

## ■ Learning Vue.js 2

**Tác giả:** Olga Filipova

**Xuất bản bởi:** Packt Publishing

**Mục tiêu:** Cuốn sách này giúp lập trình viên web nhanh chóng nắm bắt Vue.js 2 – một framework JavaScript tiên bộ dùng để xây dựng giao diện người dùng hiện đại, nhẹ và dễ mở rộng.

### ■ Chương 1: Làm Quen Với Vue.js

Vue.js là một framework JavaScript hiện đại giúp bạn xây dựng các giao diện người dùng động và thân thiện. Không giống như các framework “nặng đô” như Angular, Vue được thiết kế để nhẹ, dễ học và dễ áp dụng, ngay cả với người mới bắt đầu.

Hãy tưởng tượng bạn đang làm một website bán hàng. Khi người dùng thêm sản phẩm vào giỏ, bạn muốn số lượng sản phẩm trên biểu tượng giỏ hàng tăng lên ngay lập tức. Vue cho phép bạn làm điều đó một cách tự nhiên mà không cần viết nhiều dòng mã phức tạp.

Vue hoạt động bằng cách kết nối dữ liệu bên trong logic JavaScript với giao diện người dùng. Khi dữ liệu thay đổi, phần hiển thị cũng tự động thay đổi. Bạn không cần phải “tự tay” cập nhật lại từng phần giao diện nữa.

**Ghi nhớ:** Vue là framework phù hợp để bắt đầu học lập trình web hiện đại vì cú pháp rõ ràng, dễ dùng và có tính phản ứng tự động cao.

---

### ■ Chương 2: Dữ Liệu và Giao Diện – Hai Thế Giới Kết Nối

Trong Vue, phần dữ liệu (data) được lưu trữ dưới dạng các biến. Khi bạn thay đổi giá trị của một biến, Vue sẽ tự động cập nhật phần giao diện đang dùng biến đó.

Ví dụ, bạn có một tiêu đề trang web hiển thị “Chào buổi sáng”. Nếu bên trong logic bạn thay đổi nội dung thành “Chúc một ngày tốt lành”, thì phần hiển thị trên website cũng đổi theo, mà bạn không cần viết thêm dòng nào xử lý giao diện.

Không chỉ văn bản, bạn còn có thể thay đổi các thuộc tính như màu nền, kích thước, hiển thị hay ẩn các phần tử,... tất cả đều dựa trên dữ liệu.

Ngoài ra, Vue hỗ trợ các công cụ giúp bạn rút gọn việc xử lý dữ liệu:

- Có thể tính toán các giá trị mới dựa trên dữ liệu có sẵn (gọi là **computed**)
- Có thể phản ứng khi dữ liệu thay đổi (gọi là **watch**)

**Ví dụ thực tế:** Khi người dùng nhập tên, bạn có thể ngay lập tức hiển thị lời chào cá nhân hóa như “Xin chào, Lan!”

---

## ■ Chương 3: Xử Lý Hành Động Người Dùng

Giao diện web hiện đại không chỉ là nơi hiển thị thông tin – nó còn cần **phản hồi các hành động** từ phía người dùng. Vue giúp bạn dễ dàng xử lý các hành động như:

- Nhấn nút
- Nhập văn bản
- Di chuột
- Gửi biểu mẫu

Khi người dùng nhấn một nút, bạn có thể định nghĩa một phương thức tương ứng sẽ được gọi để xử lý hành động đó, ví dụ: hiển thị thông báo, cập nhật dữ liệu, chuyển trang,...

Bạn cũng có thể tùy chỉnh hành vi: ví dụ, chỉ xử lý khi nhấn Enter thay vì mọi phím, hoặc ngăn không cho form gửi đi khi người dùng chưa nhập đủ thông tin.

## ■ Chương 4: Hiển Thị Danh Sách Và Điều Kiện

### 👉 Lập qua danh sách

Một ứng dụng thường cần hiển thị nhiều mục giống nhau – như danh sách sản phẩm, bài viết, bình luận,... Vue cho phép bạn hiển thị danh sách bằng cách “lặp” qua mảng dữ liệu và tự động tạo ra các phần tử tương ứng trên giao diện.

