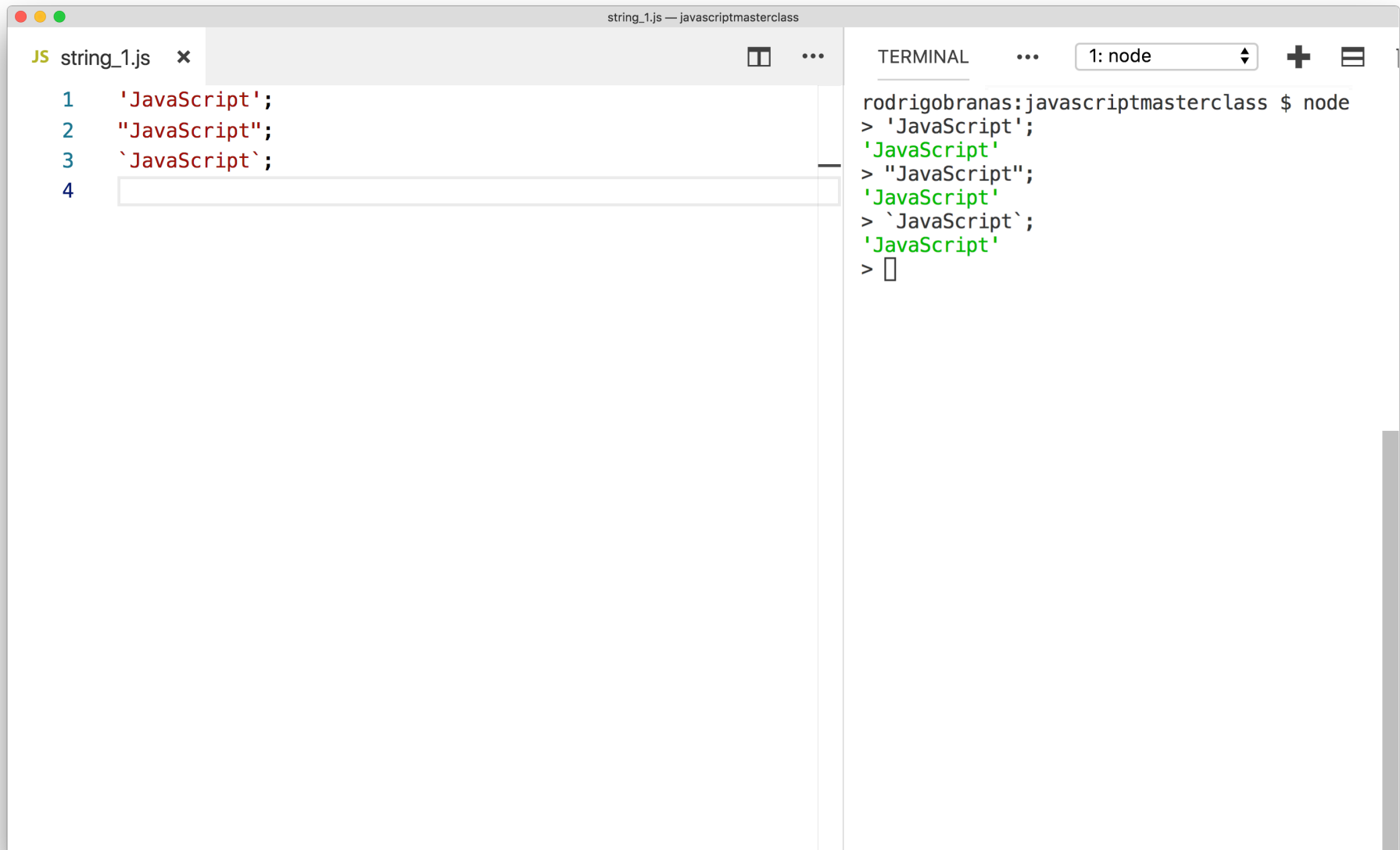




String

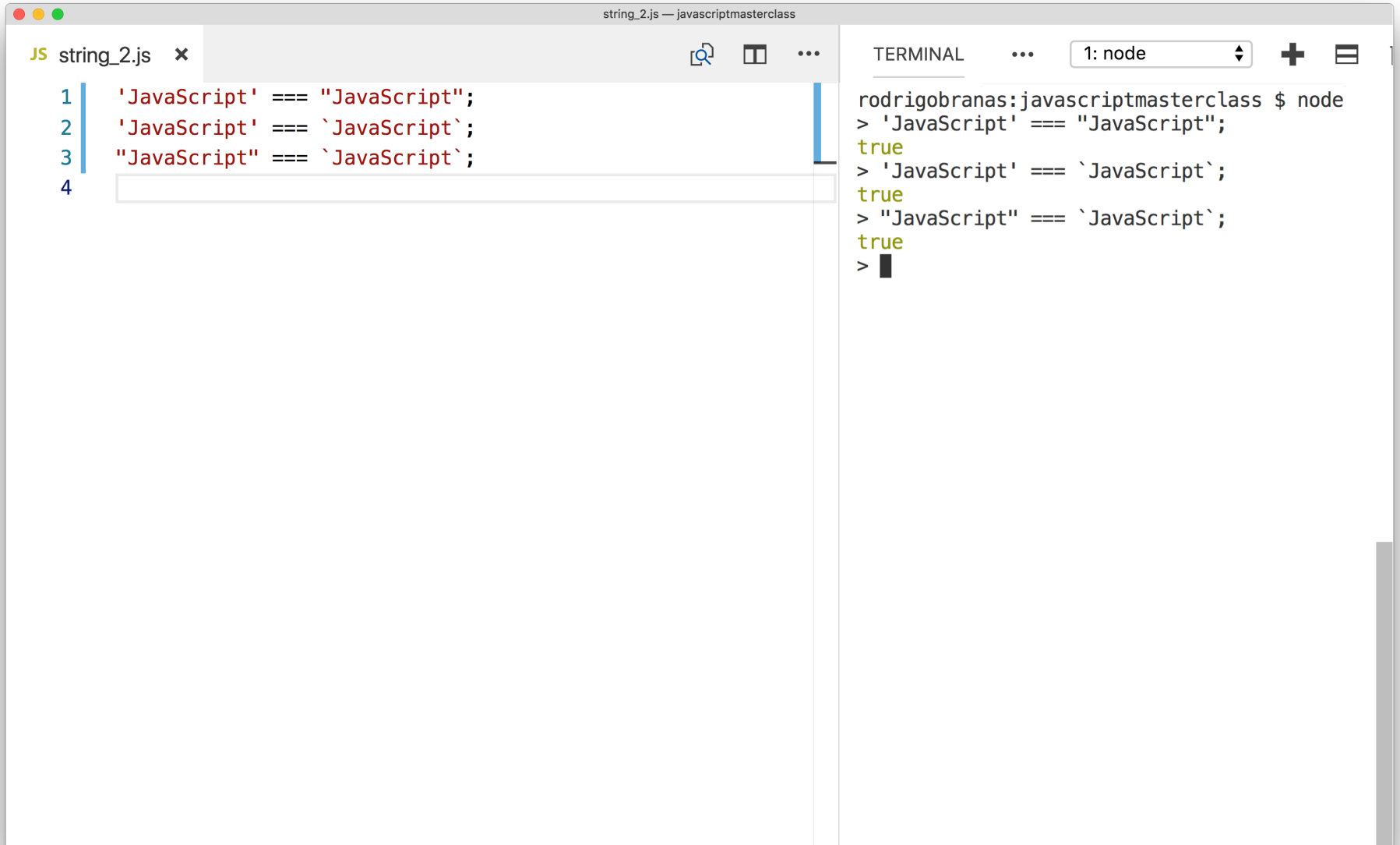
O tipo String é primitivo, imutável e é representado internamente pelo padrão **Unicode**, codificado em UTF-16

Existem 3 formas de declarar uma String de forma literal: **aspas simples**, **aspas duplas** ou **acento grave**

