

$$F = G \frac{m_1 m_2}{d^2}$$

# AULA 01 - JAVA SCRIPT INTRODUÇÃO

PROF. ANDERSON VANIN

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

$$\frac{df}{dt} = \lim_{h \rightarrow 0} \frac{f(t+h) - f(t)}{h}$$

## Client x Server



## Client x Server



# Client x Server



# Client x Server



## Client x Server





## Client x Server



## Client x Server





## Client x Server



## Website



Jornalista



Designer



Programador



## Website



Conteúdo



Estilo



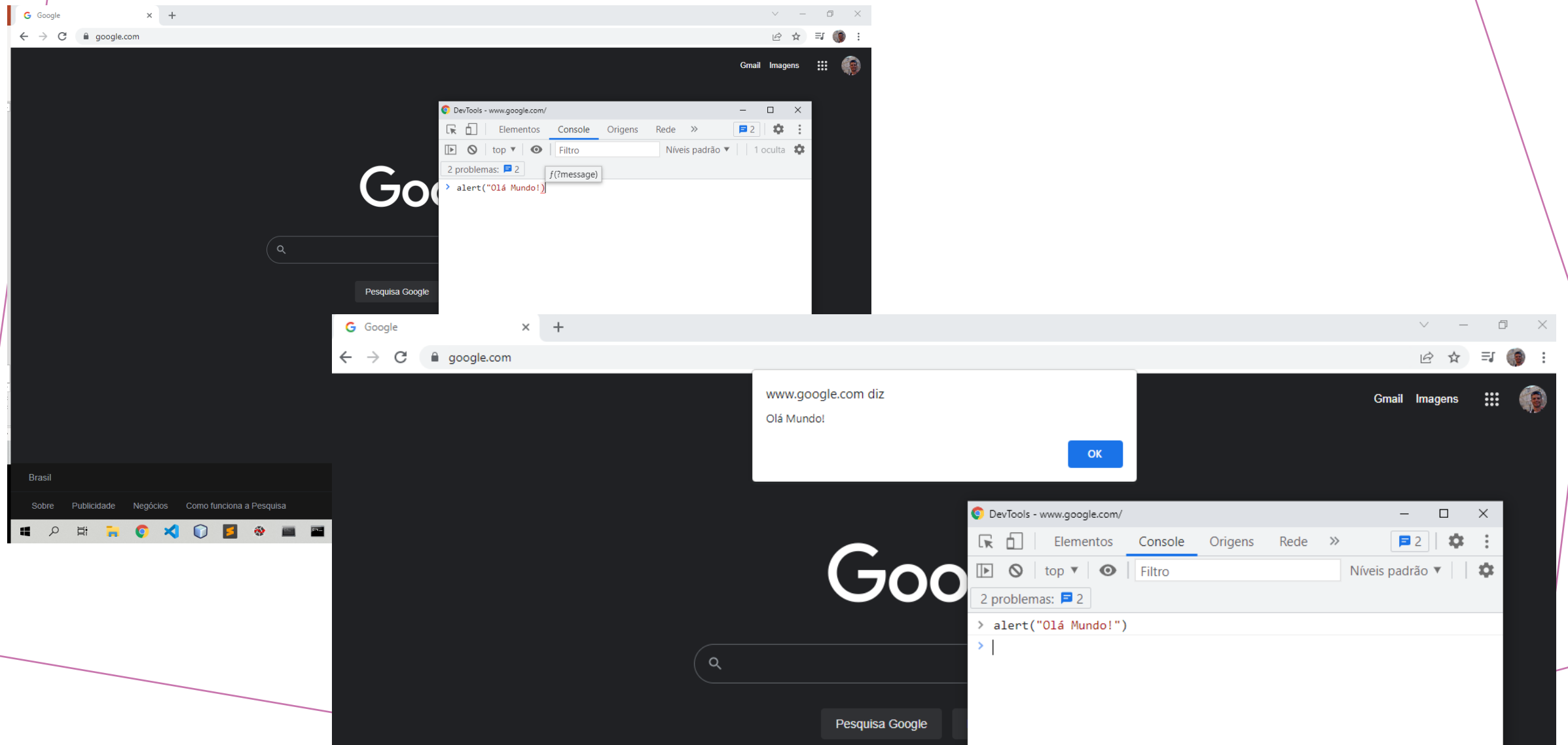
Interação

# Demonstração – Desabilitando o JS no Google Chrome

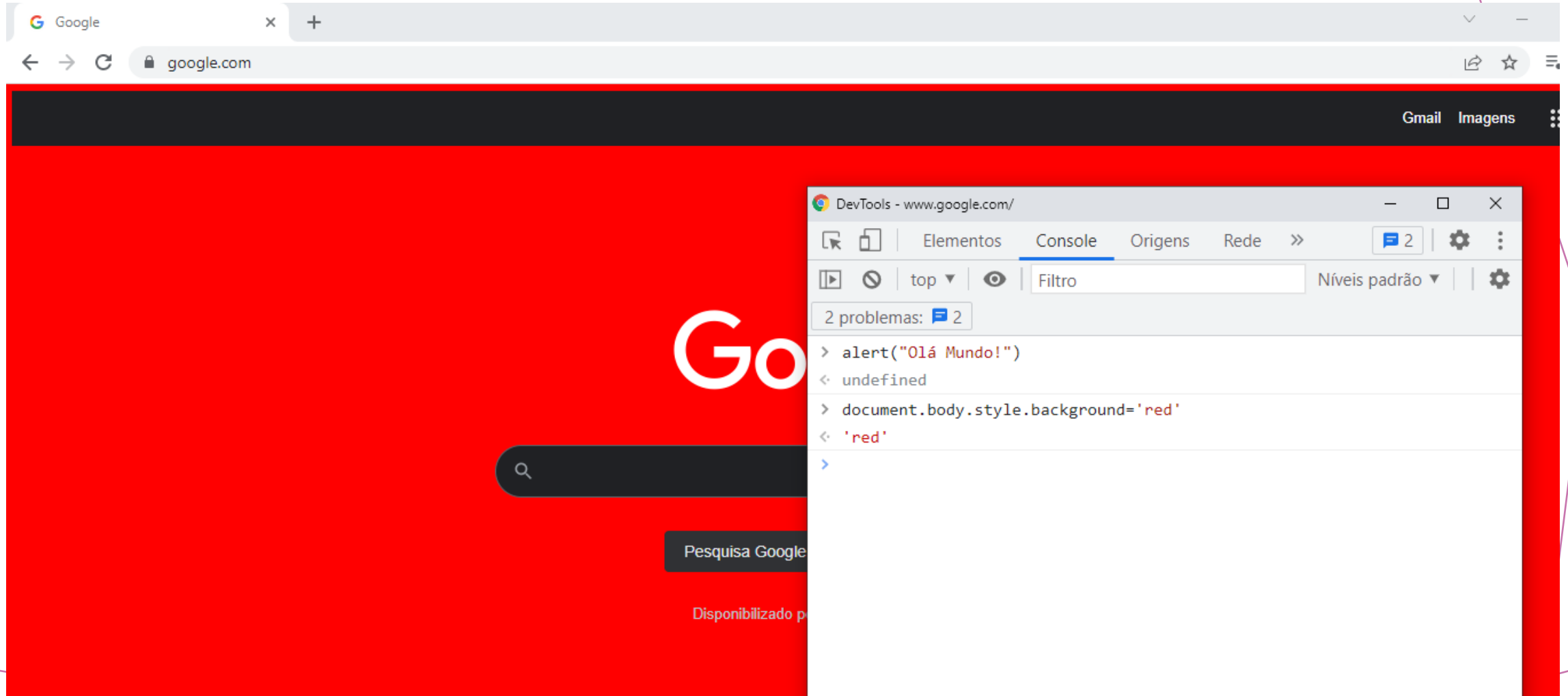
*EXEMPLO  
MÃO NA  
MASSA –  
“HACK”!*

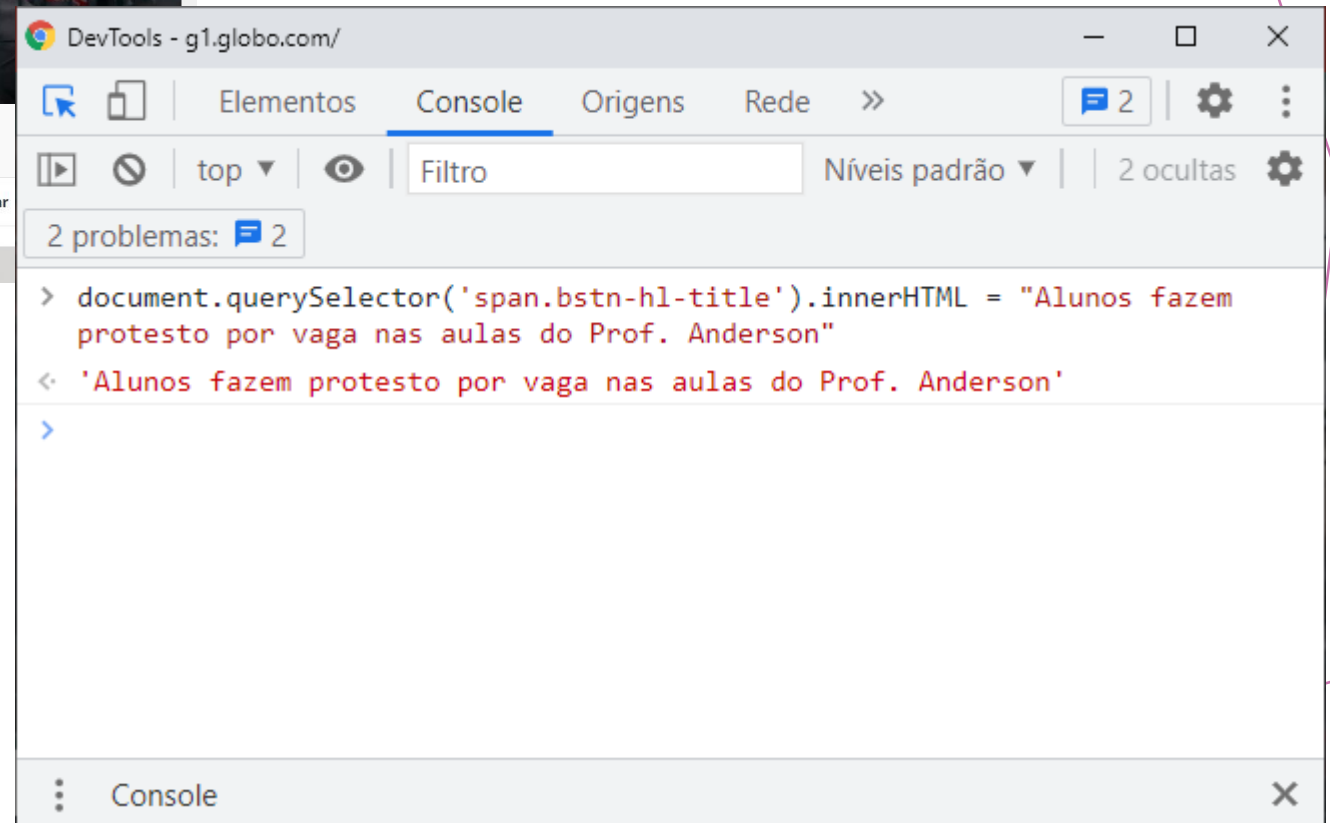


