`02-Node-Module-System/README.md`

📚 Chapter 2: Node Module System

📋 Course Rules Reminder

🎯 Important Guidelines

✅ **Attendance** - Not mandatory but highly recommended
❌ **Remote troubleshooting** - Very difficult, attend class for best support

💬 Questions

- **During class** - Always welcome! Ask immediately
- **After class** - Use the Toledo forum
- **Stuck?** - Independent research (Google, Stack Overflow, AI tools)

📝 Labs

✅ **Complete all GitHub Classroom Labs** - Essential for understanding

---

🎯 Chapter Overview

Understanding the Node Module System

Learn how to organize your code into reusable modules

🤔 Key Questions

This chapter answers:

- 🔍 **What are modules?** - Building blocks of Node applications
- 🎯 **Why do we need them?** - Code organization and reusability
- 💡 **How do we use them?** - `require()` and `module.exports`
- 🛠️ **How do we create them?** - Building your own modules

📚 Modules We've Seen

You've already encountered some built-in modules:

Module	Purpose
os	Operating system information
fs	File system operations
events	Event handling
http	Create web servers

🎯 What You'll Learn

🌍 Global Objects

- Browser vs Node.js globals
- Understanding scope
- The `global` object

📦 Module System

- What is a module?
- Module object
- Creating modules

🔧 Built-in Modules

- Path module
- OS module
- File System module

⚡ Events

- EventEmitter
- Event arguments
- Creating event-driven code

📖 Chapter Slides

1. **Global Objects** - Understanding global scope in Node.js
2. **Module System Basics** - What modules are and why we need them
3. **Creating Custom Modules** - Build and export your own modules
4. **Built-in Modules** - Path, OS, and File System

5. **Events & EventEmitter** - Working with events
6. **HTTP Module** - Creating web servers

💡 Key Concepts

Module Benefits

✅ **Encapsulation** - Keep variables and functions private
✅ **Reusability** - Use the same code in multiple places
✅ **Maintainability** - Easier to update and debug
✅ **Organization** - Logical code structure

🚀 Let's Get Started!

Ready to master the Node.js module system?

Start with Global Objects →