

UTILIZANDO O JAVASCRIPT EM UMA PÁGINA WEB

JavaScript Interno

Dentro do HTML, pode-se utilizar uma tag `<script>` com o código JavaScript dentro dela, conforme o exemplo a seguir.

```
<script>  
    //Aqui fica a codificação Javascript  
</script>
```

No exemplo que segue, o código JavaScript apresenta uma mensagem para o usuário.

```
<script>  
    alert("Aprendendo JavaScript");  
</script>
```



JavaScript Externo

Esta é a forma mais adequada para se utilizar o JavaScript, pois separamos todo o código JavaScript em um arquivo específico, evitando de misturá-lo com HTML e CSS.

Dentro do arquivo HTML, adicione a marcação a seguir para chamar um arquivo criado com a extensão “.js”.

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

Adicione-a preferencialmente no final do seu arquivo HTML, pois isso faz com que o usuário tenha uma sensação de carregamento mais rápido da página. Normalmente os arquivos JavaScript demoram um pouco para carregar em razão de seu tamanho.

No arquivo “script.js”, você pode adicionar apenas a linha abaixo.

```
alert ("Adicionando Javascript externo na página");
```

O JavaScript também pode ser utilizado *inline*. Isso quer dizer que podemos colocar o código JavaScript dentro do código HTML. No entanto, isso seria ir contra o que queremos fazer para utilizar boas práticas de programação, que é separar os códigos HTML, CSS e JavaScript.

Incluindo JavaScript em páginas HTML5

JavaScript Interno

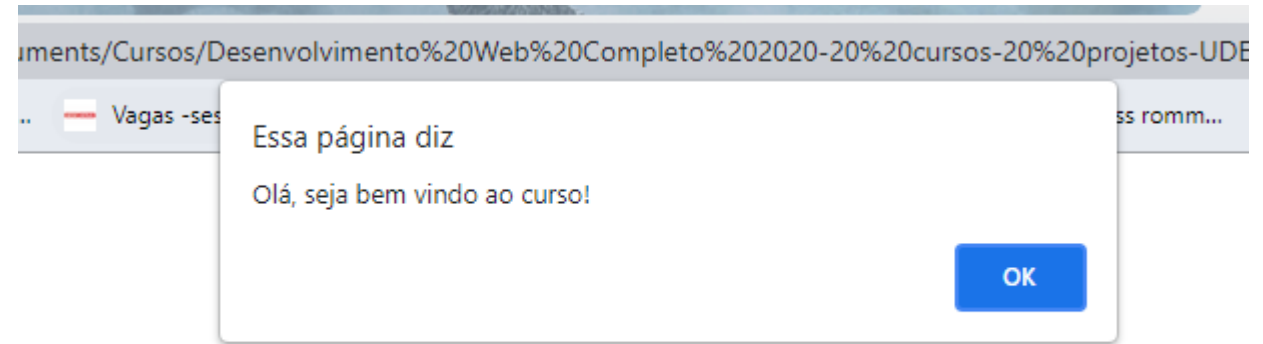
```
<head>
  <meta charset="utf-8">
  <title>JavaScript</title>

  <script>
    alert('Olá, seja bem vindo ao curso!')
  </script>
</head>
```

