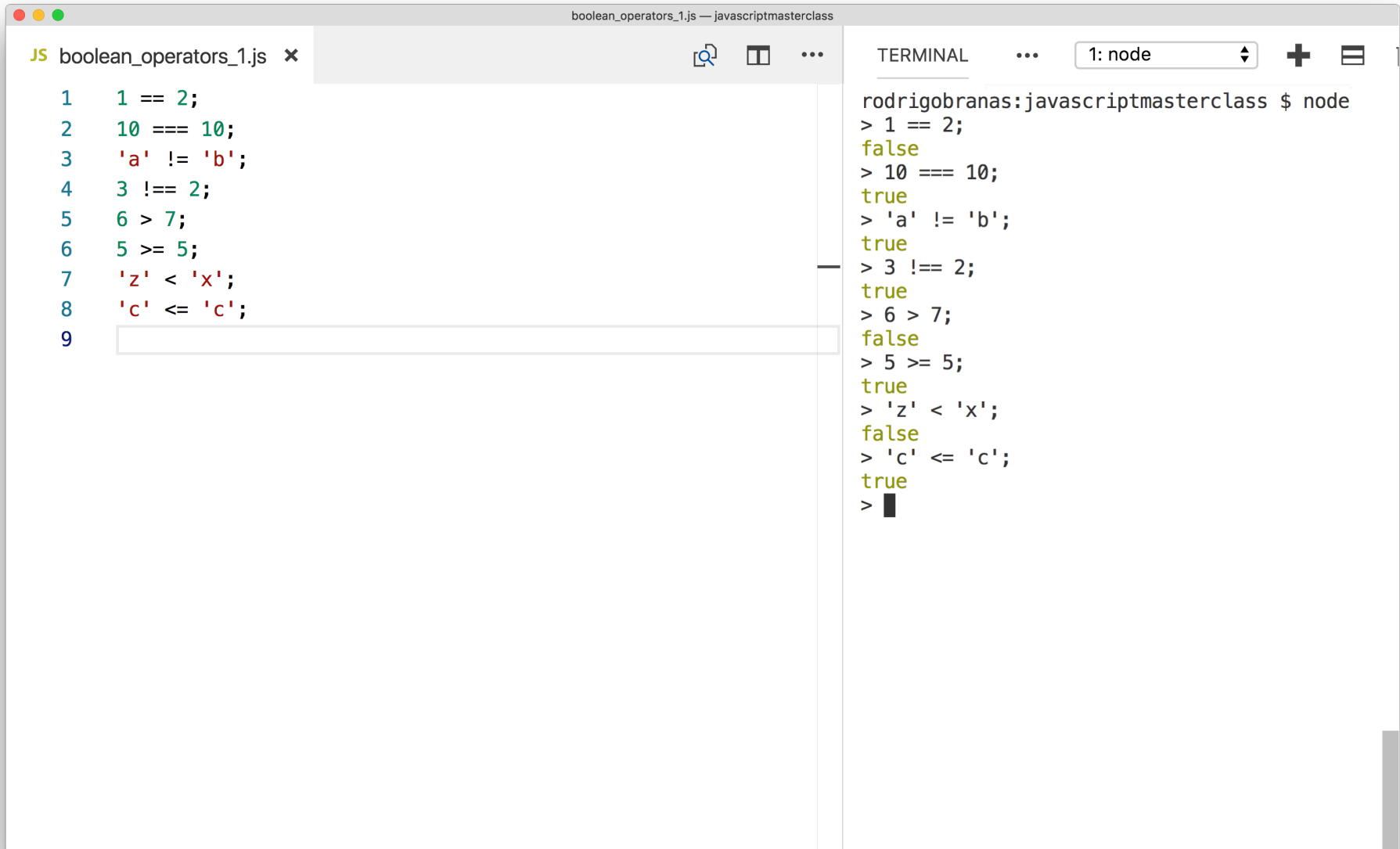


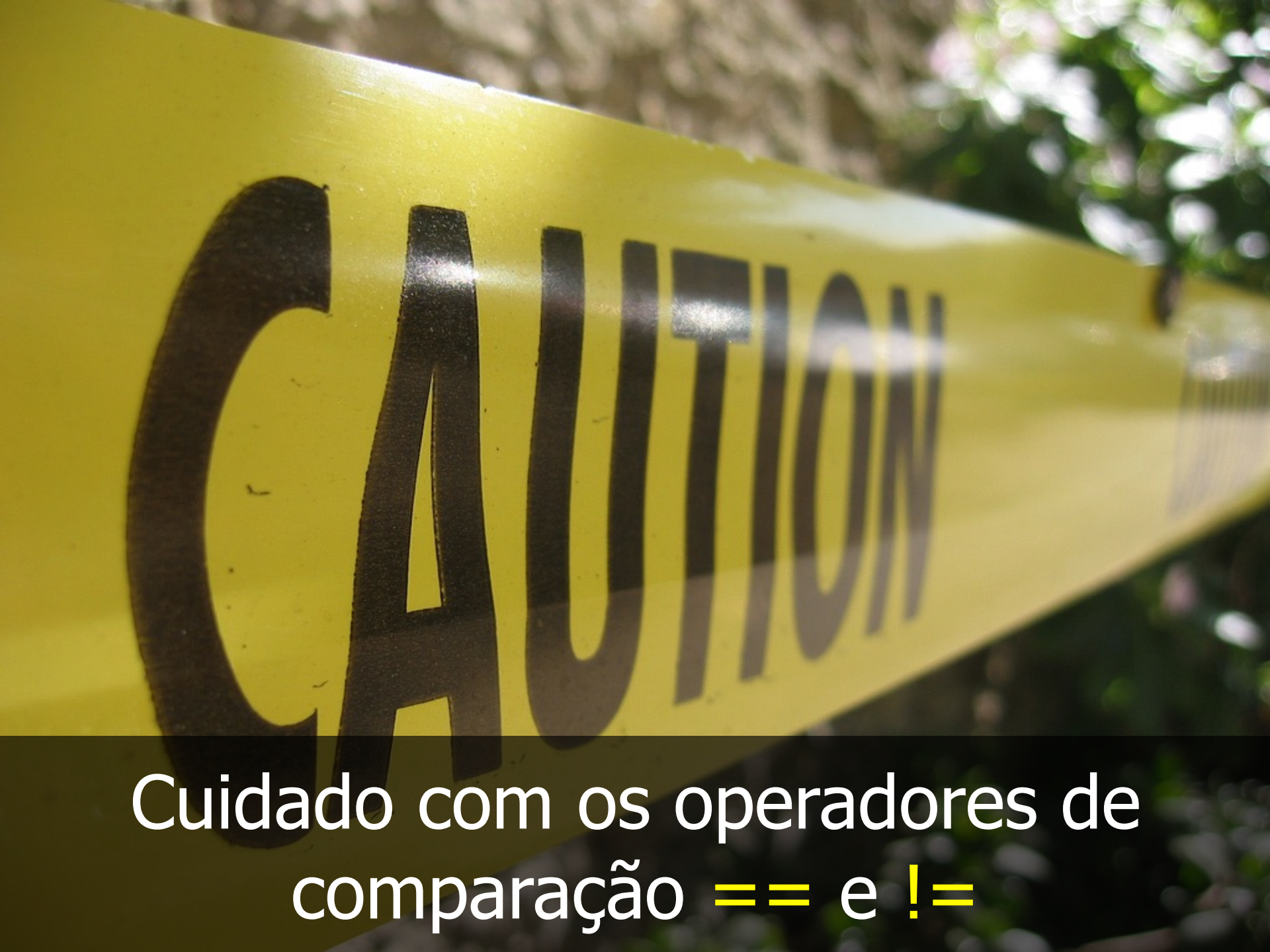


Operadores Booleanos

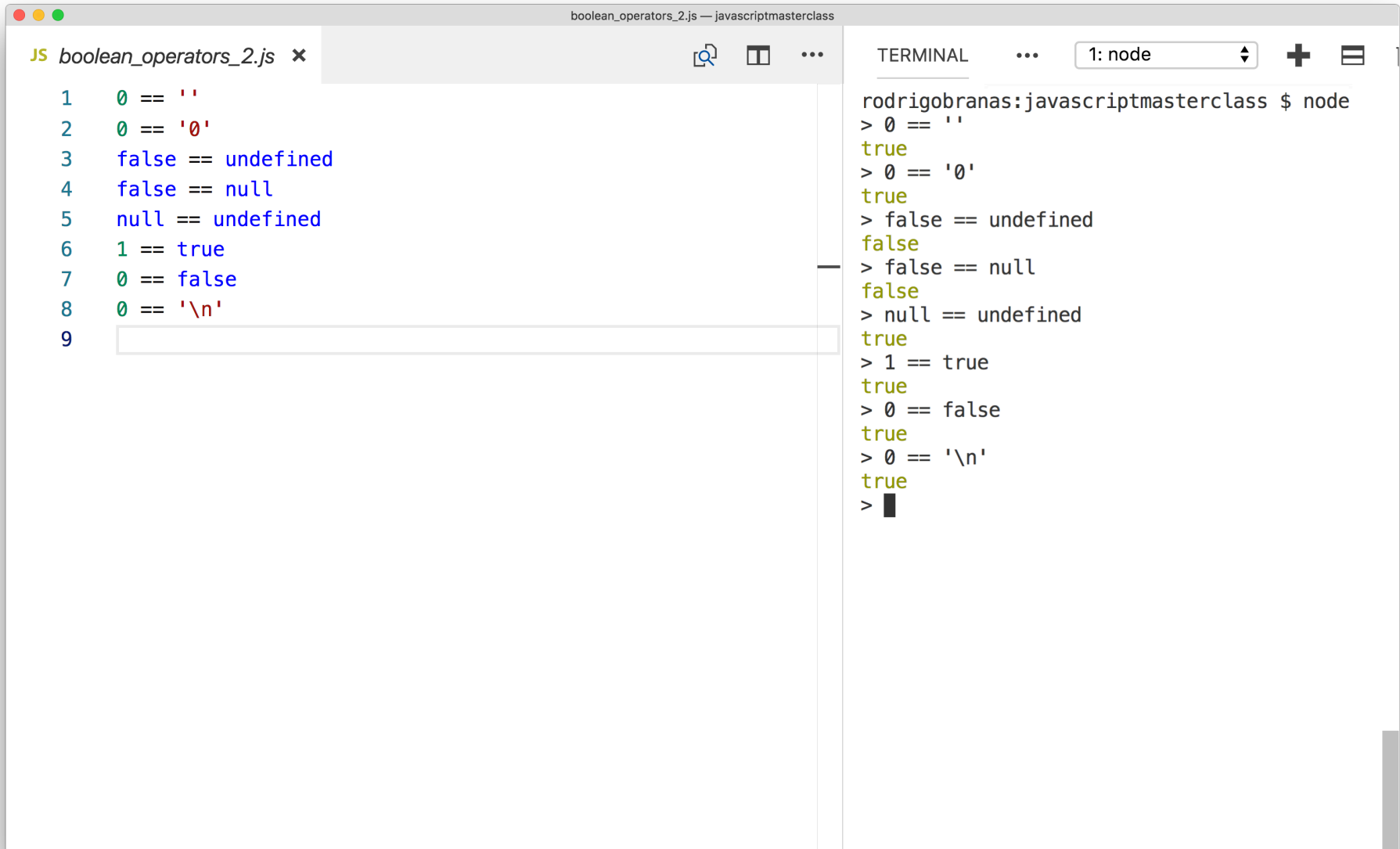
Operadores de comparação

==, **===**, **!=**, **!==**, **<**, **>**, **<=** e **>=**

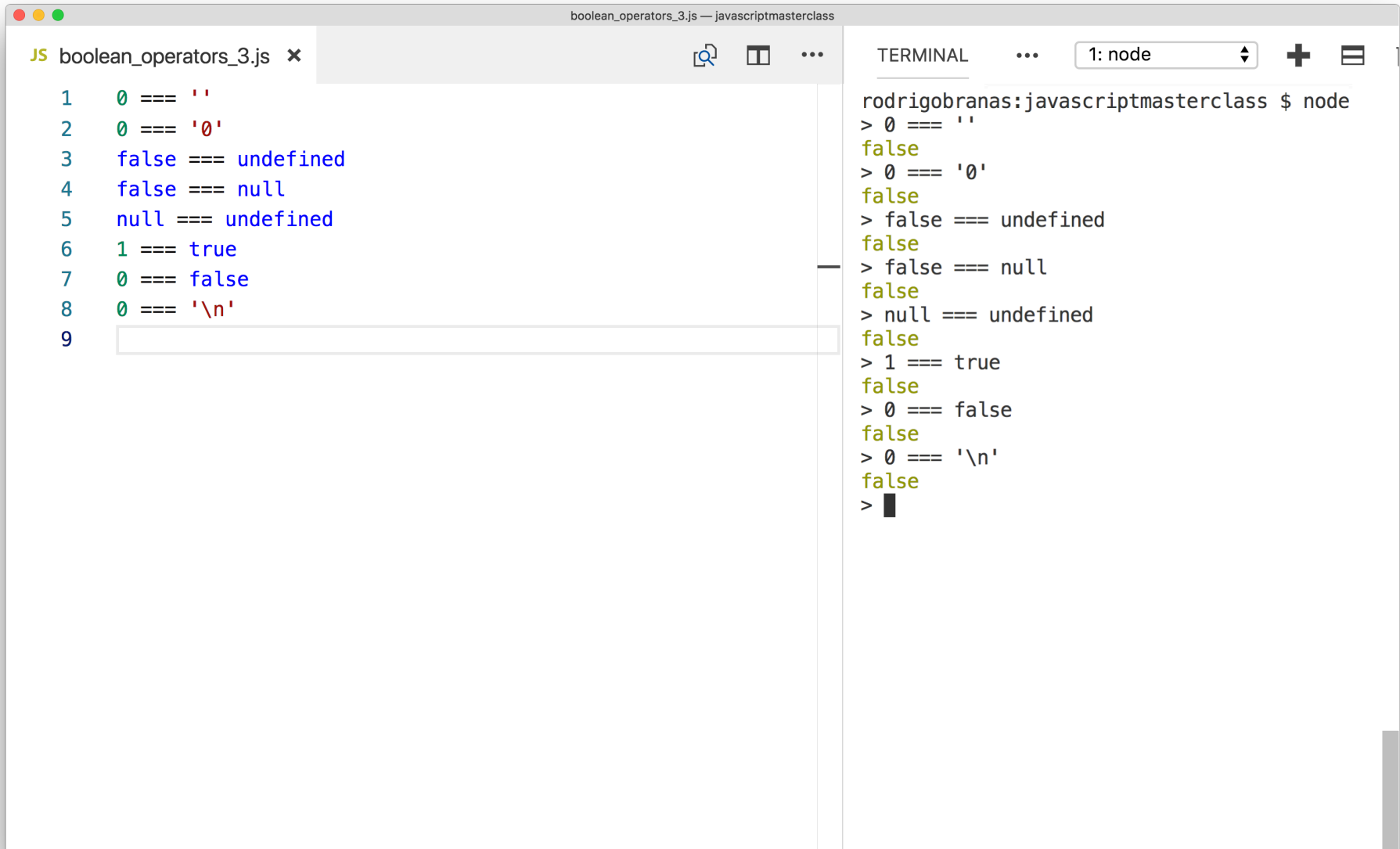




Cuidado com os operadores de
comparação **==** e **!=**

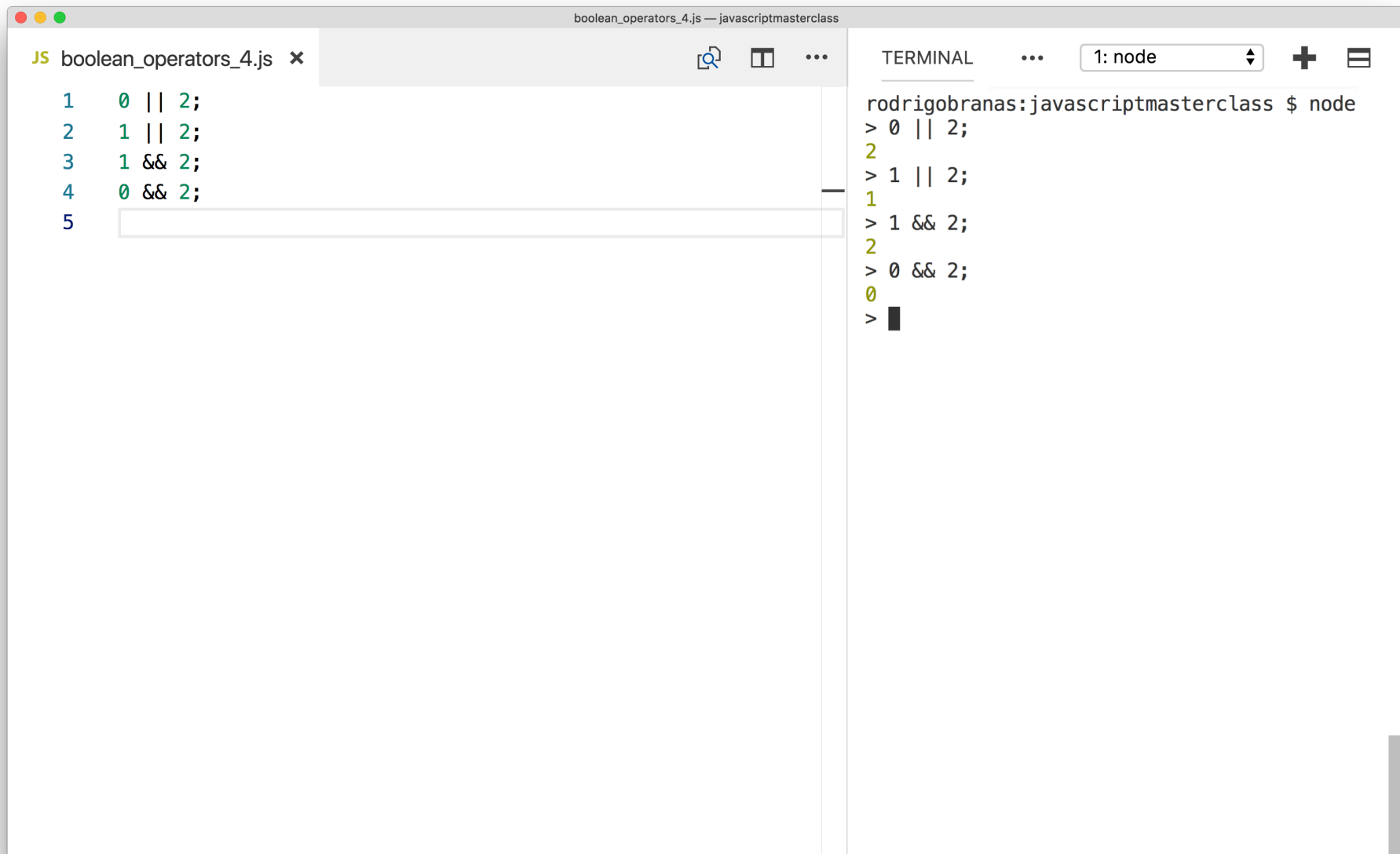


Prefira sempre a utilização dos
operadores **===** e **!==**



Operadores lógicos

`||` e `&&`



boolean_operators_5.js — javascriptmasterclass

JS boolean_operators_5.js

1 function generateSerial (max) {