- Abra uma página do Google para pesquisa e ative o **Dev Tools (F12)** e clique em console e digite o comando *alert("Olá Mundo!")* e tecele enter.



- Abra uma página do Google para pesquisa e ative o Dev Tools (F12) e clique em console e digite o comando *document.body.style.background='red'* e tecle enter.







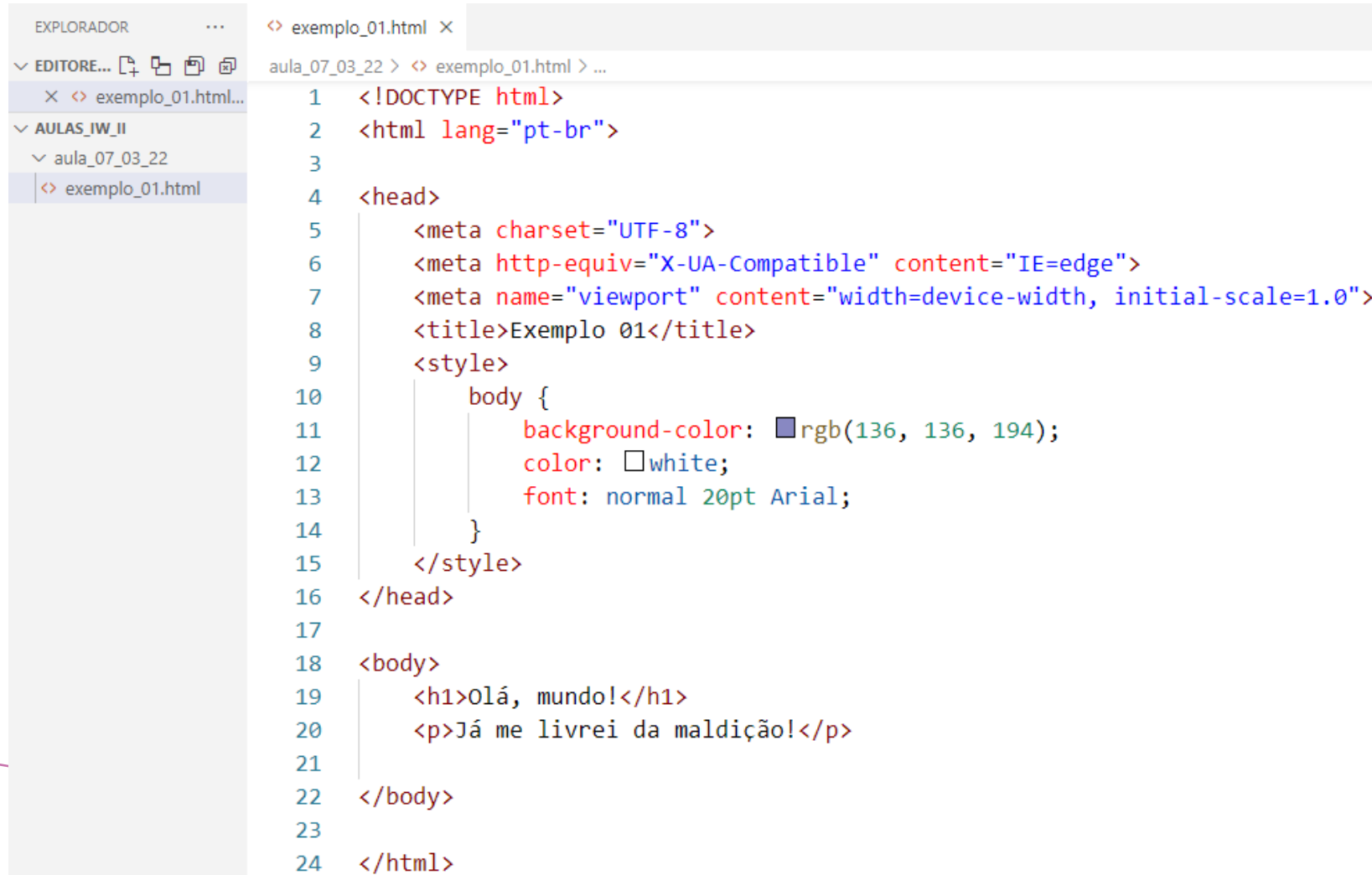


# PRIMEIROS PASSOS

# *REQUISITOS*

- Navegador Google Chrome
- IDE de desenvolvimento VS Studio Code
- Node.js ([nodejs.org](https://nodejs.org)) → **serve para rodar o JS fora do navegador.**
- Ser usuário administrador na máquina.
- **Organização dos Arquivos utilizados nas aulas!**

- Crie uma pasta chamada: **aulas\_iw\_ii**
- Crie uma subpasta chamada: **aula\_07\_03\_22**
- Crie um arquivo chamado: **exemplo\_01.html**



