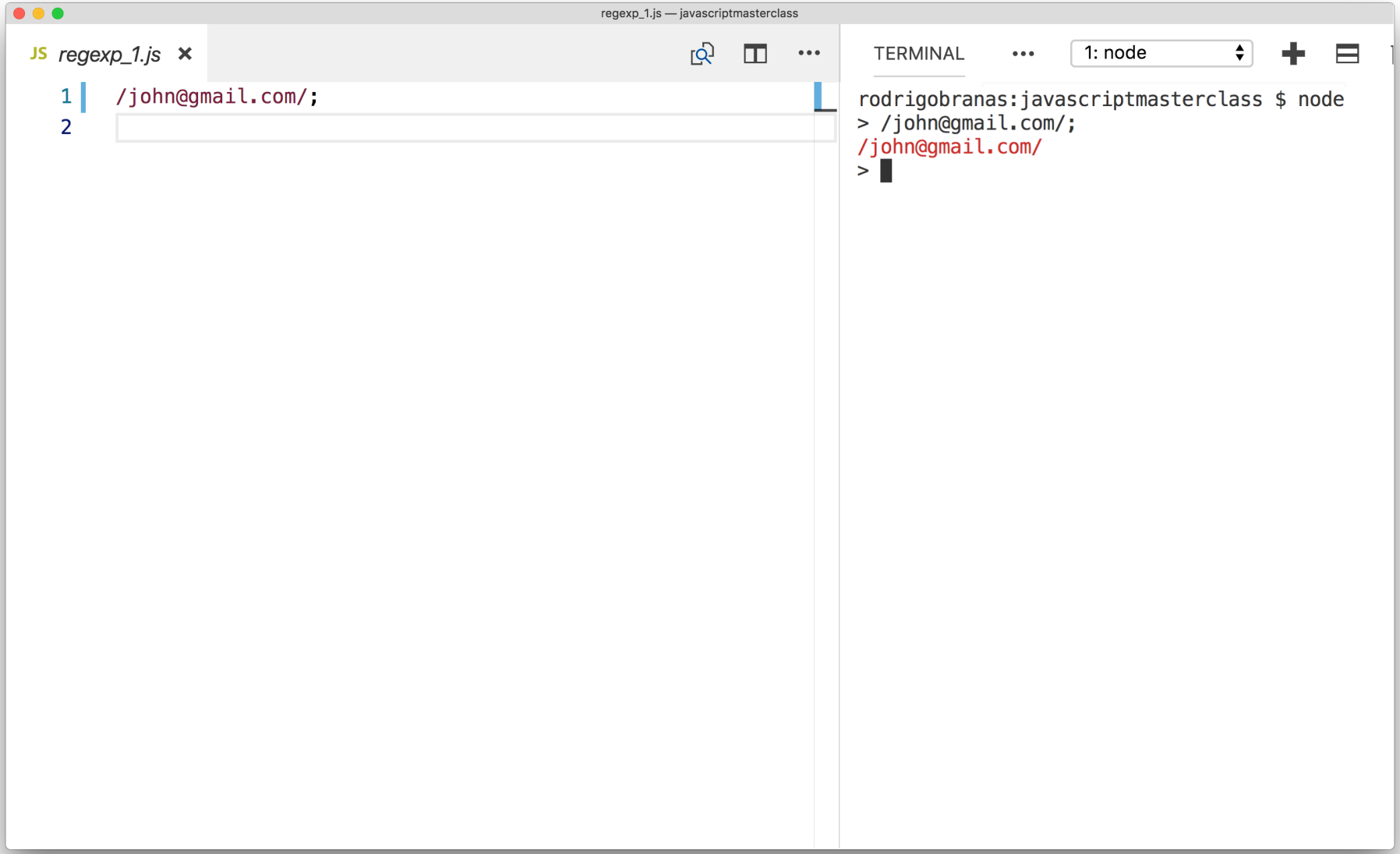
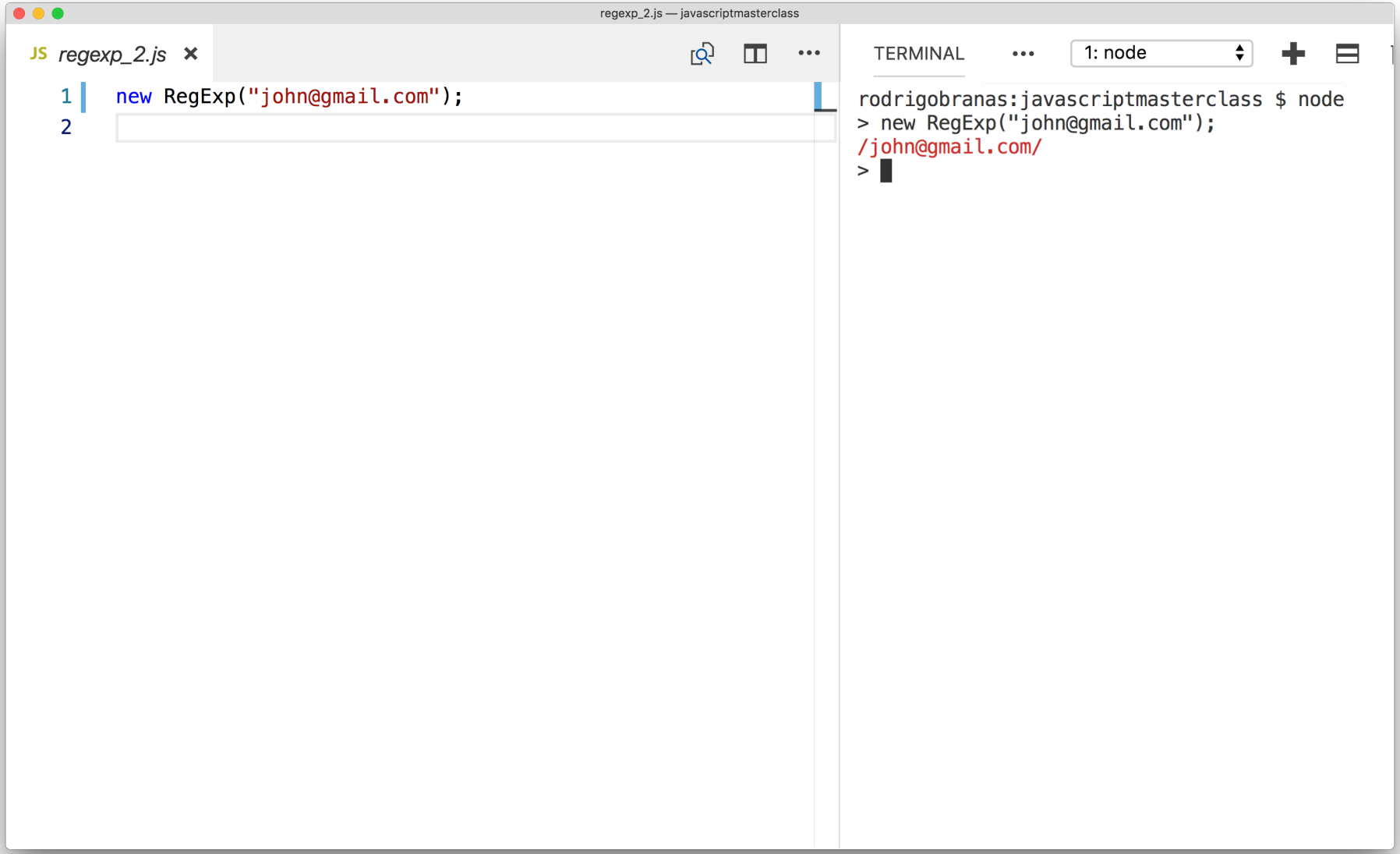