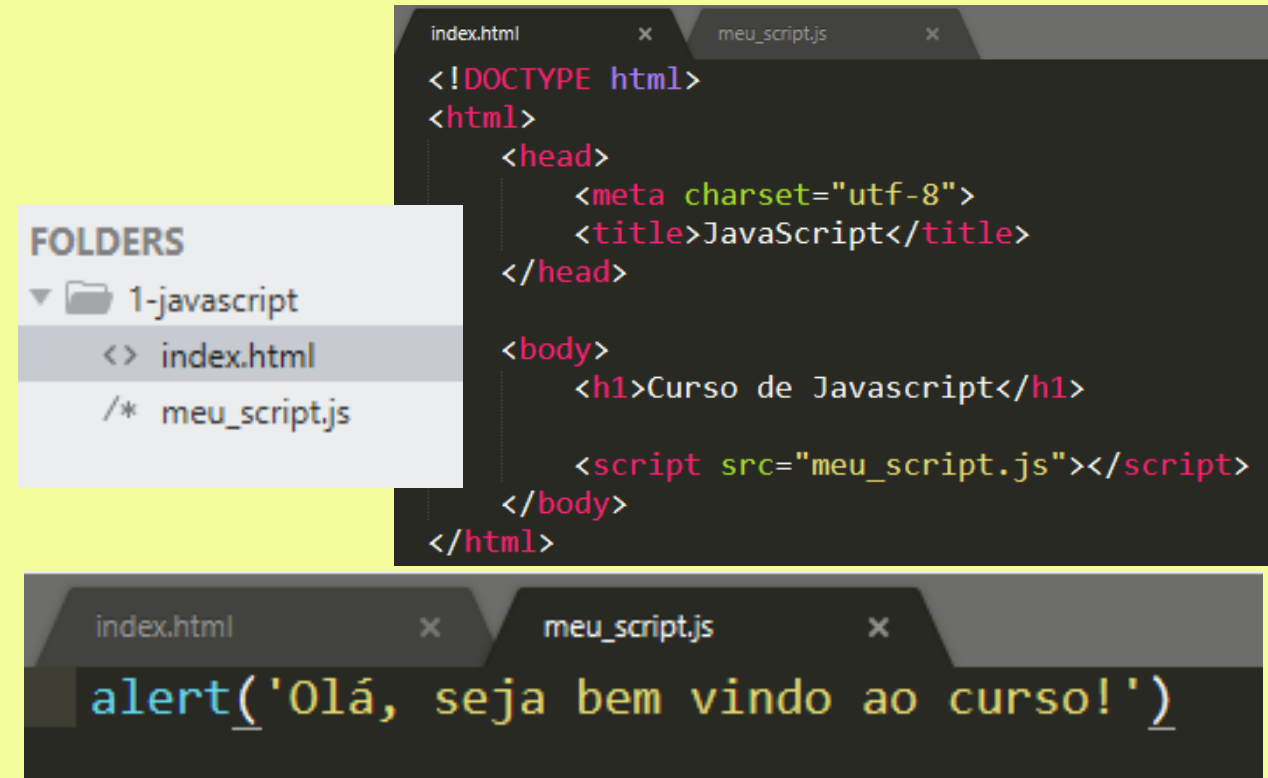
Ou

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>JavaScript</title>
  </head>
  <body>
    <h1>Curso de Javascript</h1>

    <script>
      alert('Olá, seja bem vindo ao curso!')
    </script>
  </body>
</html>
```



JavaScript Externo



Precedência de Execução

- O Javascript é interpretado, logo é processado em tempo de execução, por isso é importante ter em mente a ideia e **lógica de precedência de execução do código**.
- Um erro comum para quem está iniciando é tentar apontar o Javascript para algum elemento que ainda não existe, não foi renderizado no browser. Quando isso ocorre a lógica é quebrado.
- Os elementos são renderizados com base em uma arvore de elementos definidos no **DOM (Documento Object Model / Modelo Objeto de Documento)**. Todos elementos de HTML ficam armazenados no DOM e podem ser acessados pelo Javascript, navegando nessa arvore de elementos.

Entendendo o DOM

<https://tableless.com.br/entendendo-o-dom-document-object-model/>

Nesse primeiro exemplo em que chamamos o script no head, o comando criado gera erro na execução:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>JavaScript</title>

    <script src="meu_script.js"></script>
  </head>

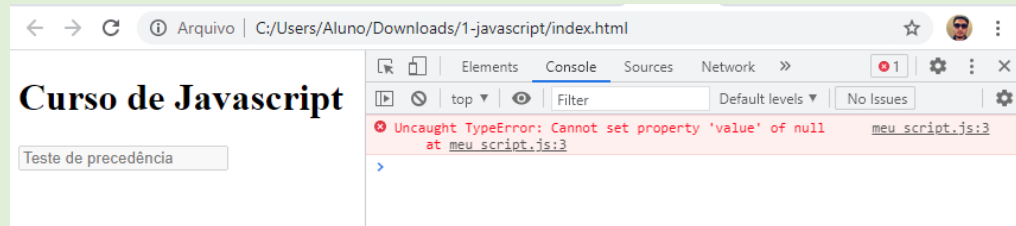
  <body>
    <h1>Curso de Javascript</h1>

    <input
      type="text"
      placeholder="Teste de precedência"
      id="nome"
      disabled="disabled"
    >
  </body>
</html>
```

```
document.getElementById('nome').value = 'Oi'
```

