



String API

- **length**: Retorna o tamanho da String
- **indexOf**: Retorna a primeira posição encontrada do caractere passado por parâmetro
- **lastIndexOf**: Retorna a última posição encontrada do caractere passado por parâmetro
- **toUpperCase**: Retorna uma nova String convertendo as letras para maiúsculas
- **toLowerCase**: Retorna uma nova String convertendo as letras para minúsculas

JS string_api_1.js x



TERMINAL



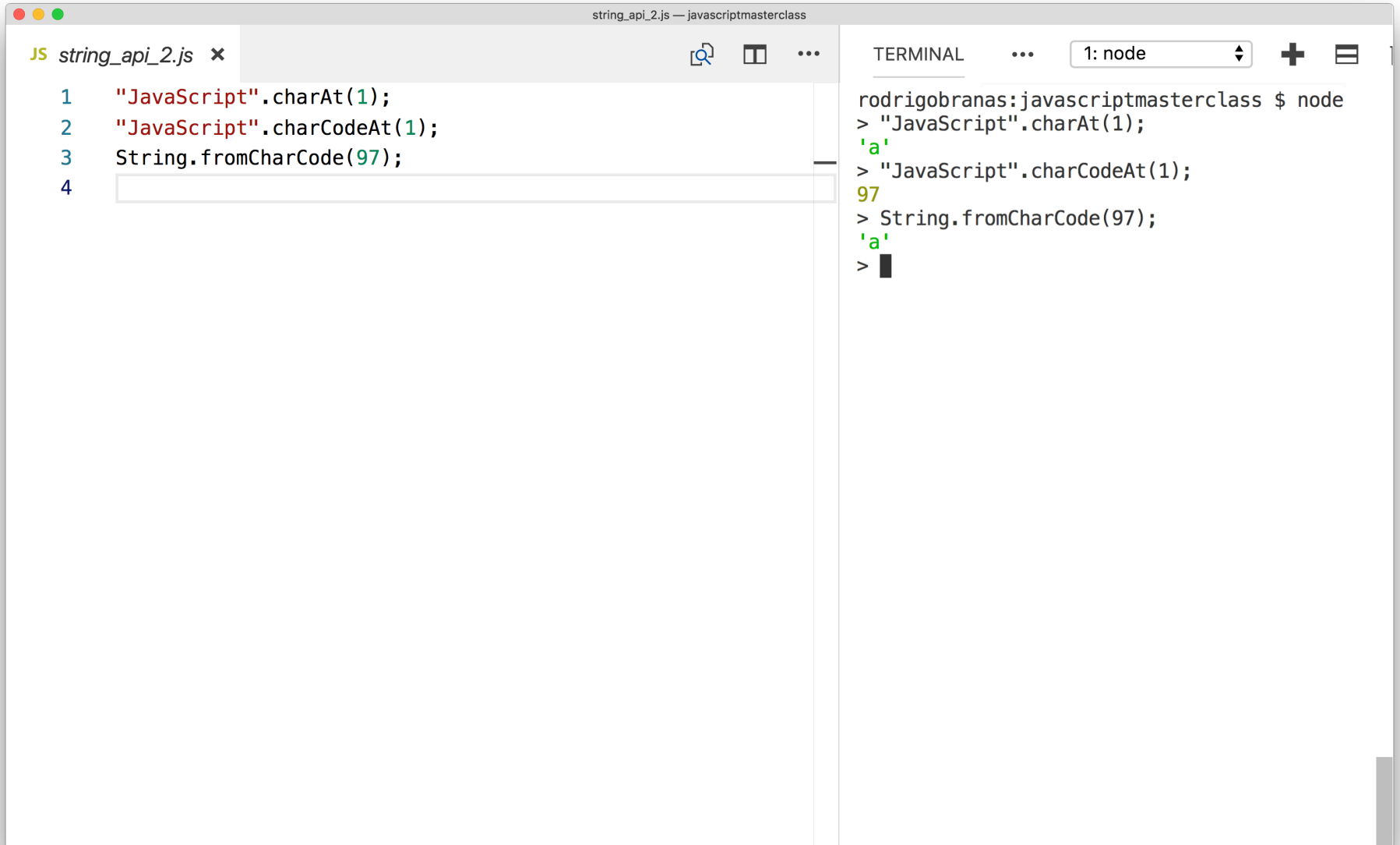
1: node



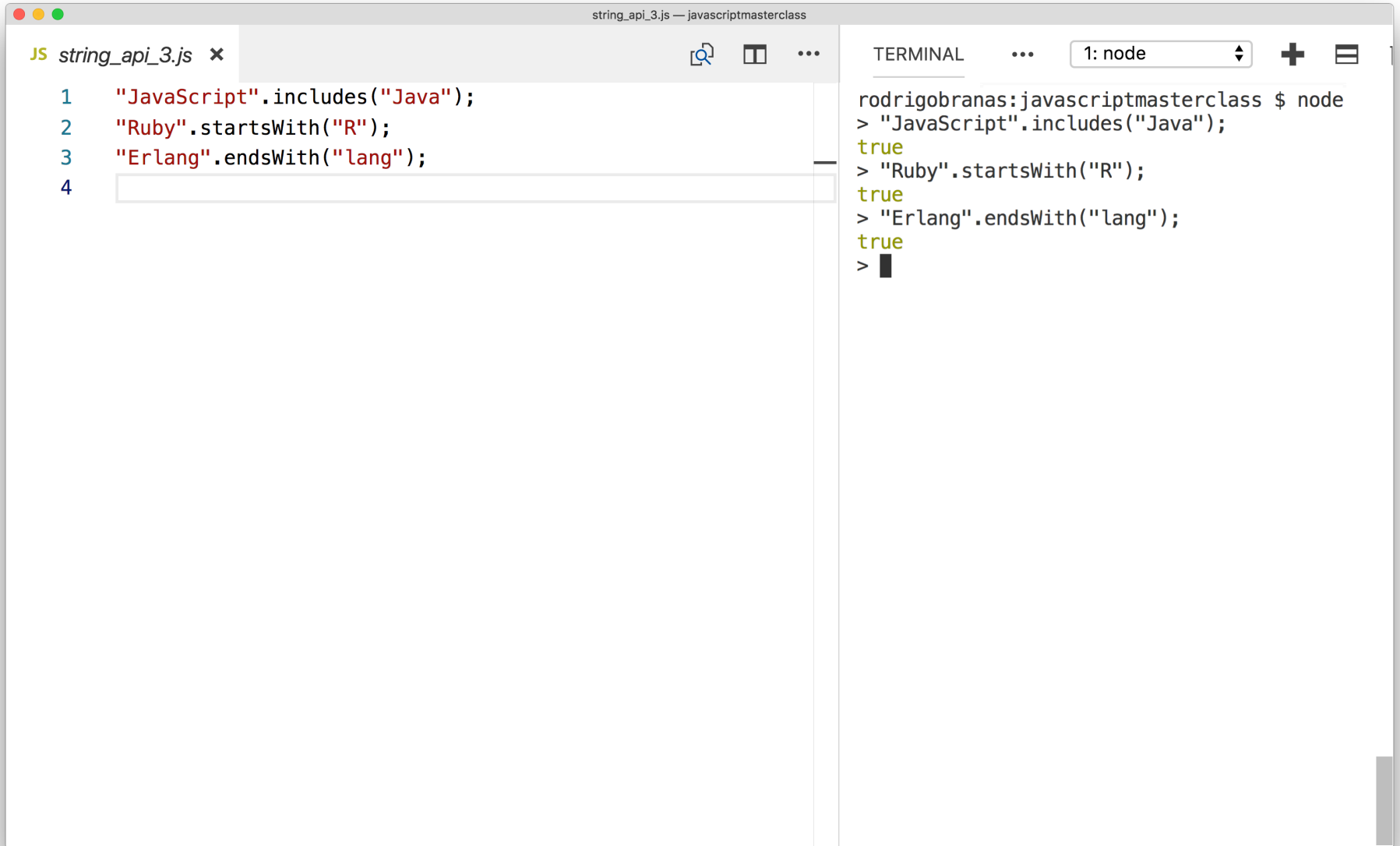
```
1  "JavaScript".length;  
2  "PHP".indexOf("P");  
3  "PHP".lastIndexOf("P");  
4  "cobol".toUpperCase();  
5  "ALGOL".toLowerCase();  
6
```

```
rodrigobranas:javascriptmasterclass $ node  
> "JavaScript".length;  
10  
> "PHP".indexOf("P");  
0  
> "PHP".lastIndexOf("P");  
2  
> "cobol".toUpperCase();  
'COBOL'  
> "ALGOL".toLowerCase();  
'algol'  
>
```

