# Node.js

## Node.js là gì

***Nodejs*** là một JavaScript runtime được xây dựng dựa trên engine javascript v8 của chrome. Nó sử dụng kiến túc hướng sự kiện, mô hình *non blocking i/o* là cho nó nhẹ và hiệu quả hơn. Hệ thống nén của nodejs, ***npm*** là hệ thống hư viện nguồn mở lớn nhất thế giới.

## Cơ chế đồng bộ

Trong cơ chế này, chương trình được thực hiện tuần tự. Mỗi khi một hàm được gọi, chương trình sẽ đợi cho đến khi hàm đó trả về trước khi tiếp tục đến dòng mã tiếp theo.

## Cơ chế bất đồng bộ

Trái ngược với cơ chế đồng bộ, các đoạn mã sẽ không chạy theo trình tự, chương trình không cần đợi đoạn code này thực hiện xong mà vẫn có thể chuyển sang được đoạn code khác

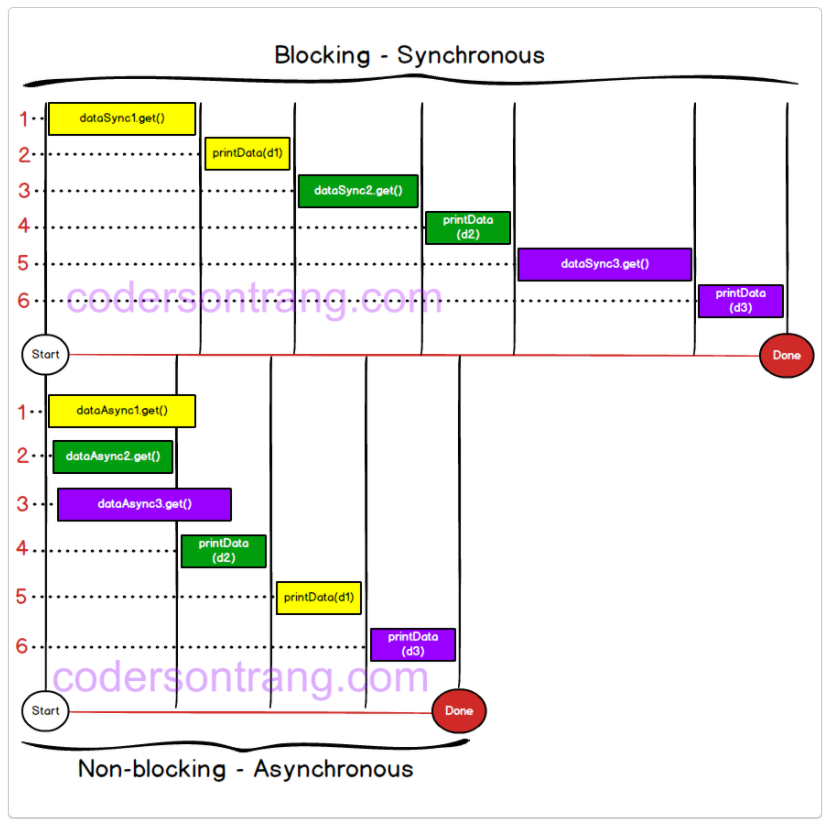
## Blocking

Với mô hình blocking thì các dòng lệnh sẽ thực hiện một cách tuần tự, các dòng lệnh phía sau phải đợi các dòng lệnh phía trước hoàn thành. Nếu các lệnh phía trước là thao tác cần nhiều thời gian xử lý nhue liên quan tới I/O hay network thì bản thân nó sẽ trờ thành vật cản trở các xử lý phía sau

## Non blocking i/o

Nếu như về mặt logic, dòng lệnh phía sau không phụ thuộc vào kết quả của dòng lệnh phía trước thì nó cũng sẽ hoàn toàn được thực hiện ngay sau khi dòng lệnh phía trước được gọi mà không cần đợi cho tới khi kết quả được sinh ra. Những dòng lệnh phía trước có thế được gọi là thực hiện theo cách không đồng bộ và đi heo mỗi dòng lệnh thường có một callback là đoạn mã sẽ được thực hiện ngay sau khi có kết quả trả về từ dòng lệnh không đồng bộ

## So sánh



***Blocking – synchronous***

dataSync1.get() được gọi ra để lấy dữ liệu, các câu lệnh tiếp theo phải đợi hàm này thực hiện xong

Sau khi lấy được dữ liệu thông qua hàm dataSync1.get(), chương trình sẽ in kết quả ra

In xong kết quả thì hàm dataSync2.get() mới được phép chạy .

Liên tục như thế, các lời gọi hàm sẽ được thực hiện tuần tự, hàm sau đợi hàm trước cho tới khi kết thúc chương trình

***Non-blocking – Asynchronous***

Hàm dataAsync1.get() được gọi để lấy dữ liệu, trong đây sẽ có một callback để in ra dữ liệu ngay sau khi việc lấy dữ liệu thành công.

Hàm dataAsync2.get() và callback của nó cũng gần như được gọi cùng lúc với dataAsync1.get().

dataAsync3.get() và callback của nó cùn tương tự, được gọi đồng thời với dataAsync1.get()

Trong khi hàm dataAsync1.get() và dataAsync3.get() đang chạy thì dataAsync2.get() thực hiện xong và gọi callback printData(d2) để in kết quả.

Trong khi printData(d2) đang chạy thì dataAsync1.get() thực hiện xong. Nó phải đợi printData(d2) in xong thì mới được gọi hàm callback của nó được. Lần lượt như vậy cho tới khi chương trình chạy xong

Ta có thể thấy tổng thời gian thực hiện non blocking nhanh hơn, không phải vì nó thực hiện ít công việc hơn, mà do nó thực hiện đồng thời nhiều việc.

## NPM (node.js project manage)

Là một chương trình quản lý thư viện, source của node.js. Nó được tích hợp sẵn vào trong node.js