



Map

Um Map é um objeto que armazena um conjunto de **chaves** e **valores** que podem ser de qualquer tipo de dado

A screenshot of a macOS desktop environment showing a terminal window. The window title is "map_1.js — javascriptmasterclass". The terminal pane displays the command "node map/map_1.js" followed by the output "Map {}".

```
JS map_1.js
1 const timeUnits = new Map();
2 console.log(timeUnits);
3
```

```
rodrigobranas:javascriptmasterclass $ node map/map_1.js
Map {}
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS terminal window titled "map_2.js — javascriptmasterclass". The window is split into two panes: a code editor on the left and a terminal on the right.

The code editor pane contains the following JavaScript code:

```
JS map_2.js  x
1 const timeUnits = new Map([['second', 1], ['minute', 60], ['hour', 3600]]);
2 console.log(timeUnits);
3
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node map/map_2.js
Map { 'second' => 1, 'minute' => 60, 'hour' => 3600 }
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS terminal window titled "map_3.js — javascriptmasterclass". The window is split into two panes: a code editor on the left and a terminal on the right.

The code editor pane contains the following JavaScript code:

```
JS map_3.js  x
1 const timeUnits = new Map([['second', 1], ['minute', 60], ['hour', 3600]]);
2 console.log(timeUnits);
3 console.log(Array.from(timeUnits));
4
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node map/map_3.js
Map { 'second' => 1, 'minute' => 60, 'hour' => 3600 }
[ [ 'second', 1 ], [ 'minute', 60 ], [ 'hour', 3600 ] ]
rodrigobranas:javascriptmasterclass $
```

- **size**: Retorna a quantidade de elementos
- **set**: Adiciona um par de chave e valor
- **forEach**: Itera sobre o mapa
- **has**: Retorna true se a chave existir
- **get**: Retorna o valor de uma determinada chave
- **delete**: Remove um par de chave e valor
- **clear**: Remove todos os elementos

A screenshot of a macOS terminal window titled "map_4.js — javascriptmasterclass". The window is split into two panes: a code editor on the left and a terminal on the right.

The code editor pane contains the following JavaScript code:

```
JS map_4.js  x
1 const timeUnits = new Map([['second', 1], ['minute', 60], ['hour', 3600]]);
2 console.log(timeUnits);
3 console.log(timeUnits.size);
4
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node map/map_4.js
Map { 'second' => 1, 'minute' => 60, 'hour' => 3600 }
3
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS desktop environment showing a terminal window. The window title is "map_5.js — javascriptmasterclass". The main pane displays the contents of a file named "map_5.js". The code defines a Map object with three entries: "second" (value 1), "minute" (value 60), and "hour" (value 3600). The bottom pane shows the output of running the script with "node map_5.js", which logs the Map object to the terminal.

```
JS map_5.js × map_5.js — javascriptmasterclass ⌂ ⌓ ⌚ TERMINAL ⌓ 1: bash ⌘ + ⌚ 1

1 const timeUnits = new Map();
2 timeUnits.set("second", 1);
3 timeUnits.set("minute", 60);
4 timeUnits.set("hour", 3600);
5 console.log(timeUnits);
6

rodrigobranas:javascriptmasterclass $ node map/map_5.js
Map { 'second' => 1, 'minute' => 60, 'hour' => 3600 }
rodrigobranas:javascriptmasterclass $
```

A screenshot of a Mac OS X desktop environment showing a terminal window. The window title is "map_6.js — javascriptmasterclass". The main pane contains the following JavaScript code:

```
1 const timeUnits = new Map();
2 timeUnits.set("second", 1);
3 timeUnits.set("minute", 60);
4 timeUnits.set("hour", 3600);
5 timeUnits.forEach(function (value, key) {
6     console.log(value, key)
7 });
8
```

The right side of the window shows the terminal output. The prompt is "rodrigobranas:javascriptmasterclass \$ node map/map_6.js". The output is:

```
1 'second'
60 'minute'
3600 'hour'
rodrigobranas:javascriptmasterclass $
```

A screenshot of a Mac OS X desktop environment showing a terminal window. The window title is "map_7.js — javascriptmasterclass". The terminal pane displays the command "node map/map_7.js" followed by the output "true" and "false".

