

Curso: Front-End do Bem



AULA 23

- Booleans
- Exemplos de Booleans
 - Método Includes()
- Exemplos do Método Includes()
- Sugestões de Prática para essa aula
- O que vamos aprender na aula 24



Booleans

Igor, Igor, Igor..... Esse termo Booleans para quem está começando e/ou migrando para essa área de Desenvolvimento é uma novidade, então poderia explicar com exemplos para visualizarmos também na prática como o Booleans funciona.

Show de bola, esse termo Booleans pode parecer algo de “outro mundo” em um primeiro contato, mas explicaremos de um jeito lúdico para que vocês possam entender como ele funciona. Mais uma vez, um ótimo questionamento. Estou gostando muitooooooooooooo disso!!!!

Então bora lá, ver esse tal de Booleans...

Booleans nos mostra 2 valores no JavaScript(JS): False or True (Falso ou Verdadeiro). É só isso que nós precisamos saber nesse momento. Conforme o decorrer dessa aula, veremos mais detalhes sobre Booleans com o auxílio dos exemplos.

E bora ver o primeiro exemplo de Booleans.....

Ah, antes disso..... dentro do nosso nobre arquivo app.js, na linha 1 criaremos um comentário: `// Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal`

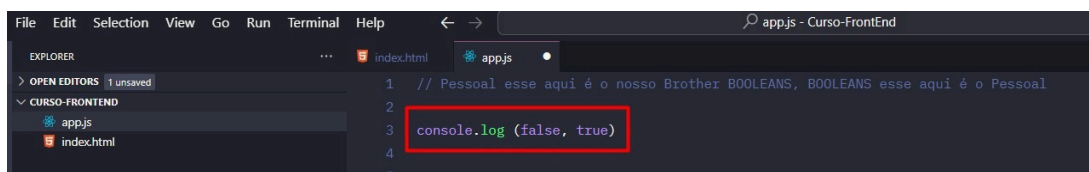
Então logo abaixo do comentário: `// Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal`, especificaremos um `console.log` e passaremos como argumento um `false` e um `true` (só para esquentarmos os nossos motores). Observem que esse `false` e esse `true` podem confundir com strings, só que esses 2 valores não são considerados strings.



Booleans

Alguém, lembra quando consideramos uma string?? Isso mesmo, para que seja considerada uma string nós precisaríamos envolver tanto false quanto true entre aspas (Sejam elas simples ou duplas). Então veremos como o false e o true sem aspas, serão declarados no console.log. Ficando assim:

```
console.log (false, true)
```



```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 console.log(false, true)
4
```

E ao salvarmos o arquivo app.js com aquele CTRL + S maroto podemos ver no console do Navegador(Chrome) que false true são mostrados.



Bora exemplificar melhor aqui (para entendermos quando é uma string com aspas e quando não é sem aspas). No Primeiro argumento partiu inserir um false, no segundo argumento partiu inserir um true, no terceiro argumento partiu inserir uma string 'false', no quarto argumento partiu inserir uma string 'true', no quinto argumento partiu inserir um true, no sexto elemento partiu inserir um false, no sétimo elemento partiu inserir uma string 'true', e no oitavo elemento partiu inserir uma string 'false' .

Bora ver na próxima página, o Passo a passo do que foi feito no texto acima:



Booleans

- 1 - Definimos que vamos usar o `console.log()`
- 2 - Abriremos o parênteses (logo após o `console.log(`
- 3 - Dentro dessa abertura dos parênteses, vamos inserir o nosso primeiro argumento que é `false`. Ficando assim: `console.log(false`
- 4 - Depois do `false`, vamos inserir uma vírgula , e daremos um espaço. Ficando assim: `console.log(false,`
- 5 - Com o espaço dado, vamos inserir o nosso segundo argumento que é `true`. Ficando assim: `console.log(false, true`
- 6 - Depois do `true`, vamos inserir uma vírgula , e daremos um espaço. Ficando assim: `console.log(false, true,`
- 7 - Com o espaço dado, vamos inserir o nosso terceiro argumento que é uma string `'false'`. Ficando assim: `console.log(false, true, 'false'`
- 8 - Depois da string `'false'`, vamos inserir uma vírgula, e daremos um espaço. Ficando assim: `console.log(false, true, 'false',`
- 9- Com o espaço dado, vamos inserir o nosso quarto argumento que é uma string `'true'`. Ficando assim: `console.log(false, true, 'false', 'true'`
- 10 - Depois da string `'true'`, vamos inserir uma vírgula, e daremos um espaço. Ficando assim: `console.log(false, true, 'false', 'true',`
- 11 - Com o espaço dado, vamos inserir o nosso quinto argumento que é o `true`. Ficando assim: `console.log(false, true, 'false', 'true', true`
- 12- Depois do `true`, vamos inserir uma vírgula , e daremos um espaço. Ficando assim: `console.log(false, true, 'false', 'true', true,`
- 13 - Com o espaço dado, vamos inserir o nosso sexto argumento que é o `false`. Ficando assim: `console.log(false, true, 'false', 'true', true, false`
- 14 - Depois do `false`, vamos inserir uma vírgula , e daremos um espaço. Ficando assim: `console.log(false, true, 'false', 'true', true, false,`



/igor-rebolla

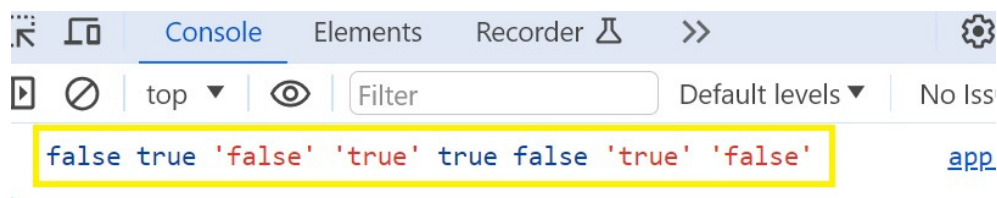
Booleans

Continuação do Passo a passo do que foi feito na página anterior:

- 15 - Com o espaço dado, vamos inserir o nosso sétimo argumento que é a string 'true'. Ficando assim: `console.log(false, true, 'false', 'true', true, false, 'true')`
- 16 - Depois da string 'true', vamos inserir uma virgula, e daremos um espaço. Ficando assim: `console.log(false, true, 'false', 'true', true, false, 'true',`
- 17 - Com o espaço dado, vamos inserir o nosso oitavo argumento que é a string 'false'. Ficando assim: `console.log(false, true, 'false', 'true', true, false, 'true', 'false')`
- 18 - Como a string 'false' é o último argumento dentro dos parênteses do `console.log...` não precisamos adicionamos uma vírgula aqui, basta fecharmos os parênteses `)` que o nosso `console.log` estará pronto. Ficando assim:

```
console.log (false, true, 'false', 'true', true, false, 'true', 'false')
```

E ao salvarmos o arquivo `app.js` com aquele CTRL + S maroto podemos ver no console do Navegador(Chrome) que `false, true, 'false', 'true', true, false, 'true', 'false'` foram mostrados.



Booleans

Resuminho Maroto do Exemplo Acima:

O primeiro valor(1º), o segundo valor(2º), o quinto valor(5º) e o sexto valor (6º) passados como argumentos, são diferentes do terceiro valor(3º), do quarto valor(4º), do sétimo valor(7º) e do oitavo valor(8º) passados como argumentos, pois tanto o false(1º) e o true(2º), quanto o true(5º) e o false (6º) são booleans não possuem as aspas, já o 'false'(3º), o 'true'(4º), o 'true'(7º) e o 'false'(8º) são strings que possuem as aspas.

Podemos observar nesse exemplo que o console do Navegador(Chrome) foi brother demais conosco ele inseriu de forma amigável: aspas nas strings, pra que pudéssemos “bater os olhos” e notarmos a diferença entres esses valores.

Igor, Igor, Igor... belezinha ficou mais claro depois dessa explicação e dos exemplos, poderia fazer um resumo.... E quando nós usaremos Booleans? Só se for agora:

Quando precisarmos verificar uma condição dentro do código que estamos desenvolvendo ou seja vão existir ocasiões que será necessário verificarmos se um bloco de código é false ou true ou seja falso ou verdadeiro.

E pra tornar esse “paranauê” de Booleans ainda mais divertido... poderemos utilizar métodos que nos mostram um Boolean que por sua vez retornará um false ou um true. Podemos utilizar métodos que nos devolvem um Boolean (false ou true). E na próxima página veremos, esse método que é o: Includes

Partiu ver sobre Includes... uhuuuuu estou animado pra vermos isso!!!



Método Includes()

Igor, Igor, Igor, Igor, Igor... E esse tal de método Includes, o que seria?
Igor, responde: O tal do método includes() vai nos informar se um array/objeto possui um determinado item ou não, retornando false ou true.