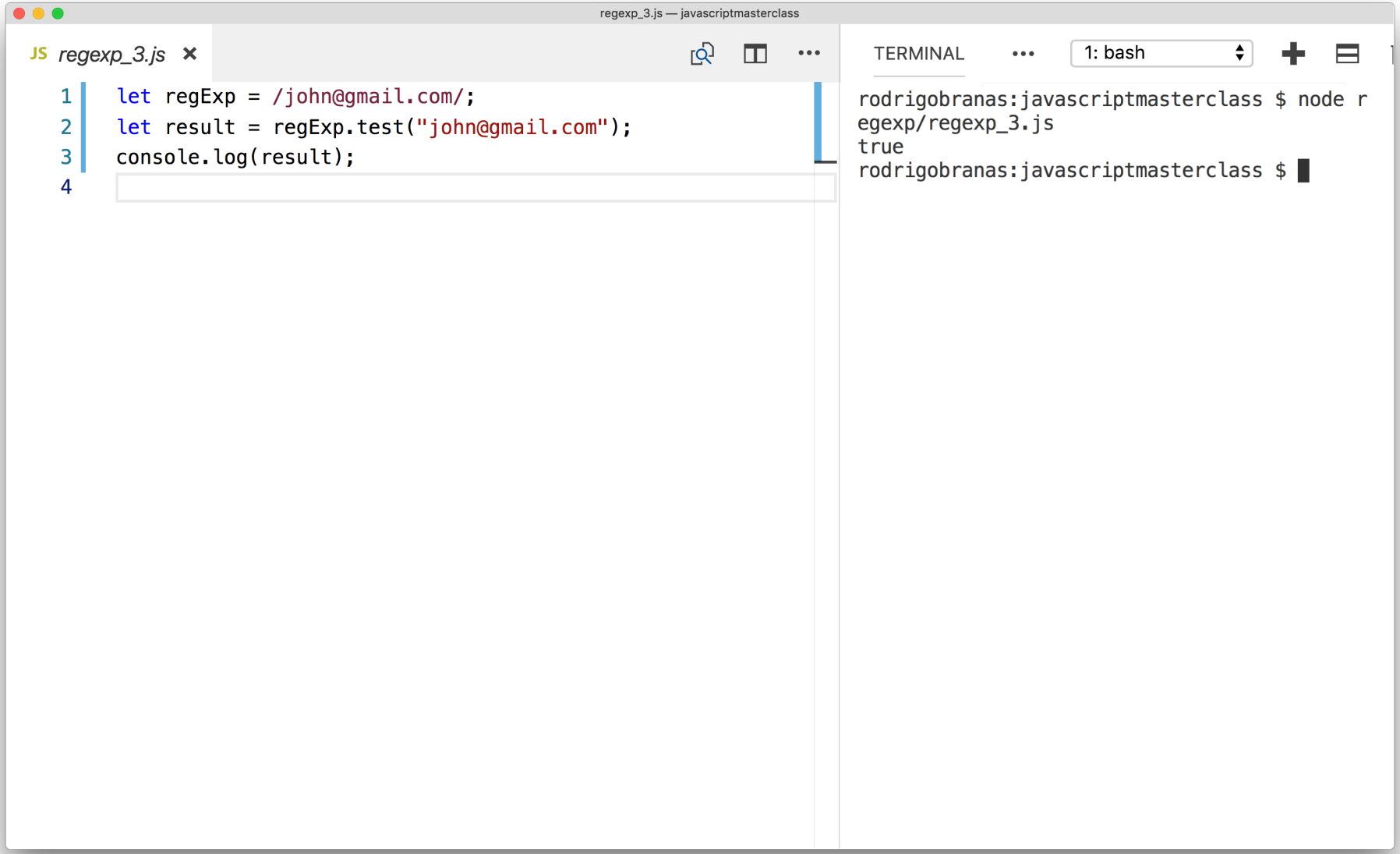
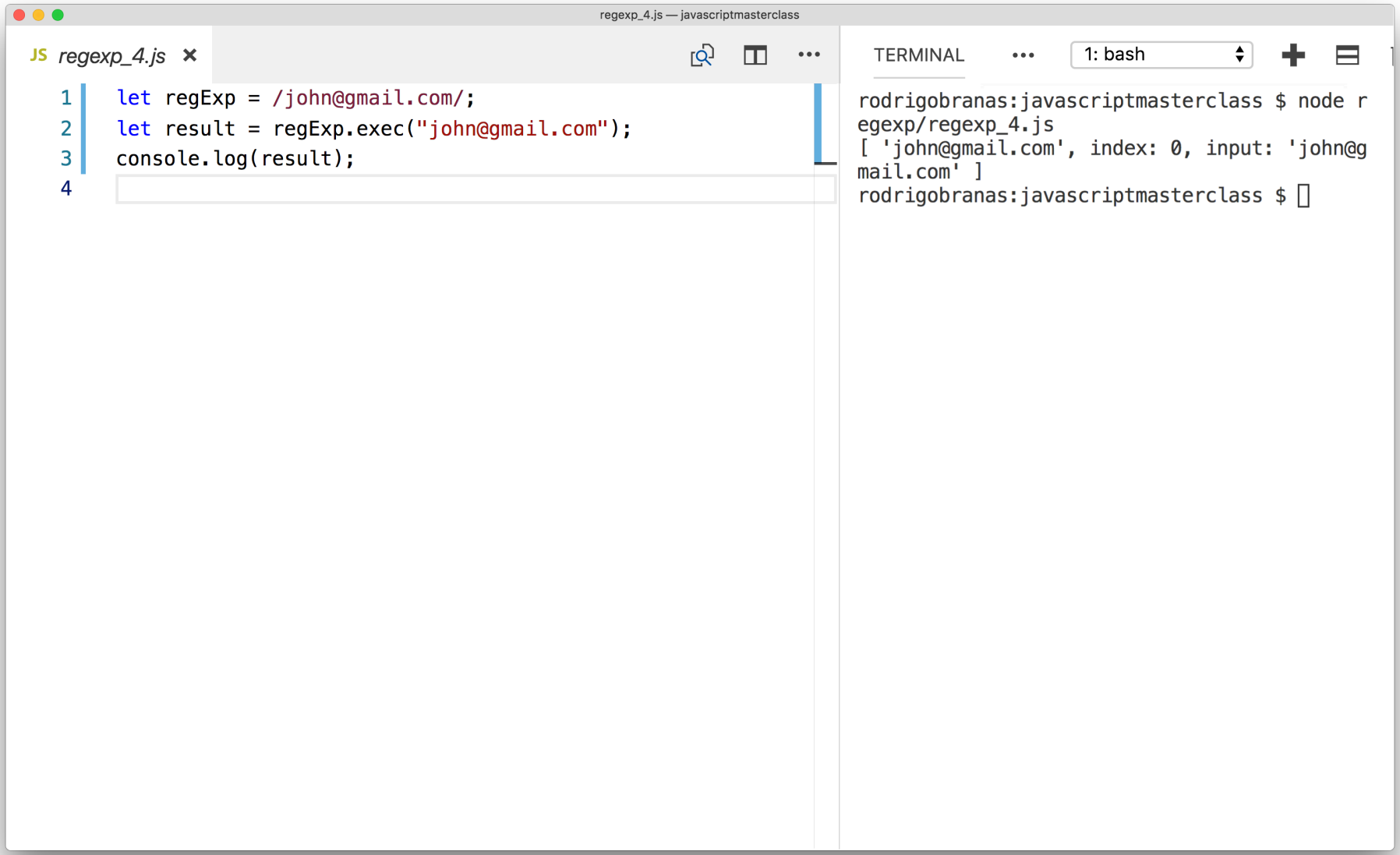
Expressões Regulares

