

ES MODULES

`import e export`

ES Modules

- Sintaxe que entrou no JS no EcmaScript 6 (ES6)
- Disponibilizar qualquer tipo de dado entre arquivos
- Separar código em diversos arquivos
- Isolar escopo e código

Preparando o ambiente para usar ES Modules

- Numa nova pasta, vamos criar os arquivos:

```
index.html, main.js, utils.js
```

- Prepare o index.html

Crie a estrutura e adicione o `main.js` com a tag `<script>`

- Habilite a tag `<script>` para utilizar a sintaxe

adicionaremos o atributo `type="module"` à tag script

```
// para usar, precisamos adicionar o atributo  
// type="module" no nosso script principal  
<script src="main.js" type="module" ></script>
```

import e export simples

Podemos importar qualquer arquivo e ele será executado imediatamente, como uma função é executada.

```
// utils.js  
const message = "mensagem" // essa variável fica isolada nesse arquivo, e não consigo usar ela em outro lugar  
alert(message)  
  
// main.js  
import './utils.js' // nessa linha ele importa e já executa o arquivo
```

default import e export

Exportamos qualquer tipo de dado e, ao importar, podemos dar qualquer nome.

```
// utils.js
export default 'algum dado'

// ou, para um dado já existente
const text = 'alguma coisa'
export default text

// main.js
import qualquerNome from './utils.js'
```

Somente 1 `default` por arquivo.

named export

Exportamos uma variável ou função e, ao importar, devemos colocar o mesmo nome.

```
// utils.js
export const today = Date.now()

// exportando um valor já existente
const double = number => number * 2
export { double }

// exportando diversos
// export { today, double }

// main.js
import { today } from './utils.js'
```

É possível ter múltiplos exports

Misto (named e default)

Poderemos usar um default export por arquivo e um (ou mais) named export.

```
//utils.js
const date = new Date()
const greeting = name => `hello ${name}`

export default function (argument) {
  console.log(date, greeting(argument))
}

export { date, greeting }

// main.js
import funcaoPadrao, { date, greeting } from './utils.js'
```

import as

É possível renomear um ou mais dados que foram exportados.

```
// utils.js
export function sum(a, b) {
  return a + b
}

// main.js
import { sum as soma } from './utils.js'
```


import * as

Podemos importar todos os named exports e dar um nome padrão.

```
//utils.js

export const sum = (a, b) => a + b
export const multiply = (a, b) => a * b
export const subtract = (a, b) => a - b
export const divide = (a, b) => a / b

// ou
// export { sum, multiply, subtract, divide }

//main.js
import * as calculator from './utils.js'
```

Referência

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/export>