**Tìm hiểu react js**

1. **Npm:**

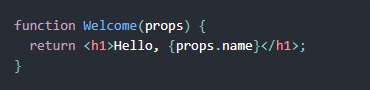
* Là viết tắt của node package manager là một công cụ tạo và quản lý các thư viện lập trình js cho Node.js
* Để cài đặt chỉ cần thực thi lệnh npm I package-name
* Có 2 cách để cài đặt một gói bằng npm:
* Local: tạo ra thư mục node\_modules nếu chưa có hoặc có rồi thì nó sẽ lấy code của gói cần cài đưa vào đây. Khi dùng thì sử dụng lệnh require
* Global: lưu trữ code của gói trong các file hệ thống cố định trong máy, chỉ có thể dùng các gói này thông qua các hàm CLI (command line interface) như gulp. Không thể dùng package thông qua require()
* Để quản lý các gói cài đặt cục bộ bằng npm thì thông qua file package.json. File này chứa các nội dung:
* Các gói thư viện lập trình mà pj sử dụng
* Xác định phiên bản các gói thư viện đó
* Các gói bạn xây dựng có thể chia sẻ dễ dàng với các lập trình viên khác trên toàn cầu thông qua npm
* Nếu muốn thêm các entry vào thuộc tính dependencies khi cài đặt sử dụng thêm flag –save, còn với thuộc tính devDependencies thì dùng flag –save-dev
* Webpack là công cụ giúp compile các module js
* Webpack-dev-server: cung cấp việc thiết lập dev web server, có một số tính năng sau:
* Live reloading –inline: khi lưu code lại thì ta có thể thấy nó refresh lại trình duyệt và cập nhật sự thay đổi
* Hot module replacement: cập nhật lại module thay đổi trong lúc đang chạy mà không cần refresh
* Dependencies: những module được dùng trong quá trình chạy sản phẩm
* devDependencies: những module dùng vào mục đích phát triển sản phẩm

1. **JSX**:

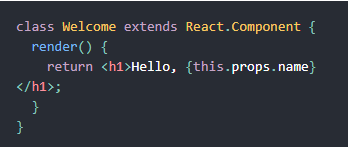
* Được dùng để tạo template thay vì js thông thường, không cần thiết phải sử dụng. Ưu điểm:
* Nhanh hơn vì tối ưu hóa trong khi biên dịch mã
* An toàn
* Viết mẫu dễ dàng và nhanh chóng hơn

1. **Components và Props**

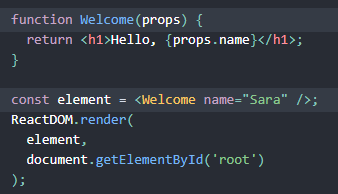
* Component:
* Components đóng vai trò như function trong js nhưng chúng hoạt động độc lập và có nhiệm vụ trả về các thành phần html thông qua hàm render
* Có 2 loại component là function component và class component
* Cách đơn giản nhất để định nghĩa component là viết function js, với response trả về là html, đây cũng là cách định nghĩa function component



* Chúng ta có thể dùng es6 class để định nghĩa component, đây được gọi là class component trong react:



* Khi react phát hiện 1 element là 1 component tự định nghĩa, nó sẽ truyền các jsx attribute vào component đó, và ta gọi nó là props



* Component có thể tham chiếu đến nhau ở output của chúng. Điều này giúp chúng ta có thể dùng kết hợp và tùy biến component, ví dụ 1 component render đến nhiều component khác



* Một component có thể được chia thành nhiều các component nhỏ khác nhau bên trong nó. Ví dụ:



Với những component có prop dạng nested như trên thường khó thay đổi logic hoặc tái sử dụng thành phần bên trong nên chúng ta nên chia nhỏ các thẻ html thành các component con tương ứng

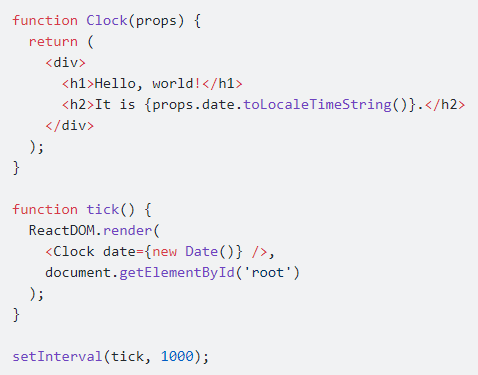
* Props:
* Dù component thuộc loại function hay class thì nó không bao giờ thay đổi props
* React có một quy tắc chung: “Tất cả các component đều phải là pure function, không được thay đổi giá trị của input và output phải luôn cùng kiểu định dạng”
* Nên dùng props cho các component nào mà luôn hiển thị cùng 1 đầu ra cho cùng 1 đầu vào để dễ kiểm soát hơn

1. **State và Lifecycle**

* State và lifecycle:
* Để giải quyết vấn đề props luôn phải là read-only, state đã ra đời
* Giống như props, state có nhiệm vụ chứa thông tin của component
* State chỉ hoạt động trong component của nó (private), là thành phần của component và là loại dữ liệu động (có thể thay đổi)
* Nếu component cần quản lý state thì com đó nên là class com hơn là function com
* Ví dụ dưới đây update UI thông qua việc gọi ReactDOM.render() nhiều lần:



Nếu dùng component và props thì ta được như sau:



Tuy nhiên ta có thể tối ưu bằng cách để component Clock tự khởi tạo biến date và update chính nó theo thời gian, điều này vi phạm qui tắc read-only của props nên chúng ta cần state và lifecycle method.