A explicação acima foi o resumo do resumo do que é o Método Includes, só para vocês terem uma ideia do que falaremos, pois com os exemplos na sequência, o entendimento ficará mais claro e depois do primeiro exemplo, passaremos um Resumo Maroto do que é o Includes em parceria com o exemplo que foi visto.

Pode ser pessoal? Pode sim, Igor!!!

Uhulllll....Então bora ver o primeiro exemplo do Includes...

Como nós já criamos dentro do arquivo app.js o nosso comentário `// Pessoal` esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal, daremos um enter para quebrarmos uma linha e na linha 3, declararemos uma `let houseRentValue` que recebe uma string `'R$ 1000 - É pra alugar hoje'`. Ficando assim:

```
let houseRentValue = 'R$ 1000 - É pra alugar hoje'
```



Método Includes()



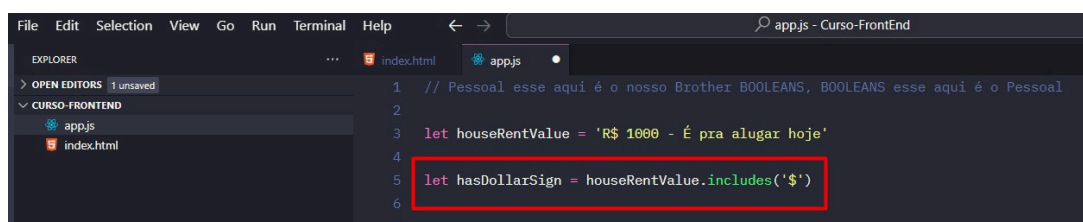
A screenshot of the Visual Studio Code editor interface. The Explorer panel on the left shows a project named 'CURSO-FRONTEND' with files 'app.js' and 'index.html'. The main editor window is open to 'app.js'. The code in the editor is as follows:

```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
4
5
```

The line `let houseRentValue = 'R$ 1000 - É pra alugar hoje'` is highlighted with a red rectangular box.

Declaremos uma `let hasDollarSign` que recebe `houseRentValue.includes()` e que vai receber dentro dos parênteses `()` por argumento uma string `'$'`. Ficando assim:

```
let hasDollarSign = houseRentValue.includes('$')
```



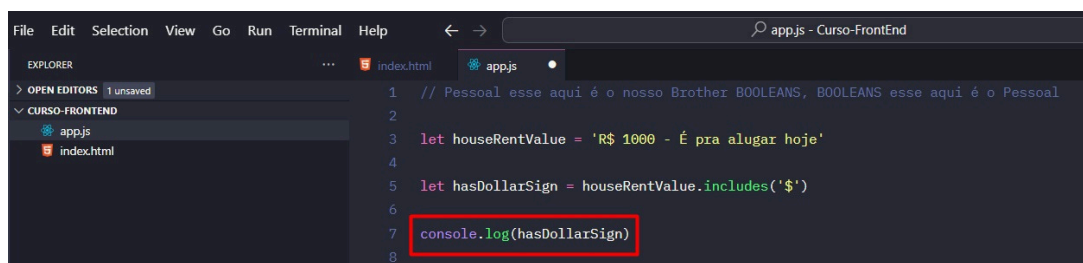
A screenshot of the Visual Studio Code editor interface, similar to the previous one. The code in 'app.js' is now:

```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
4
5 let hasDollarSign = houseRentValue.includes('$')
6
7
```

The line `let hasDollarSign = houseRentValue.includes('$')` is highlighted with a red rectangular box.

Adicionaremos um `console.log` passando a `hasDollarSign`. Ficando assim:

```
console.log(hasDollarSign)
```



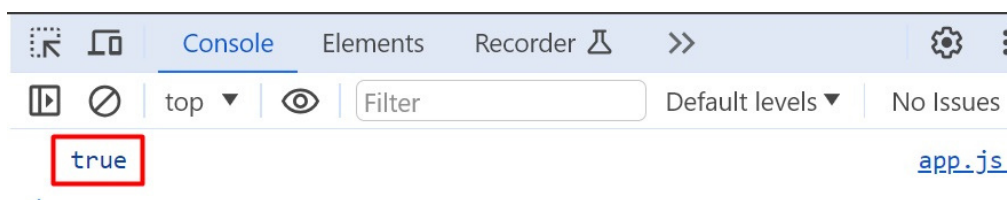
A screenshot of the Visual Studio Code editor interface. The code in 'app.js' is now:

```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
4
5 let hasDollarSign = houseRentValue.includes('$')
6
7 console.log(hasDollarSign)
8
```

The line `console.log(hasDollarSign)` is highlighted with a red rectangular box.

