



JONAS.IO
SCHMEDTMANN

THE COMPLETE JAVASCRIPT COURSE

FROM ZERO TO EXPERT!



@JONASSCHMEDTMAN

SECTION

MODERN JAVASCRIPT DEVELOPMENT:
MODULES AND TOOLING

LECTURE

AN OVERVIEW OF MODULES IN
JAVASCRIPT

JS

AN OVERVIEW OF MODULES

MODULE

- 👉 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.

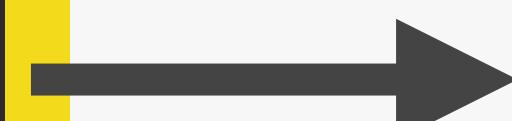
IMPORT
(DEPENDENCY)



MODULE

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

Module code



EXPORT
(PUBLIC API)

NATIVE JAVASCRIPT (ES6) MODULES

ES6 MODULES

Modules stored in files, **exactly one module per file.**

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

import and export syntax

ES6 MODULE

SCRIPT

👉 Top-level variables	Scoped to module	Global
👉 Default mode	Strict mode	“Sloppy” mode
👉 Top-level this	undefined	window
👉 Imports and exports	YES	NO
👉 HTML linking	<script type="module">	<script>
👉 File downloading	Asynchronous	Synchronous

👉 Need to happen at top-level
Imports are hoisted!

HOW ES6 MODULES ARE IMPORTED

