

Los módulos son la forma de cargar las distintas clases y funciones de nuestro proyecto en el lugar que lo necesitemos. Los ficheros de nuestro proyecto *exportan* las funciones, clases o constantes que otros *importaran*.

Antes de EcmaScript 2015, la manera de exportar e importar módulos era mediante el uso de `require` y `module.exports`. Por ejemplo, vamos a declarar una función `hello` y la vamos a exportar:

```
// hello.js
hello = text => 'Hello ' + text;

// Exportamos la funcion
module.exports = hello;
```

Para utilizar esta función en `app.js` la importamos con `require`:

```
// app.js
var hello = require('./hello');

hello('World');
```

Si quisiéramos exportar varios elementos, necesitaríamos declarar un objeto en como elemento a exportar.

```
// hello.js
hello = text => 'Hello ' + text;
helloUppercase = text => hello(text).toUpperCase();

module.exports = {
  hello, helloUppercase
}
```

## EcmaScript 2015

En ES2015 esta tarea se simplifica gracias a la adición de `import` y `export`. Con `export` podemos exportar tantos elementos como necesitemos.

```
// hello.js
export hello = text => 'Hello ' + text;
export helloUppercase = text => hello(text).toUpperCase();
```

`import` nos permite importar una de las funciones o todas:

```
// app.js
import { hello, helloUppercase } from './hello';

hello('World');
```

También podemos declarar estas funciones como un objeto.

```
// app.js
import * as HelloUtils from './hello';

HelloUtils.hello('World');
```

**Otra ventaja de esta sintaxis es que podemos exportar una función por defecto.** Si el fichero que importa las funciones no especifica ninguna (mediante el uso de `{}`), recibirá la que se exporta por defecto.

```
// hello.js

// Exportamos hello por defecto mediante "default"
export default hello = text => 'Hello ' + text;
export helloUppercase = text => hello(text).toUpperCase();
```

En `app.js`:

```
// app.js
import hello from './hello';

hello('World');
```

Si necesitamos importar todas las funciones, al exportar una por defecto tenemos que combinar los ejemplos anteriores:

```
// app.js
import hello, { helloUppercase } from './hello';

hello('World');
helloUppercase('World');
```