



# Lập trình ứng dụng web bằng NodeJS

## Bài 1. *Tổng Quan Lập Trình NodeJS*

<http://csc.edu.vn/>





# Nội dung

---

1. Giới thiệu NodeJS
2. Môi trường phát triển ứng dụng
3. Ôn tập một số khái niệm lập trình trong Javascript
4. Ôn tập lập trình trong Javascript
5. Cài đặt NodeJS
6. Xây dựng ứng dụng đầu tiên và thực thi chương trình



# 1. Giới Thiệu NodeJS

---

❑ Lịch sử phát triển

❑ Đặc điểm NodeJS

❑ Ưu điểm

❑ Nhược điểm



## 1.1 Lịch sử phát triển

---

- ❑ NodeJS được Ryan Dahl tạo ra từ năm 2009, và chỉ hỗ trợ chạy trên Linux.
- ❑ Không lâu sau khi ra đời, duy trì bởi Ryan Dahl thì NodeJS được tài trợ bởi công ty Joyent
- ❑ Vào tháng 1/2010, gói quản lý npm được giới thiệu cho việc lập trình ngôn ngữ NodeJS.
- ❑ Tháng 6/2011 Joyent và Microsoft bàn luận về NodeJS cho hệ điều hành Window, và chỉ sau đó 1 tháng thì chính thức NodeJS chạy trên Window



## 1.2 Đặc điểm NodeJS

---

- ☐ NodeJS chạy trên môi trường Google V8 Javascript Runtime
- ☐ NodeJS viết bằng ngôn ngữ javascript nhưng chạy phía server-side.
- ☐ Hiện tại có thể viết NodeJS bằng một số ngôn ngữ khác như Coffescript, Dart (Google), Typescript (Microsoft), hay bất kỳ ngôn ngữ nào có thể biên dịch sang javascript
- ☐ Nodejs sử dụng lập trình hướng sự kiện, mô hình giao tiếp non-blocking I/O



## 1.3 Ưu điểm

---

- ❑ Khả năng xử lý nhiều request đồng thời và thời gian phản hồi phải nhanh phù hợp với nhiệm vụ cho xây dựng service cho trang Single Page Application
- ❑ Cơ chế event-driven, non-blocking IO (Input/Output), mô hình kết hợp với Javascript là điều tuyệt vời để xây dựng Web Service API (JSON) và cả xây dựng một ứng dụng thời gian thực



## 1.3 Ưu điểm

---

- ❑ **Shelling tools unix: NodeJS tận dụng tối đa khả năng của Unix, có thể xử lý hàng ngàn process và trả ra một luồng dữ liệu khiến cho hiệu suất đạt cao nhất.**
- ❑ **Streaming data: Luồng dữ liệu lớn được NodeJS xây dựng proxy phân vùng xử lý không ảnh hưởng đến các luồng xử lý khác**



## 1.4 Khuyết điểm

---

- ❑ Vì NodeJS thường được dùng chung với CSDL NoSQL (VD: MongoDB) nên nếu với lập trình viên đã quen với các kiểu lập trình trước sẽ tương đối khó khăn
- ❑ Nếu xử lý ứng dụng nặng, NodeJS sẽ tốn rất nhiều tài nguyên
- ❑ Ngoài ra vì NodeJS bất đồng bộ nên xử lý lập trình sẽ rất dễ xảy ra sai sót nếu lập trình viên chưa quen với lập trình javascript.





# Nội dung

---

1. Giới thiệu NodeJS
2. Môi trường phát triển ứng dụng
3. Ôn tập một số khái niệm lập trình trong Javascript
4. Ôn tập lập trình trong Javascript
5. Cài đặt NodeJS
6. Xây dựng ứng dụng đầu tiên và thực thi chương trình



## 2. Môi trường phát triển ứng dụng

### ❑ Visual Studio Code và một số Plugin NodeJS

- Ưu điểm:
  - Nhẹ, nhanh
  - Phù hợp với Coder
  - Có gợi ý code tương đối đầy đủ
  - Debug, check lỗi tốt
  - Chạy terminal trực tiếp
- Khuyết điểm:
  - Không có hỗ trợ tạo project hoàn thiện
  - Phải nhớ các plugin hỗ trợ





