**BÁO CÁO TÌM HIỂU VỀ VUE JS**

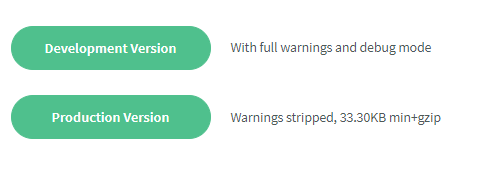
1. **Giới thiệu**

Vue.js là một framework của Javascript, là một Progressive framework dùng để xây dựng giao diện người dùng. Đối với Full-featured framework hay Monolithic framework cung cấp tất cả mọi thứ cần có để xây dựng app trong một framework duy nhất, thì progresive framework lại chia thành các thành phần nhỏ khác nhau, và ta có thể dần dần lựa chọn các thành phần tham gia vào sao cho phù hợp. Phần lõi của Vue.js tập trung chủ yếu vào phần view.

1. **Cài đặt để sử dụng**

Để sử dụng được vue js thì trước hết chúng ta phải cài đặt. Có 4 cách cài đặt là chủ yếu:

**Cách 1:** Sử dụng thẻ **<script>** trực tiếp bằng cách dowload trực tiếp trên trang chủ Vue js về.



Hình 1. Trang download vue.js

**Cách 2:** Cài đặt bằng **CDN**



Hình 2. Cài đặt bằng cdn.

Ta có thể xem được thêm các mã nguồn của package trên npm tại: <https://cdn.jsdelivr.net/npm/vue/>

**Cách 3:** Cài đặt vue js bằng **NPM**

Ta nên sử dụng npm để dụng khi muốn xây dụng một ứng dụng lớn với Vue.



Hình 3. Cài đặt bằng npm.

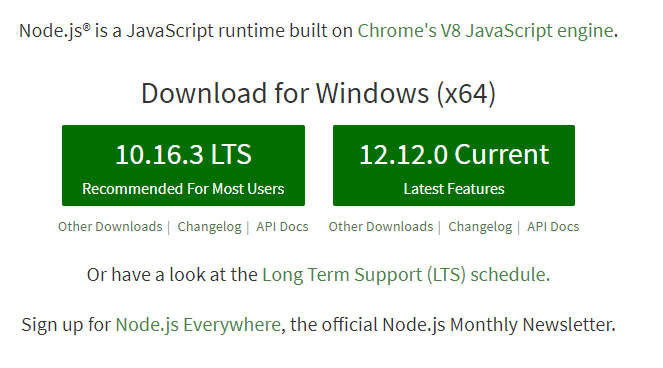
**Cách 4:** Cài đặt **Vue - CLI**

Các công cụ dòng lệnh (CLI – Command Line Interface) hiện là những tính năng không thể thiếu cho một framework, nó giúp xây dựng các template một cách nhanh chóng.

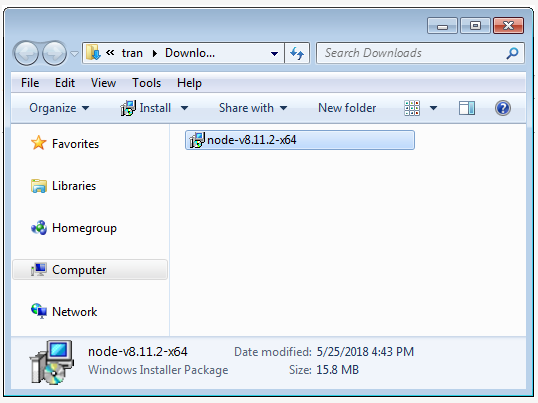
Để cài đặt được Vue-CLI thì trước hết phải cài đặt Nodejs.

Các bước cài đặt Nodejs.

**Bước 1:** Download nodejs về máy bằng cách vào trang chủ của nodojs có địa chỉ như sau: <https://nodejs.org/en/> và tìm đúng phiên bản thích hợp để download.