Método Includes()

E ao salvarmos o arquivo app.js com aquele CTRL + S maroto podemos ver no console do Navegador(Chrome) que true está sendo mostrado.



Hora do Resumo Maroto:

O que é o tal do Método Includes em Parceria com o exemplo visto acima.

O ilustríssimo método verificará se uma string '\$' que utilizamos no nosso exemplo, se ela(string) consta dentro da nossa string: houseRentValue. Se essa string '\$' fizer parte ou seja se ela estiver dentro da string houseRentValue, a expressão houseRentValue.includes('\$') retornará true, agora se essa string '\$' não estiver dentro da string houseRentValue, essa expressão houseRentValue.includes('\$') retornará false.

Fim do Resumo Maroto!!!

No Resumo Maroto nós usamos o método includes para verificarmos quando consta um valor, no nosso exemplo utilizamos o '\$'.

Agora será que é possível fazermos algo para verificarmos caso não tenha o valor dentro da string?? Fica a pergunta do amigo Internauta - Igor (No caso eu mesmo)!!!

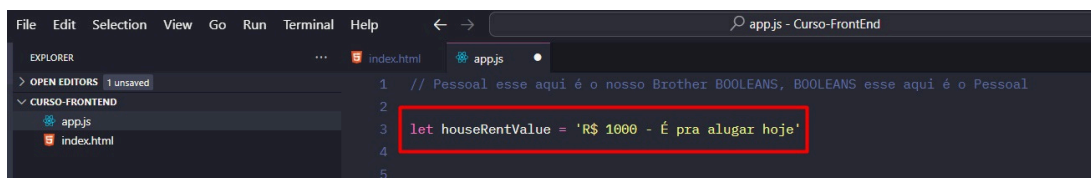
Bora ver isso na próxima página....



Método Includes()

Antes manteremos sem nenhuma alteração a nossa primeira `let` `houseRentValue = 'R$ 1000 - É pra alugar hoje'`. Ficando assim:

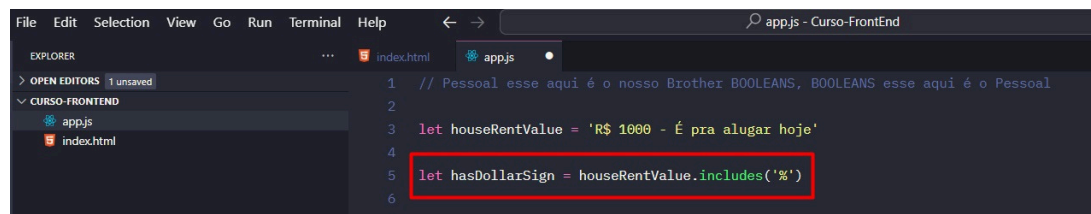
```
let houseRentValue = 'R$ 1000 - É pra alugar hoje'
```



```
File Edit Selection View Go Run Terminal Help
appjs - Curso-FrontEnd
EXPLORER
  > OPEN EDITORS 1 unsaved
  > CURSO-FRONTEND
    appjs
    index.html
  appjs
    1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
    2
    3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
    4
    5
```

E na nossa segunda `let`, passaremos a string `'%'`, como argumento dentro do método `includes()`. Ficando assim:

```
let hasDollarSign = houseRentValue.includes('%')
```

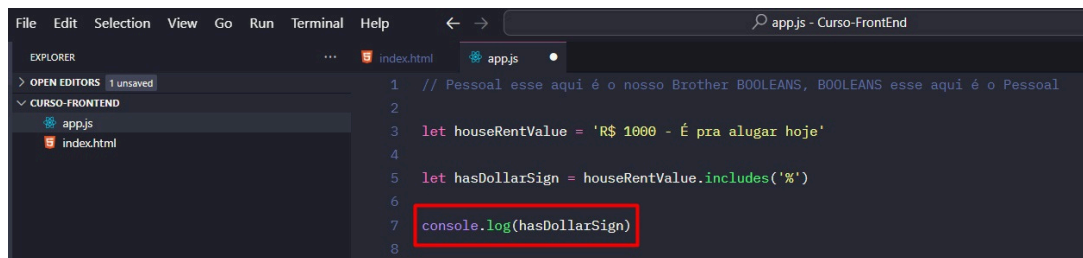


```
File Edit Selection View Go Run Terminal Help
appjs - Curso-FrontEnd
EXPLORER
  > OPEN EDITORS 1 unsaved
  > CURSO-FRONTEND
    appjs
    index.html
  appjs
    1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
    2
    3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
    4
    5 let hasDollarSign = houseRentValue.includes('%')
    6
    7
```