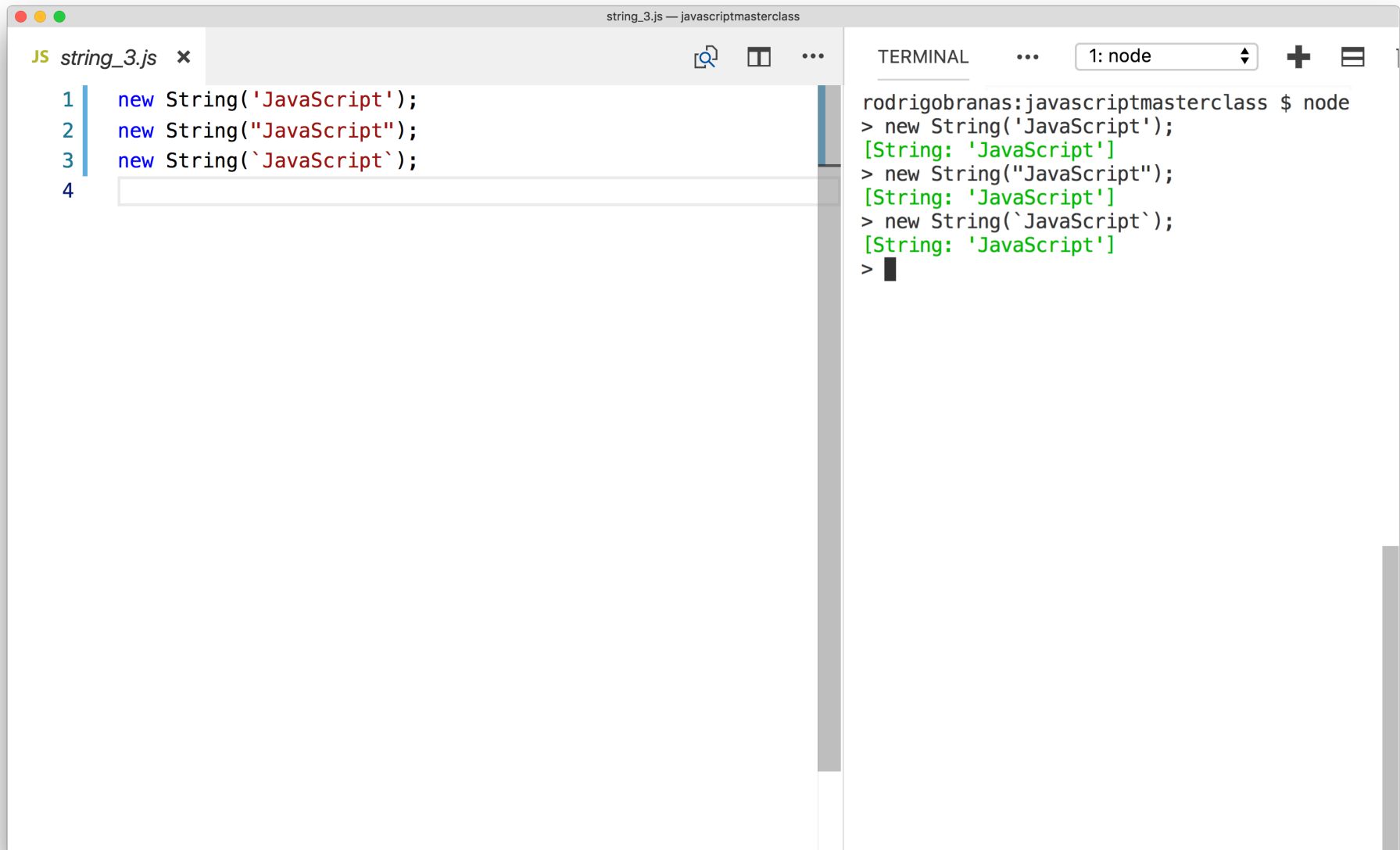




Existe **diferença** entre essas formas?



Também é possível declarar uma String
por meio da sua **função construtora**





CAUTION

A função construtora é mais lenta e
pode prejudicar a performance

JS string_4.js x



TERMINAL



1: bash



```
1 let counter = 0;
2 console.time("performance");
3 while (counter < 100000) {
4   "JavaScript";
5   counter++;
6 }
7 console.timeEnd("performance");
8
```

```
rodrigobranas:javascriptmasterclass $ node s
tring/string_4.js
performance: 2.168ms
rodrigobranas:javascriptmasterclass $
```

JS string_5.js x



TERMINAL



1: bash



```
1 let counter = 0;
2 console.time("performance");
3 while (counter < 100000) {
4   new String("JavaScript");
5   counter++;
6 }
7 console.timeEnd("performance");
8
```

```
rodrigobranas:javascriptmasterclass $ node s
tring/string_5.js
performance: 4.253ms
rodrigobranas:javascriptmasterclass $
```

Alguns tipos de caracteres não são permitidos e **precisam ser escapados**

JS string_6.js x



TERMINAL



1: bash



```
1 console.log("Each constructor is a function that has a
2 console.log('An object\'s prototype chain should have
3 console.log(`The proper way to cause a line terminator
4 be part of the String value of a string literal is to
5 use an escape sequence such as \n or \u000A`);
```

```
rodrigobranas:javascriptmasterclass $ node s
tring/string_6.js
```

Each constructor is a function that has a property named "prototype" that is used to implement prototype-based inheritance and shared properties.

An object's prototype chain should have finite length.

