



Modules

No ES6, ou ECMAScript 2015, foi especificado na própria linguagem, baseado no conceito de **importação** e **exportação**

The screenshot shows a Mac OS X desktop environment with a terminal window open. The terminal window has a title bar "modules_1.js — javascriptmasterclass". The main pane of the terminal displays the command "node modules/modules_1.js" followed by the output: "Circle { radius: 10 }" and "314.1592". The terminal window also includes standard OS X window controls (red, yellow, green) and a menu bar.

```
modules_1.js x modules_1.js — javascriptmasterclass TERMINAL ... 1: bash + = rodrigobranas:javascriptmasterclass $ node modules/modules_1.js Circle { radius: 10 } 314.1592 rodrigobranas:javascriptmasterclass $
```

```
1 const PI = 3.141592;
2 const pow = function(base, exponential) {
3     if (exponential === 0) return 1;
4     return base * pow(base, exponential - 1);
5 }
6 class Circle {
7     constructor(radius) {
8         this.radius = radius;
9     }
10
11     get area() {
12         return PI * pow(this.radius, 2);
13     }
14 }
15 const circle = new Circle(10);
16 console.log(circle);
17 console.log(circle.area);
18
```

Por meio da palavra-chave **export** é
possível **exportar qualquer tipo de dado**
existente dentro de um módulo

A screenshot of a macOS desktop environment showing a terminal window. The window has a dark theme and is titled "math.mjs — javascriptmasterclass". The main pane displays the following JavaScript code:

```
1  export const PI = 3.141592;
2  export function pow(base, exponential) {
3      if (exponential === 0) return 1;
4      return base * pow(base, exponential - 1);
5  }
6
```

The right side of the window contains a "TERMINAL" tab with a single line of text: "rodrigobranas:javascriptmasterclass \$".

A palavra-chave **import** faz a importação
de qualquer tipo de dado exportado para
dentro do módulo

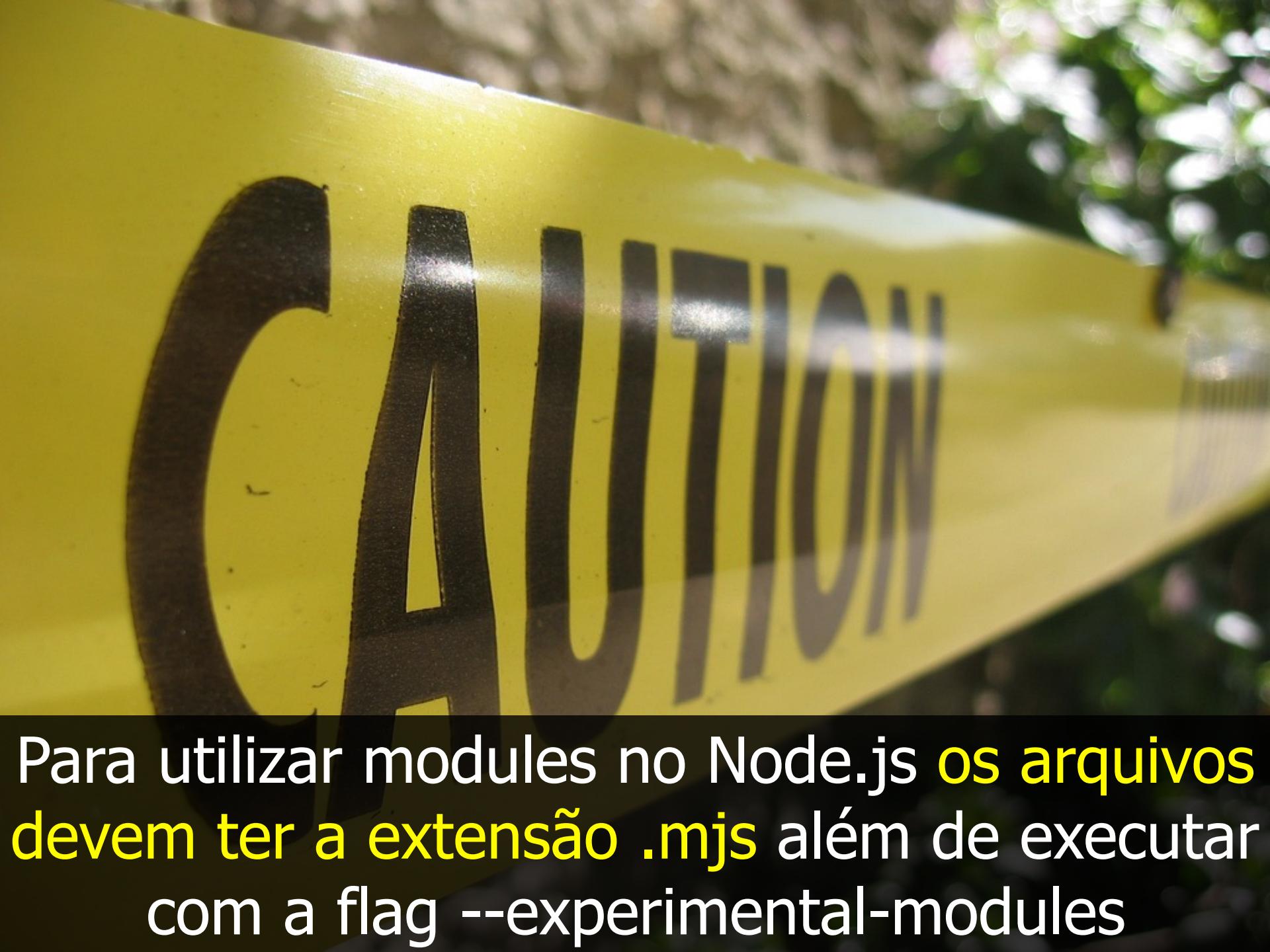
The screenshot shows a Mac OS X desktop environment with a terminal window open. The terminal window has a title bar "modules_2.js — javascriptmasterclass". The main pane displays the following JavaScript code:

```
JS modules_2.js ×
1 import {PI, pow} from './math';
2 class Circle {
3     constructor(radius) {
4         this.radius = radius;
5     }
6
7     get area() {
8         return PI * pow(this.radius, 2);
9     }
10}
11const circle = new Circle(10);
12console.log(circle);
13console.log(circle.area);
14
```