Adicionaremos um `console.log` passando a `hasDollarSign`. Ficando assim:

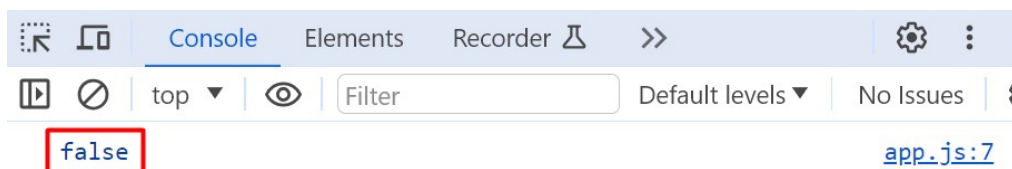
```
console.log(hasDollarSign)
```

Método Includes()



```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
4
5 let hasDollarSign = houseRentValue.includes('%')
6
7 console.log(hasDollarSign)
8
```

E ao salvarmos o arquivo app.js com aquele CTRL + S maroto podemos ver no console do Navegador(Chrome) que false está sendo mostrado.



Igor, Igor, Igor.... E porque false foi mostrado no console do navegador? Por que não temos nenhum caractere '%' dentro da nossa ilustríssima string houseRentValue.

Ah.... até o momento aprendemos como usamos o método includes para fazer aquela verificação marota (toc toc, tem algum caractere aí?) se existe ou não apenas um caractere dentro da string.

Quem lembra o conceito de Strings? Isso mesmo como uma string é um conjunto de caracteres um do lado do outro(sequência), então isso nos permite passarmos seja uma frase completa ou uma única palavra como argumento do includes. Eita, que o negócio tá ficando interessante!!!!

Método Includes()

Antes de vermos esse exemplo, vamos manter a nossa let houseRentValue = 'R\$ 1000 - É pra alugar hoje'. Ficando assim:



```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
4
5
```

E na nossa segunda let, passaremos a string 'vender', como argumento dentro do método includes(). Ficando assim:

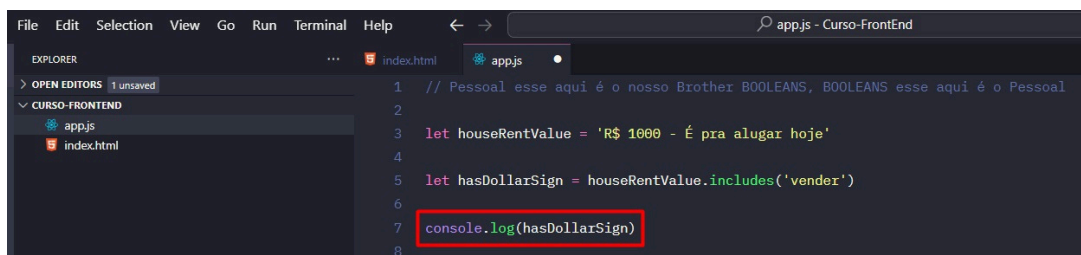
let hasDollarSign = houseRentValue.includes('vender')



```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
4
5 let hasDollarSign = houseRentValue.includes('vender')
6
```

Adicionaremos um console.log passando a hasDollarSign. Ficando assim:

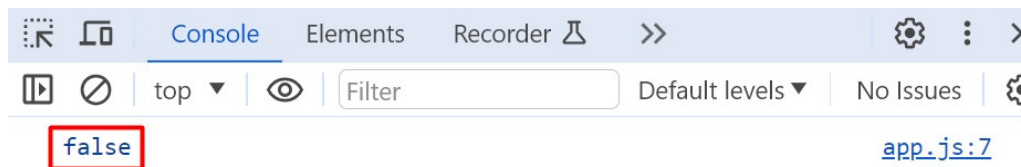
console.log(hasDollarSign)



```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
4
5 let hasDollarSign = houseRentValue.includes('vender')
6
7 console.log(hasDollarSign)
8
```

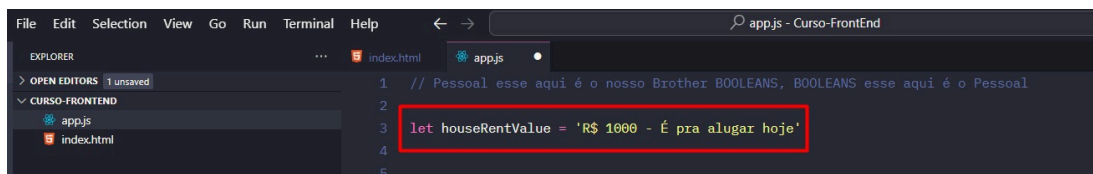
Método Includes()

E ao salvarmos o arquivo app.js com aquele CTRL + S maroto podemos ver no console do Navegador(Chrome) que false está sendo mostrado.