2 | return Math.floor(Math.random() * max);

3 }

4

5 generateSerial(10);

6 generateSerial(100);

7 generateSerial(1000);

8 generateSerial();

9

TERMINAL

1: node

rodrigo-branas:javascriptmasterclass \$ node

> function generateSerial (max) {

... return Math.floor(Math.random() * ma

x);

... }

undefined

>

> generateSerial(10);

4

> generateSerial(100);

79

> generateSerial(1000);

712

> generateSerial();

NaN

>

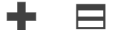
JS boolean_operators_6.js x



TERMINAL



1: node



```
1  function generateSerial (max) {
2    if (max === undefined) {
3      max = 1000;
4    }
5    return Math.floor(Math.random() * max);
6  }
7
8  generateSerial(10);
9  generateSerial(100);
10 generateSerial(1000);
11 generateSerial();
12
```

```
rodrigobranas:javascriptmasterclass $ node
> function generateSerial (max) {
...   if (max === undefined) {
...     max = 1000;
...   }
...   return Math.floor(Math.random() * ma
x);
... }
undefined
>
> generateSerial(10);
9
> generateSerial(100);
30
> generateSerial(1000);
151
> generateSerial();
852
>
```

JS boolean_operators_7.js ✕



TERMINAL



1: node



```
1  function generateSerial (max) {
2    if (!max) {
3      max = 1000;
4    }
5    return Math.floor(Math.random() * max);
6  }
7
8  generateSerial(10);
9  generateSerial(100);
10 generateSerial(1000);
11 generateSerial();
12
```

```
rodrigobranas:javascriptmasterclass $ node
> function generateSerial (max) {
...   if (!max) {
...     max = 1000;
...   }
...   return Math.floor(Math.random() * ma
x);
... }
undefined
>
> generateSerial(10);
2
> generateSerial(100);
4
> generateSerial(1000);
61
> generateSerial();
107
>
```

JS boolean_operators_8.js x



TERMINAL

...

1: node



```
1  function generateSerial (max) {
2    max = max || 1000;
3    return Math.floor(Math.random() * max);
4  }
5
6  generateSerial(10);
7  generateSerial(100);
8  generateSerial(1000);
9  generateSerial();
10
```

```
rodrigobranas:javascriptmasterclass $ node
> function generateSerial (max) {
...   max = max || 1000;
...   return Math.floor(Math.random() * ma
x);
... }
undefined
>
> generateSerial(10);
8
> generateSerial(100);
88
> generateSerial(1000);
112
> generateSerial();
879
>
```

Operador ternário
(expressão) ? true : false

