Modules

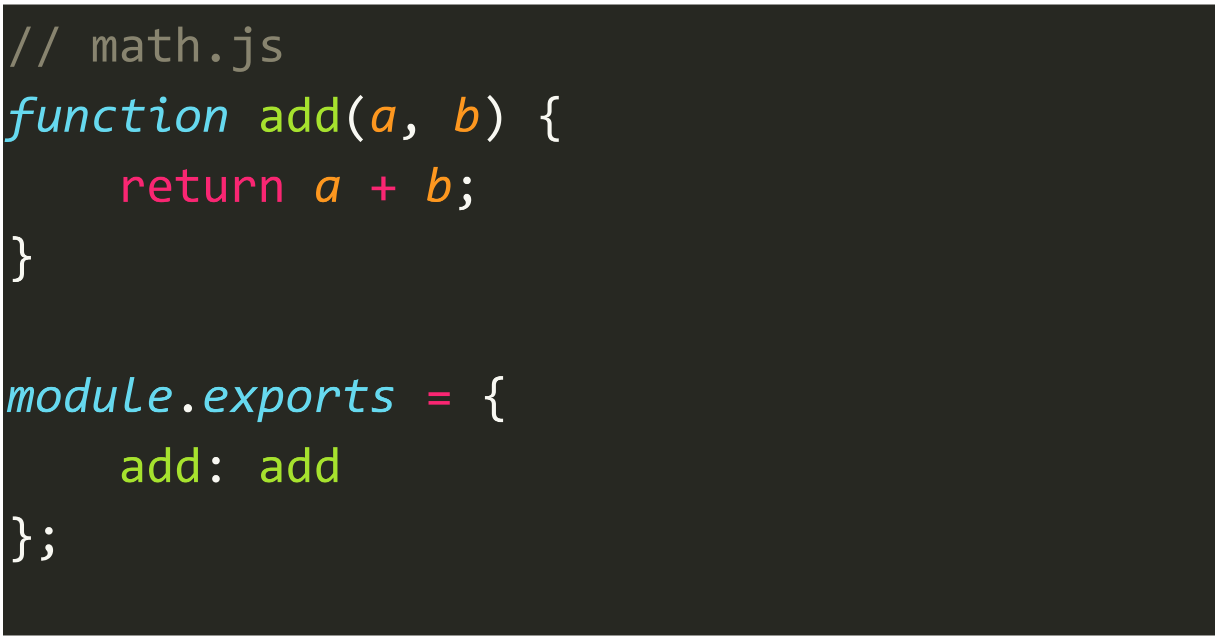
In JavaScript, modules are a way to organize code into separate files, each containing related functionality. This modular approach helps in better structuring, maintaining, and reusing code within a project. Modules allow you to encapsulate variables, functions, and classes, making them accessible only where needed and preventing unintended global scope pollution.

There are two primary standards for JavaScript modules:

CommonJS:

CommonJS is a module format used primarily in server-side JavaScript environments, like Node.js. It uses require() to import modules and module.exports or exports to export functionality.

Example of exporting in CommonJS:



```
// math.js
function add(a, b) {
    return a + b;
}

module.exports = {
    add: add
};
```

Example of importing in CommonJS:

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

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

ES Modules (ESM):

ES Modules are the standardized module system for JavaScript introduced in ECMAScript 6 (ES6). They use import and export statements to handle module dependencies.

Example of exporting in ES Modules:

```
// utlity.js
export function add(a, b) {
    return a + b;
}
```

Example of importing in ES Modules:

```
// script.js
import { add } from "./utlity.js";

console.log(add(2, 3)); // Output: 5
```

ES Modules have become increasingly popular due to their native support in modern browsers and in Node.js with appropriate configurations.

Modules provide benefits such as:

- Encapsulation: Variables and functions within modules are scoped, preventing unintended global variable access.
- Reusability: Modules can be imported into multiple files, promoting code reuse.
- Maintainability: Code is organized into smaller, manageable units, making it easier to understand and maintain.
- Dependency management: Modules explicitly define their dependencies, making it easier to track and manage dependencies between different parts of the codebase.
- Overall, modules are an essential feature in modern JavaScript development, promoting code organization, reusability, and maintainability.