E se quisermos (assim como quem não quer nada) que o valor da string seja retornado como true. Bora ver isso agora...

Antes de vermos esse exemplo, vamos manter a nossa let houseRentValue = 'R\$ 1000 - É pra alugar hoje'. Ficando assim:



E na nossa segunda let, passaremos a string 'alugar', como argumento dentro do método includes(). Ficando assim:

```
let hasDollarSign = houseRentValue.includes('alugar')
```



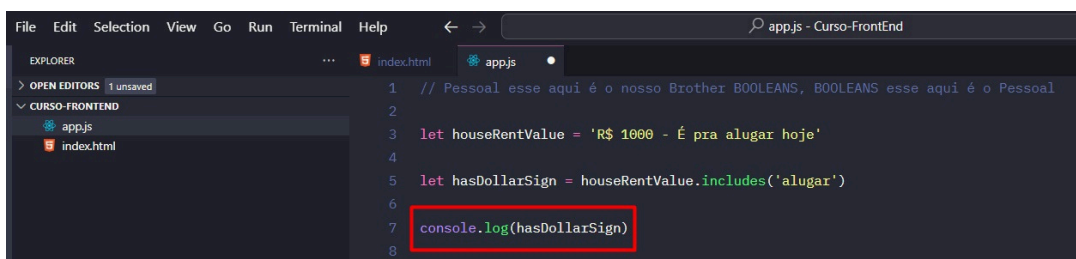
Método Includes()



```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
4
5 let hasDollarSign = houseRentValue.includes('alugar')
6
```

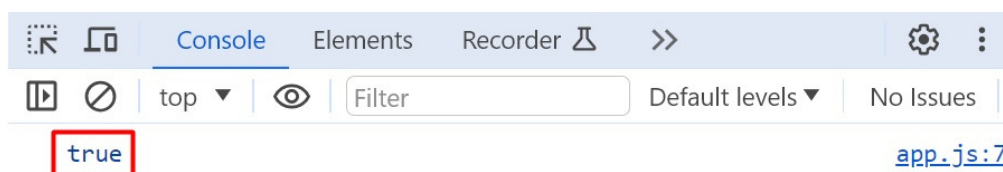
Adicionaremos um `console.log` passando a `hasDollarSign`. Ficando assim:

```
console.log(hasDollarSign)
```



```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let houseRentValue = 'R$ 1000 - É pra alugar hoje'
4
5 let hasDollarSign = houseRentValue.includes('alugar')
6
7 console.log(hasDollarSign)
8
```

E ao salvarmos o arquivo `app.js` com aquele CTRL + S maroto podemos ver no console do Navegador(Chrome) que `true` está sendo mostrado.

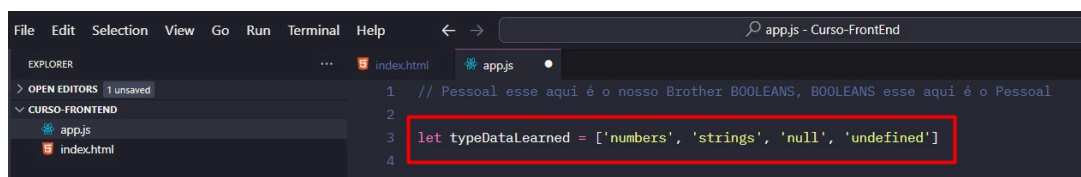


Método Includes()

Agora que já entendemos o conceito de Booleans (false true), vimos exemplos usando o método includes(), tanto para um único caractere quanto para uma string (em ambos os casos retornando tanto false como true).... bora juntar tudo isso, dentro de um array maroto.