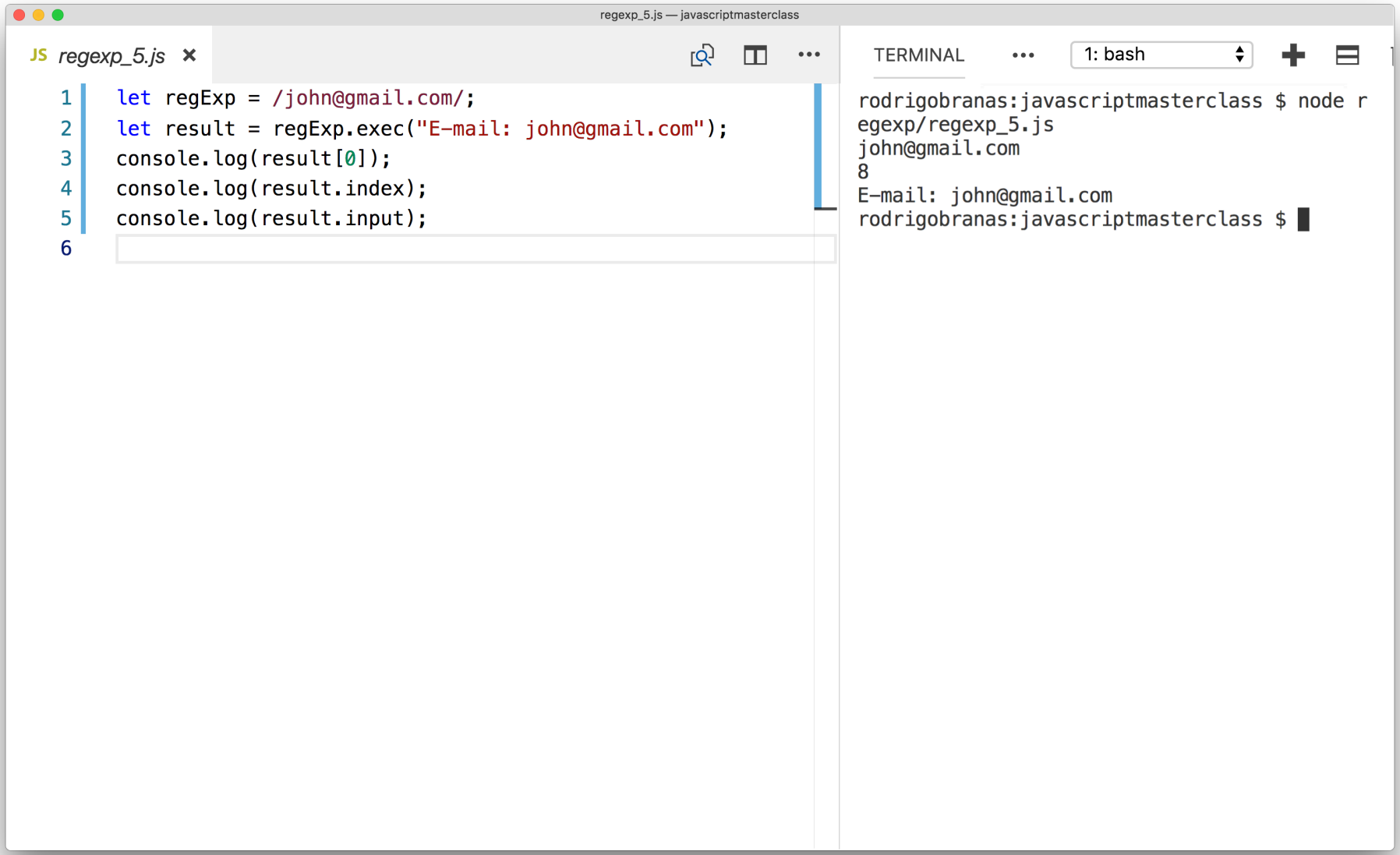
As expressões regulares são estruturas formadas por uma sequência de caracteres que especificam um padrão formal que servem para validar, extrair ou mesmo substituir caracteres dentro de uma String