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a folder named 'aulas\_iw\_ii' containing a subfolder 'aula\_07\_03\_22' and a file 'exemplo\_01.html'. The code editor shows the content of 'exemplo\_01.html' with the following HTML code:

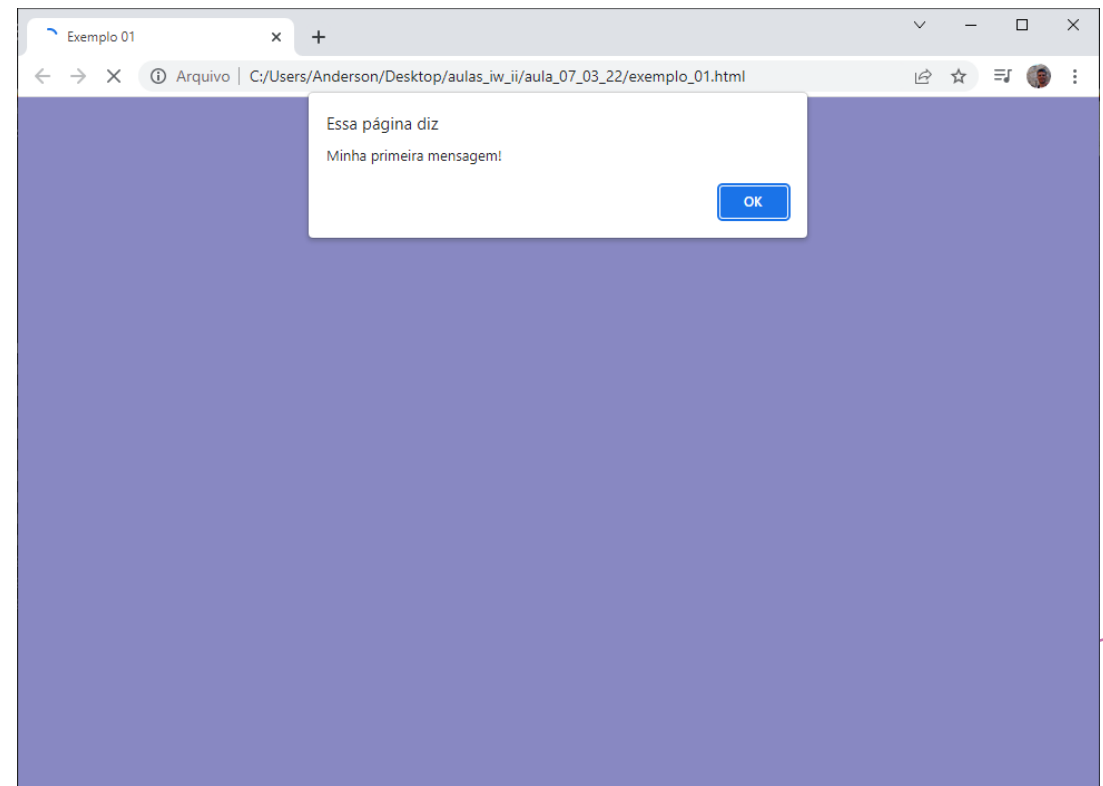
```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Exemplo 01</title>
9     <style>
10         body {
11             background-color: rgb(136, 136, 194);
12             color: white;
13             font: normal 20pt Arial;
14         }
15     </style>
16 </head>
17
18 <body>
19     <h1>Olá, mundo!</h1>
20     <p>Já me livre da maldição!</p>
21
22 </body>
23
24 </html>
```

# Olá, mundo!

Já me livrei da maldição!

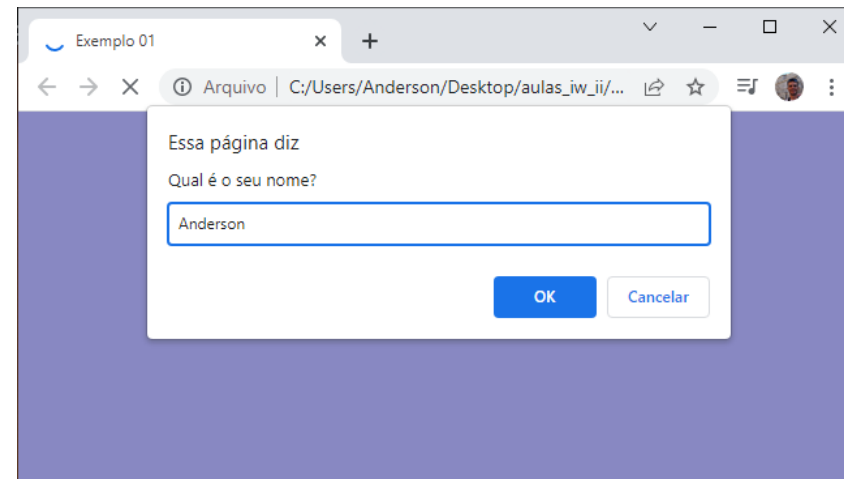
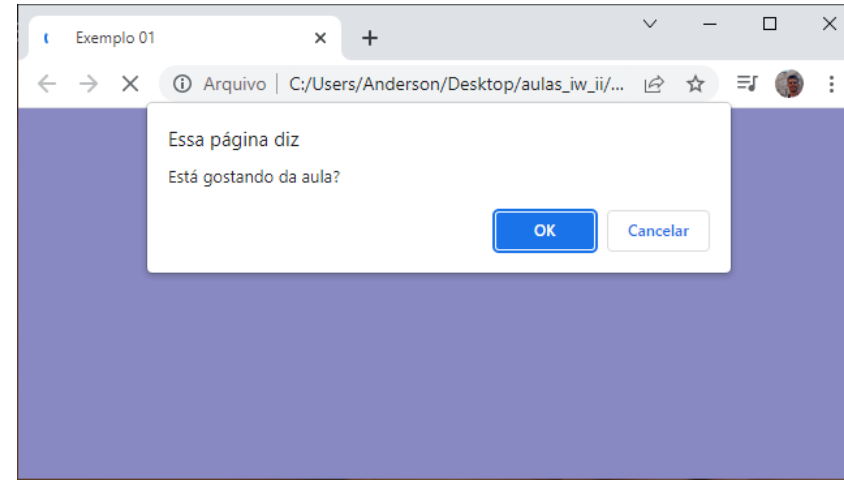
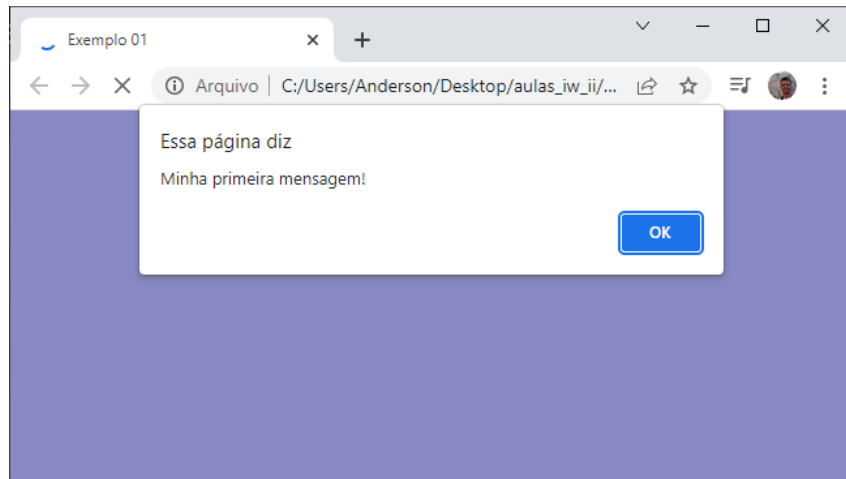
- Agora que colocamos o conteúdo e o estilo, vamos adicionar a parte de interação com o uso de JavaScript. Adicione o seguinte trecho ao código **exemplo\_01.html**:

```
18 <body>
19     <h1>Olá, mundo!</h1>
20     <p>Já me livrei da maldição!</p>
21     <script>
22         alert('Minha primeira mensagem!')
23     </script>
24
25 </body>
```



- Vamos conhecer mais alguns comandos em JS.

```
18 <body>
19   <h1>Olá, mundo!</h1>
20   <p>Já me livrei da maldição!</p>
21   <script>
22     alert('Minha primeira mensagem!')
23     confirm('Está gostando da aula?')
24     prompt('Qual é o seu nome?')
25   </script>
```