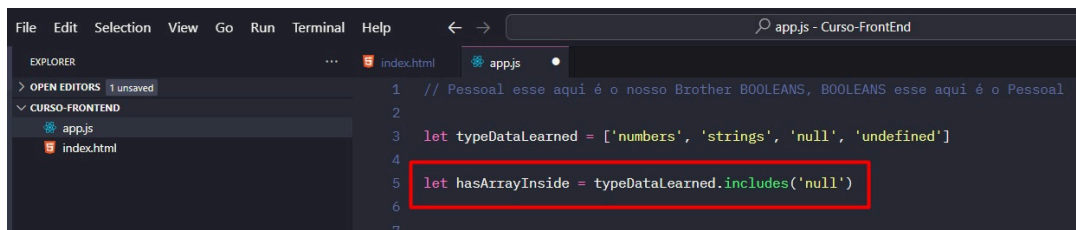
Logo abaixo do nosso comentário // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal, daremos um enter maroto e na linha 3, criaremos uma let typeDataLearned que receberá um array [] com os tipos de dados que já aprendemos até aqui (aproveitamos o exemplo aqui para essa pequena revisão) 'numbers', 'strings', 'null', 'undefined'. Ficando assim:

```
let typeDataLearned = ['numbers', 'strings', 'null', 'undefined']
```

A screenshot of a code editor interface. The Explorer panel on the left shows a project named 'CURSO-FRONTEND' with files 'app.js' and 'index.html'. The main editor area shows a file named 'app.js' with the following content: Line 1: // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal; Line 2: (empty); Line 3: let typeDataLearned = ['numbers', 'strings', 'null', 'undefined']; Line 4: (empty); Line 5: (empty). The line 3 code is highlighted with a red box.

Daremos mais um enter maroto nessa linha 3, para mais uma quebra de linha e na linha 5, declararemos uma nova let hasArrayInside que receberá typeDataLearned.includes() e dentro dos parênteses passaremos a string 'null'. Ficando assim:

Método Includes()

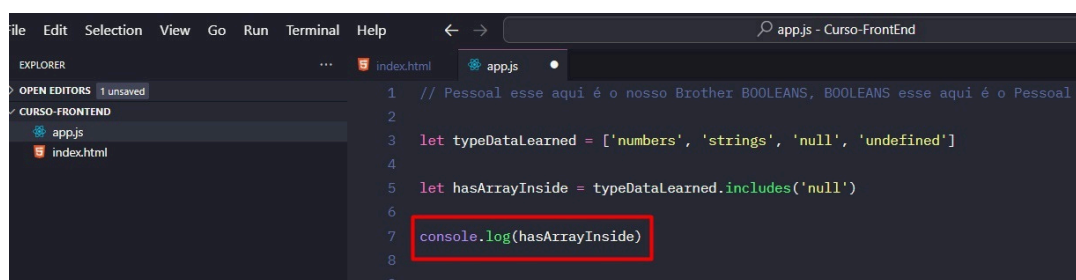


```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let typeDataLearned = ['numbers', 'strings', 'null', 'undefined']
4
5 let hasArrayInside = typeDataLearned.includes('null')
6
7
```

E o nosso nobre método `includes()` verificará se esse valor `'null'` consta dentro do array `typeDataLearned`.

Adicionaremos um `console.log` passando a `hasArrayInside`. Ficando assim:

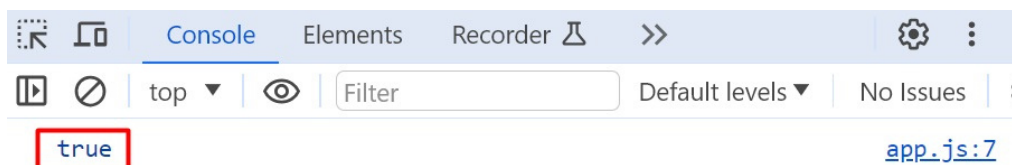
`console.log(hasArrayInside)`



```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let typeDataLearned = ['numbers', 'strings', 'null', 'undefined']
4
5 let hasArrayInside = typeDataLearned.includes('null')
6
7 console.log(hasArrayInside)
8
9
```

E ao salvarmos o arquivo `app.js` com aquele CTRL + S maroto podemos ver no console do Navegador(Chrome) que `true` está sendo mostrado.

Método Includes()

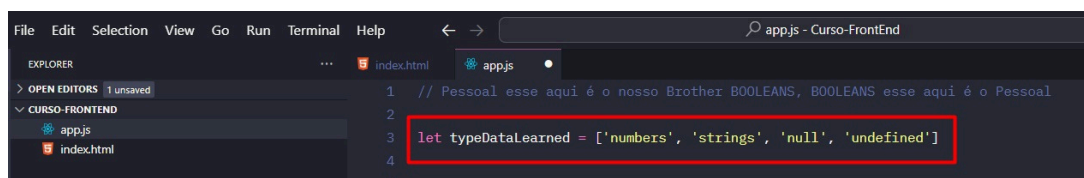


Igor, Igor, Igor..... E por que true apareceu ali?

