

Diferenças entre CommonJS e ES Modules

CommonJS (CJS) e ES Modules (ESM) são dois sistemas de módulos usados no JavaScript, mas têm diferenças importantes em funcionamento, sintaxe e uso.

Aspecto	CommonJS (CJS)	ES Modules (ESM)
Origem	Criado para o Node.js	Padrão oficial do JavaScript (ES2015)
Sintaxe	<code>`require()`</code> e <code>`module.exports`</code>	<code>`import`</code> e <code>`export`</code>
Carregamento	Síncrono	Assíncrono
Disponibilidade	Node.js e pacotes antigos	Browsers modernos e Node.js (\geq v12)
Escopo	Exporta um objeto único	Permite exportações nomeadas e default
Tree-shaking	Não suporta	Suporta
Compatibilidade	Legado	Recomendado em novos projetos
Extensão	.js	.js, .mjs ou via package.json

■ Quando usar cada um?

■ **Use CommonJS (CJS):** Em projetos Node.js legados que já estão estruturados com require. Ao usar bibliotecas antigas do NPM que só exportam em CJS. Quando precisa de carregamento síncrono (ex: scripts pequenos no Node).

■ **Use ES Modules (ESM):** Em projetos modernos (front-end e back-end). Se precisar de tree-shaking e otimização de bundlers. Para rodar direto em navegadores sem transpilar. Em Node.js moderno (com "type": "module" no package.json ou arquivos .mjs).

■ **Resumo:** CommonJS é o padrão histórico do Node.js, mas ES Modules é o padrão oficial da linguagem e recomendado hoje em dia.