



---

# Tổng Quan về ReactJS

Khóa học: ReactJS

# Mục tiêu

---



- Giới thiệu tổng quan về ReactJS
- Tìm hiểu Server Side Rendering (SSR) và Client Side Rendering(CSR)
- Biết được một số Toolchain phổ biến
- Cách cài đặt môi trường phát triển

# ReactJS là gì?

---



## Giới thiệu

- ReactJS là thư viện JavaScript được sử dụng để xây dựng các thành phần UI có thể tái sử dụng.
- ReactJS được phát triển bởi Facebook. thường được sử dụng để phát triển ứng dụng single-page, di động, server-side render.
- ReactJS trừu tượng hóa DOM. cung cấp một mô hình lập trình đơn giản hơn và hiệu suất tốt hơn.

# ReactJS là gì?

---



## Tính năng của ReactJS

- JSX – là phần mở rộng cú pháp JavaScript. Không nhất thiết phải sử dụng JSX trong phát triển React, nhưng nó được khuyến khích.
- Component – Làm việc với React là làm việc với các Component, nó giúp bạn dễ dàng chỉnh sửa, phát triển khi làm việc trên các dự án quy mô lớn.
- One-way và Flux – React triển khai luồng dữ liệu một chiều giúp bạn dễ dàng suy luận về ứng dụng của mình. Flux là một mẫu giúp giữ dữ liệu của bạn theo một phương hướng duy nhất

# ReactJS là gì?

---



## Ưu điểm của ReactJS

- Bạn hoàn toàn có thể sử dụng ReactJS để viết ứng dụng trực tiếp bằng JavaScript.
- Biến các thành phần UI phức tạp trở thành những component độc lập.
- Chuyển các dữ liệu đã được tùy biến đến một UI component cụ thể.
- Giúp bạn có thể thay đổi trạng thái cho nhiều component con và không gây ảnh hưởng đến component gốc.
- Khả năng xác định chính xác khi nào cần render lại cũng như khi nào bỏ đi phần tử DOM.

# ReactJS là gì?

---



## Nhược điểm của ReactJS

- Reactjs chỉ là thư viện giúp render ra phần view. React không phải là một MVC framework, vì thế nó không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax.
- Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
- React khá nặng nếu so với các framework khác React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.
- Khó tiếp cận cho người mới học Web.

# Server Side Rendering (SSR)

---



## Giới thiệu

- Khái niệm Server side rendering (SSR) đã được biết tới ngay thời kỳ đầu khi web mới phát triển. Bạn request một trang web, server xử lý nội dung thành HTML, và trả lại cho browser hiển thị lại lên màn hình.

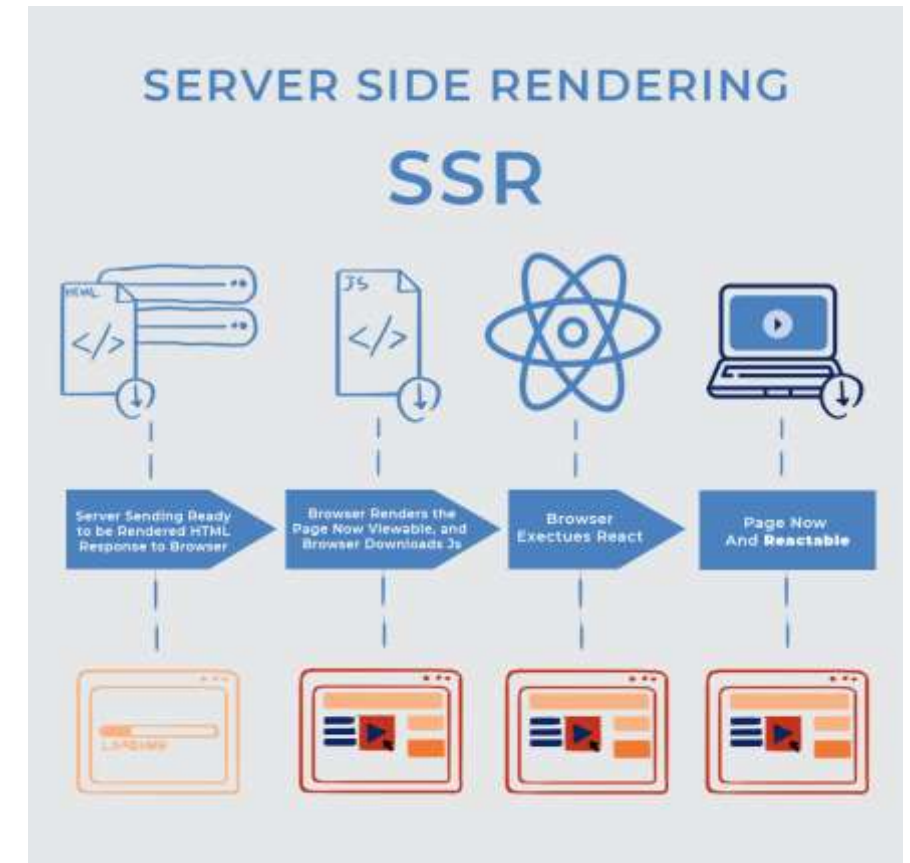
# Server Side Rendering (SSR)



## Cách thức hoạt động

Nội dung HTML được xử lý trên server và trả lại cho browser nội dung HTML đã hoàn chỉnh và được hiển thị ngay khi nó được load về máy.

Tuy nhiên lúc này trang web mới chỉ xem được nội dung, người dùng chưa thể tương tác được. Trình duyệt vẫn âm thầm tải tiếp JS và thực thi nó ngay khi hoàn tất, một khi hoàn tất quá trình này người dùng mới có thể tương tác được.



Nguồn: Kruschecompany



# Server Side Rendering (SSR)

---



## Nhược điểm

- Trang web phải xử lý lại hoàn toàn và load lại từ đầu nếu chỉ có một thay đổi nhỏ trong nội dung. (Ví dụ tiêu đề thay đổi ...)
- Việc xử lý nội dung HTML khiến hao tốn tài nguyên server, gây chậm trễ khi xử lý các request khác.
- Lượng request lên server rất nhiều, do mọi tác vụ đều phải xử lý lại trên server và render lại HTML.
- TTFB (Time To First Byte) cao do cần phải xử lý trên server, sẽ ảnh hưởng tới trải nghiệm của người dùng.

# Client Side Rendering (CSR)

---



## Giới thiệu

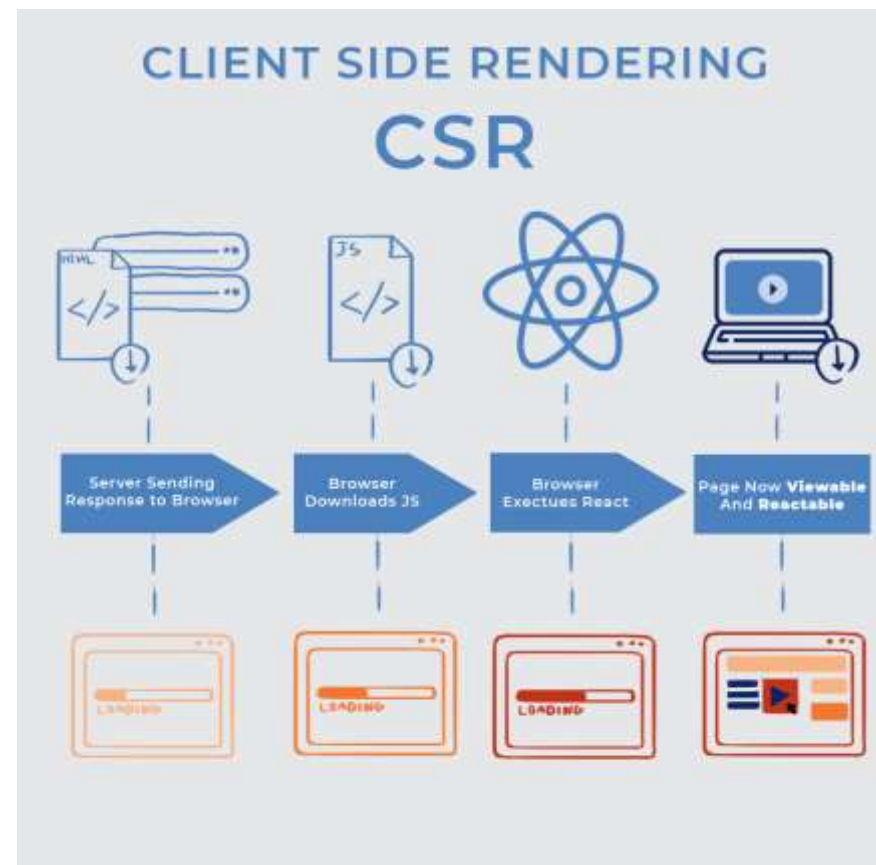
- Hiện nay những chiếc máy tính cá nhân ngày càng trở nên mạnh mẽ, việc render HTML thuần không thể tận dụng được hết sức mạnh của chúng, việc sử dụng chính chúng để xử lý nội dung thay vì đổ tất cả cho server trở thành một lựa chọn của các nhà phát triển.

# Client Side Rendering (CSR)



## Cách thức hoạt động

Như trên hình, ta có thể nhận thấy trình duyệt sẽ load tất cả nội dung về máy, một khi đã load thành công, nó sẽ thực thi code, lúc này nội dung trang web mới được hiển thị ra. Việc sinh nội dung ... đều được thực hiện ngay trên client, server chỉ có nhiệm vụ trả mã nguồn JS và dữ liệu thô cho client.



