Beginner's Guide to Node Fundamentals

Introduction

Node.js is a popular JavaScript runtime environment that allows you to run JavaScript code outside of a web browser. It's commonly used for building server-side applications, command-line tools, and more. In this beginner's guide, we'll explore some of the fundamental concepts of Node.js, including require, exports, modules, events, and event emitters.

Prerequisites

Before we get started, make sure you have the following:

- Node.js and npm installed
- Basic knowledge of JavaScript

Step 1: Getting Started with Node.js

Before we dive into Node.js fundamentals, make sure you have Node.js installed on your computer. You can download it from the official Node.js website: nodejs.org.

To check if Node.js is installed, open your terminal or command prompt and run:

```
node -v
```

This command should print the installed Node.js version.

2. Modules in Node.js

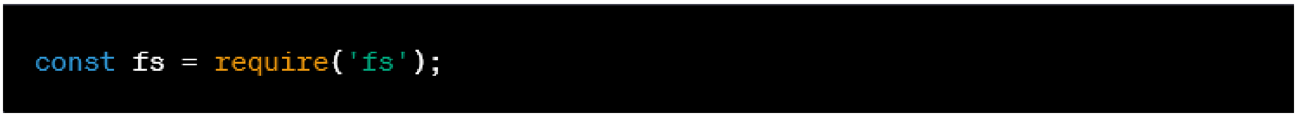
In Node.js, code is organized into modules. Modules are reusable blocks of code that can be loaded into other parts of your application. Node.js provides a module system to help manage code separation and reusability.

3. Using require and exports

`require`: Importing Modules

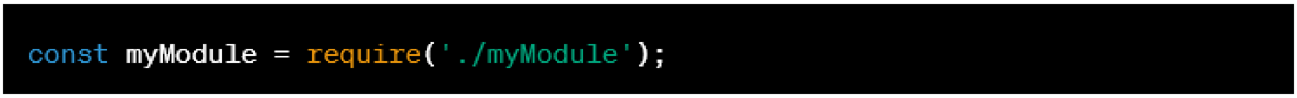
The require function is used to import modules in Node.js. You can require built-in modules or your own custom modules.

Example of requiring a built-in module (in this case, the **fs module** for file system operations):



```
const fs = require('fs');
```

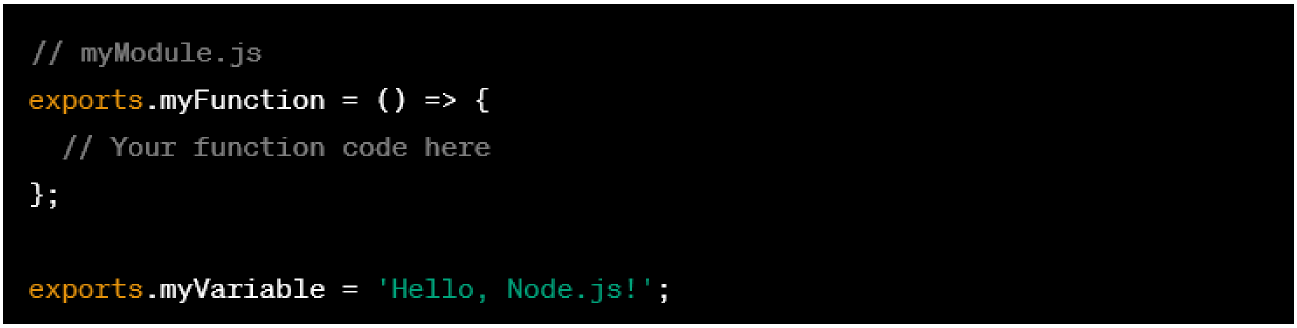
Example of requiring a custom module (**myModule.js** located in the same directory as your script):



```
const myModule = require('./myModule');
```

