

 [Go back to the homepage \(https://flaviocopes.com/\)](https://flaviocopes.com/)

Introduction to ES Modules

ES Modules is the ECMAScript standard for working with modules. While Node.js has been using the CommonJS standard since years, the browser never had a module system, as every major decision such as a module system must be first standardized by ECMAScript and then implemented

Published Jun 13, 2018

- [Introduction to ES Modules \(https://flaviocopes.com/#introduction-to-es-modules\)](https://flaviocopes.com/#introduction-to-es-modules)
- [The ES Modules Syntax \(https://flaviocopes.com/#the-es-modules-syntax\)](https://flaviocopes.com/#the-es-modules-syntax)
- [Other import/export options \(https://flaviocopes.com/#other-importexport-options\)](https://flaviocopes.com/#other-importexport-options)
- [CORS \(https://flaviocopes.com/#cors\)](https://flaviocopes.com/#cors)
- [What about browsers that do not support modules? \(https://flaviocopes.com/#what-about-browsers-that-do-not-support-modules\)](https://flaviocopes.com/#what-about-browsers-that-do-not-support-modules)
- [Conclusion \(https://flaviocopes.com/#conclusion\)](https://flaviocopes.com/#conclusion)

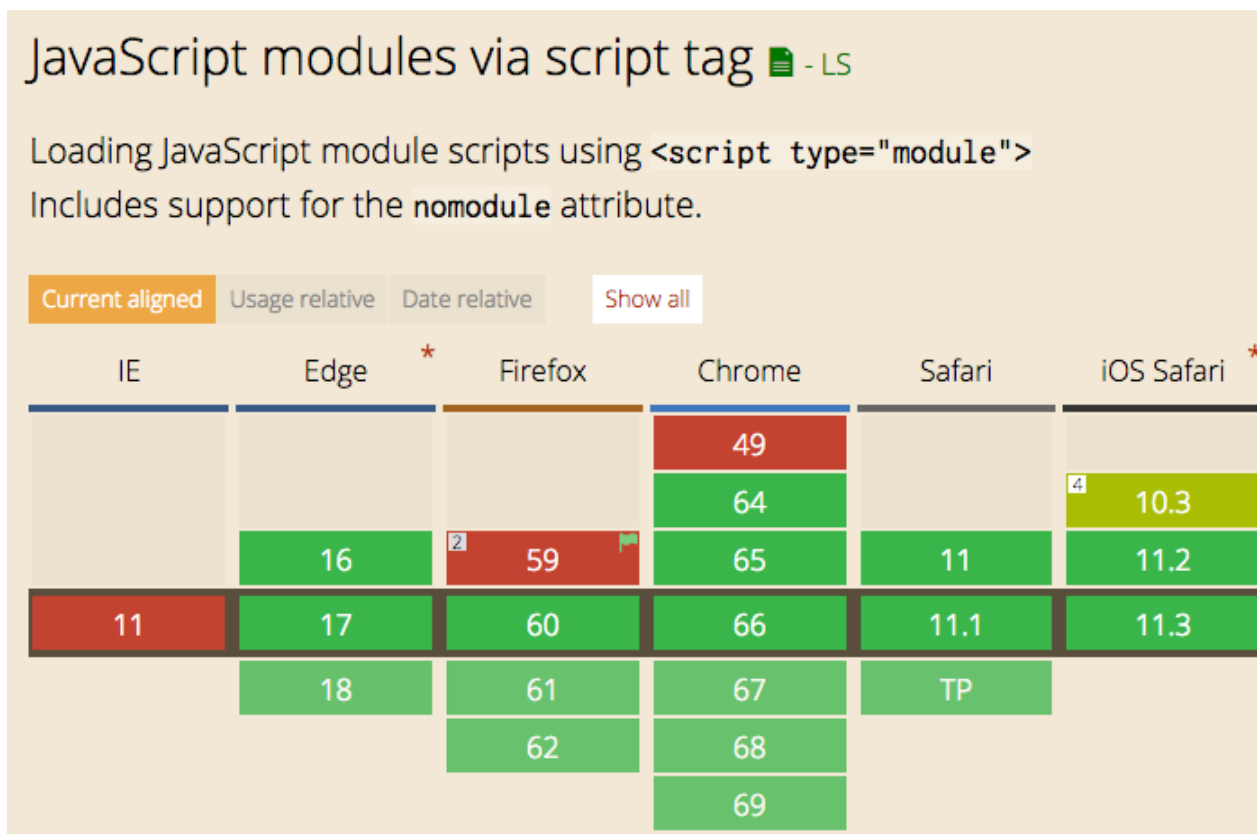
Introduction to ES Modules

ES Modules is the ECMAScript standard for working with modules.

While Node.js (<https://flaviocopes.com/nodejs/>) has been using the CommonJS standard for years, the browser never had a module system, as every major decision such as a module system must be first standardized by ECMAScript and then implemented by the browser.

This standardization process completed with ES6 (<https://flaviocopes.com/es6/>) and browsers started implementing this standard trying to keep everything well aligned, working all in the same way, and now ES Modules are supported in Chrome, Safari, Edge and Firefox (since version 60).

Modules are very cool, because they let you encapsulate all sorts of functionality, and expose this functionality to other JavaScript files, as libraries.



The ES Modules Syntax

The syntax to import a module is:

```
import package from 'module-name'
```

while CommonJS uses

```
const package = require('module-name')
```

A module is a JavaScript file that **exports** one or more values (objects, functions or variables), using the `export` keyword. For example, this module exports a function that returns a string uppercase:

uppercase.js

```
export default str => str.toUpperCase()
```

In this example, the module defines a single, **default export**, so it can be an anonymous function. Otherwise it would need a name to distinguish it from other exports.