regexp_1.js — javascriptmasterclass
JS regexp_1.js
1 /john@gmail.com/;
2
TERMINAL
1: node
rodrigobranas:javascriptmasterclass $ node
> /john@gmail.com/;
/john@gmail.com/
>


regexp_2.js — javascriptmasterclass
JS regexp_2.js
1 new RegExp("john@gmail.com");
2
TERMINAL
1: node
rodrigobranas:javascriptmasterclass $ node
> new RegExp("john@gmail.com");
/john@gmail.com/
>

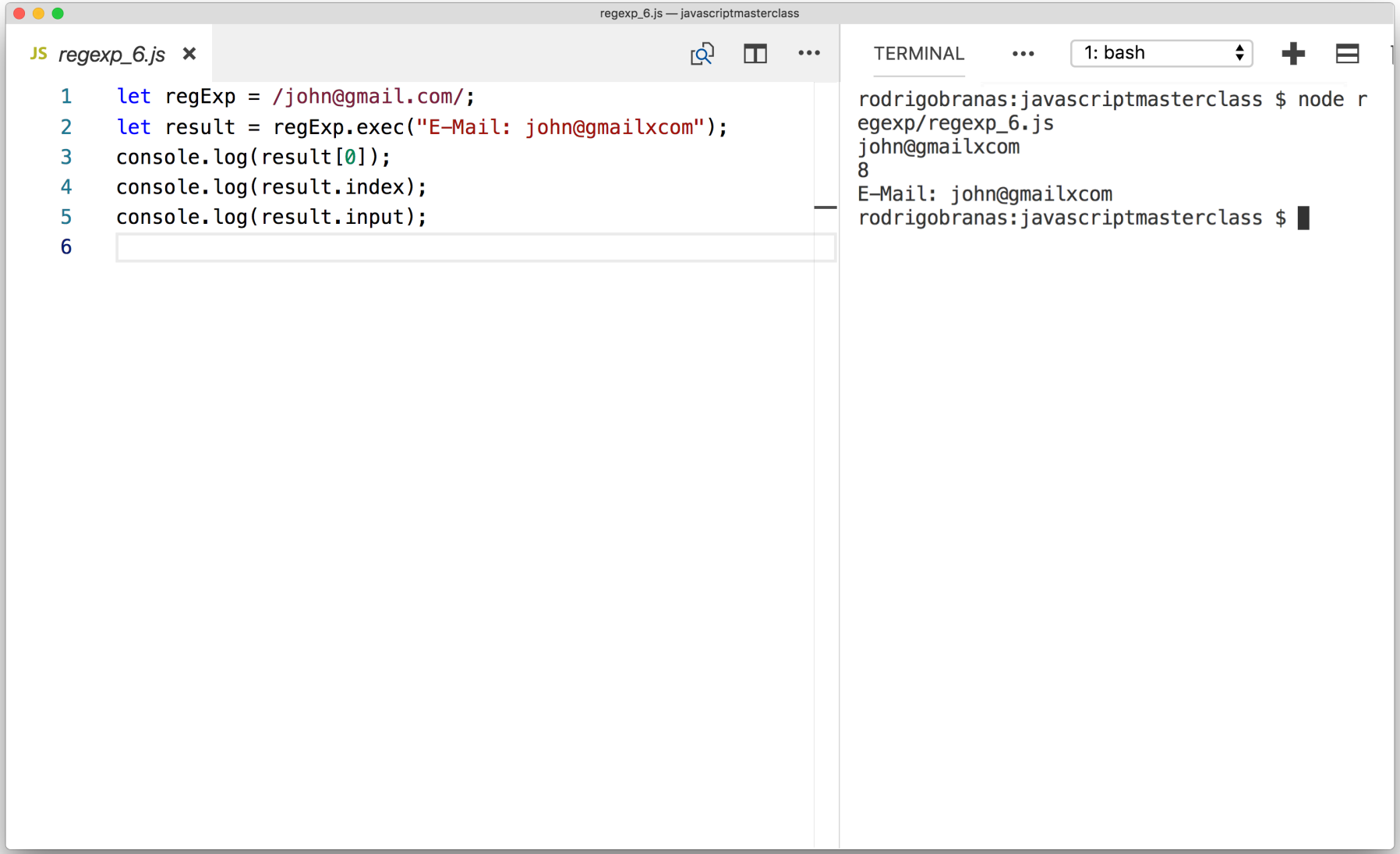

regexp_3.js — javascriptmasterclass
JS regexp_3.js
let regExp = /john@gmail.com/;
let result = regExp.test("john@gmail.com");
console.log(result);
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_3.js
true
rodrigobranas:javascriptmasterclass $


regexp_4.js — javascriptmasterclass
JS regexp_4.js
1 let regExp = /john@gmail.com/;
2 let result = regExp.exec("john@gmail.com");
3 console.log(result);
4
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_4.js
['john@gmail.com', index: 0, input: 'john@g
mail.com']
rodrigobranas:javascriptmasterclass $


regexp_5.js — javascriptmasterclass
JS regexp_5.js
let regExp = /john@gmail.com/;
let result = regExp.exec("E-mail: john@gmail.com");
console.log(result[0]);
console.log(result.index);
console.log(result.input);
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_5.js
john@gmail.com
8
E-mail: john@gmail.com
rodrigobranas:javascriptmasterclass $

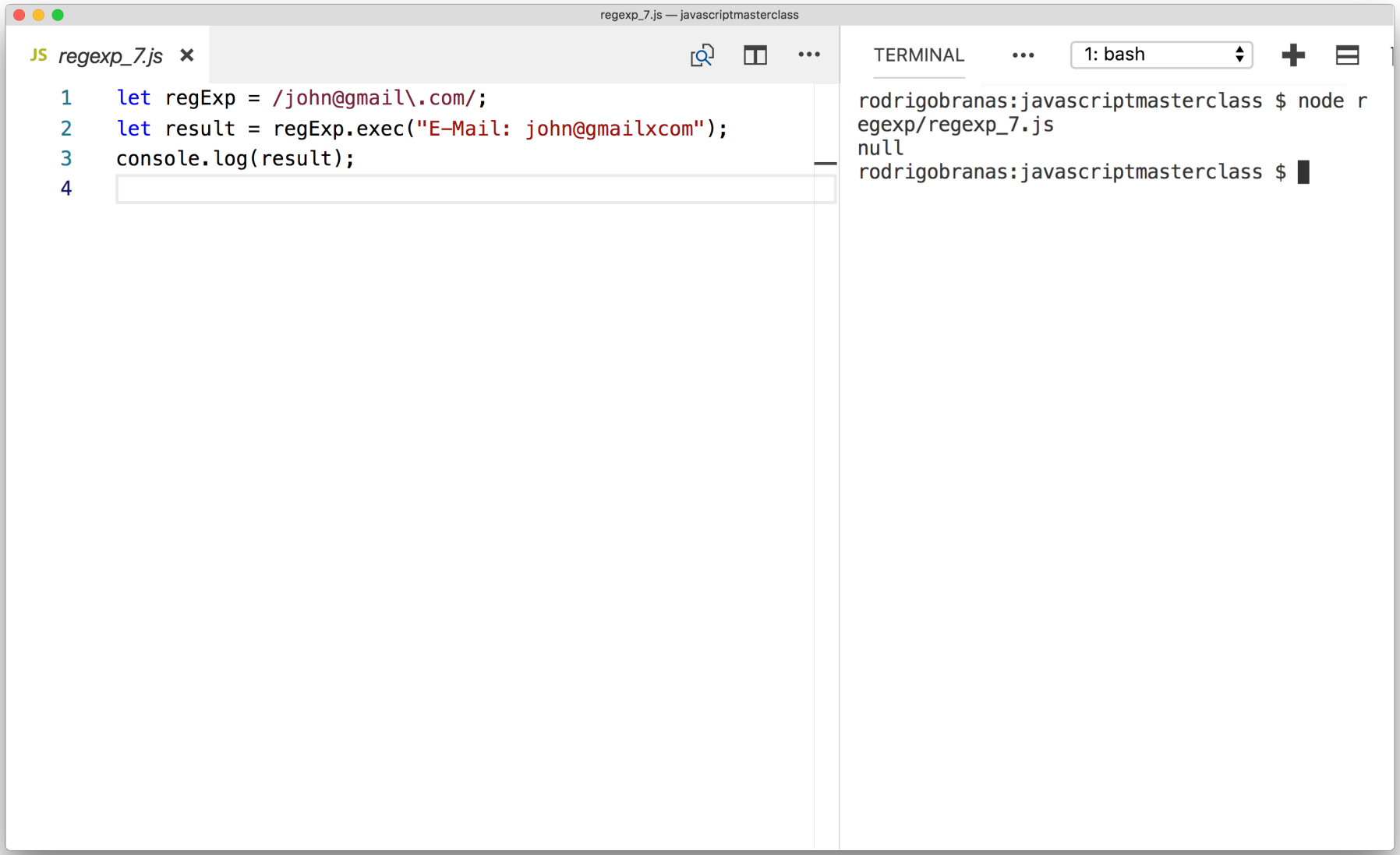
Metacaracteres - Parte 1

. - Representa qualquer caractere


regexp_6.js — javascriptmasterclass
JS regexp_6.js
1 let regExp = /john@gmail.com/;
2 let result = regExp.exec("E-Mail: john@gmailxcom");
3 console.log(result[0]);
4 console.log(result.index);
5 console.log(result.input);
6
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_6.js
john@gmailxcom
8
E-Mail: john@gmailxcom
rodrigobranas:javascriptmasterclass $

