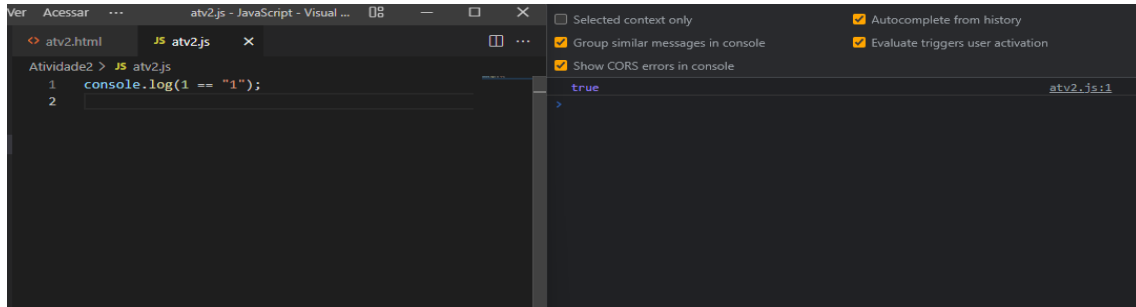


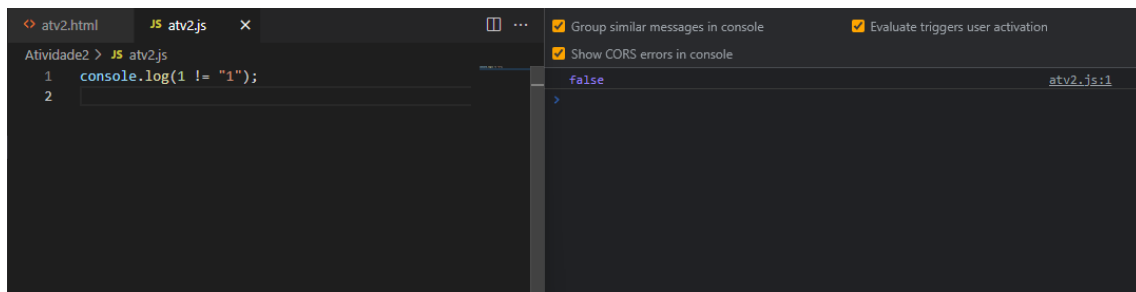
Jean Carlos Nesi

## 2. Qual a diferença entre os operadores == e ===? Quando usar um ou outro?

O == converte as variáveis que estão sendo comparadas para o mesmo tipo. Ou seja, o operador só compara o valor das variáveis e não compara os tipos (int, string, etc);

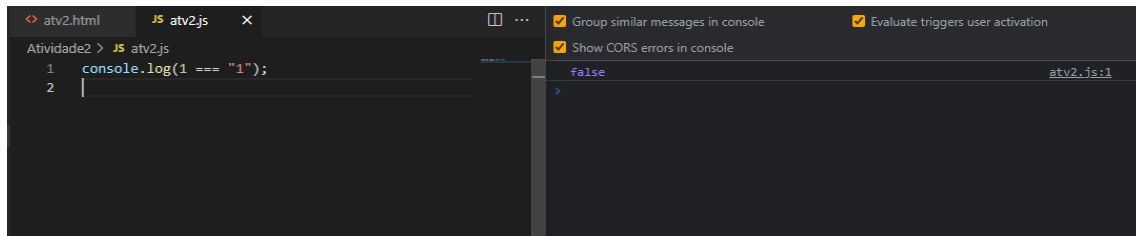


```
Ver  Acessar  ...  atv2.js - JavaScript - Visual ...  [icon]  [x]
atv2.html  JS  atv2.js  x
Atividade2 > JS  atv2.js
1  console.log(1 == "1");
2
true  atv2.js:1
```

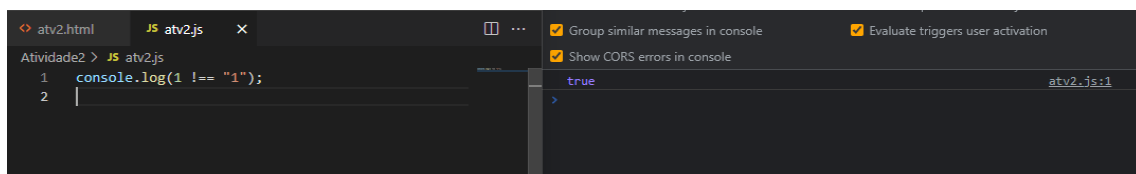


```
atv2.html  JS  atv2.js  x
Atividade2 > JS  atv2.js
1  console.log(1 != "1");
2
false  atv2.js:1
```

