## Cơ bản về JavaScript

## ****JavaScript là gì?****

**JavaScript là một ngôn ngữ lập trình website, được tích hợp và nhúng trong HTML giúp website sống động hơn**. **JavaScript** cho phép kiểm soát các hành vi của trang web tốt hơn so với khi chỉ sử dụng mỗi HTML. Vậy ứng dụng thực tiễn của **JavaScript là gì**? Các slideshow, pop-up quảng cáo và tính năng autocomplete của Google là những ví dụ dễ thấy nhất cho bạn, chúng đều được viết bằng **JavaScript.**

**JavaScript** là ngôn ngữ lập trình được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, Safari, … thậm chí các trình duyệt trên thiết bị di động.

## Lợi ích khi dùng JavaScript

Là một ngôn ngữ lập trình phổ biến sử dụng trên 92% nền tảng website hiện nay, **JavaScript** đã thể hiện vai trò quan trọng với lĩnh vực này. Tầm quan trọng của nó thể hiện qua các thao tác, công dụng như:

* ***Triển khai tập lệnh phía máy khách:*** Nhờ **Javascript**, các lập trình viên có thể dễ dàng viết tập lệnh phía máy khách, tích hợp các tập lệnh một cách liền mạch vào HTML, cho phép website tương tác, trả lời người dùng ngay lập tức và tạo ra giao diện hiển thị phong phú hơn.
* ***Viết mã phía máy chủ:*** Các lập trình viên có thể viết mã phía máy chủ bằng **JavaScript**.
* ***Đơn giản hóa phát triển ứng dụng web phức tạp:*** Javascript cho phép các nhà phát triển đơn giản hóa thành phần của ứng dụng, qua đó đơn giản hóa việc phát triển các ứng dụng web phức tạp.
* ***Thiết kế web responsive:*** **JavaScript** cho phép **thiết kế web responsive** – tối ưu trên cả máy tính và thiết bị di động chỉ với một bộ mã.
* ***Nhiều bộ chuyển đổi:*** Mặc dù thiếu một số tính năng phức tạp được cung cấp bởi các ngôn ngữ lập trình hiện đại như Java và C#, **JavaScript**vẫn có thể dễ dàng mở rộng bằng cách sử dụng các bộ chuyển đổi như CoffeeScript, TypeScript, DukeScript và Vaadin.

## ****Các thư viện Javascript phổ biến****

Hiện nay có rất nhiều thư viện và framework được viết từ Javascript như:

* **jQuery**: Chuyên về hiệu ứng.
* **NodeJS**: dùng để lập trình cho server.
* **AngularJS**: Chuyên xây dựng ứng dụng trang cá nhân.
* **ReactJS**: Chuyên viết ứng dụng mobile.
* Một số thư viện khác như **ExtJS**, **Sencha Touch**,….

## ****Ưu điểm của JavaScript****

JavaScript có rất nhiều ưu điểm vượt trội hơn so với các đối thủ khác, chẳng hạn như:

* **JavaScript** là ngôn ngữ lập trình **dễ học**.
* **Lỗi của JavaScript dễ phát hiện** hơn và vì vậy dễ sửa hơn.
* **JavaScript hoạt động trên nhiều trình duyệt**, nền tảng.
* **JavaScript giúp website tương tác tốt hơn** với khách truy cập.
* **JavaScript nhanh hơn và nhẹ hơn** các ngôn ngữ lập trình khác.

Không ngôn ngữ lập trình nào là hoàn hảo và không có khuyết điểm. Bởi **JavaScript** là ngôn ngữ lập trình phổ biến, được sử dụng rộng rãi trên nền tảng website, vì vậy nó cũng là đối tượng tiềm năng của một lượng lớn hacker, scammer – những kẻ chuyên tìm kiếm lỗ hổng và các lỗi bảo mật và thâm nhập, đánh cắp dữ liệu nội bộ.