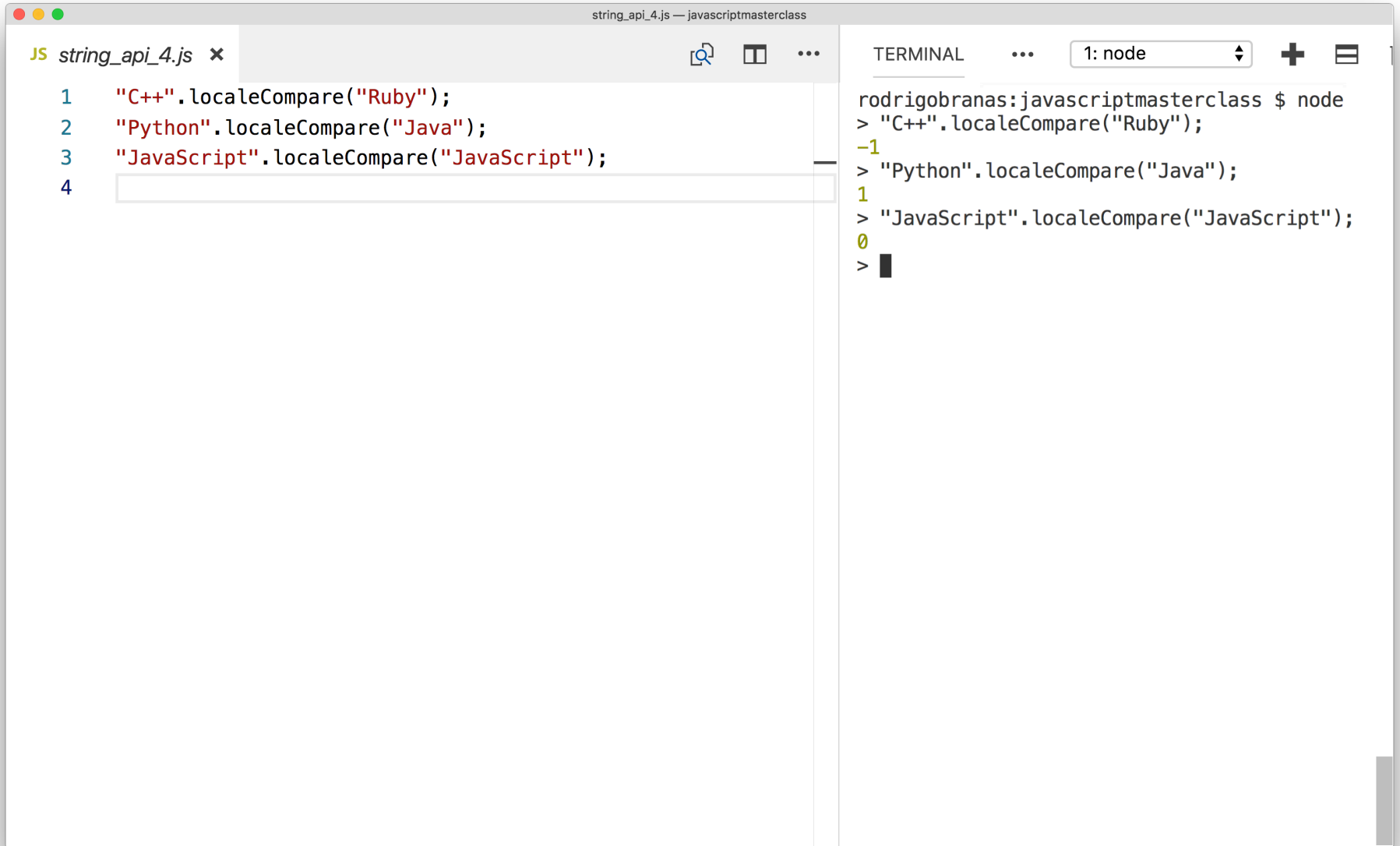
- **charAt**: Retorna o caractere na posição passada por parâmetro
- **charCodeAt**: Retorna o código com base na posição passada por parâmetro
- **fromCharCode**: Retorna um caractere com base no código passado por parâmetro