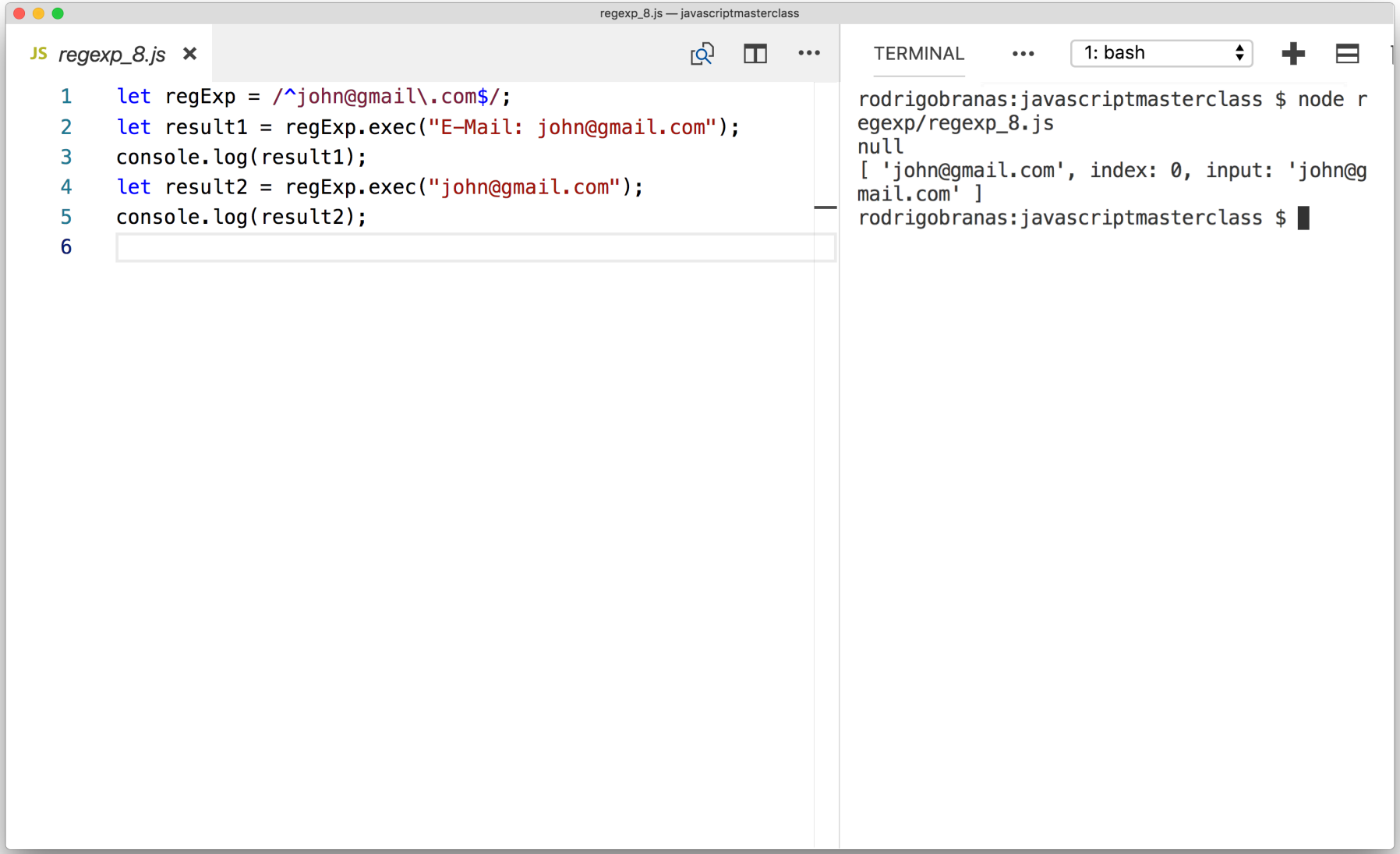
Escapando caracteres especiais

\ - A barra é utilizada antes de caracteres especiais, com o objetivo de escapá-los


regexp_7.js — javascriptmasterclass
JS regexp_7.js
1 let regExp = /john@gmail\.com/;
2 let result = regExp.exec("E-Mail: john@gmailxcom");
3 console.log(result);
4
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_7.js
null
rodrigobranas:javascriptmasterclass $

Iniciando e finalizando com um determinado caractere

^ - Inicia com um determinado caractere
$ - Finaliza com um determinado caractere


regexp_8.js — javascriptmasterclass
JS regexp_8.js
1 let regExp = /^john@gmail\.com$/;
2 let result1 = regExp.exec("E-Mail: john@gmail.com");
3 console.log(result1);
4 let result2 = regExp.exec("john@gmail.com");
5 console.log(result2);
6
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_8.js
null
['john@gmail.com', index: 0, input: 'john@g
mail.com']
rodrigobranas:javascriptmasterclass $

