**LÝ THUYẾT CƠ BẢN VỀ NODEJS**

I – CÀI ĐẶT NODEJS

**1. Cài nodejs**

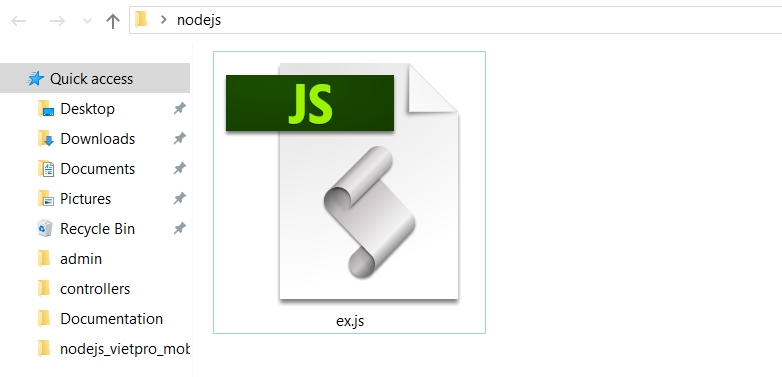
* Download bản mới nhất của nodejs trên trang chủ <https://nodejs.org/en/>

**2. Cài Git**

* Mục tiêu của Git trong khóa học này, chúng ta chỉ sử dụng để thay thế cho Command Prompt mà thôi.
* Download bản mới nhất của Git trên trang chủ <https://git-scm.com/>

II – CHẠY CHƯƠNG TRÌNH ĐẦU TIÊN

**1. Tạo cấu trúc thư mục**



**2. Viết một đoạn code js đơn giản trong file ex.js**

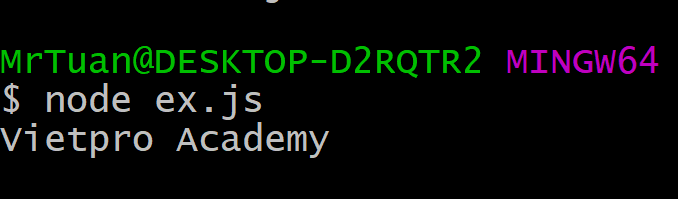
|  |
| --- |
| console.log('Vietpro Academy'); |

**3. Thực thi file ex.js bằng nodejs**

* B.1 Right click chuột vào thư mục nodejs chưa file ex.js và chọn **Git Bash Here**
* B2. Gõ đoạn lệnh sau và thực thi bằng cách Enter

|  |
| --- |
| node ex.js |

* Kết quả:



III – NODE MODULE

**1. Khái niệm**

*Trong nodejs có 3 kiểu module, một là module được tích hợp sẵn trong nodejs, hai là module chúng ta tải về qua npm, ba là module do chúng ta tự viết riêng sau đó sử dụng lại (nó gần giống với khái niệm function được viết ra các file riêng để dùng lại khi cần)*

Cài đặt và sử dụng một module thông qua npm, chúng ta sẽ nói đến ở phần sau đây. Còn phần này chúng ta tập trung tìm hiểu cách viết một module mới trong nodejs và cách gọi (sử dụng lại) nó như thế nào

**2. Viết một module đơn giản**

* Tạo file simple\_module.js với nội dung sau

|  |
| --- |
| function message(){  return 'Vietpro Academy';  } |

**3. Module export**

*Trong nodejs, chúng ta muốn gọi (sử dụng lại) một module nào đó thì module này (file này) bắt buộc phải được export kết quả cuối cùng ra. Nếu không export thì các file khác sẽ không gọi lại được nó.*

Để export một module trong nodejs chúng ta có các cách sau:

**a. Export ra một đối tượng**

|  |
| --- |
| function message(){  return 'Vietpro Academy';  }  module.exports = {  'message': message  } |

**b. Export ra một hàm**

|  |
| --- |
| function message(){  return 'Vietpro Academy';  }  module.exports = message; |

**4. Module require**

Để gọi (sử dụng lại) một Module nào đó chúng ta sử dụng hàm require(), với tham số chính là tên file module

|  |
| --- |
| const message = require('./simple\_module'); |

* Sử dụng module được export ra một function

|  |
| --- |
| const message = require('./simple\_module');  console.log(message()); |