Curso de Javascript

Teste de precedência



Solucionando:

O script é chamado somente após todos elementos HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>JavaScript</title>
  </head>

  <body>
    <h1>Curso de Javascript</h1>

    <input
      type="text"
      placeholder="Teste de precedência"
      id="nome"
      disabled="disabled"
    >

    <script src="meu_script.js"></script>
  </body>
</html>
```

Curso de Javascript

Oi

Variáveis - Introdução

O que são variáveis?

Tipos de variáveis

string

“ “ / “ “

number

Int / Float

boolean

false/true

Regras para declaração de variáveis

Não podem ser iniciadas com números, apenas com letras ou “_”

Não podem ser utilizados caracteres especiais como “ç”, “^”, “~”

Não podem ser iguais as palavras reservadas da linguagem

Declarando variáveis com a instrução “var”

Case sensitive

var curso, var Curso e var CURSO são variáveis diferentes

Exemplos

var nome, var AulaVariaveis, var _Idade

var 123nome, var número

Errado!

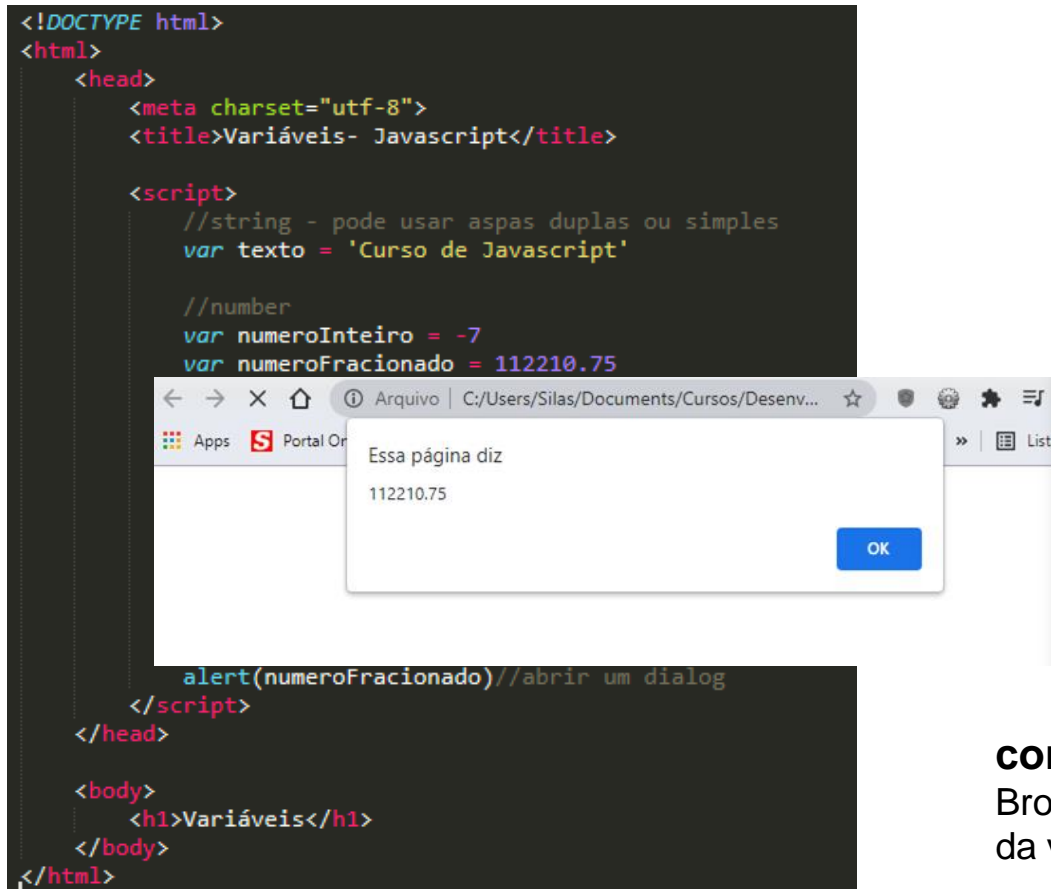
```
1 var nome;  
2 nome = “Fulano de Tal”;  
3 var idade = 30;  
4 idade = 30 + 20 - 10*5;
```

