



# Destructuring

Por meio do **destructuring** podemos  
extrair valores de arrays e objetos de  
uma forma mais simples e direta

destructuring\_1.js — javascriptmasterclass

JS destructuring\_1.js x

```
1 const language = "C;Dennis Ritchie;1972".split(";");
2 const name = language[0];
3 const author = language[1];
4 const year = language[2];
5 console.log(name, author, year);
6
```

TERMINAL ... 1: bash + =

```
rodrigobranas:javascriptmasterclass $ node d
estructuring/destructuring_1.js
C Dennis Ritchie 1972
rodrigobranas:javascriptmasterclass $
```

É possível **extrair os valores de um array**  
criando variáveis em ordem, de acordo  
com a posição de cada elemento

destructuring\_2.js — javascriptmasterclass

JS destructuring\_2.js x

```
1 const language = "C;Dennis Ritchie;1972".split(";");
2 const [name, author, year] = language;
3 console.log(name, author, year);
4
```

TERMINAL ... 1: bash + =

```
rodrigobranas:javascriptmasterclass $ node d
estructuring/destructuring_2.js
C Dennis Ritchie 1972
rodrigobranas:javascriptmasterclass $
```

destructuring\_3.js — javascriptmasterclass

JS destructuring\_3.js x

```
1 const [name, author, year] = "C;Dennis Ritchie;1972".split(";");
2 console.log(name, author, year);
3
```

TERMINAL ... 1: bash

```
rodrigobranas:javascriptmasterclass $ node d
estructuring/destructuring_3.js
C Dennis Ritchie 1972
rodrigobranas:javascriptmasterclass $
```

Podemos **ignorar** um elemento do array  
deixando de estabelecer um nome  
para a variável

A screenshot of a macOS desktop environment showing a terminal window. The window title is "destructuring\_4.js — javascriptmasterclass". The terminal pane displays the command "node destructuring/destructuring\_4.js" followed by the output "Dennis Ritchie 1972".

```
destructuring_4.js — javascriptmasterclass
JS destructuring_4.js ×
1 const [, author, year] = "C;Dennis Ritchie;1972".split(";");
2 console.log(author, year);
3

TERMINAL ... 1: bash
rodrigobranas:javascriptmasterclass $ node d
estructuring/destructuring_4.js
Dennis Ritchie 1972
rodrigobranas:javascriptmasterclass $
```

Assim como nas funções, é possível definir **valores padrão** para cada uma das variáveis

A screenshot of a macOS terminal window titled "destructuring\_5.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 destructuring_5.js ×
1 const language = "C;Dennis Ritchie;1972".split(";");
2 const [name = "-", author = "-", year = "-"] = language;
3 console.log(name, author, year);
4
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node d
estructuring/destructuring_5.js
C Dennis Ritchie 1972
rodrigobranas:javascriptmasterclass $ █
```

A screenshot of a macOS terminal window titled "destructuring\_6.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:

```
1 const language = "C;Dennis Ritchie".split(";");
2 const [name = "-", author = "-", year = "-"] = language;
3 console.log(name, author, year);
4
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node d
estructuring/destructuring_6.js
C Dennis Ritchie -
rodrigobranas:javascriptmasterclass $
```

Para **extrair** os **valores** de um **objeto** é  
necessário referenciar a chave de cada  
uma das propriedades

A screenshot of a macOS desktop environment. On the left is a code editor window titled "destructuring\_7.js — javascriptmasterclass". The code editor contains the following JavaScript code:

```
1 const language = {  
2     name: "C",  
3     author: "Dennis Ritchie",  
4     year: 1972  
5 };  
6 const name = language.name;  
7 const author = language.author;  
8 const year = language.year;  
9 console.log(name, author, year);  
10
```

The code defines an object named "language" with properties "name" (value "C"), "author" (value "Dennis Ritchie"), and "year" (value 1972). It then declares three constants: "name", "author", and "year", each assigned to a specific property of the "language" object. Finally, it logs the values of "name", "author", and "year" to the console.

To the right of the code editor is a terminal window titled "TERMINAL" with a tab labeled "1: bash". The terminal shows the command "node destructuring/destructuring\_7.js" being run, followed by the output "C Dennis Ritchie 1972".

A screenshot of a macOS desktop environment showing a terminal window and a code editor.

The terminal window is titled "1: bash" and contains the following command and output:

```
rodrigobranas:javascriptmasterclass $ node destructuring/destructuring_8.js
C Dennis Ritchie 1972
rodrigobranas:javascriptmasterclass $
```

The code editor window is titled "destructuring\_8.js — javascriptmasterclass". It contains the following JavaScript code:

```
1 const language = {
2     name: "C",
3     author: "Dennis Ritchie",
4     year: 1972
5 };
6 const {name, author, year} = language;
7 console.log(name, author, year);
8
```

É possível definir **nomes diferentes** para  
as **variáveis** em relação as **chaves das**  
**propriedades do objeto**

A screenshot of a macOS desktop environment. On the left is a code editor window titled "destructuring\_9.js — javascriptmasterclass". The code contains the following JavaScript:

```
1 const language = {  
2     name: "C",  
3     author: "Dennis Ritchie",  
4     year: 1972  
5 };  
6 const {name: n, author: a, year: y} = language;  
7 console.log(n, a, y);  
8
```

The code defines a constant `language` object with properties `name`, `author`, and `year`. It then destructures this object into variables `n`, `a`, and `y` respectively, and logs them to the console.

To the right of the code editor is a terminal window titled "TERMINAL" with a tab labeled "1: bash". The terminal output shows the command `node destructuring/destructuring_9.js` being run, followed by the output "C Dennis Ritchie 1972".

Também podemos referenciar as propriedades de **objetos** que estão dentro de outros objetos

A screenshot of a macOS desktop environment. On the left is a code editor window titled "destructuring\_10.js — javascriptmasterclass". The file contains the following JavaScript code:

```
1 const language = {  
2     name: "C",  
3     author: "Dennis Ritchie",  
4     year: 1972,  
5     company: {  
6         name: "Bell Labs"  
7     }  
8 };  
9 const {name: n, author: a, year: y, company: {name: c}} = la  
10 console.log(n, a, y, c);  
11
```

The code defines a constant `language` which is an object with properties: `name`, `author`, `year`, and `company`. The `company` property is another object with a single property `name`. In the last line, the object is destructured into variables `n`, `a`, `y`, and `c`. The variable `language` is misspelled as `la` in the code.

On the right is a terminal window titled "TERMINAL" with a tab labeled "1: bash". The terminal shows the command `node destructuring/destructuring_10.js` being run, followed by the output: "C Dennis Ritchie 1972 Bell Labs".

Podemos aplicar destructuring também nos **parâmetros de uma função**, tanto com arrays quanto com objetos

A screenshot of a macOS desktop environment showing a terminal window and a code editor.

The terminal window (top right) has tabs for "TERMINAL" and "1: bash". The command entered is:

```
rodrigobranas:javascriptmasterclass $ node destructuring/destructuring_11.js
```

The output of the command is:

```
4
```

The code editor (left side) shows a file named "destructuring\_11.js" with the following content:

```
1 function sum(a, b) {  
2     return a + b;  
3 }  
4 console.log(sum(2, 2));  
5
```

The screenshot shows a macOS desktop environment with a terminal window open. The terminal window has a title bar "destructuring\_12.js — javascriptmasterclass". The main pane displays the following code:

```
JS destructuring_12.js ×
1  function sum([a, b]) {
2    return a + b;
3  }
4  console.log(sum([2, 2]));
5
```

Below the code, there is a large empty white area, likely a scrollable workspace or a placeholder for other content.

The right side of the window contains a "TERMINAL" tab and a "1: bash" dropdown. The terminal output is as follows:

```
rodrigobranas:javascriptmasterclass $ node d
estructuring/destructuring_12.js
4
rodrigobranas:javascriptmasterclass $ █
```

A screenshot of a macOS desktop environment showing a terminal window and a code editor.

The terminal window is titled "destructuring\_13.js — javascriptmasterclass". It contains the following text:

```
rodrigobranas:javascriptmasterclass $ node d
estructuring/destructuring_13.js
4
rodrigobranas:javascriptmasterclass $
```

The code editor window is titled "destructuring\_13.js". It contains the following JavaScript code:

```
1  function sum({a, b}) {
2    return a + b;
3  }
4  console.log(sum({a: 2, b: 2}));
5
```