* Sử dụng module được export ra một đối tượng

|  |
| --- |
| const message = require('./simple\_module');  console.log(message.message()); |

IV – TẠO SERVER BẰNG NODEJS

**1. Module HTTP trong NodeJS**

*HTTP là một module được tích hợp sẵn trong Node.js, module này cho phép chúng ta chuyển dữ liệu trên giao thức HTTP (Hyper Text Transfer Protocol), có nghĩa là nó khởi tạo một cổng kết nối HTTP Server để trả dữ liệu về cho Client.*

|  |
| --- |
| const http = require('http'); |

**2. Khởi tạo Server bằng module HTTP trong NodeJS**

|  |
| --- |
| const server = http.createServer(function(req, res){    res.write('Hi NodeJS !');  res.end();  console.log('Server running on port '+port);  }).listen(port=8080); |

**Trong đó**

* **req:** Tham số nhận đối tượng Request từ HTTP, đây là đối tượng lưu trữ tất cả các thông tin có được khi có một yêu cầu (request) từ Client gửi lên
* **res:** Tham số nhận đối tượng Response từ HTTP, đây là đối tượng lưu trữ tất cả các thuộc tính và phương thức để thao tác và trả kết quả về cho Client
* **port:** Cổng kết nối

***Chú ý:*** *Chúng ta cũng có thể khởi tạo Server theo một cách thức khác như sau*

|  |
| --- |
| const http = require('http');  const app = http.createServer();  const server = app.listen(port=8080, function(){    console.log('Server running on port '+port);  }); |

V – QUẢN LÝ PACKAGE & MODULE TRONG DỰ ÁN NODEJS BẰNG NPM

**1. Khái niệm NPM**

*NPM viết tắt của Node Package Manager, quản lý các gói cài đặt, module phục vụ cho nodejs*

**2. Cách thức sử dụng**

* Bước 1: Tạo file config (default.json): Đây là file lưu thông tin của project nodejs cũng như thông tin các Package & Module phục vụ cho dự án mà chúng ta tải về và cài đặt

|  |
| --- |
| npm init -y |

* Bước 2: Cài đặt một module bất kỳ cho nodejs. Có thể tham khảo các module và cách sử dụng trên trang chủ <https://www.npmjs.com/>

|  |
| --- |
| npm install express --save |

* Các bạn cũng có thể gỡ một module nào đó thông qua câu lệnh sau

|  |
| --- |
| npm uninstall express |

* Bước 3: Để sử dụng một module sau khi đã cài đặt thông qua NPM chúng ta sử dụng hàm require() với tham số truyền vào chính là tên module như đã được tìm hiểu ở phần Module Export

|  |
| --- |
| var express = require(‘express’); |

**3. Hướng dẫn cài đặt và cấu hình Parkage Nodemon**

Trước đây, khi thay đổi một đoạn code, chúng ta sẽ phải thực hiện thao tác khởi động lại Server để Server nhận code mới. Với Nodemon, chúng ta sẽ không cần làm điều đó, Server sẽ reset một cách hoàn toàn tự động mỗi khi chúng ta save code.

**Bước 1**: Cài đặt Nodemon Parkage

**Code:**

|  |
| --- |
| npm i nodemon |

**Bước 2**: Cấu hình Nodemon

**Code:**

|  |
| --- |
| "scripts": {  "start": "nodemon app.js",  "test": "echo \"Error: no test specified\" && exit 1"  }, |

**Trong đó:**

* **start, nodemon**: là các key bắt buộc
* **app.js**: file chạy chính

**Bước 3**: Khởi động Server đã được sử dụng Nodemon bằng cách bật cmd và chạy đoạn code sau

**Code:**

|  |
| --- |
| npm start |

**4. Cấu trúc thư mục & file**

Dưới đây là cấu trúc một dự án thực tế của NodeJS được mô phỏng lại từ mô hình MVC trong Laravel Framework (Laravel là một Framework phổ biến và điển hình cho việc xây dựng các ứng dụng web từ đơn giản đến phức tạp)