# Comandos básicos





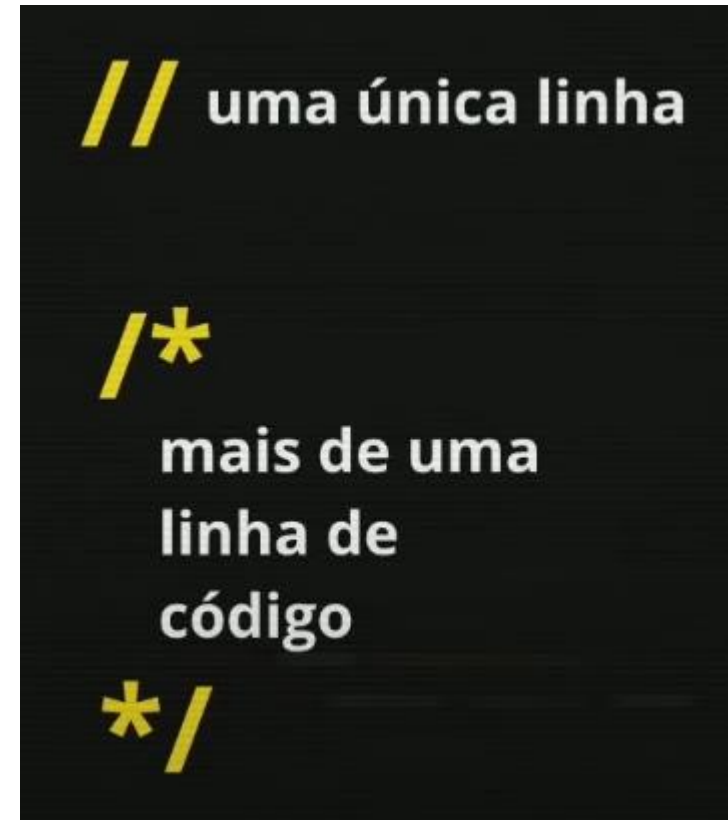
# MÓDULO B

**Armazenando  
dados**

**Tratamento de  
dados**

**Operações com  
dados**

# *TIPOS DE DADOS E VARIÁVEIS*





**n1**

5

```
var n1 = 5
```

**n2**

8.5

```
var n2 = 8.5
```

**n3**

15

```
var n3 = 15
```

```
var s1 = “ETEC MCM”
```

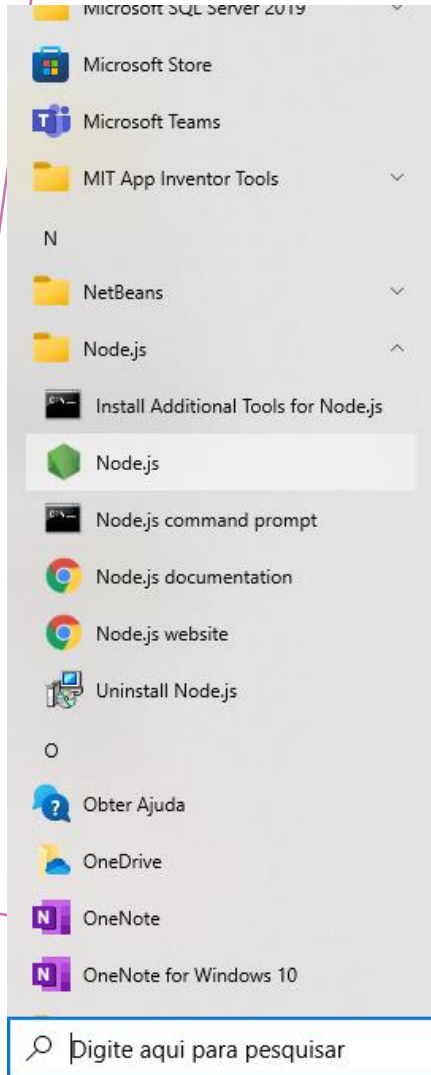
```
var s2 = ‘IW-II’
```

```
var s3 = `Anderson`
```

## Identificadores

- Podem começar com **letra, \$ ou \_**
- Não podem começar com **números**
- É possível usar **letras** ou **números**
- ~~É possível usar **acentos e símbolos**~~
- Não podem conter **espaços**
- Não podem ser **palavras reservadas**

# UTILIZANDO O NODEJS



A screenshot of the Node.js REPL terminal window. The window title is 'Node.js'. The terminal shows the following text:

```
Welcome to Node.js v14.17.0.  
Type ".help" for more information.  
> 'Oi'  
'Oi'  
> 3+2  
5  
> var nome = "Anderson"  
undefined  
> nome  
'Anderson'  
> 'nome'  
'nome'  
> nome = 'Etec MCM'  
'Etec MCM'  
> nome  
'Etec MCM'  
> _  
>
```

The text 'CTRL+L limpa o terminal' is overlaid on the bottom right of the terminal window.

```
> .exit
```



-15.9 5 0.5

8.0 "Google" 3.14

true 'JavaScript' 18

`Maria` -12 false



5 18 -12

0.5 -15.9 3.14 8.0

number

true false

boolean

"Google"

'JavaScript'

`Maria`

string

## Data Types

number

Infinity

NaN

string

boolean

null

undefined

object

Array

function

## typeof

number

string

boolean

null

undefined

object

function

Node.js

```
Welcome to Node.js v14.17.0.  
Type ".help" for more information.  
> var n = 20  
undefined  
> n  
20  
> typeof n  
'number'  
> n = "Google"  
'Google'  
> typeof n  
'string'  
> typeof 6  
'number'  
> typeof "6.5"  
'string'  
> typeof []  
'object'  
> typeof {}  
'object'  
> typeof function(){}  
>
```

The background of the slide is a collage of financial data visualizations. It includes a bar chart at the top left, a line chart with a shaded area in the middle, and a candlestick chart at the bottom left. There are also faint tables of numbers and text scattered throughout the background, suggesting a complex data environment.

# Manipulação de dados

Qual é seu nome?

OK

```
window.prompt('Qual é seu nome?')
```

- Faça uma cópia do arquivo **exemplo\_01.html**, e renomeie para **exemplo\_02.html**
- Remova um trecho de código que não precisaremos neste momento.

```
18  <body>
19      <h1>Olá, mundo!</h1>
20      <p>Já me livre! da maldição!</p>
21      <script>
22          prompt('Qual é o seu nome?')
23      </script>
```

