1. **Chuyển thư mục trong cmd để chạy JavaScript:**

**Cách 1:**

E:

cd PhatTrienUngDungWeb\BaiTap\JavaScript

node tinhToanCanBan.js

**Giải thích:**

* E: → Chuyển sang ổ **E**
* cd PhatTrienUngDungWeb\BaiTap\JavaScript → Di chuyển đến thư mục **chứa file**
* node tinhToanCanBan.js → Chạy file bằng **Node.js**

**Cách 2:**

node "E:\PhatTrienUngDungWeb\BaiTap\JavaScript\tinhToanCanBan.js"

**Lưu ý:**

* Đường dẫn có khoảng trắng thì đặt trong **dấu ngoặc kép** "..."
* Kiểm tra lại tên file có đúng không (phải là tinhToanCanBan.js, không có lỗi .js.js hoặc .txt)

***\*Xóa lịch sử code vửa chạy: cls***

**Hàm mũi tên:**

Có 3 cách để viết hàm mũi tên:

- [1] Không tham số: () => { ... }

- [2] Một tham số: x => { ... } (không cần dấu ngoặc cho một tham số)

- [3] Nhiều tham số: (x, y) => { ... }

1. **Nginx: Lắng nghe trên cổng 65536 được không? Tại sao**

Không thể lắng nghe trên cổng 65536

Lý do:

* Cổng hợp lệ trong TCP/UDP chỉ nằm trong khoảng 0 - 65535.
* 65536 là số không hợp lệ vì nó vượt quá giới hạn của hệ thống mạng.
* Khi cố gắng sử dụng listen 65536;, Nginx sẽ báo lỗi "invalid port".

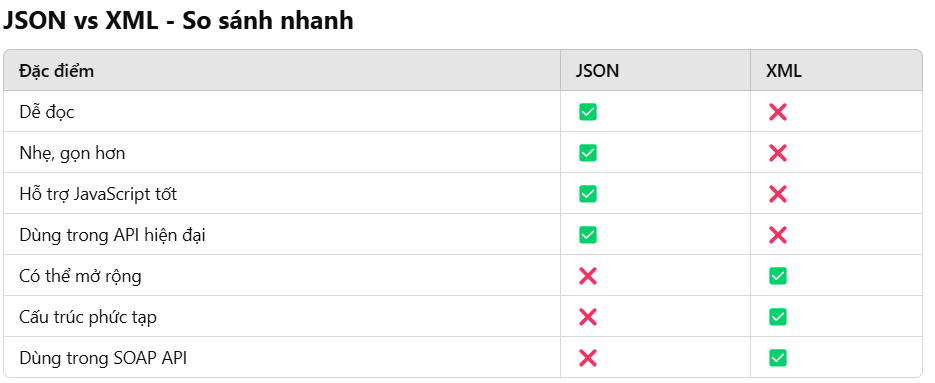
1. **Tổng quan về git:**

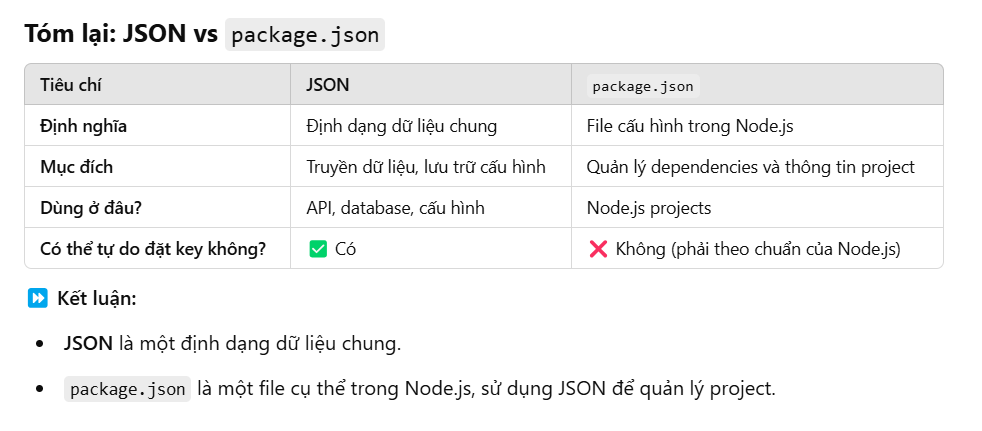
Khởi tạo git: git init (tạo kho chứa)

1. **Nodejs và NPM**

Xem phiên bản của Nodej: node -v

Xem phiên bản của NPM: npm -v







Tắt nodejs: Ctrl + C

Lệnh: ***npm init -y*** được sử dụng để tự động tạo một tập tin package.json trong dự án Node.js mà không cần phải trả lời các câu hỏi từng bước.