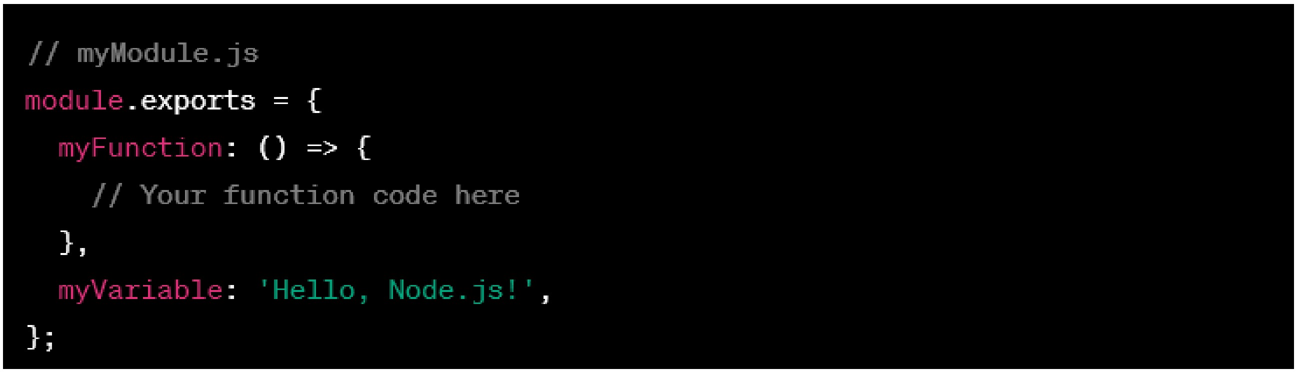
`exports`: Exporting Modules

To make functions, variables, or objects available for use in other modules, you can use the exports object. In your custom module (myModule.js), you can export functions or objects like this:



```
// myModule.js
exports.myFunction = () => {
  // Your function code here
};

exports.myVariable = 'Hello, Node.js!';
```

You can also use the **module.exports** approach to export a single object or function:

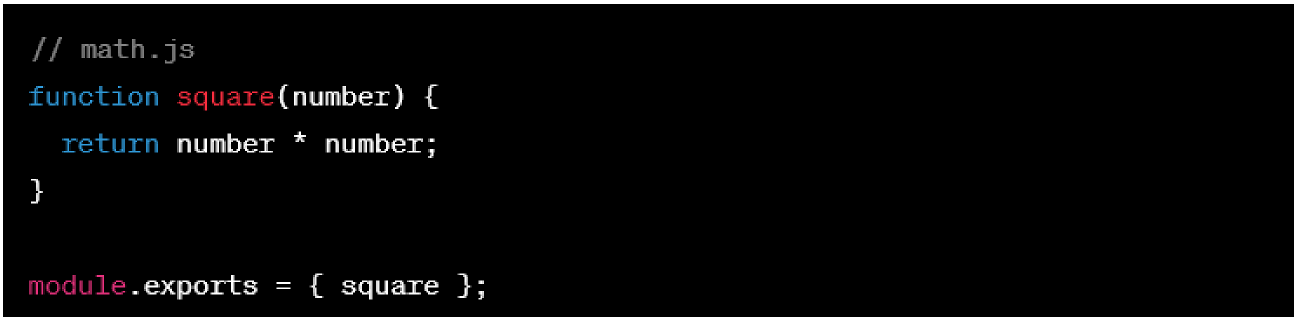


```
// myModule.js
module.exports = {
  myFunction: () => {
    // Your function code here
  },
  myVariable: 'Hello, Node.js!',
};
```

4. Creating a Custom Module

In Node.js, you can break your code into reusable modules. Let's create a simple module that provides a function to calculate the square of a number.