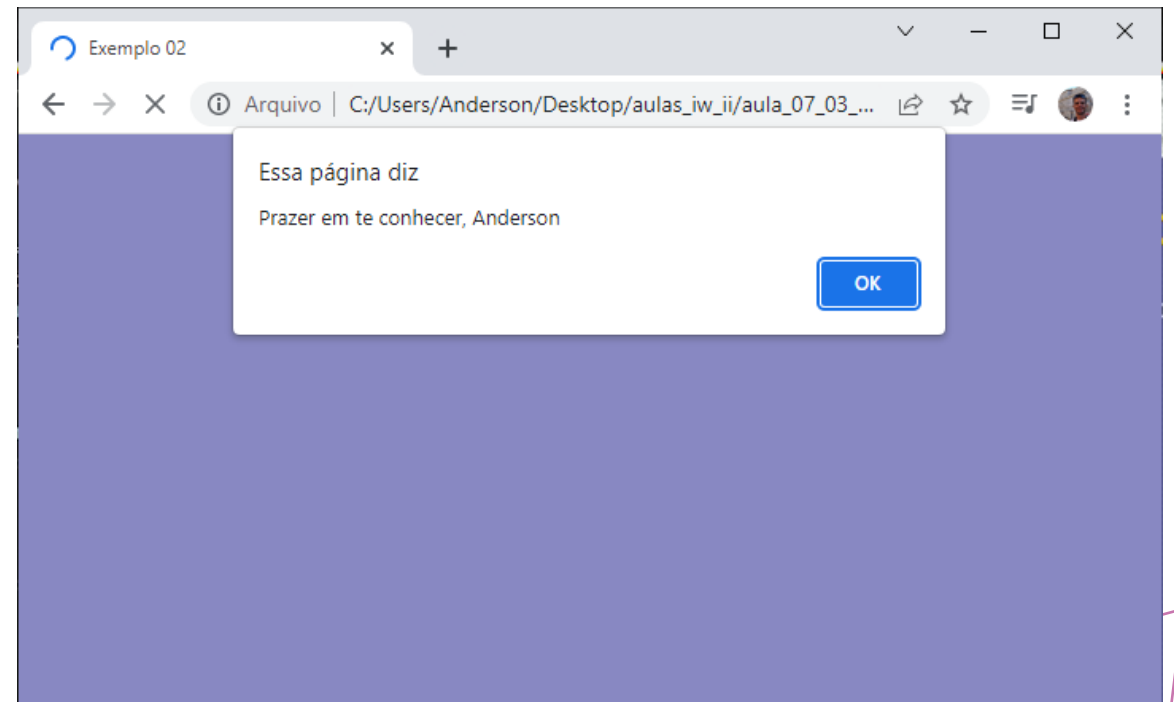
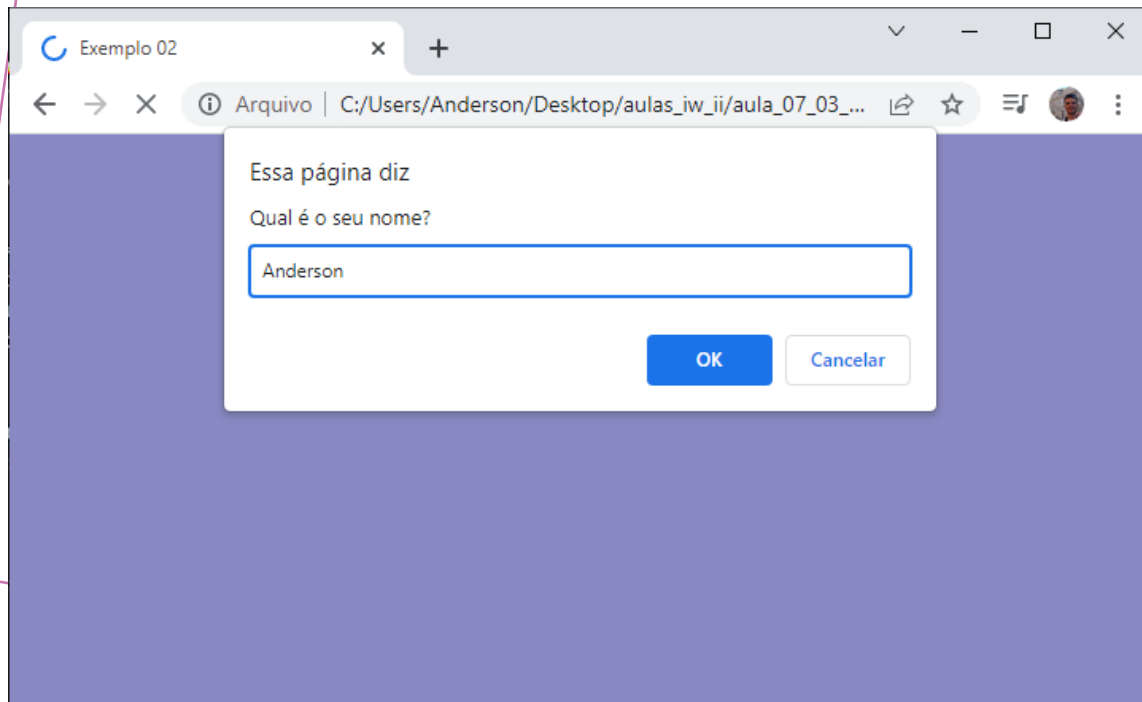
- Ao executar esse comando os valores digitados são perdidos. Precisamos guardar em uma variável. Altere a linha 22 para:

```
21  ✓  <script>
22      var nome = prompt('Qual é o seu nome?')
23  </script>
```



- Para exibir o valor da variável nome na tela, vamos utilizar o comando **alert**. Adicione o seguinte trecho de código:

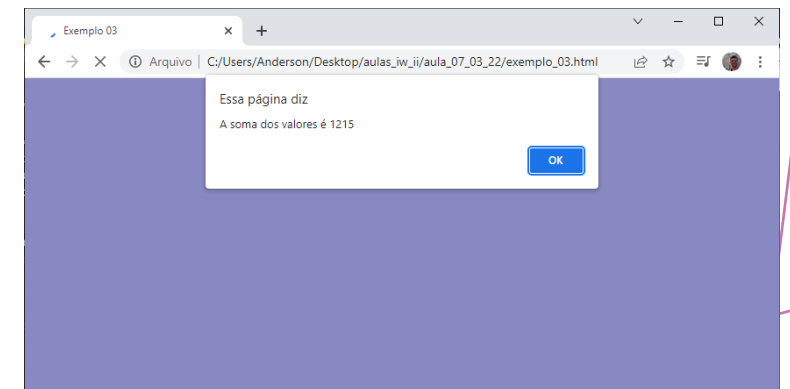
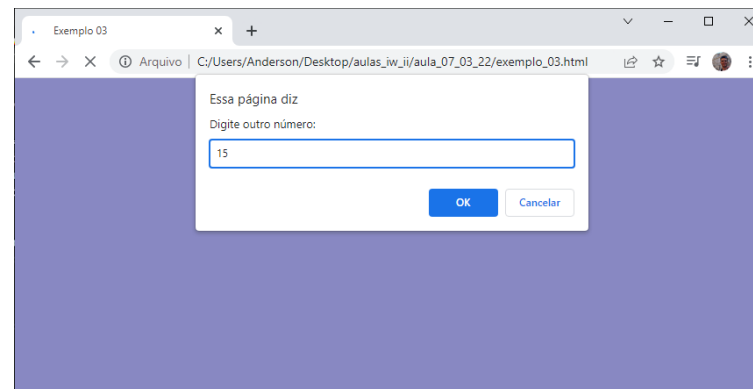
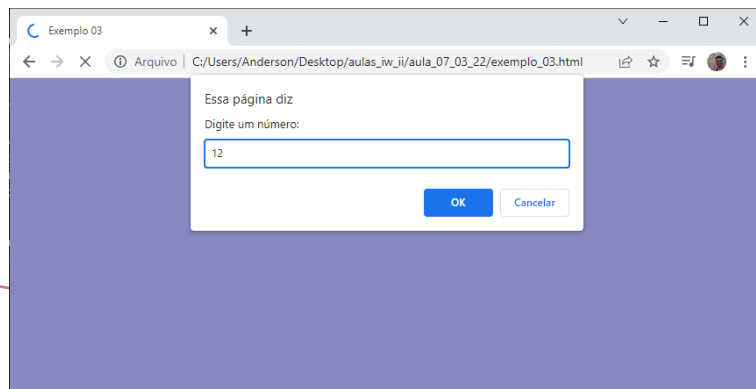
```
21 <script>
22     var nome = prompt('Qual é o seu nome?')
23     alert('Prazer em te conhecer, ' + nome)
24 </script>
```



- Faça uma cópia do arquivo **exemplo\_02.html**, e renomeie para **exemplo\_03.html**
- Remova um trecho de código que não precisaremos neste momento e adicione o seguinte:

```
21 <script>
22     var n1 = prompt('Digite um número: ')
23     var n2 = prompt('Digite outro número: ')
24     var soma = n1 + n2
25     alert('A soma dos valores é ' + soma)
26 </script>
```

- Execute em seu navegador e se atente ao resultado.