Porque o item 'null' faz parte ou seja está dentro do nosso array typeDataLearned.

E se (assim como quem não quer nada) quisermos alterar o valor para 'object', por exemplo.

Só que antes de vermos esse exemplo, manteremos a nossa let typeDataLearned = ['numbers', 'strings', 'null', 'undefined'] . Ficando assim:

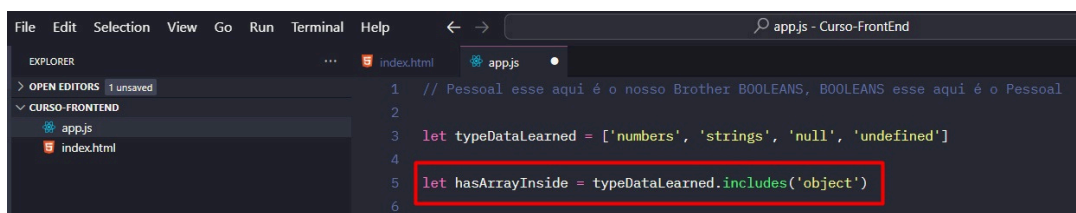


E na nossa segunda let, passaremos a string 'object', como argumento dentro do método includes(). Ficando assim:

```
let hasArrayInside = typeDataLearned.includes('object')
```



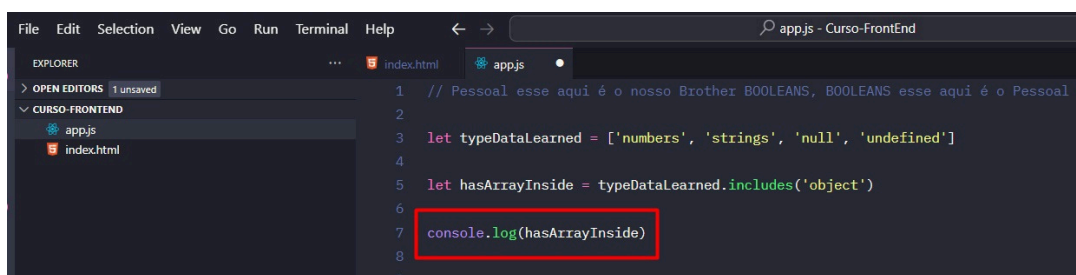
Método Includes()



```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let typeDataLearned = ['numbers', 'strings', 'null', 'undefined']
4
5 let hasArrayInside = typeDataLearned.includes('object')
6
```

Adicionaremos um `console.log` passando a `hasArrayInside`. Ficando assim:

```
console.log(hasArrayInside)
```



```
1 // Pessoal esse aqui é o nosso Brother BOOLEANS, BOOLEANS esse aqui é o Pessoal
2
3 let typeDataLearned = ['numbers', 'strings', 'null', 'undefined']
4
5 let hasArrayInside = typeDataLearned.includes('object')
6
7 console.log(hasArrayInside)
8
```

E ao salvarmos o arquivo `app.js` com aquele `CTRL + S` maroto podemos ver no console do Navegador(Chrome) que `false` está sendo mostrado.



Método Includes()

Igor, Igor, Igor.... E por que isso aconteceu?

Por que o item 'object' não faz parte da turminha ou seja não consta dentro do nosso array `typeDataLearned`.



/igor-rebolla

Sugestões de prática para essa aula:

1. Refazer e Entender os exemplos dessa aula;
2. Usar valores definidos por vocês para praticar mais sobre Booleans
3. Usar valores definidos por vocês para praticar mais sobre o Método Includes
4. Notem que nós exemplos vistos nessa aula, utilizamos o let e se tivessemos usados 2 consts ou 1 let e 1 const os resultados teriam sido(s) os mesmos?
Fica mais essa sugestão para vocês realizarem!!!

Colocar isso em prática no VSCode e ver o resultado no console do Navegador.

Programação é prática, curiosidade e repetição!!!!



O que vamos aprender na aula 24:

Na vigésima quarta aula do curso (Décima aula de JavaScript) o que veremos: É surpresa – Parte 5 (Vem ++++++ coisa boa por ai!!!)

Feedbacks dessa vigésima terceira aula são bem-vindos.

Nos vemos na vigésima terceira aula que será disponibilizada no dia 02/05/2024 às 8h30 (Horário de Brasília) – Dentro do meu perfil no LinkedIn.