Now, **any other JavaScript module** can import the functionality offered by `uppercase.js` by importing it.

An HTML page can add a module by using a `<script>` tag with the special `type="module"` attribute:

```
<script type="module" src="index.js"></script>
```

Note: this module import behaves like a `defer` script load. See [efficiently load JavaScript with defer and async \(https://flaviocopes.com/javascript-async-defer/\)](https://flaviocopes.com/javascript-async-defer/)

It's important to note that any script loaded with `type="module"` is loaded in strict mode (<https://flaviocopes.com/javascript-strict-mode/>).

In this example, the `uppercase.js` module defines a **default export**, so when we import it, we can assign it a name we prefer:

```
import toUpperCase from './uppercase.js'
```

and we can use it:

```
toUpperCase('test') //'TEST'
```

You can also use an absolute path for the module import, to reference modules defined on another domain:

```
import toUpperCase from 'https://flavio-es-modules-example.glitch.me/uppercase.js'
```

This is also valid import syntax:

```
import { toUpperCase } from '/uppercase.js'  
import { toUpperCase } from '../uppercase.js'
```

This is not:

```
import { toUpperCase } from 'uppercase.js'  
import { toUpperCase } from 'utils/uppercase.js'
```

It's either absolute, or has a `./` or `/` before the name.

Other import/export options

We saw this example above:

```
export default str => str.toUpperCase()
```

This creates one default export. In a file however you can export more than one thing, by using this syntax:

```
const a = 1
const b = 2
const c = 3

export { a, b, c }
```

Another module can import all those exports using

```
import * from 'module'
```

You can import just a few of those exports, using the destructuring assignment (<https://flaviocopes.com/es6/#destructuring-assignments>):

```
import { a } from 'module'
import { a, b } from 'module'
```

You can rename any import, for convenience, using `as` :

```
import { a, b as two } from 'module'
```

You can import the default export, and any non-default export by name, like in this common React import:

```
import React, { Component } from 'react'
```

You can see an ES Modules example here: <https://glitch.com/edit/#!/flavio-es-modules-example?path=index.html> (<https://glitch.com/edit/#!/flavio-es-modules-example?path=index.html>)

CORS

Modules are fetched using CORS (<https://flaviocopes.com/cors/>) . This means that if you reference scripts from other domains, they must have a valid CORS header that allows cross-site loading (like `Access-Control-Allow-Origin: *`)

What about browsers that do not support modules?

Use a combination of `type="module"` and `nomodule` :

```
<script type="module" src="module.js"></script>  
<script nomodule src="fallback.js"></script>
```

Conclusion

ES Modules are one of the biggest features introduced in modern browsers. They are part of ES6 but the road to implement them has been long.

We can now use them! But we must also remember that having more than a few modules is going to have a performance hit on our pages, as it's one more step that the browser must perform at runtime.

Webpack (<https://flaviocopes.com/webpack/>) is probably going to still be a huge player even if ES Modules land in the browser, but having such a feature directly built in the language is huge for a unification of how modules work client-side and on Node.js as well.

FREE EBOOKS 

JavaScript
(<https://flaviocopes.com/page/javascript-handbook>)

Python (<https://flaviocopes.com/page/python-handbook/>)

React (<https://flaviocopes.com/page/react-handbook/>)

HTML (<https://flaviocopes.com/page/html-handbook/>)

CSS (<https://flaviocopes.com/page/css-handbook/>)

Node.js (<https://flaviocopes.com/page/node-handbook/>)

Linux (<https://flaviocopes.com/page/linux-commands-handbook/>)

C (<https://flaviocopes.com/page/c-handbook/>)

Express.js (<https://flaviocopes.com/page/express-handbook/>)

Next.js (<https://flaviocopes.com/page/nextjs-handbook/>)

Vue.js (<https://flaviocopes.com/page/vue-handbook/>)

Svelte (<https://flaviocopes.com/page/svelte-handbook/>)

Deno (<https://flaviocopes.com/page/deno-handbook/>)

ES5 to ESNext (<https://flaviocopes.com/page/es5-to-esnext/>)

COURSES 

JavaScript Masterclass (<https://javascript.flaviocopes.com>)

React Masterclass (<https://react.flaviocopes.com>)

BOOTCAMPS

The Web Development Bootcamp
(<https://bootcamp.flaviocopes.com>)

The Advanced Web Development Bootcamp
(<https://advancedbootcamp.flaviocopes.com>)

You might also like

[How to write text into to an HTML canvas \(<https://flaviocopes.com/how-to-write-text-html-canvas/>\)](https://flaviocopes.com/how-to-write-text-html-canvas/)

[Wait for all promises to resolve in JavaScript \(<https://flaviocopes.com/javascript-resolve-all-promises/>\)](https://flaviocopes.com/javascript-resolve-all-promises/)

[The Object seal\(\) method \(<https://flaviocopes.com/javascript-object-seal/>\)](https://flaviocopes.com/javascript-object-seal/)

[How to get the value of a CSS property in JavaScript \(<https://flaviocopes.com/javascript-get-css-property-value/>\)](https://flaviocopes.com/javascript-get-css-property-value/)

[What's the difference between a method and a function? \(<https://flaviocopes.com/javascript-difference-method-function/>\)](https://flaviocopes.com/javascript-difference-method-function/)

[Gatsby, fix the "cannot find module gatsby-cli/lib/reporter" error \(<https://flaviocopes.com/gatsby-fix-lib-reporter-error/>\)](https://flaviocopes.com/gatsby-fix-lib-reporter-error/)

[The JavaScript Switch Conditional \(<https://flaviocopes.com/javascript-switch/>\)](https://flaviocopes.com/javascript-switch/)

[The Object freeze\(\) method \(<https://flaviocopes.com/javascript-object-freeze/>\)](https://flaviocopes.com/javascript-object-freeze/)

[How to join two arrays in JavaScript \(<https://flaviocopes.com/how-to-join-arrays-javascript/>\)](https://flaviocopes.com/how-to-join-arrays-javascript/)

How to deep clone a JavaScript object (<https://flaviocopes.com/how-to-clone-javascript-object/>)

© 2021 Flavio Copes (<https://flaviocopes.com/>)