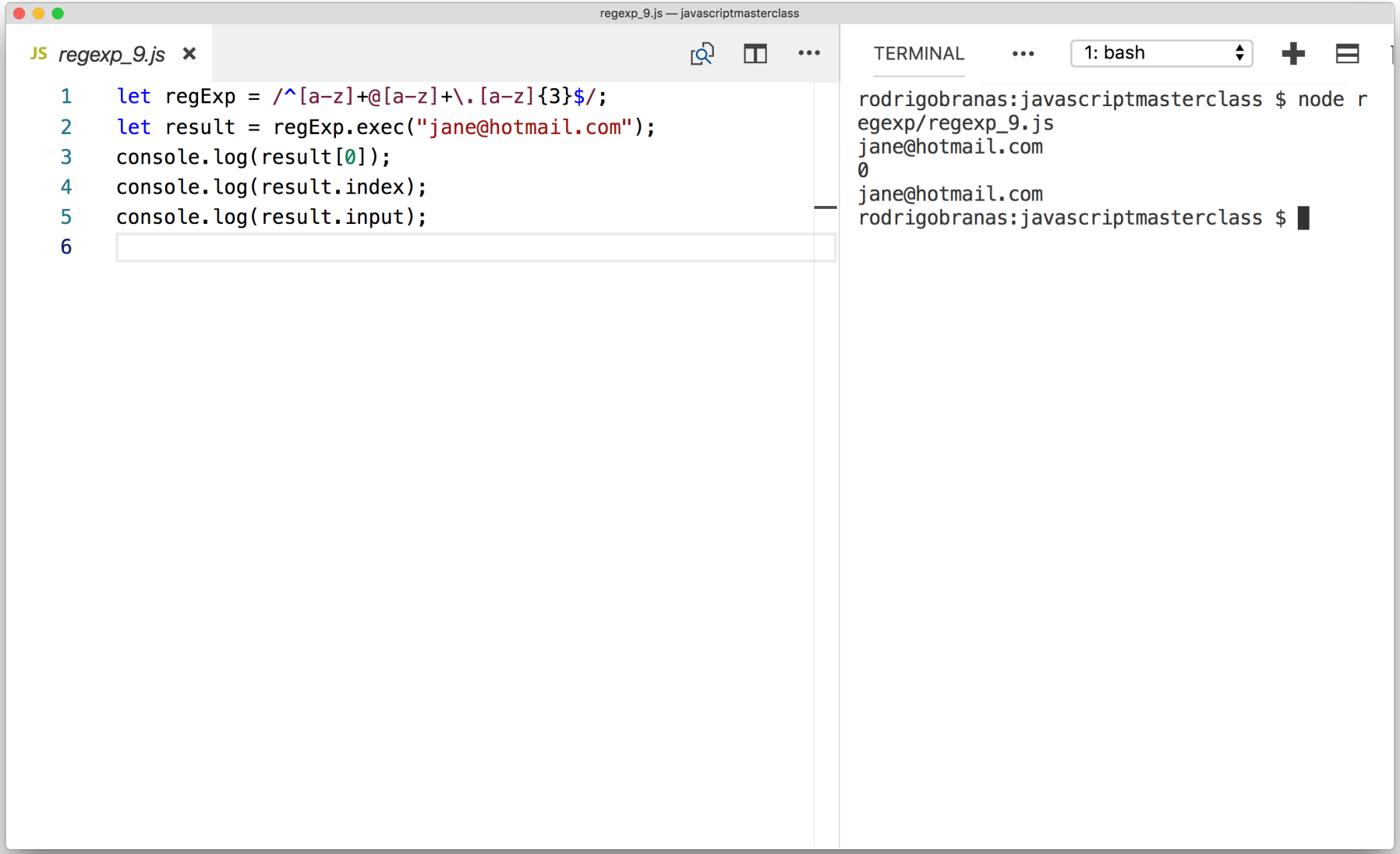
Grupos de caracteres

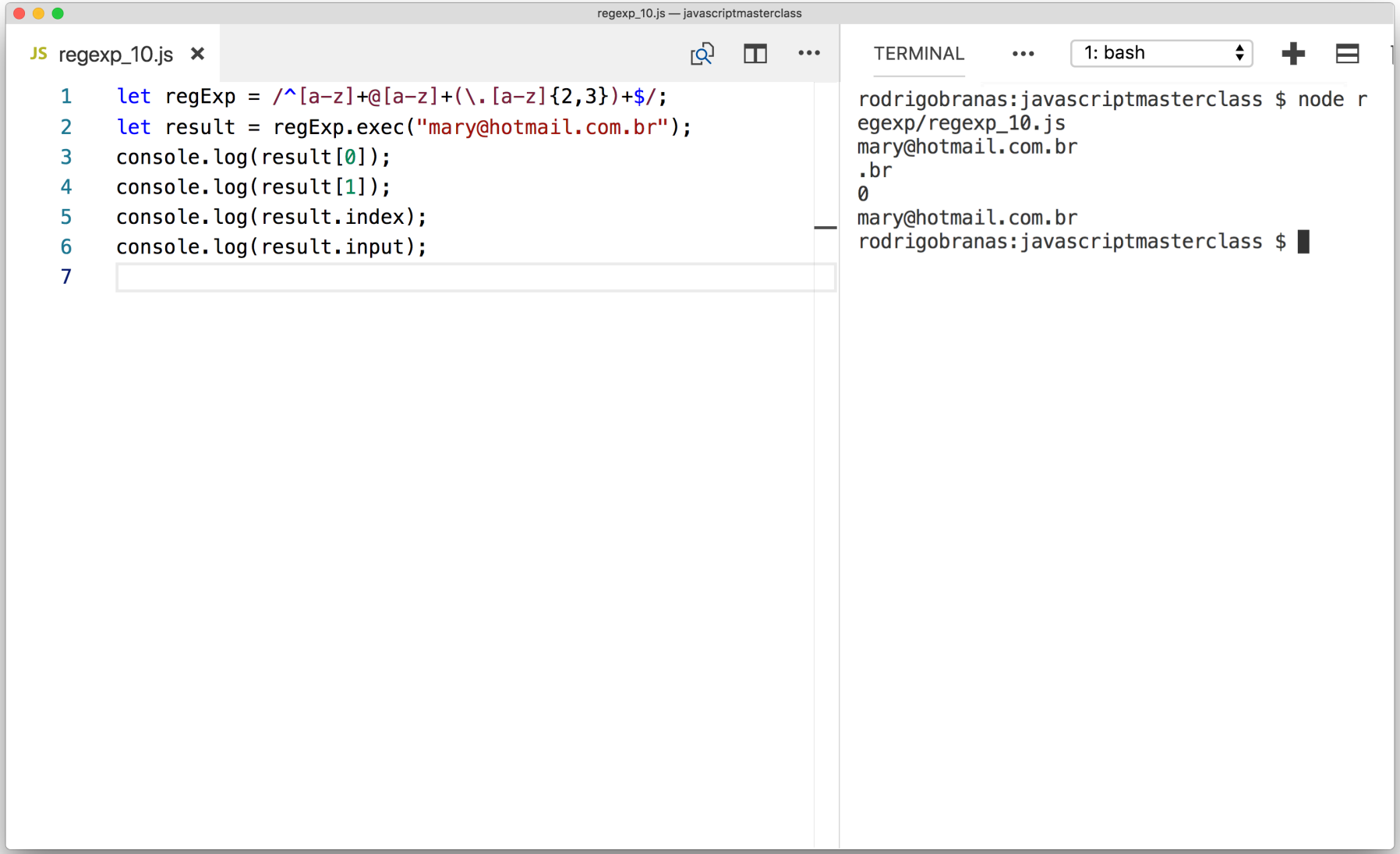
[abc] - Aceita qualquer caractere dentro do grupo, nesse caso a, b e c
[^abc] - Não aceita qualquer caractere dentro do grupo, nesse caso a, b ou c
[0-9] - Aceita qualquer caractere entre 0 e 9
[^0-9] - Não aceita qualquer caractere entre 0 e 9

Quantificadores

Os quantificadores podem ser aplicados a caracteres, grupos, conjuntos ou metacaracteres.