## Tìm hiểu về NodeJS

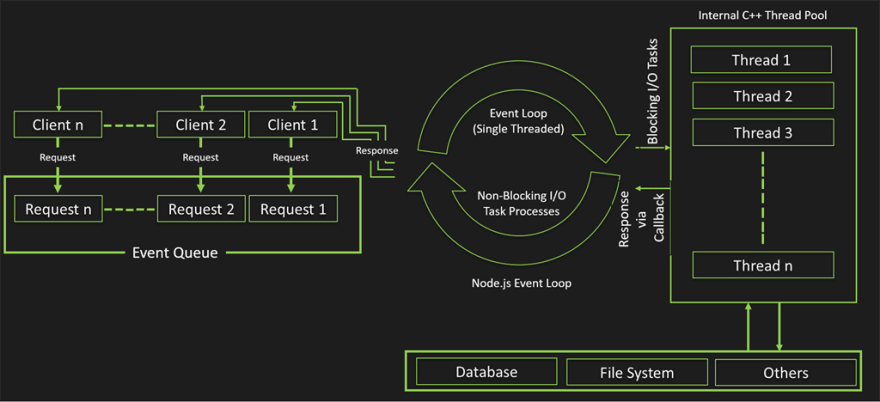
## Nodejs là gì?

* NodeJS là một nền tảng (Platform) được xây dựng trên nền tảng Javascript V8 Engine. Được xây dựng để phát triển những ứng dụng server side.
* Phần core sử dụng Javascript và C++ cho phép xử lý với hiệu năng cao.
* Phù hợp với các ứng dụng xử lý nhanh, real time hoặc những ứng dụng cần thay đổi công nghệ nhanh.
* Chạy sigle thread nhưng có cơ chế non blocking giúp chạy bất đồng bộ, tăng khả năng xử lý.
* Thường phát triển theo hướng module hóa thành các phần nhỏ của ứng dụng và có thể thêm module có sẵn bên ngoài thông qua NPM, yarn....

## Đặc điểm cơ bản của Nodejs:

* **Không đồng bộ:** Đặc điểm đầu tiên của Nodejs là tính bất đồng bộ. Node.js không cần đợi API trả dữ liệu về, vậy nên mọi APIs nằm trong thư viện Node.js đều không được đồng bộ, hiểu đơn giản là chúng không hề blocking (khóa). Server có cơ chế riêng để gửi thông báo và nhận phản hồi về các hoạt động của Node.js và API đã gọi.
* **Tốc độ nhanh:** Với phần core phía dưới lập trình gần như toàn bộ bằng ngôn ngữ C++, kết hợp với V8 Javascript Engine mà Google Chrome cung cấp, tốc độ vận hành, thực hiện code của thư viện Node.js diễn ra rất nhanh.
* **Đơn giản - Hiệu quả:** Tiến trình vận hành của Node.js đơn giản song lại mang đến hiệu năng cao nhờ ứng dụng mô hình single thread và các sự kiện lặp. Một loạt cơ chế sự kiện cho phép server trả về phản hồi bằng cách không block, đồng thời tăng hiệu quả sử dụng. Các luồng đơn cung cấp dịch vụ cho nhiều request hơn hẳn Server truyền thống.
* **Không đệm**: Nền tảng Node.js không có vùng đệm, tức không cung cấp khả năng lưu trữ dữ liệu buffer.
* **Chạy trên nhiều thiết bị:** Nodejs vận hành trên đa nền tảng của Server, bởi vậy nó hoàn toàn có thể chạy trên nhiều thiết bị.
* **Có giấy phép:**Đây là nền tảng đã được cấp giấy phép, phát hành dựa trên MIT License.

## Kiến trúc



# **Những khái niệm cần biết trong NodeJS**

## Npm và package

NPM viết tắt của **N**ode **P**ackage **M**anager là một công cụ tạo và quản lý các thư viện lập trình Javascript cho NodeJS. Trong cộng đồng Javascript, các lập trình viên chia sẻ hàng trăm nghìn các thư viện với các đoạn code đã thực hiện sẵn một chức năng nào đó. Nó giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản, các thư viện lập trình hay thậm chí cả các framework.