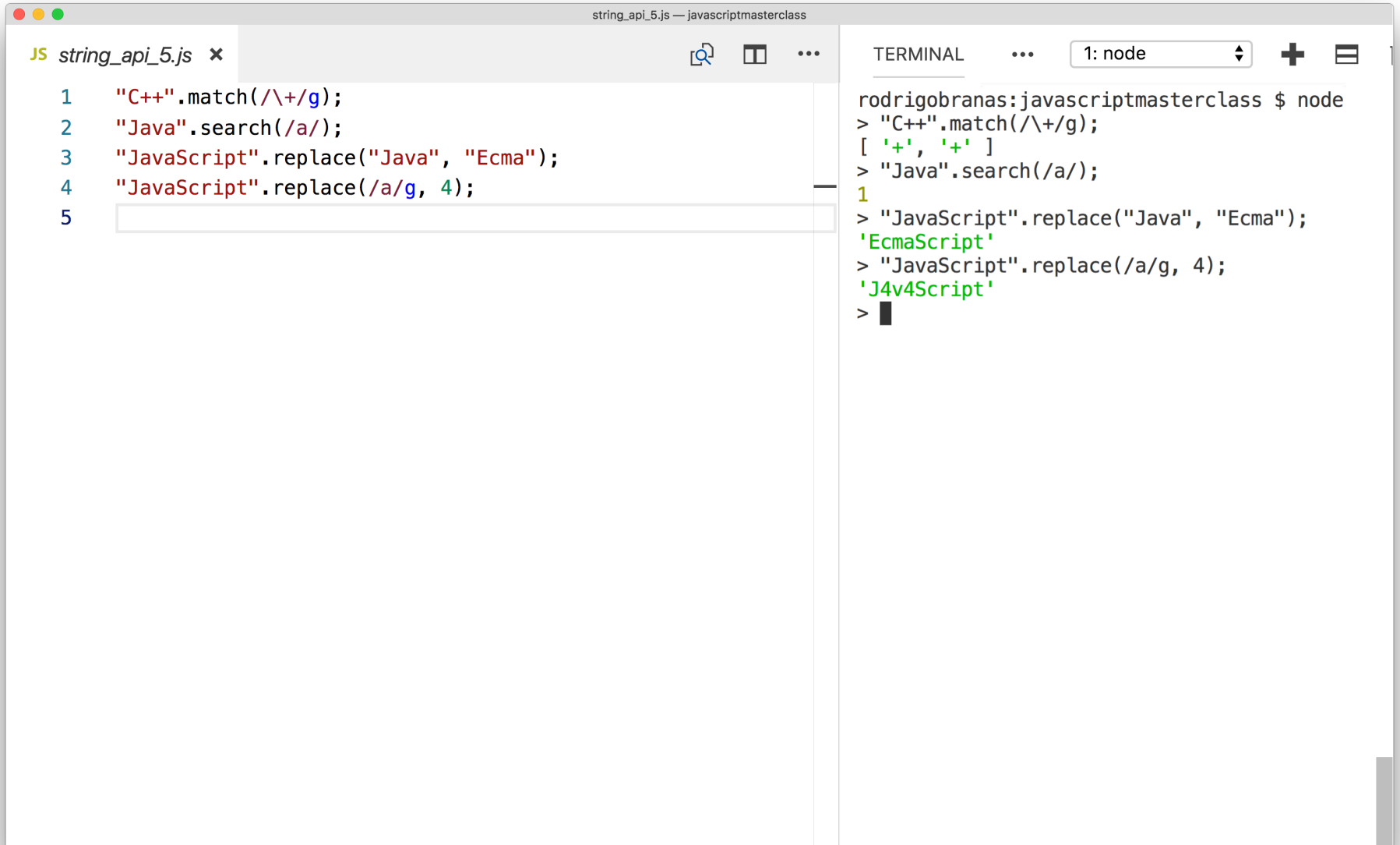
- **includes**: Retorna verdadeiro se a String contém a String passada por parâmetro
- **startsWith**: Retorna verdadeiro se a String inicia com a String passada por parâmetro
- **endsWith**: Retorna verdadeiro se a String termina com a String passada por parâmetro