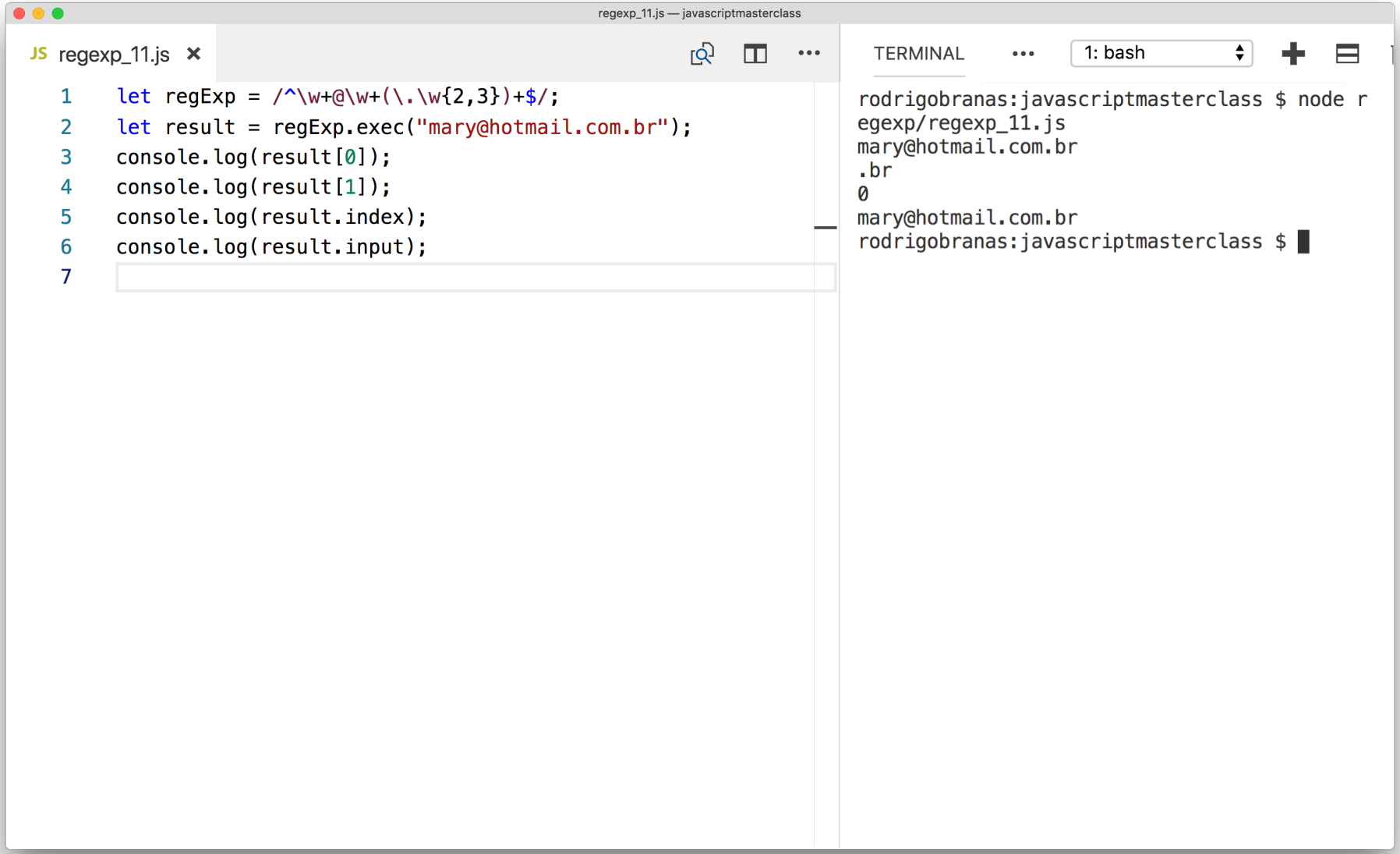
{n} - Quantifica um número específico
{n,} - Quantifica um número mínimo
{n,m} - Quantifica um número mínimo e um número máximo
? - Zero ou um
* - Zero ou mais
+ - Um ou mais


regexp_9.js — javascriptmasterclass
regexp_9.js
let regExp = /^[a-z]+@[a-z]+\.[a-z]{3}$/;
let result = regExp.exec("jane@hotmail.com");
console.log(result[0]);
console.log(result.index);
console.log(result.input);
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_9.js
jane@hotmail.com
0
jane@hotmail.com
rodrigobranas:javascriptmasterclass $


regexp_10.js — javascriptmasterclass
regexp_10.js
let regExp = /^[a-z]+@[a-z]+(\.[a-z]{2,3})+$/;
let result = regExp.exec("mary@hotmail.com.br");
console.log(result[0]);
console.log(result[1]);
console.log(result.index);
console.log(result.input);
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_10.js
mary@hotmail.com.br
.br
0
mary@hotmail.com.br
rodrigobranas:javascriptmasterclass $

