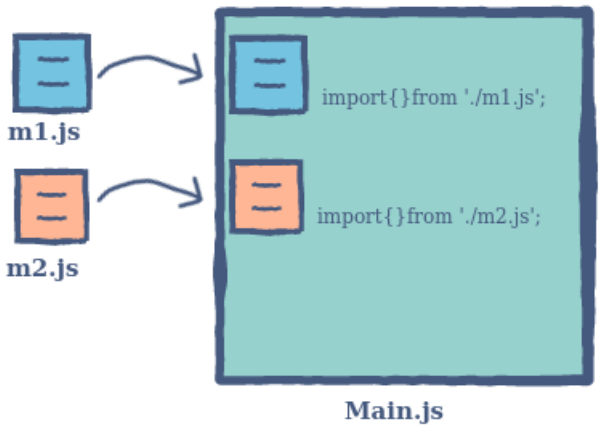
Module

In Node.Js, files are called modules. Modularity is a technique where one program has separate parts each providing a single piece of the overall functionality - like pieces of a puzzle coming together to complete a picture.

A Module is nothing more than a chunk of JavaScript code written in a file. The variables, functions in a module are not available for use unless they are exported from one module to another.

The **`export` statement** is used in Javascript modules to export functions, objects, or primitive values from one module so that they can be used in other files.

Module exports are the instructions that tell Node.js which bits of code (functions, objects, strings, etc.) to **`export`** from a given file so that other files are allowed to access the exported code.



Syntax

Let's say you have a file called **`Log.js`** . The file has a function called **method1**. If you want to use this function in another file, you have to assign this function to `module.exports` as shown below:

```
function method1 (msg) {
    console.log(msg);
};
module.exports = method1;
```

`require()` is a function which is used to bring one module into another. If you want to use the above function in another file, you can do it by using **`require`** keyword and module name (`Log.js`) and assign it to a new variable as shown below:

```
var msg = require('./Log.js');
msg('Hello World');
// Output
// Hello World
```

You can also use **`import`** keyword in place of `require()` and do the same task as shown below.

`Note:` **You don't have to use it for now. As it will be explained to you further in detail** .

```
import greetings from "./Log.js";
greetings('Hello World');
// Output
// Hello World
```