Create a new file named **math.js**:



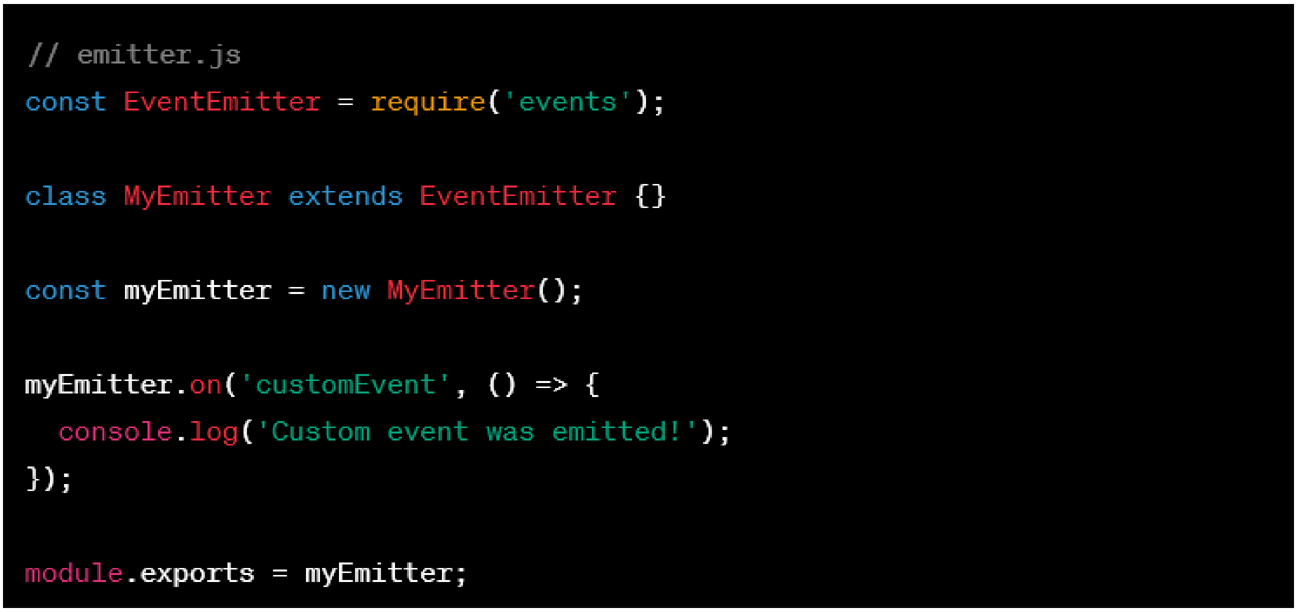
```
// math.js
function square(number) {
  return number * number;
}

module.exports = { square };
```

In this module, we define a square function and export it using **module.exports**.

5. Creating a Custom Event Emitter

Node.js includes the built-in events module, which allows you to work with events and event emitters. Let's create a custom event emitter that emits a custom event.

Create a new file named **emitter.js**:



```
// emitter.js
const EventEmitter = require('events');

class MyEmitter extends EventEmitter {}

const myEmitter = new MyEmitter();

myEmitter.on('customEvent', () => {
  console.log('Custom event was emitted!');
});

module.exports = myEmitter;
```

In this module, we create a custom event emitter class MyEmitter, create an instance of it, and define an event listener for the customEvent event.

6. Importing and Using Modules

Now, let's create an app.js file where we import the custom module and custom event emitter:

```
// app.js
const math = require('./math'); // Import the custom module
const myEmitter = require('./emitter'); // Import the custom event emitter

// Use the custom module
const result = math.square(5);
console.log(`Square of 5 is: ${result}`);

// Emit the custom event
myEmitter.emit('customEvent');
```

In this app.js file, we import the math module and the myEmitter event emitter. We then use the square function from the math module to calculate the square of 5 and emit the customEvent using the event emitter.

7. Running the Application

Open your terminal and navigate to the directory where your app.js file is located. Run the following command:

```
node app.js
```

You should see the following output:

```
Square of 5 is: 25
Custom event was emitted!
```

This confirms that your custom module and event emitter are working as expected.

Conclusion

Congratulations! You've now learned the basics of working with modules and custom event emitters in Node.js. These concepts are fundamental to building Node.js applications and can help you create organized and efficient code.