



- Suportado nativamente em navegadores modernos e no Node.js (desde a versão 12+ com "type": "module" no package.json).
- Carregamento é **assíncrono** (melhor para browsers).
- Mais otimizado: permite **tree-shaking** (remover código não usado em build).
- Usa `export` e `import` .

◆ Diferenças principais

Aspecto	CommonJS	ES Modules
Sintaxe	<code>require</code> , <code>module.exports</code>	<code>import</code> , <code>export</code>
Execução	Síncrona	Assíncrona
Suporte	Node.js legado	Padrão oficial JS (Node.js e browsers)
Tree-shaking	✗ Não	✓ Sim
Configuração	Sem config extra no Node	Precisa "type": "module" no package.json ou .mjs

◆ Quando usar cada um?

- **Use CommonJS (CJS):**
 - Se está mantendo projetos antigos em Node.js.
 - Se precisa de compatibilidade com bibliotecas que ainda não migraram para ESM.
- **Use ES Modules (ESM):**
 - Para novos projetos (tanto em browser quanto Node.js).
 - Quando precisa de tree-shaking e builds otimizados.
 - Se quer seguir o padrão oficial e ter maior compatibilidade futura.



Pergunte alguma coisa



O ChatGPT pode cometer erros. Por isso, lembre-se de conferir informações relevantes.