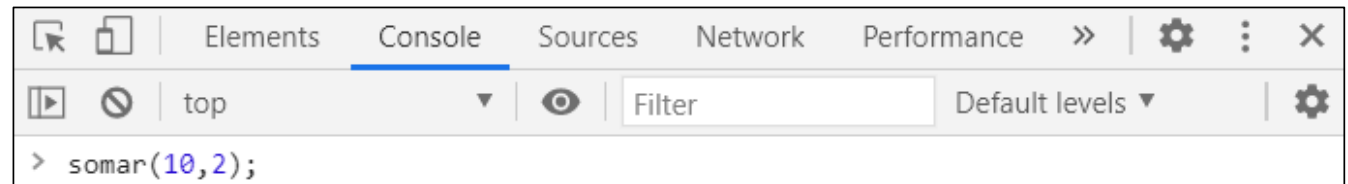
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <link rel="stylesheet" href="style.css"/>
6   </head>
7   <body>
8     <h1 id="titulo">Seja bem vindo(a)</h1>
9     <input id="campo" type="text" name="usuario" value="digite o nome"/>
10    <script src="script.js"></script>
11  </body>
12 </html>
```

```
JS script.js > ...
1 function alterar(titulo){
2   document.getElementById("titulo").innerHTML = titulo;
3   document.getElementById("campo").value = titulo;
4 }
```



Funções

```
JS script.js > ...  
1  function somar(x,y){  
2      let total = x + y;  
3      //se fosse var total a variavel ficaria visivel fora da função  
4      document.getElementById("campo").value = total;  
5  }
```



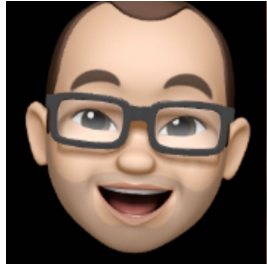
Seja bem vindo(a)

12



Técnicas de Programação. - Prof. Msc. Luiz C M Lozano

Siga-me...



<https://www.facebook.com/luizmlozano>



<https://www.instagram.com/luizmlozano>



Técnicas de Programação. - Prof. Msc. Luiz C M Lozano

Créditos:

Material parcialmente elaborado
pelo Prof. Alexander Gobatto e
adaptado pelo Prof. Luiz Lozano



Técnicas de Programação. - Prof. Msc. Luiz C M Lozano