THE COMMONJS MODULE SYSTEM

- Each JavaScript file is treated as a separate module;
- Node.js uses the **CommonJS module system**: `require()`, `exports` or `module.exports`;
- **ES module system**: `import/export`;

```
// math.js
function add(x, y) {
  return x + y;
}

exports.add = add;
```

```
// index.js
const math = require('./math.js');

console.log(math.add(2, 3)); // outputs 5
```

Where does it come from?

WHAT HAPPENS WHEN WE REQUIRE() A MODULE

```
require('test-module');
```

RESOLVING & LOADING → WRAPPING → EXECUTION → RETURNING EXPORTS → CACHING

👉 **Core modules**

```
require('http');
```

👉 **Developer modules**

```
require('./lib/controller');
```

👉 **3rd-party modules (from NPM)**

```
require('express');
```

PATH RESOLVING: HOW NODE DECIDES WHICH MODULE TO LOAD

1️⃣ Start with **core modules**;

2️⃣ If begins with '`./`' or '`../`' 👉 Try to **load developer module**;

3️⃣ If no file found 👉 Try to **find folder** with `index.js` in it;

4️⃣ Else 👉 Go to **`node_modules/`** and try to find module there.

WHAT HAPPENS WHEN WE REQUIRE() A MODULE

```
require('test-module');
```

RESOLVING & LOADING → WRAPPING → EXECUTION → RETURNING EXPORTS → CACHING

Where does it come from?

```
(function (exports, require, module, __filename, __dirname) {
  // Module code lives here ...
});
```

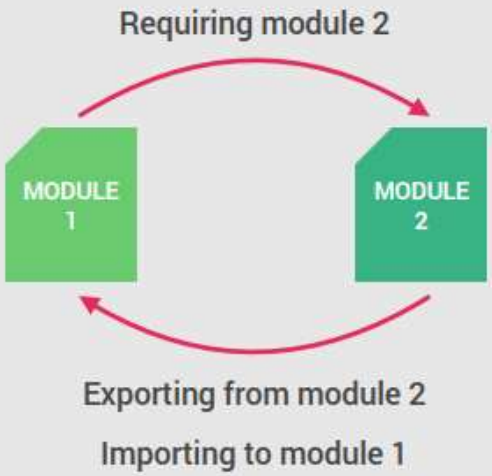
- **exports**: a reference to `module.exports`, used to export object from a module;
- **require**: function to require modules;
- **module**: reference to the current module;
- **__filename**: absolute path of the current module's file;
- **__dirname**: directory name of the current module.

WHAT HAPPENS WHEN WE REQUIRE() A MODULE

```
require('test-module');
```

RESOLVING & LOADING → WRAPPING → EXECUTION → RETURNING EXPORTS → CACHING

- `require` function returns **exports** of the required module;
- `module.exports` is the returned object (important!);
- Use `module.exports` to export one single variable, e.g. one class or one function (`module.exports = Calculator`);
- Use `exports` to export multiple named variables (`exports.add = (a, b) => a + b`);
- This is how we import data from one module into another;



In ES Modules = export default

In ES Modules = export

WHAT HAPPENS WHEN WE REQUIRE() A MODULE

```
require('test-module');
```

ES MODULES