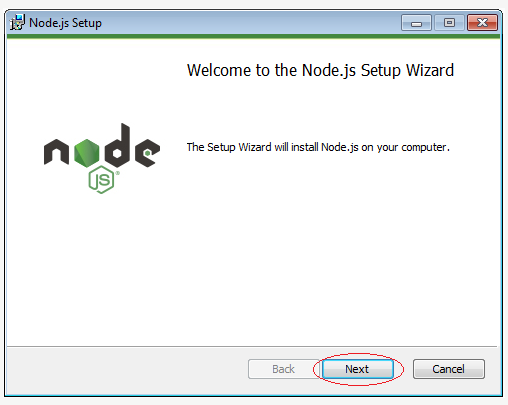


Hình 4. Trang chủ download Nodejs.



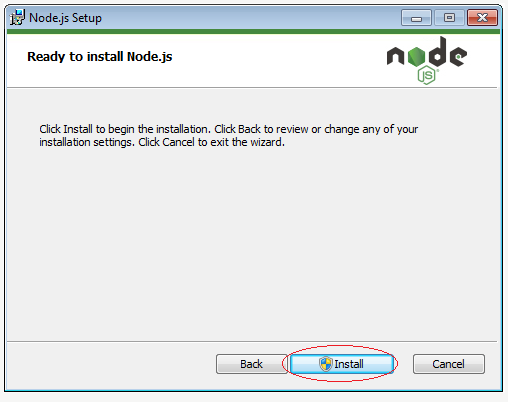
Hình 5. File nodejs.

**Bước 2:** Sau khi download thành công thì tiến hành cài đặt nodejs bằng cách double click vào file vừa download về sau đó cứ làm đúng yêu cầu rồi **Next** để cài đặt.



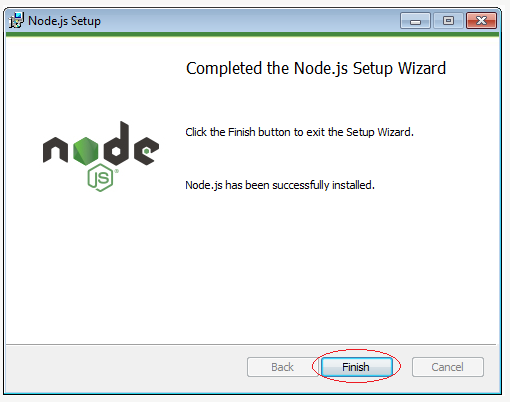
Hình 6. Giao diện cài đặt nodejs.

**Bước 3:** Nhấn Install để cài đặt nodejs.



Hình 7. Cài đặt nodejs.

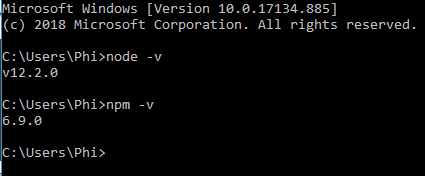
**Bước 4:** Chờ đến khi quá trình cài đặt hoàn tất và nhấn **Finish** để hoàn thành.



Hình 8. Giao diện hoàn thành cài đặt nodejs.

**Bước 5:** Kiểm tra và cấu hình Nodejs.

Vào cửa sổ **CMD** để thực hiện kiểm tra phiên bản của **Node Js** và **NPM**



Hình 9. Kiểm tra phiên bản nodejs và nmp.

Để có thể run anywhere với vue-cli hãy install nó global



Hình 10. Cài đặt Vue-CLI

1. **Rendering**

Lõi của Vue.js là một hệ thống cho phép chúng ta render dữ liệu đến DOM bằng cú pháp template rõ ràng, đơn giản:



Hình 11. Hiển thị dữ liệu đơn giản với Vue js.

Dữ liệu của các DOM được liên kết với nhau. Khi ta sửa app.message=”hello world 12345”trong console thì lập tức chuỗi sẽ chuyển sang thành “hello world 12345”

Ngoài việc quản lí và chỉnh sửa văn bản, chúng ta cũng có thể bind (ràng buộc) các thuộc tính của phần tử web, như sau:

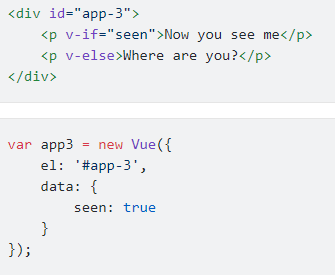


Hình 12. Demo v-bind.

Khi rê chuột lên chữ “Loading” một vài giây thì sẽ hiển thị thuộc tính title được bind. Thuộc tính v-bind được gọi là một directive trong Vue, các directive sẽ được bắt đầu bằng v- . Trong ví dụ trên, directive về cơ bản đang giữ thuộc tính của title và luôn đồng nhất với đối tượng vue được khởi tạo.

