

ECMAScript Modules Cheat Sheet

This cheat sheet introduces ECMAScript modules (ESM), an alternative to CommonJS modules for sharing functions in JavaScript programs. ESM are the future of modules in JavaScript. ESM are already the standard format for front-end development, and now can be used on the back end with recent versions of Node.js. Although the old module system (CommonJS) has worked for many years, ESM come with many benefits such as asynchronous loading, top-level await, and improved static analysis.

EXPORTING A MODULE

The following code exports the module in the funTimes.mjs file:

```
function funTimes() {  
  return 'Fun'  
}
```

```
export { funTimes };
```

IMPORTING A FUNCTION

The following code imports and runs the funTimes function from the funTimes.mjs file:

```
import { funTimes } from './funTimes.mjs'
```

```
funTimes();
```

ENABLING ESM

Because Node.js treats code as CommonJS modules by default, module authors will need to tell Node.js to treat the code as ECMAScript Modules by either using the .mjs file extension (as shown in earlier examples) or by using the type field in the package.json file:

```
// package.json
```

```
{  
  name: "funTimes",  
  type: "module"  
}
```

DEFAULT EXPORTS

A default export allows a module to provide a default function:

```
// funTimes.mjs
```

```
function funTimes() {  
  return 'Fun'  
}
```

```
export default funTimes;
```

Importing this default function could look something like this:

```
// app.mjs
```

```
import funTimes from './funTimes.mjs'
```

```
funTimes();
```