## 2. Môi trường phát triển ứng dụng

### ❑ Eclipse Neon 2 và Nodeclipse Plugin:

- Ưu điểm:
  - Hỗ trợ tạo project tốt
  - Debug lỗi Node tốt nhờ Nodeclipse
  - Chạy terminal trực tiếp
  - Phù hợp với Leader
- Khuyết điểm
  - Tương đối chậm
  - Dễ bị dump nếu máy yếu





# Nội dung

---

1. Giới thiệu NodeJS
2. Môi trường phát triển ứng dụng
3. Ôn tập một số khái niệm lập trình trong Javascript
4. Ôn tập lập trình trong Javascript
5. Cài đặt NodeJS
6. Xây dựng ứng dụng đầu tiên và thực thi chương trình



### 3. Ôn tập một số khái niệm trong javascript

- ❑ Đặc điểm và các kiểu dữ liệu
- ❑ Hàm ẩn danh (Anonymous Function)
- ❑ Hàm lồng hàm (Closure Function)
- ❑ Đồng bộ và bất đồng bộ



# 3. Ôn tập khái niệm lập trình trong Javascript

---

## ❑ Đặc điểm ngôn ngữ Javascript:

- Là ngôn ngữ kịch bản (scripting language)
- Là ngôn ngữ bất đồng bộ
- Có phân biệt Hoa thường
- Lập trình không cần biên dịch trước

## ❑ Các kiểu dữ liệu:

- String
- Number
- Boolean
- Array
- Object



### 3. Ôn tập khái niệm lập trình trong Javascript

---

#### ❑ Anonymous function:

- Là hàm ẩn danh.
- Một trong những ưu điểm mạnh nhất của Javascript giúp người lập trình code các function trở nên linh hoạt hơn
- Cấu trúc khai báo:

```
function(// tham số truyền vào){  
    // xử lý hàm  
}
```

Hay có thể dùng

```
Ten_bien_chua_ham = function(// tham số truyền vào){  
    // xử lý hàm  
}
```



## 3. Ôn tập khái niệm lập trình trong Javascript

---

### ❑ Hàm lồng hàm (closure function):

- Là một khái niệm quan trọng trong lập trình Javascript.
- Tạo bởi các hàm ẩn danh lồng vào nhau
- Mở ra một cách lập trình đồng bộ trong javascript khi cần thiết
- Ví dụ:

```
var add = (function () {  
    var counter = 0;  
    return function () {return counter += 1;}  
})();
```

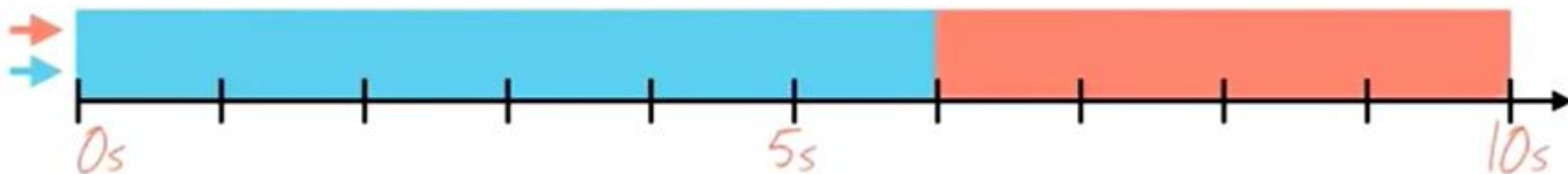




### 3. Ôn tập khái niệm lập trình trong Javascript

#### ❑ Đồng bộ:

```
var contents = fs.readFileSync('/etc/hosts');  
console.log(contents);  
console.log('Doing something else');
```

