

Módulos Javascript

En sus inicios, un programa en Javascript no tenía manera de cargar código de otros archivos.

O colocabamos todo el código en un solo archivo js y lo cargabamos en el html.

O creabamos varios archivos js y los cargabamos en el html en varias etiquetas `<script>`

Los sistemas de módulos surgen de la necesidad de encapsular funcionalidad de Javascript (funciones) y crear bloques de código que podamos reutilizar.

Sistemas de módulos en Javascript

A lo largo de la historia de Javascript han surgido diferentes especificaciones para importación y exportación de módulos, con algunas diferencias entre si.

Sistema de módulos	Implementación
Patrón Módulo Revelador	Patrón de diseño
CommonJS	NodeJS/npm
Asynchronous Module Definition	Require.js
Módulos Nativos de Javascript	ES6/ES2015

Módulos preincluidos en Node

Node ya viene con módulos incluidos que proveen cierta funcionalidad y que no es necesario instalar (`npm install`):

Módulo	Descripción
<code>fs</code>	Trabajar con el sistema de archivos
<code>path</code>	Trabajar con rutas
<code>url</code>	Resolver y parsear URLs
<code>http</code>	Crear un servidor http
<code>events</code>	Manejo de eventos

Para poder usarlos en nuestras aplicaciones solo debemos importarlos (requerirlos) con la función `require()` que nos devolverá un objeto con las funciones y propiedades que el módulo esté exportando. Asignamos ese objeto a una variable y accedemos a las funciones y propiedades del módulo con la notación de punto.

```
var fs = require('fs'); // Importamos y asignamos a una variable
fs.writeFile('archivo.txt', 'Hola Mundo', function (err) { // Usamos las funciones del modulo
  if (err) { console.log(err); }
  else { console.log('Saved!'); }
});
```

```
var http = require('http'); //Importamos y asignamos a una variable
var servidor = http.createServer(function(req, res){...}); //Usamos sus funciones
```

module.exports

En NodeJS cada archivo Javascript posee un objeto especial llamado `module.exports`.

module: Es una variable que representa el módulo actual.

exports: Es el objeto que será expuesto.

Los módulos que creamos los podemos "empaquetar" (`$ npm init`)
y publicar en el registro npm (`$ npm publish`)

Módulos CommonJS (**exports/require**)

Módulo Exportador:

```
exports.cuadrado = (x) => x * x;
```

Importando el módulo en otro programa:

```
const cuadrado = require('./cuadrado.js');  
console.log(cuadrado(2));
```

Módulos ECMAScript2015 (**export/import**)

export : Crea un módulo de exportación que contiene constantes y funciones.

import : Lee un módulo para utilizar lo que exporta.

Módulo Exportador:

```
export function cuadrado(x) {  
  return x * x;  
}
```

Importando el módulo en otro programa:

```
import { cuadrado } from 'cuadrado.js';  
console.log(cuadrado(2));
```