The right pane is a "TERMINAL" window showing the output of the command "node modules/modules_2.js". The output shows the following error message:

```
rodrigobranas:javascriptmasterclass $ node modules/modules_2.js
/Users/rodrigobranas/development/workspace/javascriptmasterclass/modules/modules_2.js:1
(function (exports, require, module, __filename, __dirname) { import {PI, pow} from './math';

~~~~~
SyntaxError: Unexpected token import
    at createScript (vm.js:80:10)
    at Object.runInThisContext (vm.js:139:10)
)
    at Module._compile (module.js:607:28)
    at Object.Module._extensions..js (module.js:654:10)
    at Module.load (module.js:556:32)
    at tryModuleLoad (module.js:499:12)
    at Function.Module._load (module.js:491:3)
    at Function.Module.runMain (module.js:684:10)
    at startup (bootstrap_node.js:187:16)
    at bootstrap_node.js:608:3
rodrigobranas:javascriptmasterclass $
```

A close-up photograph of a yellow caution tape. The word "CAUTION" is printed in large, bold, black capital letters. The tape is slightly curved, and the background shows some green foliage.

Para utilizar modules no Node.js os arquivos devem ter a extensão .mjs além de executar com a flag --experimental-modules

A screenshot of a terminal window titled "modules_3.mjs — javascriptmasterclass". The window is divided into two main sections: a code editor on the left and a terminal on the right.

The code editor contains the following JavaScript code:

```
1 import {PI, pow} from './math';
2 class Circle {
3     constructor(radius) {
4         this.radius = radius;
5     }
6
7     get area() {
8         return PI * pow(this.radius, 2);
9     }
10}
11const circle = new Circle(10);
12console.log(circle);
13console.log(circle.area);
14
```

The terminal section shows the output of running the code with Node.js:

```
rodrigobranas:javascriptmasterclass $ node --experimental-modules modules/modules_3.mjs
(node:57426) ExperimentalWarning: The ESM module loader is experimental.
Circle { radius: 10 }
314.1592
rodrigobranas:javascriptmasterclass $
```

É possível utilizar um **alias** na importação,
renomeando o que estiver sendo
importado

modules_4.mjs (deleted from disk) — javascriptmasterclass

JS modules_4.mjs x

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node --experimental-modules modules/modules_4.mjs
(node:57493) ExperimentalWarning: The ESM module loader is experimental.
Circle { radius: 10 }
314.1592
rodrigobranas:javascriptmasterclass $
```

```
1 import {PI as pi, pow} from './math';
2 class Circle {
3     constructor(radius) {
4         this.radius = radius;
5     }
6
7     get area() {
8         return pi * pow(this.radius, 2);
9     }
10}
11const circle = new Circle(10);
12console.log(circle);
13console.log(circle.area);
14
```

Por meio do * é possível importar tudo
que estiver sendo exportado em um único
objeto

modules_5.mjs (deleted from disk) — javascriptmasterclass

JS modules_5.mjs x

```
1 import * as math from './math';
2 class Circle {
3     constructor(radius) {
4         this.radius = radius;
5     }
6
7     get area() {
8         return math.PI * math.pow(this.radius, 2);
9     }
10}
11const circle = new Circle(10);
12console.log(circle);
13console.log(circle.area);
14
```

TERMINAL ... 1: bash

```
rodrigobranas:javascriptmasterclass $ node --experimental-modules modules/modules_5.mjs
(node:57535) ExperimentalWarning: The ESM module loader is experimental.
Circle { radius: 10 }
314.1592
rodrigobranas:javascriptmasterclass $
```

Também podemos importar e exportar de forma padrão utilizando a palavra-chave **default**

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

```
JS circle.mjs  x
1 import * as math from './math';
2 export default class Circle {
3     constructor(radius) {
4         this.radius = radius;
5     }
6
7     get area() {
8         return math.PI * math.pow(this.radius, 2);
9     }
10}
11
```

The code imports the "math" module and defines a "Circle" class with a constructor and a computed "area" property.

To the right of the code editor is a terminal window titled "TERMINAL" with the identifier "1: bash". The terminal shows the command "rodrigobranas:javascriptmasterclass \$".

A screenshot of a terminal window titled "modules_6.mjs — javascriptmasterclass". The window is split into two panes. The left pane contains the code:

```
JS modules_6.mjs x
modules_6.mjs — javascriptmasterclass
1 import Circle from './circle';
2 const circle = new Circle(10);
3 console.log(circle);
4 console.log(circle.area);
5
```

The right pane shows the output of running the script with Node.js:

```
rodrigobranas:javascriptmasterclass $ node --experimental-modules modules/modules_6.mjs
(node:57578) ExperimentalWarning: The ESM module loader is experimental.
Circle { radius: 10 }
314.1592
rodrigobranas:javascriptmasterclass $ █
```



A **ordem** da importação é relevante?

A screenshot of a macOS terminal window titled "modules_7.mjs — 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 circle = new Circle(10);
2 console.log(circle);
3 console.log(circle.area);
4 import Circle from './circle';
5
```

The terminal pane shows the output of running the code with Node.js:

```
rodrigobranas:javascriptmasterclass $ node --experimental-modules modules/modules_7.mjs
(node:57647) ExperimentalWarning: The ESM module loader is experimental.
Circle { radius: 10 }
314.1592
rodrigobranas:javascriptmasterclass $
```



CAUTION

Não é permitido realizar a importação e
exportação dentro de blocos

The screenshot shows a terminal window with the following details:

- Title Bar:** modules_8.mjs — javascriptmasterclass
- Code Editor:** A file named "modules_8.mjs" containing the following code:

```
1 function fn1() {  
2     import Circle from './circle';  
3     const circle = new Circle(10);  
4     console.log(circle);  
5     console.log(circle.area);  
6 }  
7 fn1();  
8
```
- Terminal:** A bash shell window titled "1: bash".
- Output:**

```
rodrigobranas:javascriptmasterclass $ node --experimental-modules modules/modules_8.mjs  
(node:57657) ExperimentalWarning: The ESM module loader is experimental.  
SyntaxError: Unexpected token import  
    at ModuleJob.loaders.set [as moduleProvider] (internal/loader/ModuleRequest.js:33:13)  
        at <anonymous>  
rodrigobranas:javascriptmasterclass $ █
```