```
<script>  
  //string - pode usar aspas duplas ou simples  
  var texto = 'Curso de Javascript'  
  
  //number  
  var numeroInteiro = -7  
  var numeroFracionado = 112210.75  
  // no navegador deve-se aplicar uma máscara para  
  mostrar os nmeros com vírgula, seguindo o nosos  
  padrão  
  
  //boolean - sao usadas em testes condicionais  
  var teste = true //true ou false  
</script>
```

3 FORMAS BÁSICAS PARA TESTAR E TRABALHAR COM VARIÁVEIS

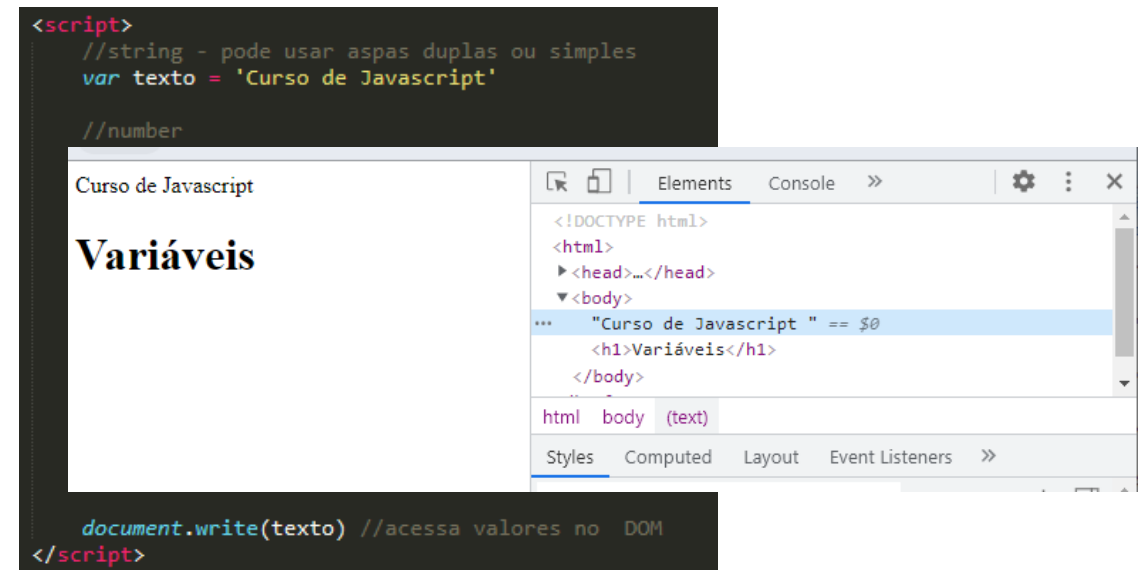
alert(nome-variavel) →

usado para abrir uma caixa de dialogo



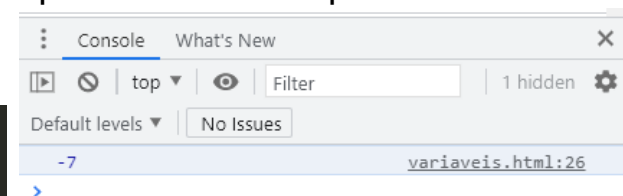
document.write(nome-variavel) → Acessa valores no DOM

No exemplo repare que ele faz aparecer o conteúdo da variável "texto" na página, pelo console é possível ver que o elemento aparece na página pois foi inserido no DOM:



console.log() → Permite realizar testes através da ferramenta de console do Browser. Lá aparecerá os resultados ou valores (ótimo para testar o comportamento da variável em determinado trecho do código)

```
console.log(numeroInteiro) //Expor o valor da
variável no CONSOLE.
```



Var, Let e Const

Var permite recriar uma variável com o mesmo nome, já **Let** não permite.

```
// identificador -palavraReservada : var e let
var a = 3
let b = 4 // de regra geral utilize let para criar variavel

var a = 30
let b = 40

console.log(a, b)
```

Tentar rodar isso apresenta um erro dizendo que a variável b já foi declarada.

```
// identificador -palavraReservada : var e let
var a = 3
let b = 4 // de regra geral utilize let para criar variavel

var a = 30
b = 40

console.log(a, b)
```

Dessa forma rodará apresentando no console **30 40**.

Const não permite mudar o valor da variável.