Việc đầu tiên chính là biến Clock từ function thành class

* Chuyển hàm thành class

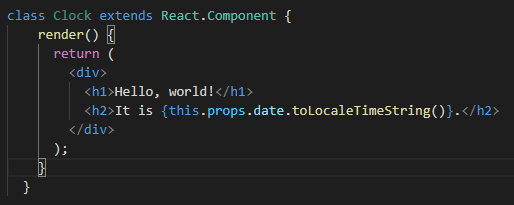
Thực hiện các bước sau:

+ Tạo một class với tên là Clock, extend React.Component

+ Dùng method render() bên trong

+ Chuyển body của function vào hàm render()

+ Dùng this.props. thay vì props.

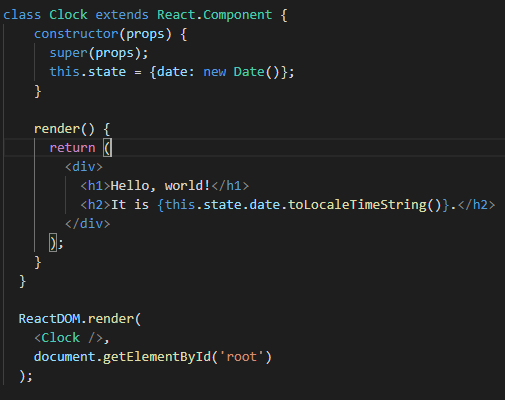


* Chuyển data từ props sang state như sau

+ Chuyển this.props.date thành this.state.date ở method render()

+ Dùng constructor khởi tạo giá trị date và gán vào state

+ Bỏ props date khỏi element <Clock />



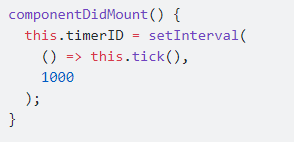
* Dùng lifecycle để update giá trị date theo thời gian và update lại UI tương ứng

+ Khi một app dùng nhiều component, việc giải phóng tài nguyên bị component chiếm dụng sau khi component bị loại bỏ rất quan trọng

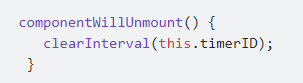
+ Trong bài trên, cần tạo method setInterval để update date, nó sẽ hoạt động ngay sau khi Clock được render vào DOM, điều này được gọi là Mounting. Quá trình xóa bỏ việc đếm này sau khi Clock bị loại bỏ gọi là unmounting

+ Để thực hiện, cần thêm logic tương ứng khi component được mount và unmount thông qua componentDidMount() và componentWillUnmount(). Đây chính là Lifecycle

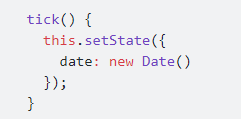
+ Với bài trên, muốn chạy setInterval ngay khi Clock được render thì thêm nó vào componentDidMount():



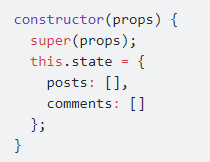
+ Để xóa quá trình đếm, dùng method clearInterval và cho vào componentWillUnmount:



+ Tạo function tick() sử dụng this.setState() để set lại giá trị của date khi tick() được gọi



* Dùng state đúng cách
* Không modify state trực tiếp, vì component tương ứng sẽ không được render lại. Thay vào đó dùng method setState()
* React có thể gộp nhiều lệnh gọi đến setState() vào một batch duy nhất nên this.props và this.state có thể được update bất đồng bộ. Để tránh điều này, có thể dùng dạng khác của setState(), nhận vào function thay vì object
* Bản chất của method setState() là hợp nhất object bên trong vào giá trị state hiện tại



Chúng ta có thể update/modified lại từng field một cách độc lập:

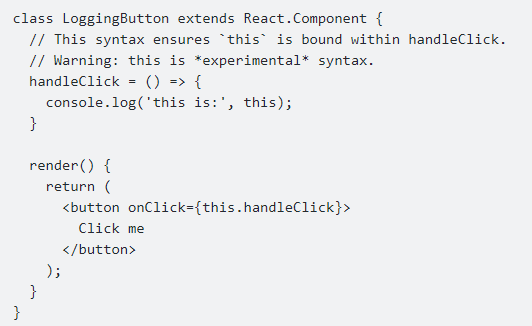


1. **Handling event**

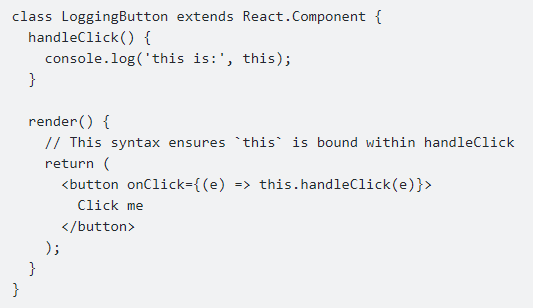
* Xử lý sự kiện trong react cũng giống xử lý sự kiện trong DOM nhưng có một số khác biệt về cú pháp
* Các sự kiện react được đặt tên bằng camelCase, thay vì chữ thường. Ví dụ: onclick -> onClick, onchange->onChange
* Với jsx, truyền một hàm để bắt sự kiện, thay vì một chuỗi như html thông thường
* Không thể dùng return false để chặn các hành động mặc định, trong react phải dùng preventDefault()
* Khi dùng react, không cần gọi function addEventListener để lắng nghe sự kiện vào DOM khi nó tạo đối tượng mới, thay vào đó chỉ cần khai báo một sự kiện lắng nghe khi nó được hiển thị lần đầu



* Trong react, nếu không gọi bằng this.handleClick mà gọi onClick thì this sẽ bị lỗi undefined khi function được gọi
* Nếu không gọi binds thì có 2 cách để gọi
* Dùng field class



* Dùng arrow function



* Để truyền tham số vào event handlers, có thể dùng theo 2 cách dưới đây:



Tham số e đại diện cho react event sẽ được truyền như tham số thứ 2 sau id

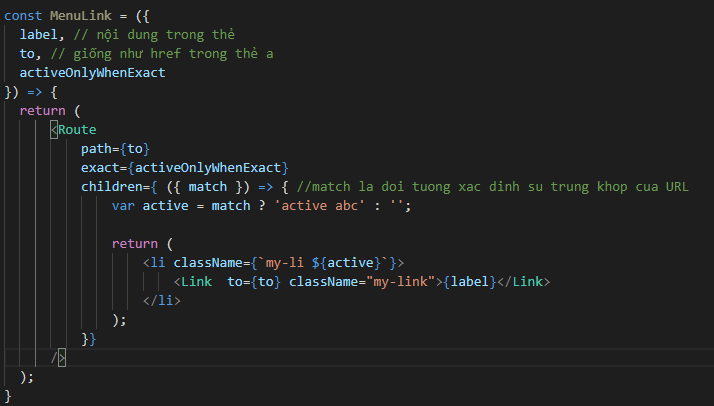
1. **Router**

* React-router là một thư viện định tuyến tiêu chuẩn trong react, giữ cho giao diện của ứng dụng đồng bộ với url trên trình duyệt
* Cho phép định tuyến luồng dữ liệu (data-flow) trong ứng dụng một cách rõ ràng
* Các thành phần:
* BrowerRouter và HashRouter: Hai thành phần này khác nhau ở kiểu url mà chúng sẽ tạo ra và đồng bộ

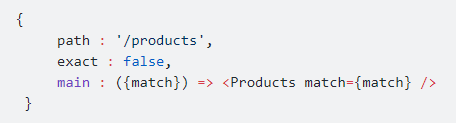
+ BrowerRouter: được dùng phổ biến hơn, nó sử dụng History API có trong html5 để theo dõi lịch sử bộ định tuyến

+ HashRouter: Sử dụng Hash của url để ghi nhớ mọi thứ

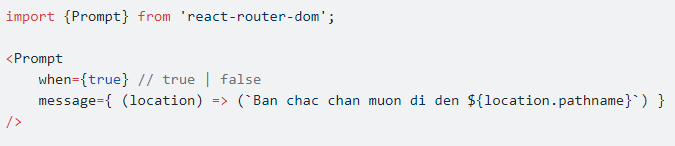
* Route: Định nghĩa một ánh xạ (mapping) giữa một url và một component, có nghĩa khi người dùng truy cập theo một url trên trình duyệt, một component tương ứng sẽ được render trên giao diện
* Link: Trong react dùng cặp thẻ <Link></Link> được import từ react-router để chuyển hướng
* NavLink: Giống với Link về cách dùng nhưng NavLink tốt hơn vì nó hỗ trợ thêm một số thuộc tính như activeClassName và activeStyle, 2 thuộc tính này giúp nó khi trùng khớp sẽ được active và style
* Custom Link: Giúp sử dụng class và style ở thẻ bao ngoài của thuộc tính



* Match: Khi muốn lấy một số thông tin trên url thì có thể dùng đối tượng Match để lấy dữ liệu về. Tại cấu hình router chỉ cần truyền thêm đối tượng match vào component mà cần sử dụng match



* Prompt – xác nhận trước khi chuyển trang: Dùng để xác nhận trước khi back hay chuyển trang. Khi dùng thì import từ react-router





* Redirect: Dùng để chuyển trang và truy xuất thông tin trang trước thông qua đối tượng location. Để dùng thì import từ react-router

+ Khi muốn dùng location thì tại cấu hình router chỉ cần truyền thêm đối tượng location vào component mà cần dùng nó