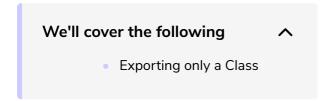# Exporting a Class or Object

This lesson teaches how to export a class or object in Node.js.

Numerous modules in the Node.js ecosystem export only a single object aggregating all of the module's functionality. To do so, they reassign the `module.exports` object instead of adding properties to it. For example, check out how the following module `calculator.js` is defined.

```javascript
1   // Declare a factory function that returns an object literal
2   const createCalc = () => {
3     // The returned object has 4 methods
4     return {
5       add(x, y) {
6         return x + y;
7       },
8       substract(x, y) {
9         return x - y;
10      },
11      multiply(x, y) {
12        return x * y;
13      },
14      divide(x, y) {
15        return x / y;
16      }
17    };
18  };
19
20  // Export the factory function
21  module.exports = createCalc;
```

In this module, the only exported element is a function that returns an object literal. Using it in another file (located in the same folder) is as follows

```
1   const calculator = require("./calculator.js");
2
3   // Create an object by calling the exported function of this module
4   const calc = calculator();
5
6   // Use the object's methods
7   console.log(`2 + 3 = ${calc.add(2, 3)}`); // "2 + 3 = 5"
```

The result of the call to `require()` is a function stored in the calculator variable, referencing the `createCalc()` function. Calling this function returns an object with several methods, which can be subsequently used.

Exporting only a Class

When you want a module to only export a specific class, you can also reassign the `module.exports` object.

Here is a module `user.js` that defines and exports a `User` class.

```
// Export a User class
module.exports = class User {
  constructor(firstName, lastName) {
    this.firstName = firstName;
    this.lastName = lastName;
    // Create user login by combining first letter of first name + last name
    this.login = (firstName[0] + lastName).toLowerCase();
  }
  describe() {
    return `${this.firstName} ${this.lastName} (login: ${this.login})`;
  }
};
```

Here's how to use this class in another file (located in the same folder).

```
// Notice the first uppercase letter, since User is a class
const User = require("./user.js");

// Create an object from this class
const johnDoe = new User("John", "Doe");

// Use the created object
console.log(johnDoe.describe());
```