*STRING* → *NÚMERO*

```
parseInt(n)
```

```
parseFloat(n)
```

```
Number(n)
```

- O sinal `+` pode ser utilizado tanto para realizar uma soma como para fazer uma concatenação. A soma é sempre realizada para tipos `number` e a concatenação para tipos `string`.
- O uso do comando ***prompt*** **SEMPRE** retorna uma **STRING**, portanto as variáveis ***n1*** e ***n2***, estarão recebendo tipos **STRING** e **não NUMBER**. Isso faz com que os números que foram digitados, sejam considerados pelo JS como **STRING**, fazendo assim com que os valores sejam **CONCATENADOS** e **não somados**!
- Faça a alteração em seu código como mostrado abaixo e execute novamente.

```
21 <script>
22     var n1 = parseInt(prompt('Digite um número: '))
23     var n2 = parseInt(prompt('Digite outro número: '))
24     var soma = n1 + n2
25     alert('A soma dos valores é ' + soma)
26 </script>
```

*NÚMERO* → *STRING*

```
String(n)
```

```
n.toString()
```

# *FORMATANDO STRING*

// Formatando Strings

```
var s = 'JavaScript'
```

```
'Eu estou aprendendo s' // não faz interpolação
```

```
'Eu estou aprendendo' + s // usa concatenação
```

```
`Eu estou aprendendo ${s}` // usa template string
```

- Abra um terminal NodeJs

```
Node.js
> var s = 'JavaScript'
undefined
> 'Eu estou estudando s'
'Eu estou estudando s'
> 'Eu estou estudando ' + s
'Eu estou estudando JavaScript'
>
```

```
Node.js
> nome = 'Anderson'
'Anderson'
> idade = 47
47
> nota = 5.5
5.5
> 'O aluno ' + nome + ' com ' + idade + ' anos, tirou a nota ' + nota
'O aluno Anderson com 47 anos, tirou a nota 5.5'
> _
```

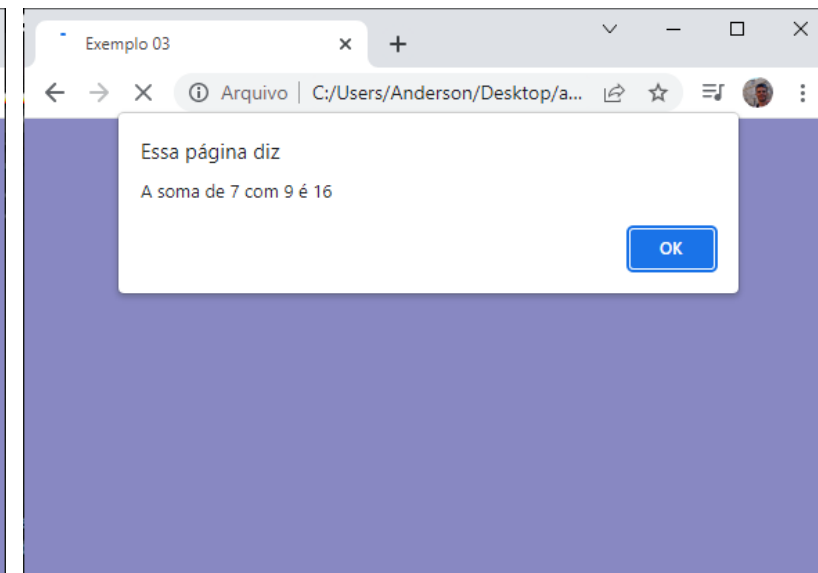
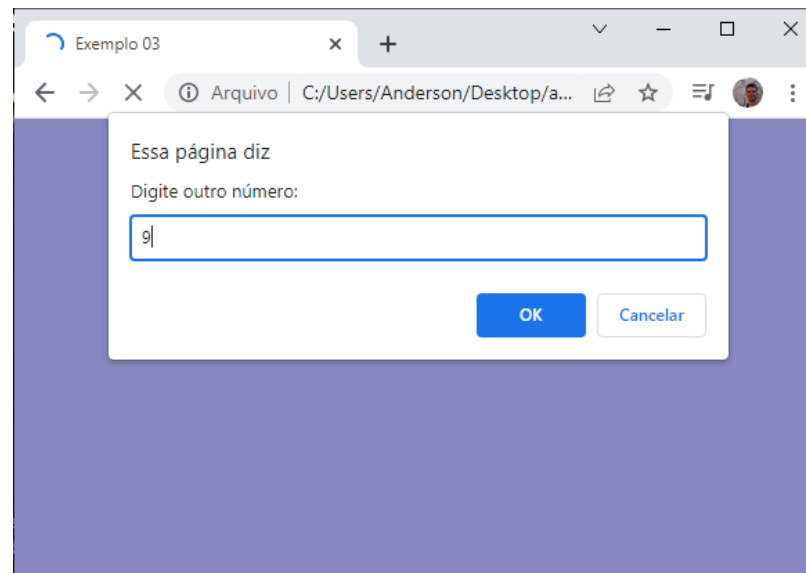
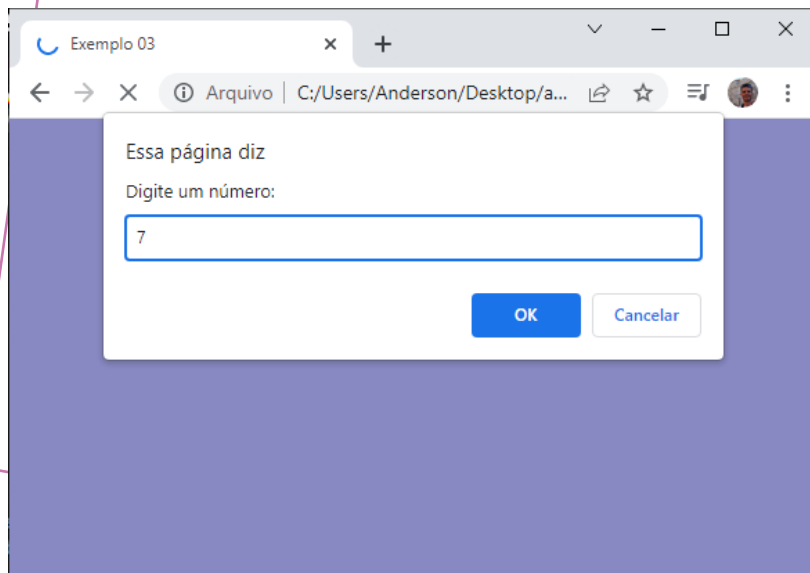
- Podemos usar o que chamamos de Place Holder dentro de uma Template string

```
Node.js
> nome = 'Anderson'
'Anderson'
> idade = 47
47
> nota = 5.5
5.5
> 'O aluno ' + nome + ' com ' + idade + ' anos, tirou a nota ' + nota
'O aluno Anderson com 47 anos, tirou a nota 5.5'
> `O aluno ${nome} com ${idade} anos, tirou a nota ${nota}`
'O aluno Anderson com 47 anos, tirou a nota 5.5'
>
```



- Vamos voltar ao **exemplo\_03.html** e alterar a visualização da soma dos dois números utilizando a técnica de *Template string*. Altere o código como mostrado abaixo:

```
21 <script>
22   var n1 = parseInt(prompt('Digite um número: '))
23   var n2 = parseInt(prompt('Digite outro número: '))
24   var soma = n1 + n2
25   //alert('A soma dos valores é ' + soma)
26   alert(`A soma de ${n1} com ${n2} é ${soma}`)
27 </script>
```



# *FORMATANDO STRING*

```
// Formatando Strings
```

```
var s = 'JavaScript'
```

```
s.length
```

```
// quantos caracteres a string tem
```

```
s.toUpperCase()
```

```
// tudo para 'MAIÚSCULAS'
```