{

"name": "ptudw",

"version": "1.0.0",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

"keywords": [],

"author": "",

"license": "ISC",

"dependencies": {

"express": "^5.1.0"

},

"devDependencies": {},

"description": ""

}

Trong đó:

* name: Tên thư mục hiện tại (dự án) sẽ được sử dụng làm tên của dự án.
* version: Phiên bản mặc định là 1.0.0.
* main: Tên file chính (tập tin đầu vào) của ứng dụng là index.js (nếu không có, bạn có thể thay đổi).
* scripts: Chứa một số script mặc định, ví dụ test, có thể tùy chỉnh sau.
* license: Mặc định là ISC.
* description: Trống.

Cài đặt lại Express bằng npm: *npm install express*

Kiểm tra lại thư mục node\_modules: *dir node\_modules*

Chạy lại ứng dụng: *node index.js* -> thành công: server dang chay tren cong 9000

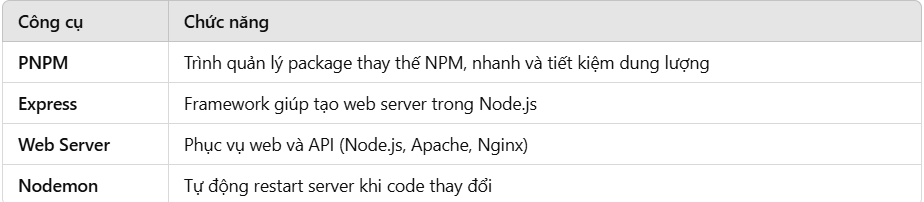
Chạy trên trình duyệt: [*http://localhost:9000/*](http://localhost:9000/)

1. **Nodemon**

Cài đặt Nodemon: *npm install --g nodemon*

Chạy nodemon: *nodemon*

**Kết luận:**



1. **Local và Global**

****

**VD:**

Cài đặt gói Express theo kiểu cục bộ: npm install express

Gỡ bỏ một gói cục bộ: npm uninstall express

Đây là mô hình hoạt động của Web Server theo kiến trúc Client-Server

Nó chia làm 2 phía: Phía Client và phía Sever, phía Client dùng 3 ngôn ngữ để làm: HTML, CSS, JavaScript, phía Sever dùng 1 trong 6 ngôn ngữ như: PHP, Node.js, Python, Java, C#, Ruby

Bên sever: dịch và thực thi mã

B1: Người dùng dùng trình duyệt(Browser) gửi yêu cầu(Request) đến Web Server dựa trên giao thức HTTP và phương thức GET để gửi yêu cầu.

B2: Phân tích Request data: URL encoding, GET/POST Data, Cookies

* URL Encoding: Mã hóa ký tự đặc biệt trong URL để truyền dữ liệu an toàn.
* GET/POST Data: Gửi dữ liệu từ client lên server. GET hiển thị trên URL, POST ẩn trong body request.
* Cookies: Lưu thông tin trên trình duyệt để duy trì đăng nhập, giỏ hàng, hoặc theo dõi người dùng.

Sau khi lấy thông tin của gói Request data thì đẩy xuống Web Application (mã nguồn Backend) để xử lý

B3: Web Application biết được yêu cầu của Web Server và xử lý yêu cầu, truy xuất dữ liệu từ cơ sở dữ liệu gửi trở về cho Web Application

B4: Đổ dữ liệu sang định dạng HTML (vì trình duyệt không hiểu ngôn ngữ)

B5: Web Application gửi HTML đã xử lý sau đó trả về Web Server

B6: Trình duyệt người dùng hiển thị nội dung response

B7: Sau khi nhận và hiển thị HTML, trình duyệt tiếp tục gửi thêm request khác để lấy các nội dung tĩnh của ứng dụng web (HTML, CSS, JavaScript, hình ảnh, và các tài nguyên khác).