```
const c = 5
/*c = 50 >>>> se tentar isso em sequência não funciona
* pois o valor da const não pode ser mudado */
console.log(c)
```


Tipagem Fraca

JavaScript é uma linguagem dinâmica, ou seja, de linguagem fracamente tipada. Não identificamos o tipo da variável, ou seja, ele pode ser string depois um número depois uma função.

```
let qualquer = 'Legal'  
console.log(qualquer)  
console.log(typeof qualquer)
```

```
qualquer = 3.1415  
console.log(qualquer)  
console.log(typeof qualquer)
```

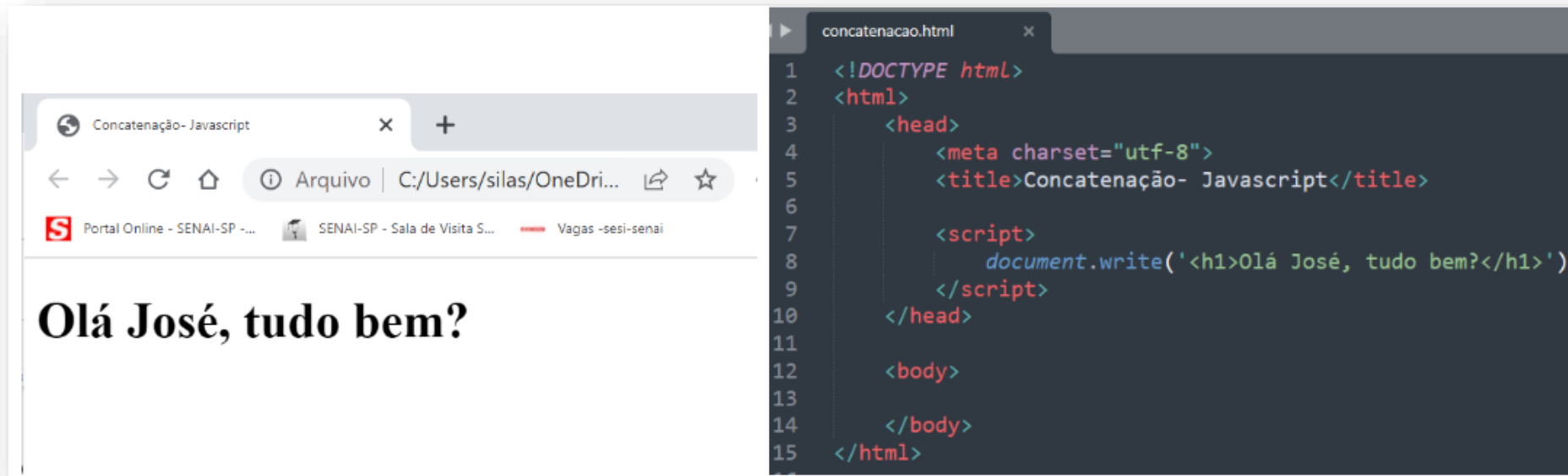
```
//Evite nomes genéricos e siglas nas variáveis. Exemplo ruim:  
let valor = ''  
let numero = 5540  
let pqp = false // PQP = Produto Químico Perigoso, rsrs....
```

`console.log(typeof qualquer)`

Typeof mostra o tipo da variável

Concatenação

Vamos usar o comando **document.write** para inserir uma tag na nossa página:



Apertando F12 na página podemos ver o elemento `<h1>` criado no corpo na página:



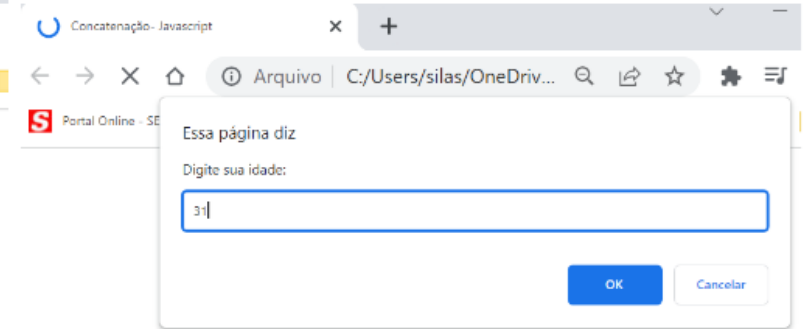
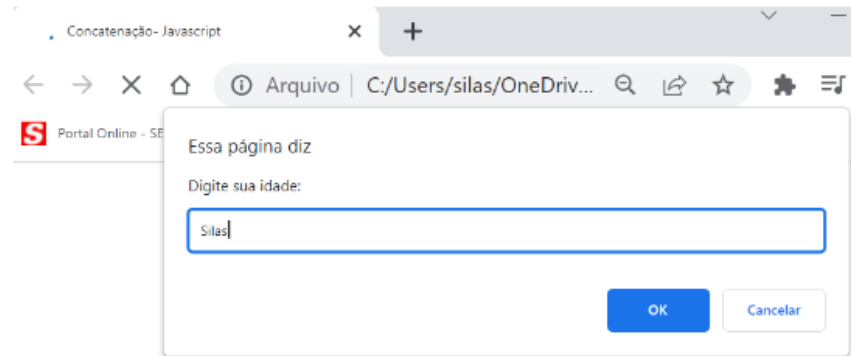
Concatenação

Concatenando para deixar o nome dinâmico:

```
7 <script>
8
9 var nome = 'Maria'
10 var idade = 27
11
12
13 document.write('<h1>Olá ' + nome + ' , tudo bem? Estou vendo
14 aqui que você possui ' + idade + ' anos.</h1>')
15 </script>
```

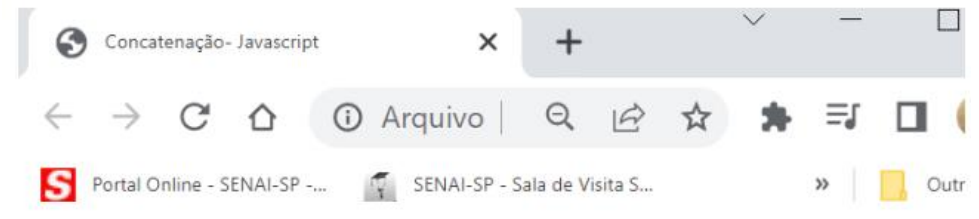
Olá Maria , tudo bem? Estou vendo aqui que você possui 27 anos.

Atribuindo valores dinâmicos, usando o o **método Prompt**:



```
var nome = prompt('Digite seu nome:')
var idade = prompt('Digite sua idade:')

document.write('<h1>Olá ' + nome + ' , tudo bem? Estou vendo
aqui que você possui ' + idade + ' anos.</h1>')
```



Olá Silas , tudo bem? Estou vendo aqui que você possui 31 anos.

Concatenação

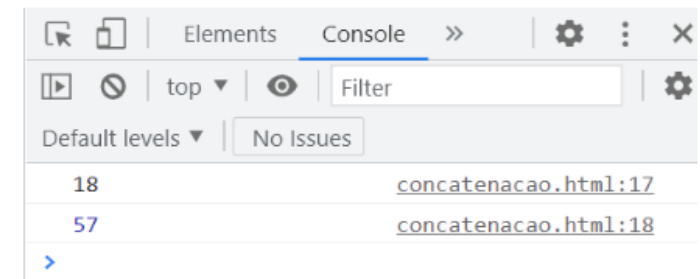
O **prompt** sempre recebe os valores digitados como tipo **string**. Olhando pelo **console.log** podemos ver a diferença de representação de dois valores, um que está como texto e o outro como numérico:

```
var nome = prompt('Digite seu nome:');
var idade = prompt('Digite sua idade:');
var idade2 = 57

console.log(idade)
console.log(idade2)

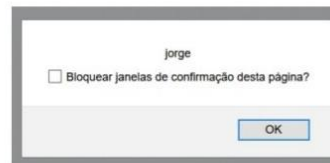
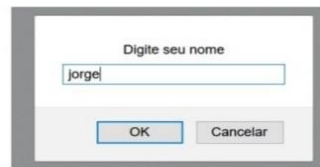
document.write('<h1>Olá ' + nome + ' , tudo bem? Estou vindo aqui que você possui ' + idade + ' anos.</h1>')
```

Olá João , tudo bem? Estou vindo aqui que você possui 18 anos.



Método Prompt()