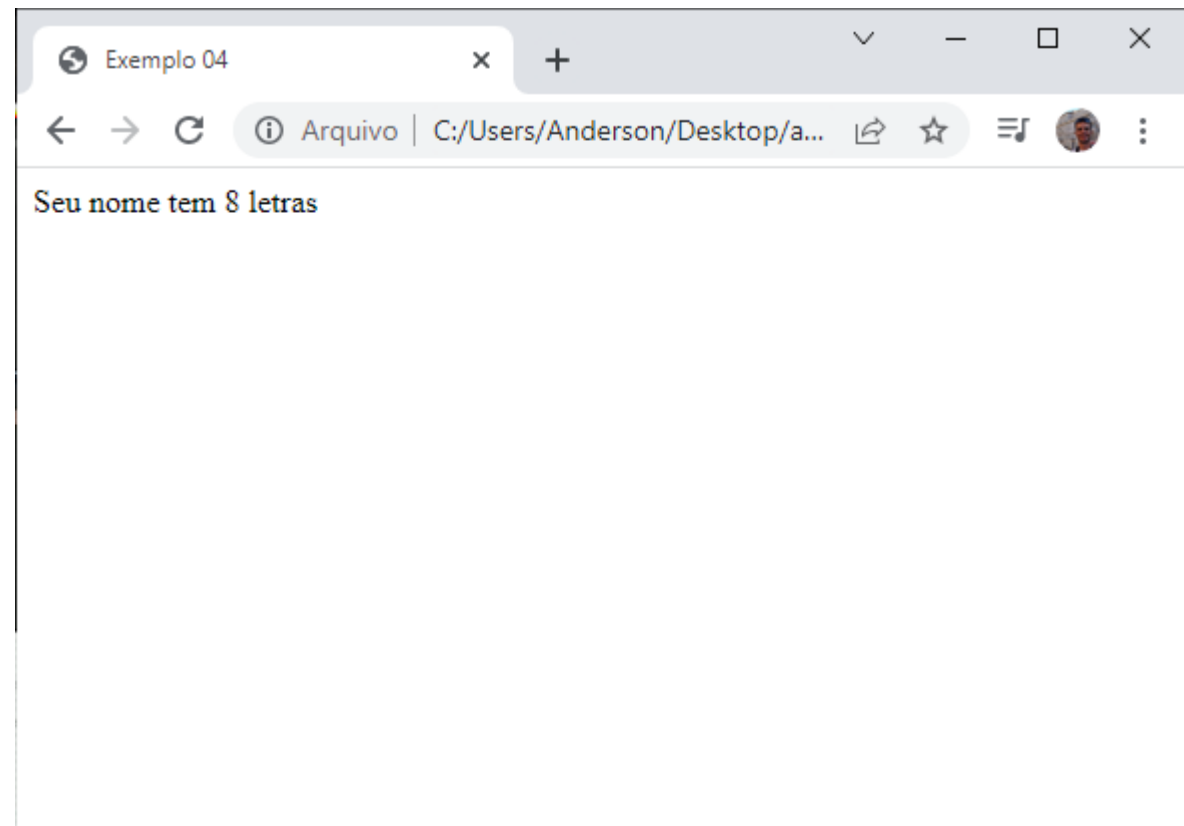
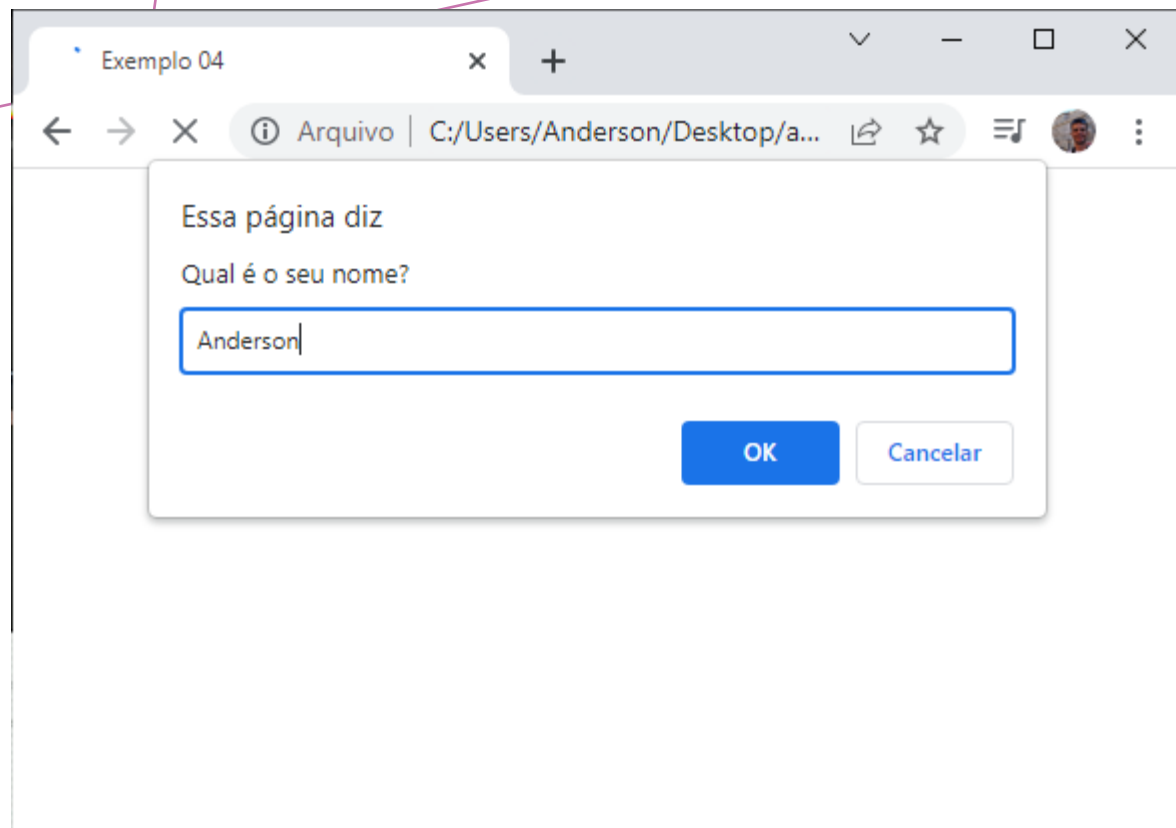
```
s.toLowerCase()
```

```
// tudo para 'minúsculas'
```

- Crie um **novo arquivo (do zero)** chamado **exemplo\_04.html** e insira o seguinte trecho de código.

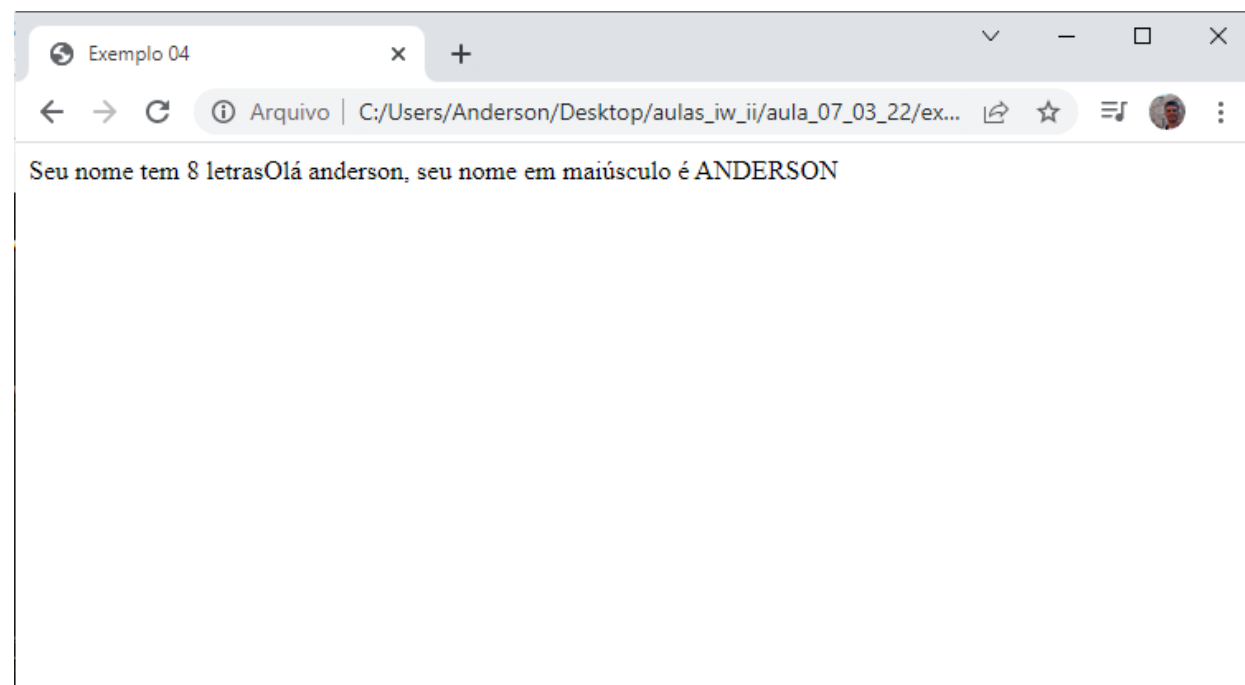
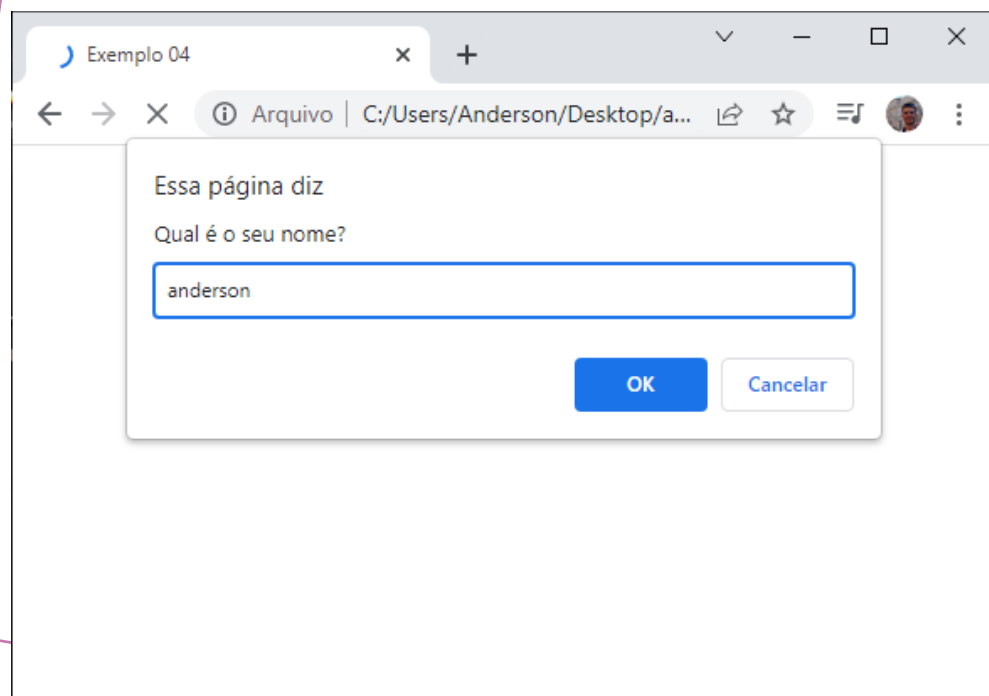
aula\_07\_03\_22 > <> exemplo\_04.html >  html >  body >  script