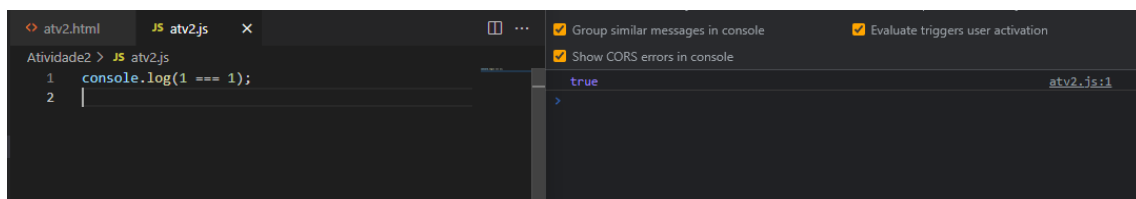
O === compara as os valores e os tipos das variáveis. Sendo assim, o console só retornará **true** quando o valor e o tipo das variáveis forem iguais.



```
atv2.html  JS  atv2.js  x
Atividade2 > JS  atv2.js
1  console.log(1 === "1");
2
false  atv2.js:1
```



```
atv2.html  JS  atv2.js  x
Atividade2 > JS  atv2.js
1  console.log(1 !== "1");
2
true  atv2.js:1
```



```
atv2.html  JS  atv2.js  x
Atividade2 > JS  atv2.js
1  console.log(1 === 1);
2
true  atv2.js:1
```

O ideal seria utilizar o === sempre, pois não teria risco de acontecer nenhum problema de comparação. Mas quando vamos comparar variáveis de mesmo tipo o == pode ser usado, como no **for** por exemplo.

3-

```
testes > JS home.js > ...
1   var num = -1;
2
3   while(num <= 10){
4       console.log(num)
5       num += 0;
6   }
7
8   if(num > 10) {
9       console.log(num);
10  }
11
```

O erro que esse algoritmo possui é um erro de looping infinito.

Como a variável num = -1, o algoritmo sempre irá entrar no while. Dentro do while a variável recebe num = -1 + 0, sendo assim a variável num nunca passara de 10, o que torna um looping infinito.

Se percebe também que na linha 4, falta um ponto e vírgula no final do comando. Porém como não há mais nenhum comando na mesma linha de código, não haverá problema de sintaxe.

4-

```
1   var a = 3;
2
3   var minhaFuncao = function(numero) {
4       numero = 9;
5       return numero;
6   };
7
8   var novoNumero = minhaFuncao(a);
9
10  console.log("novoNumero = " + novoNumero);
11  console.log("a = " + a);
12
```

Análise:

```
1   var a = 3;
2
3   var minhaFuncao = function(numero) {
4       numero = 9;
5       return numero;
6   };
7
8   var novoNumero = minhaFuncao(a); /* Chama a função e armazena o valor retornado na nova variável*/
9
10  console.log("novoNumero = " + novoNumero);
11  /* Console irá exibir 9, porque minhaFuncao sempre retorna numero = 9 e armazena na variável novoNumero*/
12
13  console.log("a = " + a);
14  /* Console irá exibir 3, porque definimos no início que a = 3*/
```

5-

```
1  var a = {
2    |   nome: 'Pedro'
3  };
4
5  var b = {
6    |   nome : a.nome
7  };
8
9  console.info(a === b);
10 console.info(a.nome === b.nome);
11
12 a.nome = 'Maria';
13
14 console.dir(a);
15 console.dir(b);
16
17 console.info(a.nome === b.nome);
```

Análise:

```
1  var a = {
2    |   nome: 'Pedro'
3  };
4
5  var b = {
6    |   nome : a.nome /* O item nome do objeto 'B', recebe o item nome do objeto 'A', que é "Pedro" */
7  };
8
9  console.info(a === b);
10 /*Imprime False*/
11 /*Aqui ele compara os objetos. Os objetos são comparados por referência que checa se o dado objeto é declarado pelo mesmo
12 espaço em memória, e por este motivo eles não são iguais */
13
14 console.info(a.nome === b.nome);
15 /*Imprime True*/
16 /*Aqui ele compara somente o item nome de cada objeto, e como b.nome = a.nome, logo são iguais*/
17
18 a.nome = "Maria"; /* O item nome do objeto 'A' recebe "Maria" */
19
20 console.dir(a); /* Aqui imprime o objeto 'A' = nome: "Maria" */
21
22 console.dir(b); /* Aqui imprime o objeto 'B' = nome: "Pedro" */
23
24 console.info(a.nome === b.nome);
25 /*Imprime False*/
26 /* Compara a.nome("Maria") com b.nome("Pedro"),logo não são iguais*/
```

6-

```
1  var x = 10;
2  var y = 20;
3
4  x += x == 20 ? x/y : y/x;
5  y -= y == 10 ? y/x : x/y;
6
7  console.log(x);
8  console.log(y);
```

Análise:

```
1  var x = 10;
2  var y = 20;
3
4  x += x == 20 ? x/y : y/x;
5  /* x = x + (y/x)
6   |   x = 10 + 20/10
7   |   x = 10 + 2
8   |   x = 12          */
9
10 y -= y == 10 ? y/x : x/y;
11 /* y = y - (x/y)
12 |   y = 20 - 12/20
13 |   y = 20 - 0.6
14 |   y = 19.4        */
15
16 console.log(x); /* Imprime X = 12*/
17 console.log(y); /* Imprime Y = 19.6*/
```