- **localeCompare**: Retorna -1 se a String passada por parâmetro for maior, 0 se for igual e 1 se for menor



- **match**: Retorna partes da String com base na RegExp passada por parâmetro
- **search**: Retorna a primeira posição encontrada com base na RegExp passada por parâmetro
- **replace**: Retorna uma nova String resultante da substituição da String ou RegExp passada no primeiro parâmetro pelo segundo parâmetro



- **slice**: Retorna uma parte da String que está invocando a função iniciando na posição passada no primeiro parâmetro até a posição final passada no segundo parâmetro, ou da posição passada no primeiro parâmetro até o fim caso o segundo parâmetro não seja informado
- **split**: Retorna um array contendo o resultado da divisão da String original de acordo com o critério passado por parâmetro
- **substring**: Similar ao slice, não aceita valores negativos como parâmetro e permite a inversão dos parâmetros

JS string_api_6.js x



TERMINAL



1: node



```
1  "JavaScript".slice(0, 4);
2  "JavaScript".slice(4);
3  "JavaScript".slice(0, -6);
4  "JavaScript".slice(-6);
5  "C;Java;JavaScript;Ruby".split(";");
6  "JavaScript".substring(0, 4);
7  "JavaScript".substring(4, 0);
8  "JavaScript".substring(4);
9
```

```
rodrigobranas:javascriptmasterclass $ node
> "JavaScript".slice(0, 4);
'Java'
> "JavaScript".slice(4);
'Script'
> "JavaScript".slice(0, -6);
'Java'
> "JavaScript".slice(-6);
'Script'
> "C;Java;JavaScript;Ruby".split(";");
[ 'C', 'Java', 'JavaScript', 'Ruby' ]
> "JavaScript".substring(0, 4);
'Java'
> "JavaScript".substring(4, 0);
'Java'
> "JavaScript".substring(4);
'Script'
>
```

- **concat**: Retorna uma nova String resultante da concatenação da que está invocando a função e da outra, passada por parâmetro
- **padStart**: Completa a String com caracteres no início
- **padEnd**: Completa a String com caracteres no fim
- **repeat**: Repete um caractere
- **trim**: Elimina espaços em branco no início e no fim
- **trimLeft**: Elimina espaços em branco no início
- **trimRight**: Elimina espaços em branco no fim

JS string_api_7.js x



TERMINAL



1: node



```
1  "Java".concat("Script");
2  "Data".concat("Flex");
3  "Script".padStart(10, "Java");
4  "C".padEnd(3, "+");
5  "C".concat("+".repeat(2));
6  " Self ".trim();
7  " Scheme ".trimLeft();
8  " Perl ".trimRight();
9
```

```
rodrigobranas:javascriptmasterclass $ node
> "Java".concat("Script");
'JavaScript'
> "Data".concat("Flex");
'DataFlex'
> "Script".padStart(10, "Java");
'JavaScript'
> "C".padEnd(3, "+");
'C++'
> "C".concat("+".repeat(2));
'C++'
> " Self ".trim();
'Self'
> " Scheme ".trimLeft();
'Scheme '
> " Perl ".trimRight();
' Perl'
>
```