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
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>Exemplo 04</title>
8  </head>
9  <body>
10     <script>
11         var nome = prompt('Qual é o seu nome?')
12         document.write(`Seu nome tem ${nome.length} letras`)
13     </script>
14 </body>
15 </html>
```



- Altere o código

```
10 <script>
11     var nome = prompt('Qual é o seu nome?')
12     document.write(`Seu nome tem ${nome.length} letras`)
13     document.write(`Olá ${nome}, seu nome em maiúsculo é ${nome.toUpperCase()}`)
14 </script>
```



# *FORMATANDO NÚMEROS*

```
// Formatando Números
```

```
var n1 = 1543.5
```

```
n1
```

```
n1.toFixed(2)
```

```
n1.toLocaleString('pt-BR', {style: 'currency', currency: 'BRL'})
```

# *FORMATANDO NÚMEROS*

- Abra um terminal do nodejs




```
Node.js
> var n1 = 1545.5
undefined
> n1
1545.5
> n1.toFixed(2)
'1545.50'
> n1.toFixed(2).replace('.', ',')
'1545,50'
>
```



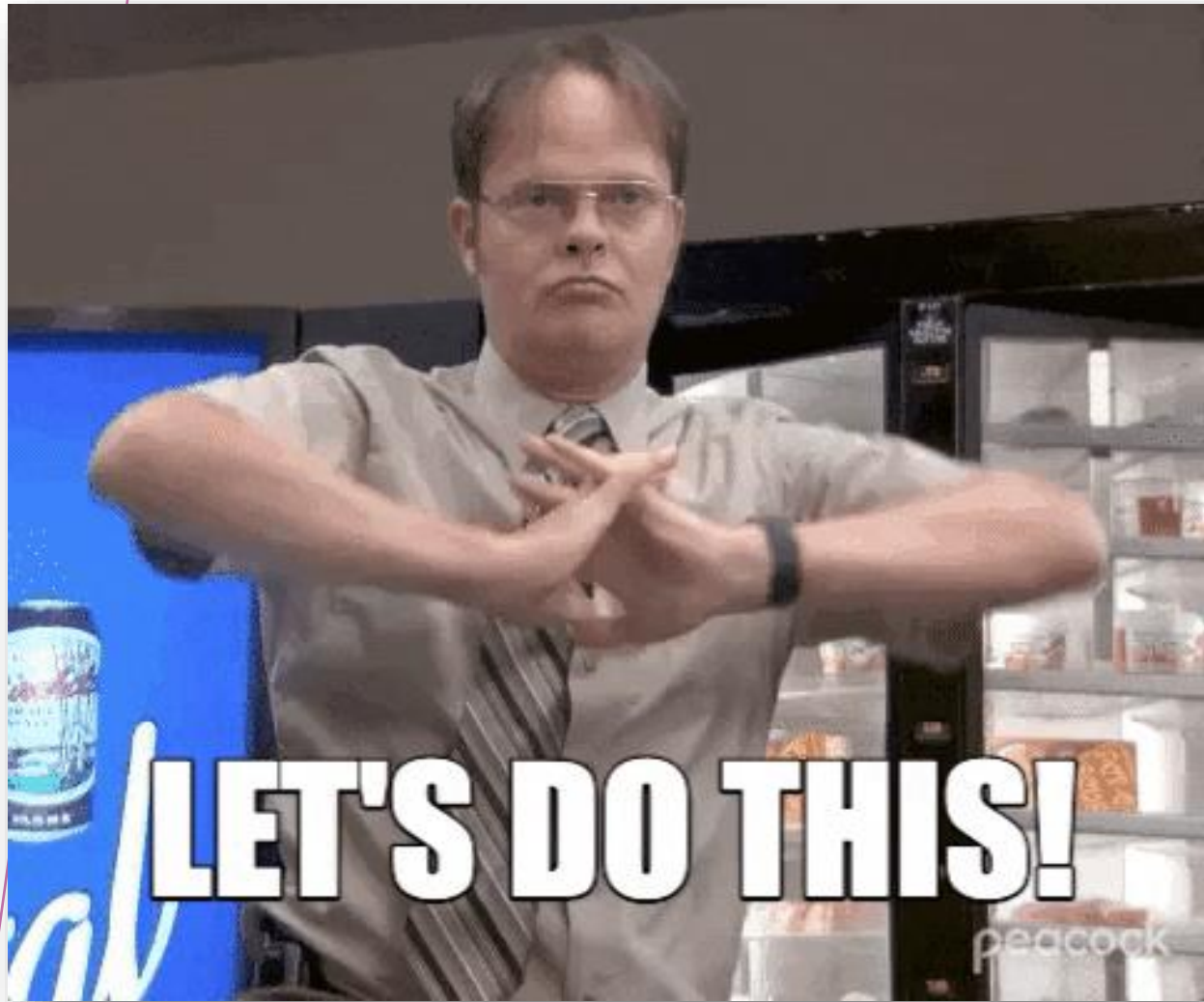
# *FORMATANDO NÚMEROS*

- Abra um terminal do nodejs



```
Node.js
> var n1 = 1545.5
undefined
> n1
1545.5
> n1.toFixed(2)
'1545.50'
> n1.toFixed(2).replace('.', ',')
'1545,50'
> n1.toLocaleString('pt-BR',{style: 'currency',currency: 'BRL'})
'R$ 1.545,50'
> n1.toLocaleString('pt-BR',{style: 'currency',currency: 'USD'})
'US$ 1.545,50'
> n1.toLocaleString('pt-BR',{style: 'currency',currency: 'EUR'})
'€ 1.545,50'
>
```





**Por hoje é isso!**  
**Dúvidas?**  
**Obrigado!**