Ví dụ: Nếu bạn có một danh sách gồm ba món ăn, Vue sẽ hiển thị cả ba – và nếu bạn thêm một món mới vào danh sách, nó sẽ lập tức hiện ra mà không cần viết lại phần giao diện.

Điểm mạnh là bạn chỉ cần viết một đoạn giao diện duy nhất, Vue sẽ lặp lại nó dựa trên dữ liệu có sẵn. Điều này giúp tiết kiệm thời gian, tránh trùng lặp và dễ bảo trì.

---

### 👉 Hiển thị có điều kiện

Không phải lúc nào cũng cần hiển thị mọi thứ – đôi khi bạn chỉ muốn hiện một phần giao diện nếu có dữ liệu, nếu người dùng đăng nhập, hoặc khi có lỗi xảy ra.

Vue hỗ trợ hiển thị có điều kiện rất linh hoạt:

- Hiện khi có dữ liệu
- Ẩn khi chưa sẵn sàng
- Thay đổi giao diện dựa vào trạng thái

Tình huống thực tế: Khi người dùng chưa đăng nhập, bạn có thể hiển thị “Vui lòng đăng nhập để xem giỏ hàng”. Khi họ đăng nhập rồi, phần giỏ hàng mới hiện ra.

**Ghi nhớ:** Vue giúp bạn xử lý cả lập và điều kiện hiển thị dễ dàng, giúp giao diện trở nên thông minh và linh hoạt hơn.

---

## ■ Chương 5: Component – Chia Nhỏ Giao Diện Để Dễ Quản Lý

Trong một ứng dụng lớn, bạn không thể viết toàn bộ giao diện trong một file HTML. Bạn cần chia nhỏ ra thành từng **component** – mỗi component là một phần độc lập như một "khối lego": nút đăng nhập, hộp tìm kiếm, mục sản phẩm,...

Vue cho phép bạn định nghĩa component và sử dụng chúng như những phần tử riêng biệt. Điều này có 3 lợi ích lớn:

1. **Đễ quản lý:** Mỗi phần tử có thể tách biệt logic và giao diện riêng.
2. **Tái sử dụng:** Bạn có thể dùng lại một component ở nhiều nơi.
3. **Tổ chức rõ ràng:** Giao diện có cấu trúc rõ, tránh rối.

Tưởng tượng: Bạn tạo một component “Thẻ sản phẩm”, bên trong có ảnh, tên, giá. Sau đó bạn có thể dùng component này 10 lần để hiện 10 sản phẩm khác nhau, mà không phải lặp lại toàn bộ mã.

**Ghi nhớ:** Component là nền tảng giúp Vue xây dựng ứng dụng lớn mà vẫn gọn gàng, rõ ràng và dễ mở rộng.

---

## ■ Chương 6: Làm Việc Với Vue CLI – Công Cụ Tạo Dự Án Chuyên Nghiệp

Khi bạn làm các dự án lớn hơn, bạn sẽ không muốn tạo file HTML và chèn Vue theo cách thủ công nữa. Thay vào đó, bạn dùng **Vue CLI** – công cụ giúp khởi tạo, cấu trúc và vận hành dự án Vue một cách bài bản.

Vue CLI giúp bạn:

- Khởi tạo dự án chỉ bằng vài lệnh
- Tự động tạo cấu trúc thư mục chuẩn
- Cài sẵn công cụ biên dịch, kiểm tra lỗi, tối ưu mã nguồn
- Chạy thử ứng dụng trên máy tính và xây dựng để đưa lên mạng

Ví dụ: Vue CLI sẽ chia thư mục thành các phần rõ ràng như components, assets, views,... giúp bạn và nhóm làm việc hiệu quả hơn.

**Ghi nhớ:** Vue CLI là bước tiến để bạn từ một người học Vue trở thành người **làm dự án Vue thật sự**.