{JS}

Clase 19

‹Índice›

Prototipos

¿Qué es un módulo?

A partir de ECMAScript se introduce una característica nativa denominada Módulos ES (ESM), que permite la importación y exportación de fragmentos de datos entre diferentes ficheros Javascript, eliminando las desventajas que teníamos hasta ahora y permitiendo trabajar de forma más flexible en nuestro código Javascript.

¿Qué problema resuelve?

Uno de los principales problemas que ha ido arrastrando Javascript desde sus inicios es la dificultad de organizar de una forma adecuada una aplicación grande, con muchas líneas de código. En muchos lenguajes de programación, cuando un programa crece, se comienza a estructurar en funciones. Posteriormente, se traslada a clases, que contienen variables (propiedades) y funciones (métodos). De esta forma organizamos de forma más lógica el código de nuestro programa. Sin embargo, no será suficiente.

Import y Export

Declaración	Descripción
`export`	Pone los datos indicados (variables, funciones, clases...) a disposición de **otros ficheros**
`import`	Incorpora datos (variables, funciones, clases...) desde otros ficheros `.js` al código actual.

Antes de usar módulos

```
<script type="module">
  import { nombre } from "./file.js";
</script>
```

Nuestro primer export

Forma	Descripción
`export ...`	Declara un elemento o dato, a la vez que lo añade al módulo de exportación.
`export { name }`	Añade el elemento `name` al módulo de exportación.
`export { name as newName }`	Añade el elemento `name` al módulo de exportación con el nombre `newName`.
`export { n1, n2, n3... }`	Añade los elementos indicados (`n1` , `n2` , `n3` ...) al módulo de exportación.
`export * from "./file.js"`	Añade todos los elementos del módulo de `file.js` al módulo de exportación.
`export default ...`	Declara un elemento y lo añade como módulo de exportación **por defecto**.

Nuestro primer export

```
let number = 42;
const hello = () => "Hello!";
const goodbye = () => "¡Adiós!";
class CodeBlock { };

export { number };                    // Se crea un módulo y se añade number
export { hello, goodbye as bye };     // Se añade saludar y despedir al módulo
export { hello as greet };            // Se añade otroNombre al módulo
```

Nuestro primer export

```
let number = 42;
const hello = () => "Hello!";
const goodbye = () => "¡Adiós!";
class CodeBlock { };

export {
  number,
  hello,
  goodbye as bye,
  hello as greet
};
```

Nuestro primer import

Forma	Descripción
`import { nombre } from "./file.js"`	Importa el elemento `nombre` de `file.js`.
`import { nombre as newName } from "./file.js"`	Importa el elemento `nombre` de `file.js` como `newName`.
`import { n1, n2... } from "./file.js"`	Importa los elementos indicados desde `file.js`.
`import nombre from "./file.js"`	Importa el elemento **por defecto** de `file.js` como `nombre`.
`import * as name from "./file.js"`	Importa todos los elementos de `file.js` en el objeto `name`.
`import "./file.js"`	Ejecuta el código de `file.js`. No importa ningún elemento.
`import { name } from "https://web.com/file.js"`	Descarga el fichero e importa el elemento `name` de su módulo.

Nuestro primer import

```
import { nombre } from "./file.js";
import { number, element } from "./file.js";
import { brand as brandName } from "./file.js";
```

Nuestro primer import

```
import nombre from "./math.js";
```

Probemos con Vite

‹Despedida›

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez