



# ReactJS

## Phần 1 – Giới thiệu ReactJS

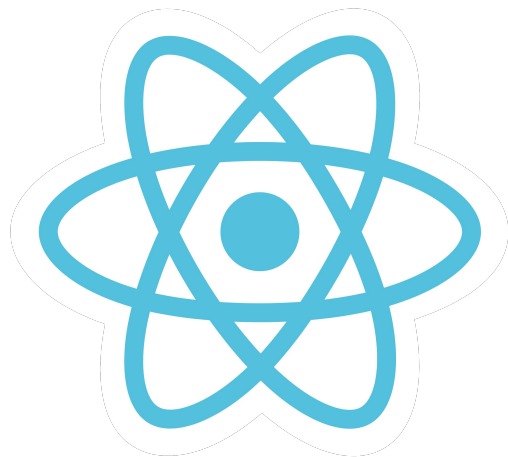
## ReactJS là gì?

- Trang chủ: <https://reactjs.org>
- Là một thư viện Javascript dùng để xây dựng giao diện người dùng.
- Được phát triển bởi **Facebook**.

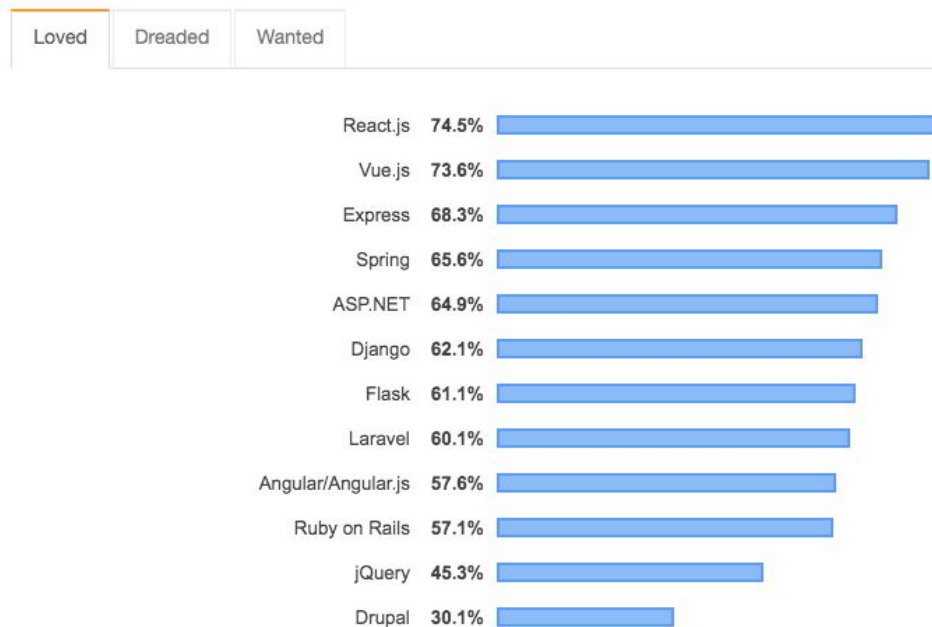


Facebook  
Open Source

Copyright © 2019 Facebook Inc.



### Most Loved, Dreaded, and Wanted Web Frameworks



Thống kê các Web Framework được ưa thích vào năm 2019 trên Stackoverflow

## Một số website nổi tiếng sử dụng ReactJS



Zing MP3

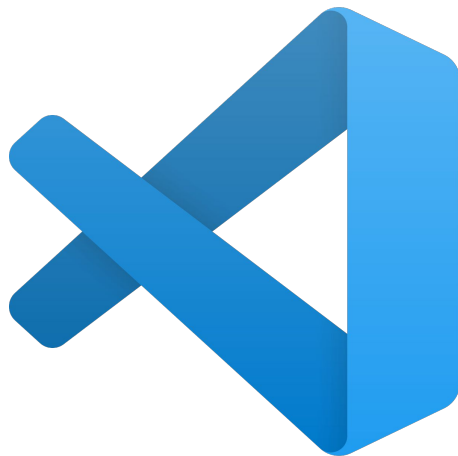


---

# Thiết lập môi trường phát triển

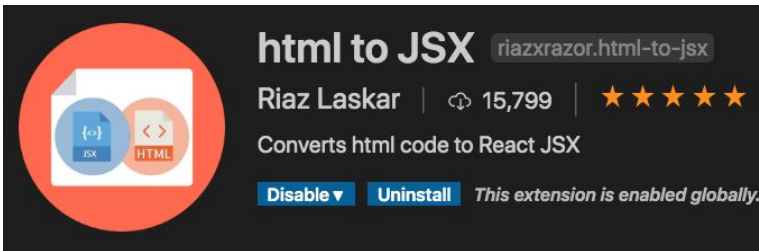
## Visual Studio Code (VS Code)

- VS Code là một Code editor được phát triển bởi Microsoft. Hiện nó đang là Code editor phổ biến nhất cho các dự án Javascript.
- VS Code có rất nhiều Extension đi kèm, giúp việc phát triển web bằng ReactJS dễ dàng hơn.
- Download và cài đặt tại đây: <https://code.visualstudio.com/>



## Một số Extension của VS Code hỗ trợ lập trình React

- Prettier - Code formatter
- HTML to JSX - Converts HTML code to React JSX



## Debug ứng dụng React với React Developer Tools

- React Developer Tools là một Dev Tools extension để debugging và profiling các ứng dụng React trên trình duyệt Chrome và Firefox.
- Link cài trên Chrome: <https://bit.ly/2t30z4q>



### React Developer Tools

Offered by: Facebook

★★★★★ 1,228

Developer Tools

👤 2,012,067 users



## Cài đặt Node.js

- Trang chủ: <https://nodejs.org/en/>
- Cài đặt phiên bản LTS (Long Term Support)
- Sau khi cài xong, kiểm tra version bằng lệnh:

```
> node --version
```





## Giới thiệu Create React App

- Tool giúp thiết lập môi trường phát triển ứng dụng React chỉ với 1 câu lệnh.
- Source code + Hướng dẫn: <https://github.com/facebook/create-react-app>

```
npx create-react-app my-app  
cd my-app  
npm start
```



Edit `src/App.js` and save to reload.

[Learn React](#)

Giao diện mặc định khi tạo project bằng Create React App

---

# Giới thiệu JSX

## Sử dụng React trên trang web

- Nhúng thư viện tương tự jQuery, cần có: **react** và **react-dom**.
- Render giao diện bằng code Javascript.



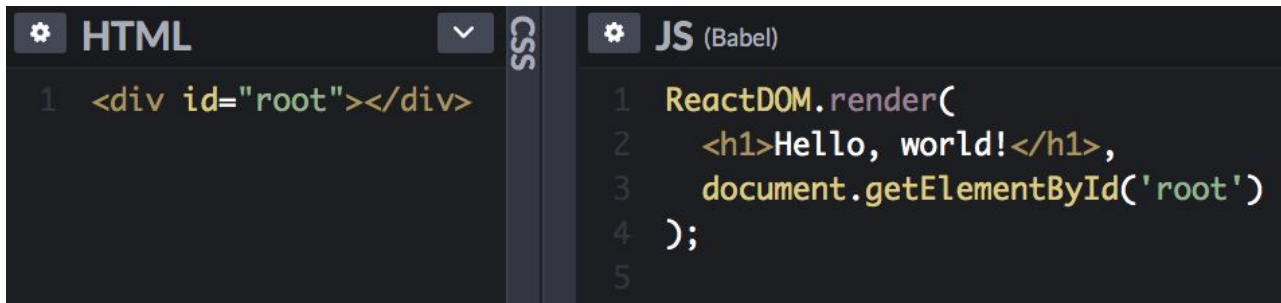
```
HTML
1 <div id="root"></div>

CSS

JS
1 ReactDOM.render(
2   React.createElement("h1", null, "Hello, world!"),
3   document.getElementById("root")
4 );
5
```

## Giới thiệu JSX

- Là cú pháp mở rộng cho Javascript, giúp viết code React dễ dàng và dễ đọc hơn.
- Để trình duyệt hiểu được code **JSX** sẽ cần 1 trình biên dịch như [Babel](#).
- Có thể coi **JSX** như 1 template language với đầy đủ chức năng của Javascript:



The image shows a code editor with two panels. The left panel is titled 'HTML' and contains a single line of code: `<div id="root"></div>`. The right panel is titled 'JS (Babel)' and contains a JavaScript snippet: `ReactDOM.render(  
 <h1>Hello, world!</h1>,  
 document.getElementById('root')  
)`. A vertical tab labeled 'CSS' is visible between the two panels.