Nguồn: Kruschecompany

# Client Side Rendering (CSR)

---



## Nhược điểm

- Lượng dữ liệu lần đầu load về khá nặng
- Nội dung hiển thị ra lần đầu chậm hơn so với SSR, tuy nhiên những lần sau thì quá nhanh do việc xử lý ngay trên client.
- SEO bị ảnh hưởng, do nội dung web được sinh trên client, khiến crawler của của Search engine không tiếp xúc được nội dung. Gần đây mới có Google giải quyết được vấn đề này, tuy nhiên dù Google nắm phần lớn thị phần tìm kiếm thì tại những quốc gia khác nhau, sẽ có ảnh hưởng nhất định.

# Chọn SSR hay CSR?

---



SSR hay CSR đều có những điểm mạnh và điểm yếu khác nhau. Do đó, việc lựa chọn còn phụ thuộc vào dự án bạn thực hiện, quốc gia cũng như những yêu cầu đặc biệt của dự án, không có cách nào áp chế hay thay thế hoàn toàn cách còn lại. Chúng song hành cùng nhau và lựa chọn sử dụng chúng một cách khôn ngoan là việc của chúng ta.

# Toolchain

---



Toolchain là chuỗi công cụ tích hợp trong quá trình phát triển giúp đem đến trải nghiệm tốt nhất cho người dùng và nhà phát triển như:

- Mở rộng quy mô cho nhiều file và component.
- Sử dụng thư viện của bên thứ ba từ npm.
- Phát hiện sớm các lỗi thường gặp.
- Chỉnh sửa trực tiếp CSS và JS đang được phát triển.
- Tối ưu hóa đầu ra cho production.

# Một số toolchain phổ biến

---



- [npm](#) (Node Package Manager) - một trình quản lý các thư viện trong môi trường Node.js. Các thư viện được đóng gói dưới dạng package và được lập trình viên đóng góp, chia sẻ trên npm registry (cơ sở dữ liệu của npm)
- [npx](#) (Node Package Execute) - là một công cụ để chạy package mà không cần phải cài đặt.
- [Yarn](#) là công cụ quản lý package tương tự như npm và được phát triển bởi Facebook. Lập trình viên có thể thực hiện các câu lệnh để cài đặt, thực thi các câu lệnh thông qua CLI.
- [webpack](#) cho phép bạn viết code mô-đun và gói nó lại với nhau thành các gói nhỏ để tối ưu hóa thời gian tải.
- [Babel](#) là một trình biên dịch javascript, nó được sử dụng để chuyển đổi mã JavaScript chuẩn mới (ECMAScript 2015+) về phiên bản cũ hơn để đảm bảo chương trình của bạn có thể hoạt động bình thường trong các trình duyệt cũ.
- [Next.js](#) là một framework phổ biến và nhẹ cho ứng dụng static và server-rendered được xây dựng bằng React.
- [Gatsby](#) giúp tạo website tĩnh với React. Nó cho phép bạn sử dụng các component của React, nhưng xuất ra HTML và CSS được render trước để đảm bảo thời gian tải nhanh nhất.

# Cách cài đặt môi trường phát triển

---



Tải và cài đặt

- NodeJS tại <https://nodejs.org>
- Visual Studio Code tại <https://code.visualstudio.com>

**Cài đặt ReactJS theo hai cách:**

- Sử dụng webpack và babel.
- Sử dụng câu lệnh *create-react-app*.



# Cách cài đặt môi trường phát triển

---



Tải và cài đặt

- NodeJS tại <https://nodejs.org>
- Visual Studio Code tại <https://code.visualstudio.com>

**Cài đặt ReactJS theo hai cách:**

- Sử dụng webpack và babel.
- Sử dụng câu lệnh *create-react-app*.

# Cách cài ReactJS bằng webpack và babel



- Khai báo file **package.json**
- Cài đặt **react, react-dom**
- Cài đặt **webpack, webpack-cli, webpack-dev-server**
- Cài đặt **@babel/core , babel-loader , @babel/preset-env, @babel/preset-react**
- Cài đặt **html-webpack-plugin**
- Tạo các file cấu hình, file code
- Khởi chạy dự án
- Build file bundle

# Cách cài ReactJS bằng create-react-app



- Sử dụng câu lệnh **npx create-react-app my-app** để tạo dự án
- Khởi chạy dự án

# Tổng kết

---



- Tổng quan về ReactJS là gì, có những ưu, nhược điểm gì.
- Cơ chế Server Side Rendering (SSR) và Client Side Rendering(CSR) cùng ưu và nhược điểm của chúng.
- Biết được một số Toolchain phổ biến
- Hai cách cài đặt môi trường phát triển