1. **Điều kiện vòng lặp**

Để ẩn hay hiển thị một phần tử trong Vue rất dễ dàng:



Hình 13. Điều kiện trong Vue js.

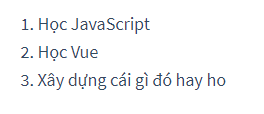
Ta thấy dòng chữ “Now you see me” nếu như giá trị của seen trong data là true còn ngược lại nếu là false thì sẽ hiển thị dòng chữ “Where are you?”.

Ngoài v-bind hay v-if thì chúng ta còn có thể v-for dùng để trình bày các giá trị dữ liệu từ một mảng:



Hình 14. Vòng lặp trong vue js.

Kết quả sẽ hiển thị là:



Hình 15. Hiển thị ví dụ vòng lặp.

Ta có thể thấy :

+ **todo**: là phần tử thứ x trong mảng.

+ **todos**: là tên mảng.

+ **{{todo.text}}**: là để lấy giá trị trong mảng lên.

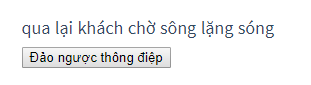
1. **Xử lý thông tin của người dùng**

Để người dùng tương tác với ứng dụng, chúng ta có thể dùng directive **v-on** để đính kèm các event listener (phần tử lắng nghe sự kiện) và gọi các method (phương thức) trên đối tượng Vue.



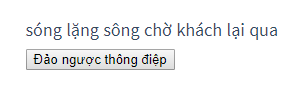
Hình 16. Sự kiện tương tác với người dùng trong Vue js.

Hiển thị:



Hình 17. Hiển thị v-on.

Khi ta click vào button “Đảo ngược ….” Thì chuỗi hiển thị sẽ đảo ngược lại.



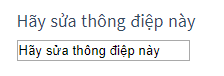
Hình 18. Sau khi click vào button.

Vue cũng cung cấp directive **v-model** giúp cho việc ràng buộc hai chiều (two-way binding) giữa form input và trạng thái ứng dụng trở nên vô cùng dễ dàng:



Hình 19. Demo v-model.

Hiển thị:

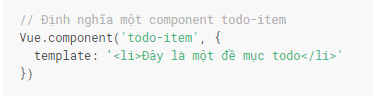


Hình 20. Hiển thị v-model.

Chúng ta có thể chỉnh sửa thông điệp trong form input.

1. **Component**

Định nghĩa về component trong Vue js



Hình 21. Định nghĩa component trong vue js.

Giờ thì bạn có thể sử dụng component todo-item trong template của các c omponent khác:



Hình 22. Tạo đỗi tượng của component để hiển thị.

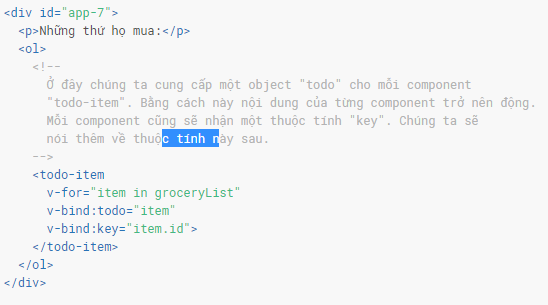
Có điều ví dụ này chưa được thú vị cho lắm, vì các đề mục todo hiện đang giống nhau hoàn toàn. Chúng ta có thể truyền dữ liệu từ scope (phạm vi) cha vào các component con bằng cách sử dụng một **prop**:



Hình 23. Ví dụ về props.

Bây giờ chúng ta có thể truyền **todo** vào mỗi component bằng cách sử dụng **v-bind**:

HTML:



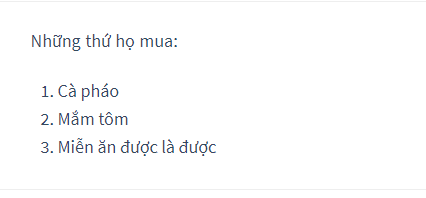
Hình 24. Ví dụ về component (HTML).

Vue js:



Hình 25. Ví dụ về component (Vue js).

Hiển thị:



Hình 26. Hiển thị ví dụ component.