## Nhúng biểu thức Javascript trong JSX

- Để nhúng biểu thức JSX trong Javascript, ta viết trong 1 cặp ngoặc nhọn { }.
- Bản thân JSX cũng được coi là 1 biểu thức.

```
const name = 'Josh Perez';  
const element = <h1>Hello, {name}</h1>;  
  
ReactDOM.render(  
  element,  
  document.getElementById('root')  
)
```

## Chú ý khi sử dụng JSX

- Các thuộc tính của thẻ phải viết dưới dạng camelCase giống Javascript DOM. Ví dụ **class** viết thành **className**, **onclick** viết thành **onClick**, ...
- Một element viết bằng JSX phải nằm trong 1 thẻ, có thể là thẻ bất kỳ hoặc thẻ [<React.Fragment>](#). Ví dụ sau là sai:

```
1  const element = <h1>Heading</h1><p>Content</p>;
```

Adjacent JSX elements must be wrapped in an enclosing tag.  
Did you want a JSX fragment <>...</>? (1:32)



---

# Làm quen React qua ví dụ Shopping Cart

# Shopping Cart

Home > Shopping Cart

3 items in the bag



## Laptop HP 348

HP Business laptops and desktops are built for security and collaboration.

**\$430.58**

2

×



## Laptop Asus VivoBook X509FJ

Experience the power of an everyday classic.

**\$702.12**

1

×

Have A Promo Code?

Subtotal **\$1,563.28**

VAT (10%) **\$156.33**

**Total \$1,719.61**

Check Out >

Ứng dụng demo trang Shopping Cart: <https://codepen.io/robinhuy/full/VOVbOw>



## Các chức năng cần bổ sung

- Thay đổi số lượng sản phẩm. Nâng cao: Có validate dữ liệu.
- Xóa sản phẩm. Nâng cao: Khi xóa sản phẩm có Confirm, nếu hết sản phẩm có thông báo.
- Tính lại số lượng sản phẩm trong giỏ hàng.
- Tính lại tổng tiền. Nâng cao: Tính tổng tiền khi có Promo Code.

=> Các bước thực hiện tham khảo: [Thinking in React](#).



## Bước 1: Chuyển từ giao diện tĩnh sang React

- Chuyển toàn bộ giao diện vào 1 Component gốc: App component.
- Chuyển code HTML thuần sang JSX (có thể dùng extension HTML to JSX của VS Code).
- Import CSS vào Component.



## Bước 2: Tách Component

- Chia nhỏ ứng dụng ra thành các phần nhỏ trên giao diện (component) cho dễ quản lý và bảo trì, đồng thời còn có thể tái sử dụng lại ở nhiều chỗ.
- Component có thể viết dưới 2 dạng: **Function Component** (khuyến khích dùng) và **Class Component**.
- **Component** có thể lồng nhau (cha con) và truyền dữ liệu lẫn nhau qua **Props**.
- Tham khảo: [Components and Props](#).



## Bước 3: Định nghĩa State

- Dựa vào giao diện, phân tích dữ liệu động (thay đổi theo thao tác của người dùng). Sau đó lưu dữ liệu vào State của Component (chú ý phân biệt giữa State và Props).
- Tham khảo các cú pháp dùng Vòng lặp, Điều kiện trong JSX để viết lại phần render theo State.
- Tham khảo:
  - [State and Lifecycle](#) (tạm bỏ qua Lifecycle).
  - [Conditional Rendering](#).
  - [Lists and Keys](#).

## Bước 4: Viết hàm thay đổi State khi có sự kiện xảy ra

- Xử lý sự kiện trong React tương tự trong HTML nhưng có 1 chút khác biệt:
  - Tên sự kiện viết theo kiểu camelCase.
  - Trong JSX thì truyền vào một hàm thay vì một chuỗi gọi hàm như HTML.

```
<button onclick="activateLasers()">  
  Activate Lasers  
</button>
```

HTML

```
<button onClick={activateLasers}>  
  Activate Lasers  
</button>
```

JSX

- Tham khảo:
  - [Handling Events](#).
  - [Forms](#).



## Bước 5: Truyền dữ liệu giữa Component cha con

- Khi các Component sử dụng dữ liệu liên quan đến nhau, ta sẽ chuyển hết dữ liệu từ các Component con lên Component cha (*single source of truth*).
- Component cha sẽ truyền dữ liệu đến Component con qua Props và Component con gọi function của Component cha để xử lý dữ liệu (dữ liệu chỉ thay đổi ở Component cha).
- Tham khảo: [Lifting State Up](#).