## ****Cài đặt và sử dụng Npm****

Npm được tích hợp sẵn có trong NodeJS, để kiểm tra xem trên hệ thống của bạn đã được cài npm chưa chúng ta sử dụng lệnh **npm -v**, nếu một phiên bản hiện ra thì hệ thống của bạn đã được cài đặt npm.

Nếu bạn tạo mới 1 project và muốn sử dụng npm, tốt hơn hết bạn nên bắt đầu với câu lệnh: **npm init**

Câu lệnh trên đơn giản là sẽ tạo ra 1 file có tên là **package.json** – thành phần này được gọi là **Local Package Database**, lưu trữ thông tin (tên package, phiên bản, các dependencies) mà project của bạn sử dụng.

Sau khi chạy câu lệnh này, Npm sẽ hỏi chúng ta một vài câu hỏi về project của bạn:

* Đầu tiên là về package name: Đây là tên dự án của bạn
* Version: Đây là phiên bản dự án của bạn, mặc định sẽ là 1.0.0
* Description: Mô tả về dự án của bạn.
* Entry point: Entry point sẽ quy định root của Node, file này sẽ chứa các câu lệnh về server, mặc định sẽ là index.js. Chú ý là khi bạn chạy server, Node sẽ đọc dữ liệu từ file này để khởi tạo. Tên này có thể thay đổi được, nhưng hãy chắc rằng file đó tồn tại ở root của dự án.
* Test command: Đây là dòng lệnh mà sẽ chạy khi bạn gọi **npm test** Mặc định để trống
* Git repository: Như tên gọi, nó là git repository ứng với dự án của bạn. Mặc định để trống.
* Keywords: Các từ khóa tương ứng với dự án của bạn. Mặc định để trống
* Author: Tên tác giả của dự án. Mặc định để trống
* License: Giấy phép. Mặc định là ISC.

### **Các loại package**

* **Dependecies:** là những packagebắt buộc phải có trong quá trình chạy sản phẩm, kiểu như một thư viện cung cấp các hàm mà code của bạn cần. Khi cài đặt dependencies, Npm sẽ tự động cài đặt tất cả các dependencies cần thiết.



* **Development dependencies:**là những package bắt buộc khi phát triển cũng như phát hành sản phẩm. Kiểu như các trình biên dịch giúp biên dịch đoạn code của bạn về javascript, rồi các framework phục vụ cho việc kiểm thử,…



### **Cài đặt các package như thế nào nhé**

* **Dependencies package:** Cú pháp như sau:

npm install [tên package]

* **Development dependencies package:** Cú pháp như sau**:**

npm install [tên package] --save-dev

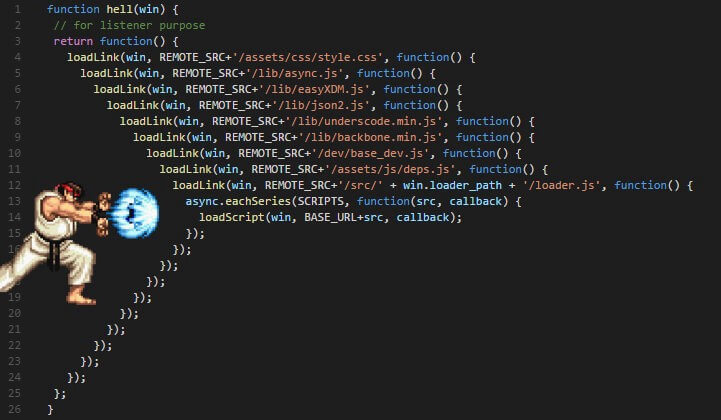
## Những khái niệm Javascript cơ bản trong NodeJS

## Async callback

Bất đồng bộ callback xuất hiện rất nhiều ở Javascript



* Ưu điểm: Dễ hiểu (vì chỉ là một callback function đơn giản)
* Nhược điểm: Khó khăn trong việc xử lý lỗi, code bị lồng vào nhau quá nhiều (callback hell)



*Hiện tượng Callback Hell (tức là có quá nhiều callback lồng nhau)*