



Somos un **ecosistema** de desarrolladores de software

Java Script POO (Object Oriented Programming)



```
<!-- BEGIN NAVIGATION  
">Home</a></li>  
.html">Home Events</a></li>  
enu.html">Multiple Column Men  
<a href="#" class="current"
```

```
utton-header.html">Tall But  
logo.html">Image Logo</a></  
href="tall-logo.html">Ta
```

```
f="#">Carousels</a>
```

```
th-slider.html">Variat  
lider.html">Testimoni
```

Module

Why Modules?

- Reusable piece of code that encapsulates implementation details
- Usually a standalone file, but it doesn't have to be.

WHY MODULES?

- ❖ Compose software: modules are small building blocks that we put together to build complex applications;
- ❖ Isolate components: modules can be developed in isolation without thinking about the entire codebase;
- ❖ Abstract code: implement low-level code in modules and import these abstractions into other modules;
- ❖ Organized code: modules naturally lead to a more organized codebase;
- ❖ Reuse code: modules allow us to easily reuse the same code, even across multiple projects.

Module

</Riwi>

IMPORT
(DEPENDENCY)

```
import { rand } from './math.js';  
const diceP1 = rand(1, 6, 2);  
const diceP2 = rand(1, 6, 2);  
const scores = { diceP1, diceP2 };  
export { scores };
```

EXPORT
(PUBLIC API)

Module

NATIVE JAVASCRIPT (ES6) MODULES

IMPORT &
EXPORT
SYNTAX

```
</Riwi>  — □ ×  
  
import { rand } from './math.js';  
const diceP1 = rand(1, 6, 2);  
const diceP2 = rand(1, 6, 2);  
const scores = { diceP1, diceP2 };  
export { scores };
```

Modules stored in
files, exactly one
module per file.

MODULE

Top-level variables Scoped to module

Default mode Strict mode HTML linking

	ES6 MODULE	SCRIPT
Top-level variables	Scoped to module	Global
Default mode	Strict mode	"Sloopy" mode
Top-level this	Undefined	Window
Imports and exports	YES	NO
HTML linking	<script type="module">	<SCRIPT>
File downloading	Asynchronous	Asynchronous

**</Be a
coder>**