

A close-up photograph of a large number of green apples arranged in a grid pattern. The apples are mostly uniform in color, appearing bright green with some water droplets and small brown stems. In the center of this grid is a single, vibrant red apple, which stands out due to its color and position. The lighting is even, highlighting the texture of the apples' skin.

Set

Um Set é um objeto que armazena  
**elementos** únicos, que podem ser de  
qualquer tipo de dado

A screenshot of a Mac OS X desktop environment showing a terminal window. The window title is "set\_1.js — javascriptmasterclass". The terminal pane displays the command "node set/set\_1.js" followed by the output "Set {}".

set\_1.js — javascriptmasterclass

TERMINAL ... 1: bash + =

```
rodrigobranas:javascriptmasterclass $ node set/set_1.js
Set {}
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS desktop environment showing a terminal window. The window title is "set\_2.js — javascriptmasterclass". The terminal tab bar shows "1: bash". The terminal content displays the output of running the JavaScript file "set\_2.js".

```
rodrigobranas:javascriptmasterclass $ node set/set_2.js
Set { 'ASCII', 'ISO-8599-1', 'UTF-8' }
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS desktop environment. On the left is a code editor window titled "set\_3.js — javascriptmasterclass". The file contains the following JavaScript code:

```
1 const charsets = new Set(["ASCII", "ISO-8599-1", "UTF-8"]);
2 console.log(Array.from(charsets));
3
```

The code uses the `Set` object to store three string values and then logs the contents of the set to the console as an array. On the right is a terminal window titled "1: bash". The terminal shows the command `node set/set_3.js` being run, followed by the output `[ 'ASCII', 'ISO-8599-1', 'UTF-8' ]`, which is the expected result of the code execution.

- **size**: Retorna a quantidade de elementos
- **add**: Adiciona um elemento
- **forEach**: Itera sobre o set
- **has**: Retorna true se o elemento existir
- **delete**: Remove um elemento
- **clear**: Remove todos os elementos

A screenshot of a macOS desktop environment showing a terminal window. The window title is "set\_4.js — javascriptmasterclass". The terminal tab bar shows "1: bash". The terminal pane displays the output of running the JavaScript file "set\_4.js".

```
rodrigobranas:javascriptmasterclass $ node set/set_4.js
Set { 'ASCII', 'ISO-8599-1', 'UTF-8' }
3
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS terminal window titled "set\_5.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 contains the following JavaScript code:

```
1 const charsets = new Set();
2 charsets.add("ASCII");
3 charsets.add("ISO-8859-1");
4 charsets.add("UTF-8");
5 console.log(charsets);
6
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node set/set_5.js
Set { 'ASCII', 'ISO-8859-1', 'UTF-8' }
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS desktop environment showing a terminal window. The window title is "set\_6.js — javascriptmasterclass". The main pane contains the following JavaScript code:

```
1 const charsets = new Set();
2 charsets.add("ASCII");
3 charsets.add("ISO-8859-1");
4 charsets.add("UTF-8");
5 charsets.forEach(function (charset) {
6     console.log(charset);
7 });
8
```

The right side of the window shows the terminal output:

```
rodrigobranas:javascriptmasterclass $ node set/set_6.js
ASCII
ISO-8859-1
UTF-8
rodrigobranas:javascriptmasterclass $
```

A screenshot of a Mac OS X desktop environment showing a terminal window and a code editor.

The terminal window is titled "set\_7.js — javascriptmasterclass". It contains the following text:

```
rodrigobranas:javascriptmasterclass $ node set/set_7.js
true
false
rodrigobranas:javascriptmasterclass $
```

The code editor window is titled "set\_7.js". It contains the following JavaScript code:

```
1 const charsets = new Set();
2 charsets.add("ASCII");
3 charsets.add("ISO-8859-1");
4 charsets.add("UTF-8");
5 console.log(charsets.has("ASCII"));
6 console.log(charsets.has("CP-1252"));
7
```

A screenshot of a macOS terminal window titled "set\_8.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 charsets = new Set();
2 charsets.add("ASCII");
3 charsets.add("ISO-8859-1");
4 charsets.add("UTF-8");
5 console.log(charsets.delete("ISO-8859-1"));
6 console.log(charsets.delete("CP-1252"));
7 console.log(charsets.has("ASCII"));
8 console.log(charsets.has("ISO-8859-1"));
9 console.log(charsets.has("CP-1252"));
10
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node set/set_8.js
true
false
true
false
false
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS desktop environment showing a terminal window. The window title is "set\_9.js — javascriptmasterclass". The main pane contains the following JavaScript code:

```
1 const charsets = new Set();
2 charsets.add("ASCII");
3 charsets.add("ISO-8859-1");
4 charsets.add("UTF-8");
5 console.log(charsets);
6 console.log(charsets.size);
7 charsets.clear();
8 console.log(charsets);
9 console.log(charsets.size);
10
```

The right pane shows the terminal output:

```
rodrigobranas:javascriptmasterclass $ node set/set_9.js
Set { 'ASCII', 'ISO-8859-1', 'UTF-8' }
3
Set {}
0
rodrigobranas:javascriptmasterclass $
```



Qual é a diferença para um array?

A screenshot of a Mac OS X desktop environment showing a terminal window. The window title is "set\_10.js — javascriptmasterclass". The main pane contains the following JavaScript code:

```
1 const array = [];
2 array.push(10);
3 array.push(10);
4 array.push(10);
5 console.log(array);
6 console.log(array.length);
7
```

The right pane is a "TERMINAL" window showing the command-line output:

```
rodrigobranas:javascriptmasterclass $ node set/set_10.js
[ 10, 10, 10 ]
3
rodrigobranas:javascriptmasterclass $
```

A screenshot of a Mac OS X desktop environment showing a terminal window. The window title is "set\_11.js — javascriptmasterclass". The main pane contains the following JavaScript code:

```
1 const set = new Set();
2 set.add("10");
3 set.add("10");
4 set.add("10");
5 console.log(set);
6 console.log(set.size);
7
```

The right pane is a "TERMINAL" window titled "1: bash". It shows the command being run and its output:

```
rodrigobranas:javascriptmasterclass $ node set/set_11.js
Set { '10' }
1
rodrigobranas:javascriptmasterclass $
```

A screenshot of a macOS desktop environment showing a terminal window. The window title is "set\_12.js — javascriptmasterclass". The main pane contains the following JavaScript code:

```
1 const array = [10, 10, 10];
2 console.log(array);
3 console.log(array.length);
4 const set = new Set(array);
5 console.log(set);
6 console.log(set.size);
7
```

The right side of the window shows the terminal output:

```
rodrigobranas:javascriptmasterclass $ node set/set_12.js
[ 10, 10, 10 ]
3
Set { 10 }
1
rodrigobranas:javascriptmasterclass $ █
```

A screenshot of a macOS terminal window titled "set\_12.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 array = [10, 10, 10];
2 console.log(array);
3 console.log(array.length);
4 const set = new Set(array);
5 console.log(set);
6 console.log(set.size);
7
```

The terminal pane shows the output of running the script:

```
rodrigobranas:javascriptmasterclass $ node set/set_12.js
[ 10, 10, 10 ]
3
Set { 10 }
1
rodrigobranas:javascriptmasterclass $ █
```

A screenshot of a Mac OS X desktop environment showing a terminal window. The window title is "set\_13.js — javascriptmasterclass". The main pane contains the following JavaScript code:

```
1 const array = [10, 10, 10];
2 const obj = {};
3 array.forEach(function (element) {
4     obj[element] = undefined;
5 });
6 const set = Object.keys(obj);
7 console.log(set);
8
```

The right side of the window shows the terminal output:

```
rodrigobranas:javascriptmasterclass $ node set/set_13.js
[ '10' ]
rodrigobranas:javascriptmasterclass $
```

The screenshot shows a macOS desktop environment with a terminal window open. The terminal window has a title bar "set\_14.js — javascriptmasterclass". The main pane displays the contents of the file "set\_14.js", which is a JavaScript code snippet. The code creates a new array from a given array by filtering out duplicate values. The right pane of the terminal window shows the output of running the script with the command "node set/set\_14.js". The output is "[ 10 ]", indicating that the script successfully removed the duplicates from the input array.

```
set_14.js  x
set_14.js — javascriptmasterclass
TERMINAL  ...  1: bash  +  =  1

JS set_14.js
1 const array = [10, 10, 10];
2 const set = [];
3 array.forEach(function (element) {
4     if (!set.includes(element)) {
5         set.push(element);
6     }
7 });
8 console.log(set);
9

rodrigobranas:javascriptmasterclass $ node set/set_14.js
[ 10 ]
rodrigobranas:javascriptmasterclass $
```