```
JS map_7.js × map_7.js — javascriptmasterclass TERMINAL 1: bash + = rodrigobranas:javascriptmasterclass $ node map/map_7.js true false rodrigobranas:javascriptmasterclass $
```

```
1 const timeUnits = new Map();
2 timeUnits.set("second", 1);
3 timeUnits.set("minute", 60);
4 timeUnits.set("hour", 3600);
5 console.log(timeUnits.has("hour"));
6 console.log(timeUnits.has("day"));
7
```

A screenshot of a Mac OS X desktop environment showing a terminal window. The window title is "map_8.js — javascriptmasterclass". The main pane contains the following JavaScript code:

```
JS map_8.js  x
map_8.js
1 const timeUnits = new Map();
2 timeUnits.set("second", 1);
3 timeUnits.set("minute", 60);
4 timeUnits.set("hour", 3600);
5 console.log(timeUnits.get("second"));
6 console.log(timeUnits.get("minute"));
7 console.log(timeUnits.get("hour"));
8
```

The right side of the window shows the terminal output:

```
rodrigobranas:javascriptmasterclass $ node map_8.js
1
60
3600
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS desktop environment showing a terminal window. The window title is "map_9.js — javascriptmasterclass". The terminal tab is labeled "1: bash". The command run is "node map/map_9.js". The output shows three lines of text: "1", "60", and "undefined".

```
JS map_9.js × map_9.js — javascriptmasterclass TERMINAL ... 1: bash + = rodrigobranas:javascriptmasterclass $ node map/map_9.js 1 60 undefined rodrigobranas:javascriptmasterclass $
```

```
1  const timeUnits = new Map();
2  timeUnits.set("second", 1);
3  timeUnits.set("minute", 60);
4  timeUnits.set("hour", 3600);
5  timeUnits.delete("hour");
6  console.log(timeUnits.get("second"));
7  console.log(timeUnits.get("minute"));
8  console.log(timeUnits.get("hour"));
9
```

A screenshot of a macOS terminal window titled "map_10.js — javascriptmasterclass". The window is split into two panes: a code editor on the left and a terminal on the right.

The code editor pane contains the following JavaScript code:

```
JS map_10.js  x
1 const timeUnits = new Map();
2 timeUnits.set("second", 1);
3 timeUnits.set("minute", 60);
4 timeUnits.set("hour", 3600);
5 console.log(timeUnits);
6 timeUnits.clear();
7 console.log(timeUnits);
8
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node map/map_10.js
Map { 'second' => 1, 'minute' => 60, 'hour' => 3600 }
Map {}
rodrigobranas:javascriptmasterclass $
```



Qual é a diferença para um objeto?

A screenshot of a macOS terminal window titled "map_11.js — javascriptmasterclass". The window is split into two panes: a code editor on the left and a terminal on the right.

The code editor pane contains the following JavaScript code:

```
JS map_11.js  x
1 const obj = {};
2 obj[10] = "Number";
3 obj["10"] = "String";
4 obj[true] = "Boolean";
5 obj["true"] = "String";
6 console.log(obj[10]);
7 console.log(obj["10"]);
8 console.log(obj[true]);
9 console.log(obj["true"]);
10
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node map/map_11.js
String
String
String
String
String
rodrigobranas:javascriptmasterclass $ █
```

A screenshot of a macOS desktop environment showing a terminal window. The window title is "map_12.js — javascriptmasterclass". The main pane contains the following JavaScript code:

```
1 const map = new Map();
2 map.set(10, "Number");
3 map.set("10", "String");
4 map.set(true, "Boolean");
5 map.set("true", "String");
6 console.log(map.get(10));
7 console.log(map.get('10'));
8 console.log(map.get(true));
9 console.log(map.get("true"));
10
```

The right pane shows the output of running the script with "node map_12.js". The output is:

```
rodrigobranas:javascriptmasterclass $ node map/map_12.js
Number
String
Boolean
String
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS terminal window titled "map_13.js — javascriptmasterclass". The window has three panes: a code editor on the left, a terminal on the right, and a status bar at the top.

The code editor pane shows the following JavaScript code:

```
JS map_13.js  x
1 const obj = {};
2 console.log("toString" in obj);
3 console.log("valueOf" in obj);
4
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node map/map_13.js
true
true
rodrigobranas:javascriptmasterclass $
```

A screenshot of a Mac OS X desktop environment showing a terminal window. The window title is "map_14.js — javascriptmasterclass". The terminal pane displays the command "node map/map_14.js" followed by the output "false" and "false".

```
JS map_14.js × map_14.js — javascriptmasterclass TERMINAL ... 1: bash + = rodrigobranas:javascriptmasterclass $ node map/map_14.js false false rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS desktop environment showing a terminal window. The window title is "map_15.js — javascriptmasterclass". The terminal pane displays the command "node map/map_15.js" followed by three lines of output: "undefined", "undefined", and "rodrigobranas:javascriptmasterclass \$".

```
JS map_15.js × map_15.js — javascriptmasterclass TERMINAL ... 1: bash + = rodrigobranas:javascriptmasterclass $ node map/map_15.js undefined undefined rodrigobranas:javascriptmasterclass $
```

```
1 const map = new Map();
2 console.log(map.get("toString"));
3 console.log(map.get("valueOf"));
4
```