```
1 <html lang="pt-br">
2 <head>
3   <meta charset="utf-8">
4   <Script type="text/javascript">
5     nome = prompt("Digite seu nome");
6     alert(nome);
7   </Script>
8 </head>
9 <body>
10  <p>Java Script Prompt</p>
11 </body>
12 </html>
```



Métodos Javascript

Javascript alert, confirm e prompt: caixas de diálogo PopUp!

<https://blog.betrybe.com/javascript/javascript-alert/>

03 - Curso de JavaScript - Métodos alert, confirm, prompt e document.write

<https://www.youtube.com/watch?v=-ip-UXa3Aj8>

Curso JavaScript #37 - Alterar conteúdo do elemento (textContent e innerHTML)

<https://www.youtube.com/watch?v=nA3QJYFGsYc>

Variáveis null e undefined

Null →

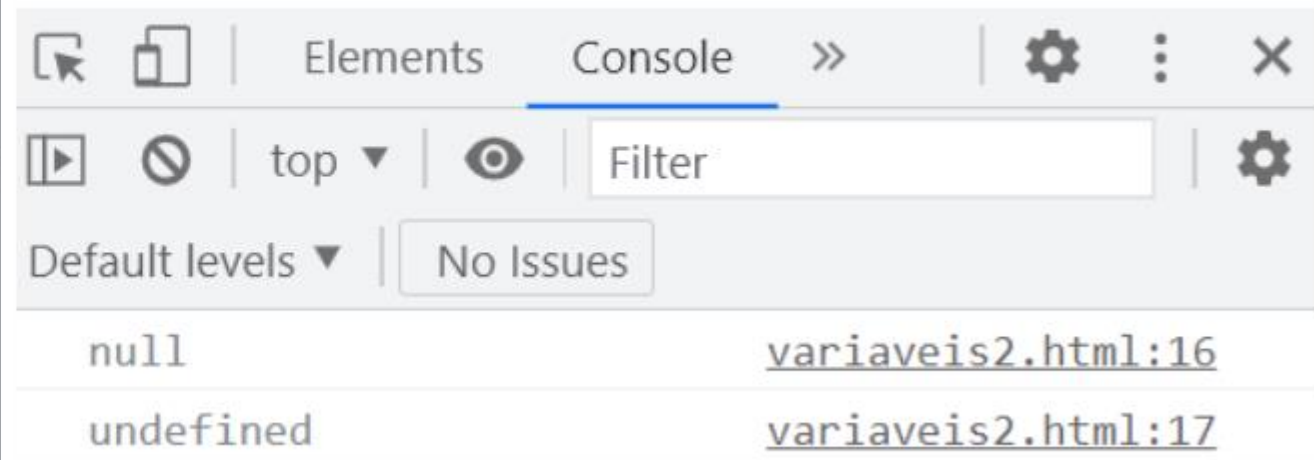
- representa a ausência intencional de um valor.
- Funções podem retorna um valor null

Undefined →

- A variável apesar de declarada não possui nenhum valor.
- quando tentamos acessar informações que estão indisponíveis. EX: Se tentamos acessar no navegador um elemento que não exista, termos como retorno “indefinido / undefined”

Podemos visualizar no console como a variável teste2 será classificada com undefined:

```
7  <script>
8
9      //Null
10     var teste = null
11
12     //Undefined
13     //var teste2 = undefined
14     var teste2
15
16     console.log(teste)
17     console.log(teste2)
18
19 </script>
```



Prática

Trocar valores



Desafio:

Criar uma página que contenha um script JavaScript com 3 variáveis onde cada uma deve receber os seguintes valores:

var a = a

var b = 20

var c = null

Feito isso, imprimir os valores usando o **método write** do objeto documento para que o usuário consiga interagir com a aplicação de modo a saber o que existe dentro de cada uma das variáveis.

O desafio é inverter os valores das variáveis **a** e **b** usando uma lógica. A variável **c** será uma variável auxiliar.

```
<script>

  var a = 10
  var b = 20
  var c = null // variavel auxiliar utilizada
na lógica

  document.write('A: ' +a+ '<br/>')
  document.write('B: ' +b+ '<br/>')
  document.write('C: ' +c+ '<br/>')

  document.write('<hr/>')

  //lógica de inversão dos valores das
variaveis A e B

  //
  document.write('A: ' +a+ '<br/>')
  document.write('B: ' +b+ '<br/>')
  document.write('C: ' +c+ '<br/>')

</script>
```