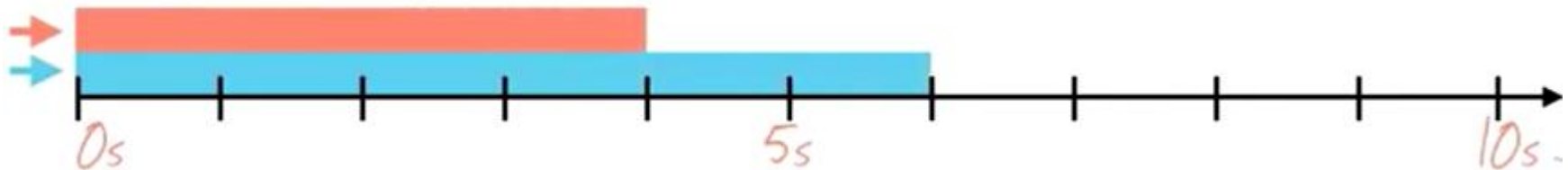




### 3. Ôn tập khái niệm lập trình trong Javascript

#### ❑ Bất đồng bộ:

```
fs.readFile('/etc/hosts', function(err, contents) {  
  console.log(contents);  
});  
console.log('Doing something else');
```



# Nội dung

---



## 1. Giới thiệu NodeJS

- Lịch sử phát triển NodeJS
- Đặc điểm NodeJS

## 2. Môi trường phát triển ứng dụng

## 3. Ôn tập một số khái niệm lập trình trong Javascript

## 4. Ôn tập lập trình trong Javascript

## 5. Cài đặt NodeJS

## 6. Xây dựng ứng dụng đầu tiên và thực thi chương trình



## 4. Ôn tập lập trình trong Javascript

### ❑ Khai báo biến:

```
var <tên biến>;  
hoặc      var <tên biến> = <giá trị>;  
hoặc      <tên biến> = <giá trị>;
```

### ❑ Để khai báo biến cục bộ, bắt buộc phải dùng từ khóa var



## 4. Ôn tập lập trình trong Javascript

---

### ❑ Phương thức trong đối tượng Javascript

- Cách khai báo thứ 1:

```
var Đối_tượng = {  
    Tên_phương_thức: function () {  
        //code xử lý  
    }  
}
```

- Cách khai báo thứ 2

```
Đối_tượng.Tên_phương_thức = function () {  
    //code xử lý  
};
```

# Nội dung

---



## 1. Giới thiệu NodeJS

- Lịch sử phát triển NodeJS
- Đặc điểm NodeJS

## 2. Môi trường phát triển ứng dụng

## 3. Ôn tập một số khái niệm lập trình trong Javascript

## 4. Ôn tập lập trình trong Javascript

## 5. Cài đặt NodeJS

## 6. Xây dựng ứng dụng đầu tiên và thực thi chương trình



## 5. Cài đặt NodeJS

---

### ❑ Download NodeJS:

<https://nodejs.org/en/download/>

- Chọn phiên bản phù hợp với hệ điều hành mà chúng ta đang sử dụng.
- Nên cài đặt phiên bản NodeJS được nhiều người sử dụng nhất nếu mới bắt đầu code NodeJS



## 5. Cài đặt NodeJS

---

### ❑ Kiểm tra version NodeJS đã cài đặt:

- Dùng lệnh: `node -v` để kiểm tra xem version của Node
- Dùng lệnh: `npm -v` để kiểm tra version npm package đã cài đặt

```
C:\Users\NXHUNG>node -v  
v7.5.0
```

```
C:\Users\NXHUNG>npm -v  
4.1.2
```



# Nội dung

---



## 1. Giới thiệu NodeJS

- Lịch sử phát triển NodeJS
- Đặc điểm NodeJS

## 2. Môi trường phát triển ứng dụng

## 3. Ôn tập một số khái niệm lập trình trong Javascript

## 4. Ôn tập lập trình trong Javascript

## 5. Cài đặt NodeJS

## 6. Xây dựng ứng dụng đầu tiên và thực thi chương trình

## 6. Xây dựng ứng dụng đầu tiên và thực thi chương trình



### ❑ Hướng dẫn:

```
http = require("http");

http.createServer(function(req, res){
    res.end("Chạy NodeJS rồi nhé!")
}).listen("8888");
```

### ❑ Kết quả khi vào trang <http://localhost:8888>



Chạy NodeJS rồi nhé!

# Thảo luận

---