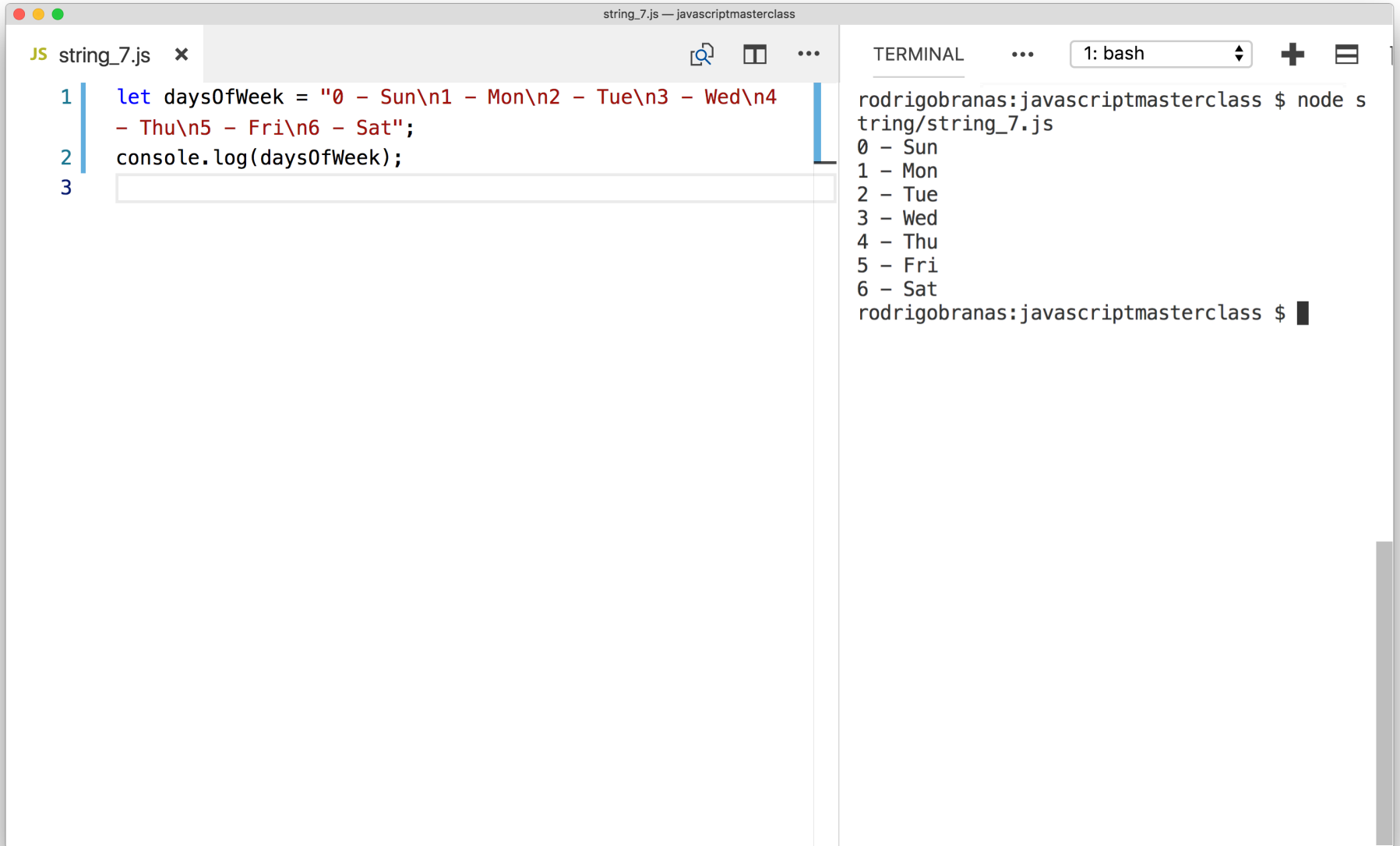
The proper way to cause a line terminator code point to

be part of the String value of a string literal is to use an escape sequence such as \n or \u000A

```
rodrigobranas:javascriptmasterclass $
```

Existem também **caracteres de controle** que podem ser utilizados para quebrar linhas, aumentar a indentação entre outras coisas

Caractere de Controle	Descrição
\b	Backspace
\f	Form Feed
\n	New Line
\r	Carriage Return
\t	Horizontal Tabulator
\v	Vertical Tabulator



É possível escrever caracteres utilizando
seu **code point** no padrão Unicode

JS string_8.js x



TERMINAL



1: bash



```
1 let daysOfWeekInCodePoints = "\u0030 \u002d
  \u0053\u0075\u006e\u000A\u0031 \u002d
  \u004d\u0066\u006e\u000A\u0032 \u002d
  \u0054\u0075\u0065\u000A\u0033 \u002d
  \u0057\u0065\u0064\u000A\u0034 \u002d
  \u0054\u0068\u0075\u000A\u0035 \u002d
  \u0046\u0072\u0069\u000A\u0036 \u002d
  \u0053\u0061\u0074";
2 console.log(daysOfWeekInCodePoints);
3
```

```
rodrigobranas:javascriptmasterclass $ node s
tring/string_8.js
0 - Sun
1 - Mon
2 - Tue
3 - Wed
4 - Thu
5 - Fri
6 - Sat
rodrigobranas:javascriptmasterclass $
```