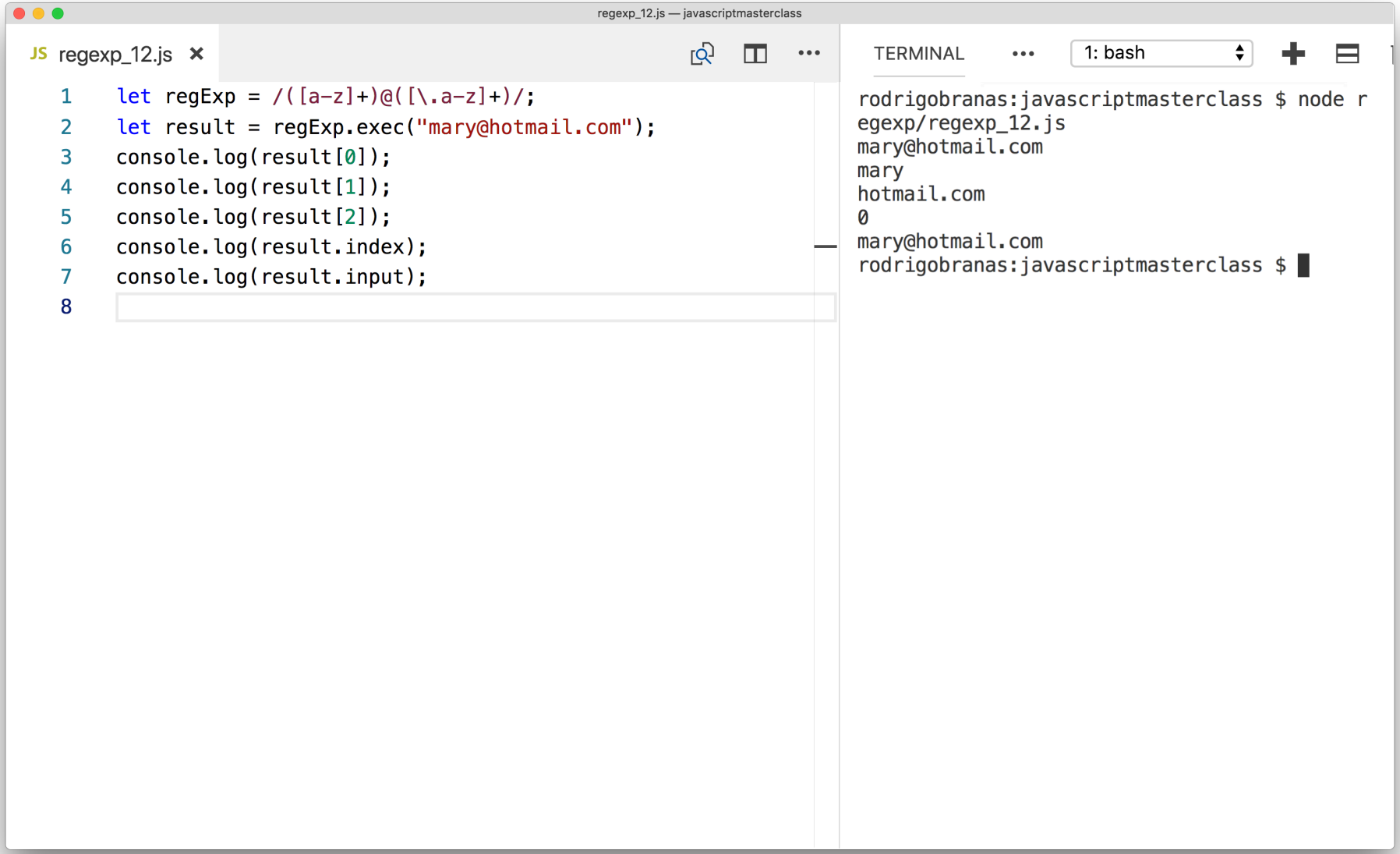
Metacaracteres

\w - Representa o conjunto [a-zA-Z0-9_]
\W - Representa o conjunto [^a-zA-Z0-9_]
\d - Representa o conjunto [0-9]
\D - Representa o conjunto [^0-9]
\s - Representa um espaço em branco
\S - Representa um não espaço em branco
\n - Representa uma quebra de linha
\t - Representa um tab


regexp_11.js — javascriptmasterclass
regexp_11.js
1 let regExp = /^\w+@\w+(\.\w{2,3})+$/;
2 let result = regExp.exec("mary@hotmail.com.br");
3 console.log(result[0]);
4 console.log(result[1]);
5 console.log(result.index);
6 console.log(result.input);
7
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_11.js
mary@hotmail.com.br
.br
0
mary@hotmail.com.br
rodrigobranas:javascriptmasterclass $

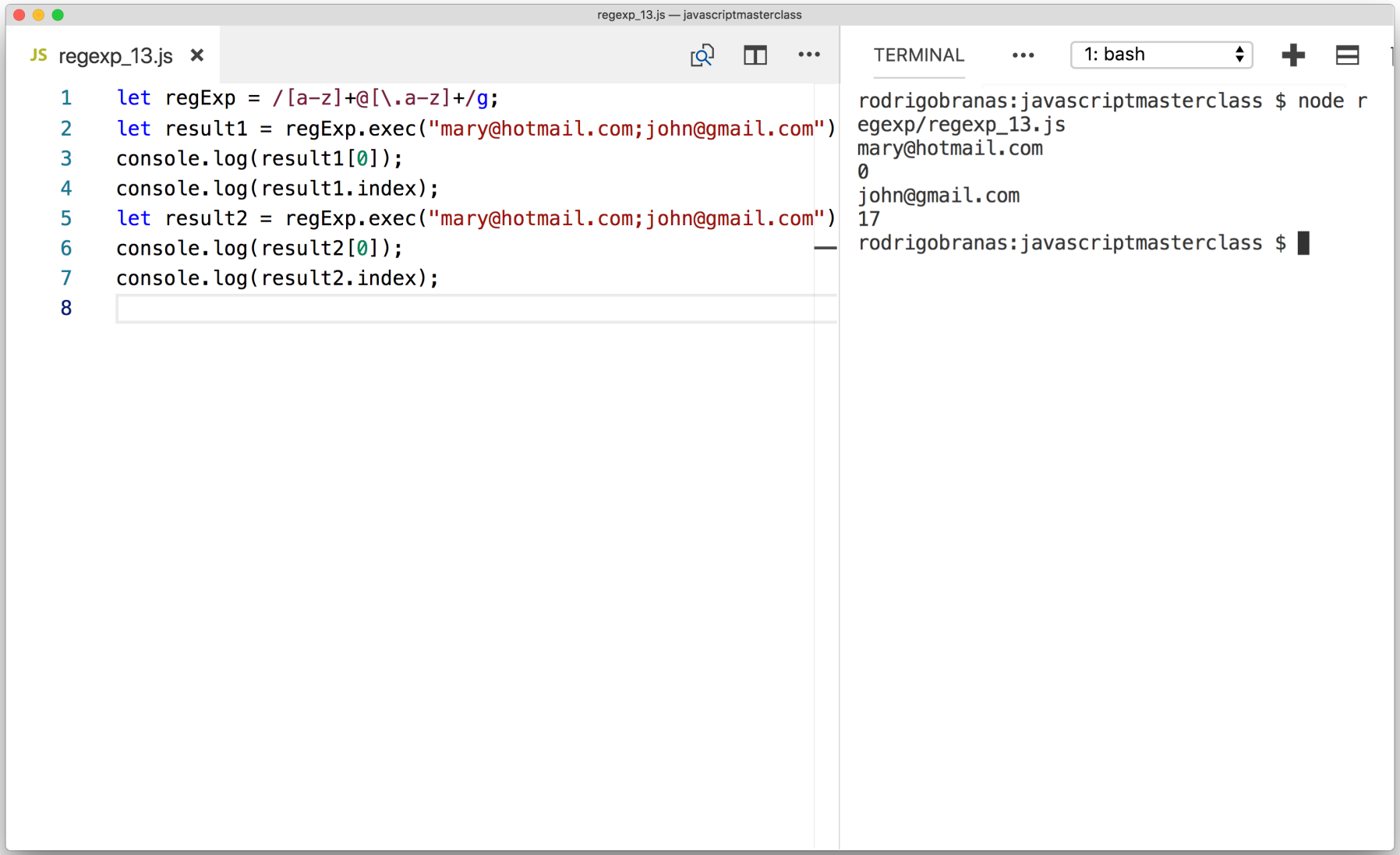
Grupos de Captura

() - Determina um grupo de captura para realizar a extração de valores de uma determinada String


regexp_12.js — javascriptmasterclass
regexp_12.js
let regExp = /([a-z]+)@([\.a-z]+)/;
let result = regExp.exec("mary@hotmail.com");
console.log(result[0]);
console.log(result[1]);
console.log(result[2]);
console.log(result.index);
console.log(result.input);
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_12.js
mary@hotmail.com
mary
hotmail.com
0
mary@hotmail.com
rodrigobranas:javascriptmasterclass $

Modificadores

i - Case-insensitive matching
g - Global matching
m - Multiline matching


regexp_13.js — javascriptmasterclass
regexp_13.js
let regExp = /[a-z]+@[\.a-z]+/g;
let result1 = regExp.exec("mary@hotmail.com;john@gmail.com")
console.log(result1[0]);
console.log(result1.index);
let result2 = regExp.exec("mary@hotmail.com;john@gmail.com")
console.log(result2[0]);
console.log(result2.index);
TERMINAL
1: bash
rodrigobranas:javascriptmasterclass $ node r
egexp/regexp_13.js
mary@hotmail.com
0
john@gmail.com
17
rodrigobranas:javascriptmasterclass $