Módulos, exportaciones e importaciones

{desafío} latam_

Módulos

Los programas JavaScript comenzaron siendo bastante pequeños — la mayor parte de su uso en los primeros días era para realizar tareas de scripting aisladas, proporcionando un poco de interactividad a tus páginas web donde fuera necesario, por lo que generalmente no se necesitaban grandes scripts. Avancemos unos años y ahora tenemos aplicaciones completas que se ejecutan en navegadores con mucho JavaScript, JavaScript ahora se usa en otros contextos (Node.js, por ejemplo).

Por lo tanto, en los últimos años se ha comenzado a pensar en proporcionar mecanismos para dividir programas JavaScript en módulos separados que se puedan importar cuando sea necesario. Node.js ha tenido esta capacidad durante mucho tiempo, y hay una serie de bibliotecas y marcos de JavaScript que permiten la utilización de módulos (por ejemplo, CommonJS y AMD otros basados en sistemas de módulos como RequireJS, y recientemente Webpack y Babel).

Fuente: https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Modules

Módulos

¿Qué es un módulo ES (ESM)?

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.

Fuente: https://lenguajejs.com/javascript/modulos/que-es-esm/

Módulos

¿Qué es un módulo ES (ESM)?

Para trabajar con módulos tenemos a nuestra disposición las siguientes palabras clave:

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.
import()	Permite importar módulos de forma más flexible, en tiempo real (imports dinámicos).

Antes de usar módulos

Debes saber que para poder utilizar export o import en nuestro código Javascript que se ejecuta directamente en el navegador, debemos cargar el fichero .js con la etiqueta y atributo <script type="module"> para indicarle que utilizaremos módulos. Si no lo hacemos, obtendremos el siguiente error:

> Uncaught SyntaxError: Cannot use import statement outside a module

Al añadir el atributo type="module" a nuestra etiqueta <script> estaremos avisando al navegador que estamos cargando un módulo en el que podemos utilizar import y export:

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

Importaciones y Exportaciones

`import` y `export` proveen un manejo más avanzado de módulos en JavaScript. Es posible importar/exportar funciones, objetos o valores primitivos. Existen 2 maneras de hacerlo: nombradas y por defecto.

Fuente: https://medium.com/@simonhoyos/ecmascript6-y-react-e3fe8150adb8

Importaciones y Exportaciones

Nombradas

Exportaciones/importaciones nombradas: son muy buenas para exportar/importar varios valores al mismo tiempo. Es necesario usar el mismo nombre para exportar como para importar.

```
export { functionOne, objectTwo, valueThree }; // inside app.js
```

```
import { functionOne, objectTwo, valueThree } from 'app'; // inside other.js
```

Importaciones y Exportaciones

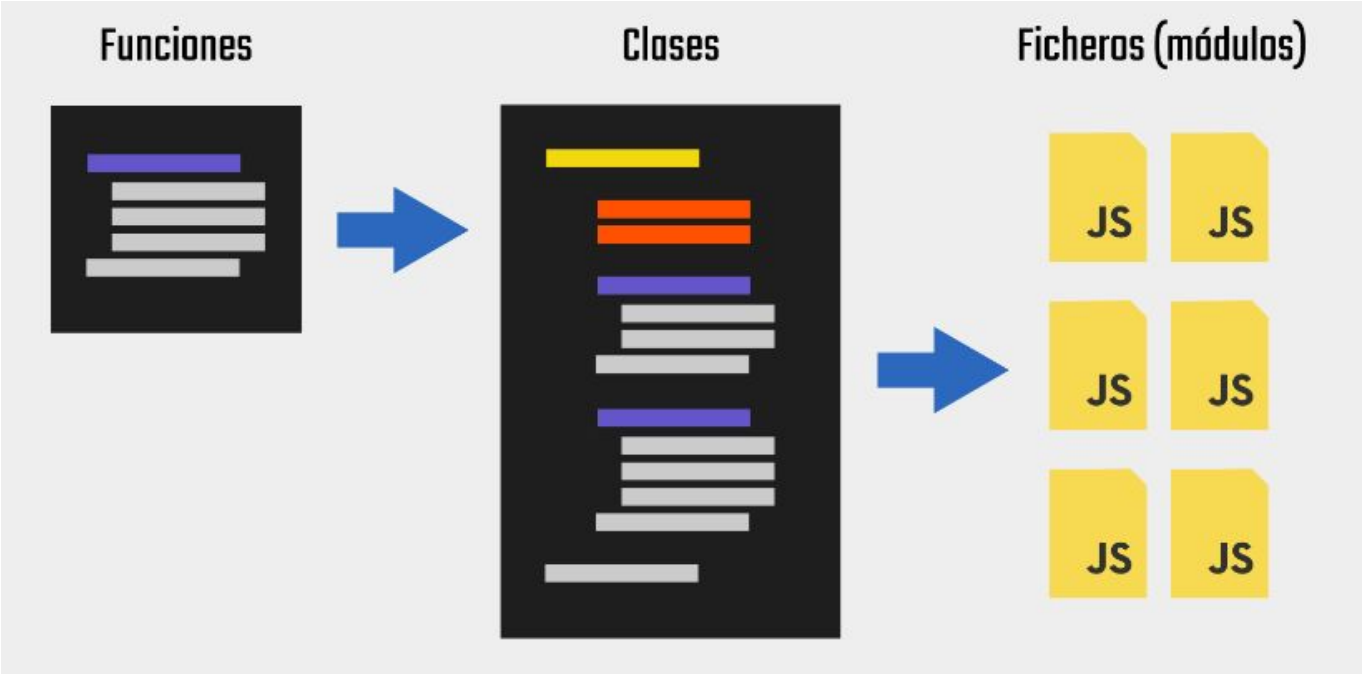
Por defecto

Exportaciones por defecto: solo puede haber una por módulo. Se puede importar con cualquier nombre, no tiene que ser el mismo que se usó para la exportación.

```
export default ClassDefault; // inside app.js
```

```
import Class from 'app'; // inside other.js
```

Si quieres saber más, puedes visitar los enlaces dejados en esta presentación y ver el video adjunto en el material complementario.



{desafío} latam_

{desafío} latam_

Academia de talentos digitales