**Reactjs**

Là 1 opensource để build user interfaces

Ko phải framework

Tập trung UI

Rich ecosystem

Kiến trúc dựa vào component

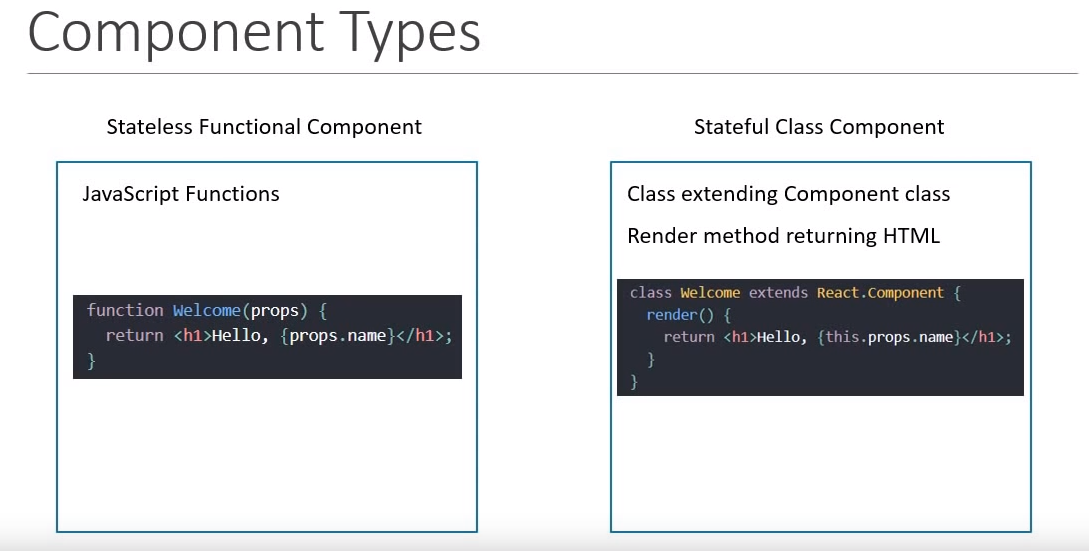
Được phát triển và bảo trì bởi Facebook

Code tái sử dụng

Packeage for VsCode: ***ES7 React/Redux/GraphQL/React-Native snippets***

Knowleage: Fundamentals, Http, Hook, routing, redux, Utilities

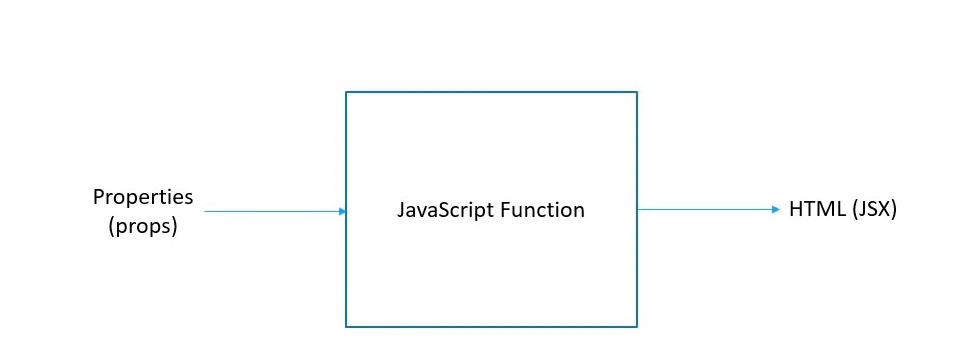
**Component**



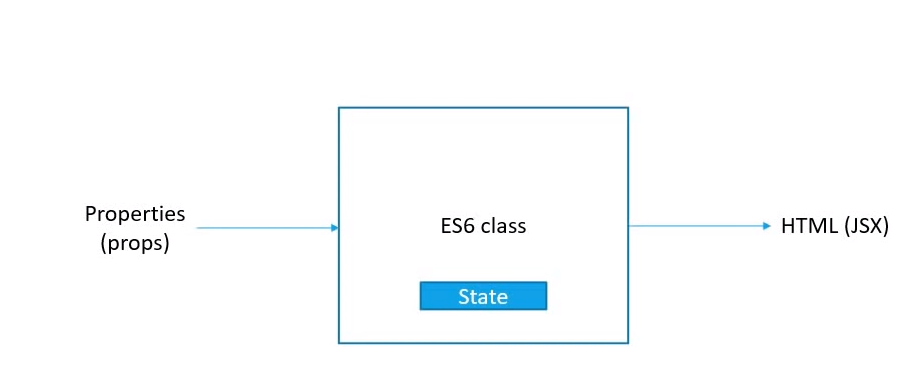
Là 1 thành phần UI, có thể lồng ghép nhau.

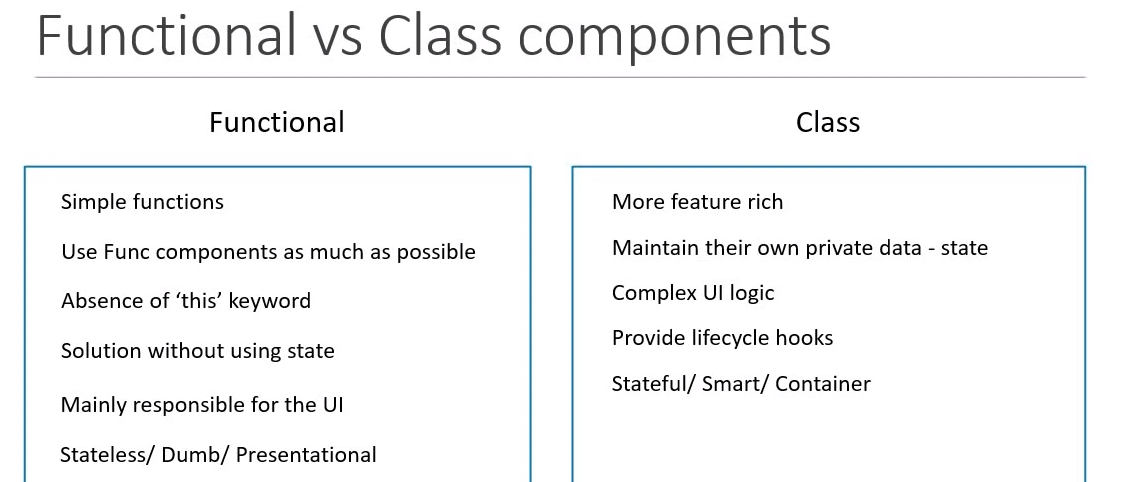
Có 2 loại component: stateless và stateful

**Function**



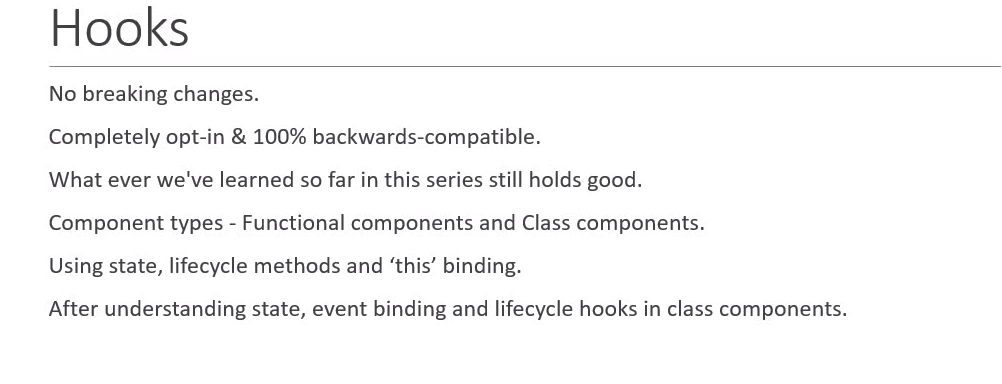
**Class**





**Hook Update (In file ReactHook)**

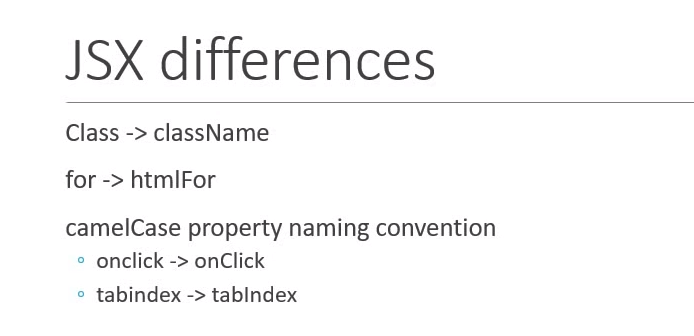
Cho phép sử dụng state, lifecycle, this binding trong function(stateless) component



**JSX**

Cú pháp mở rộng của js để hiển thị UI component

Gồm: tag name, attributes, children

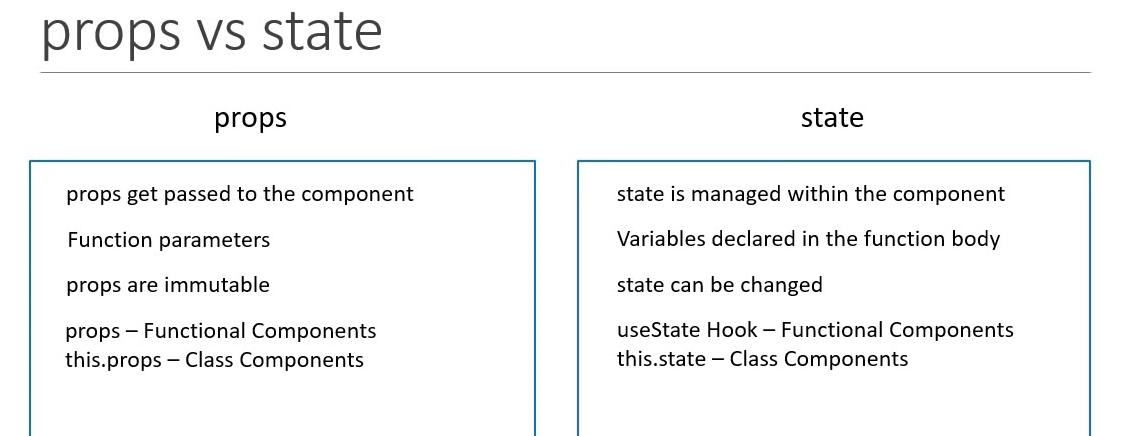


**Props**

Props có thể đến từ parent component hoặc từ chính nó (defaultProp)

Ko nên thay đổi props bằng setProps hoặc replaceProps => dễ dàng kiểm soát nó

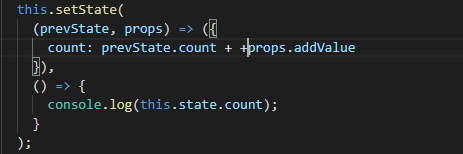
**State**



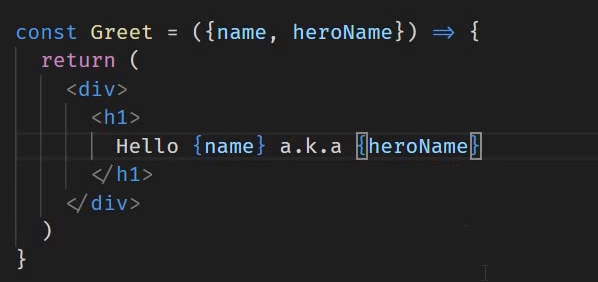
Luôn sử dụng setState và ko thay đổi trực tiếp state

Setstate là async => viết code trong callback

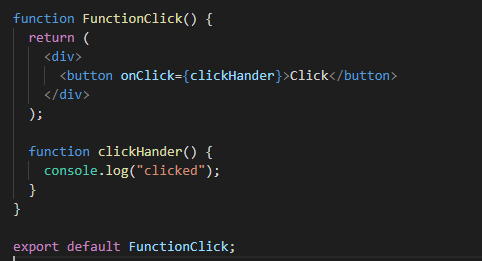
Khi update state dựa vào giá trị state trước đó => pass function như là 1 argument

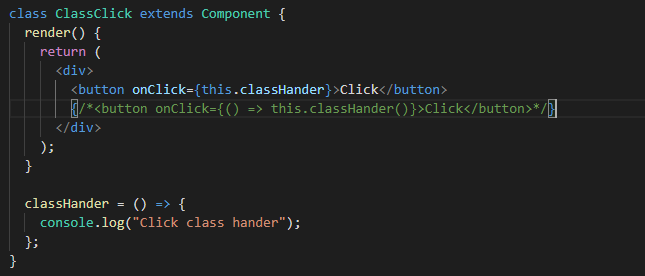


**Destructuring props and state**



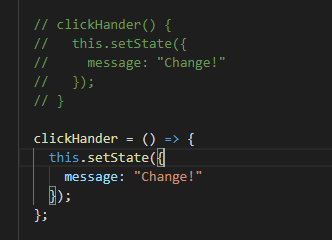
**Event Handing**





**Binding Event Handlers**

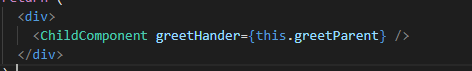
Chú ý từ khóa this

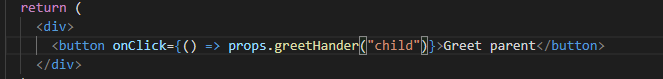






**Methods as prop**





**Conditional Rendering**

**List Rendering**

**Lists and Keys**

Key giúp React xác định item đã đc thay đổi, xóa hoặc thêm.

**Index as key anti-pattern**

**Styling and Css Basics**

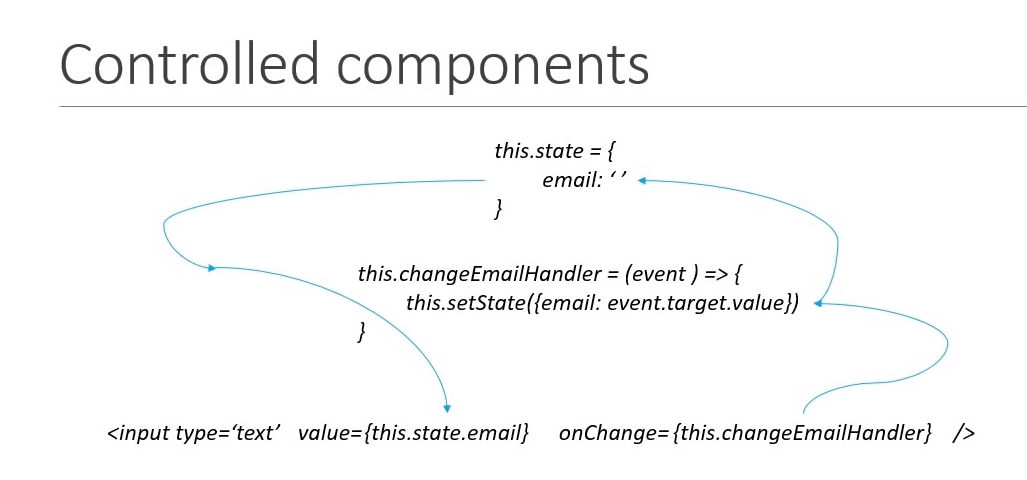
+ Css stylesheet

+ Inline styling

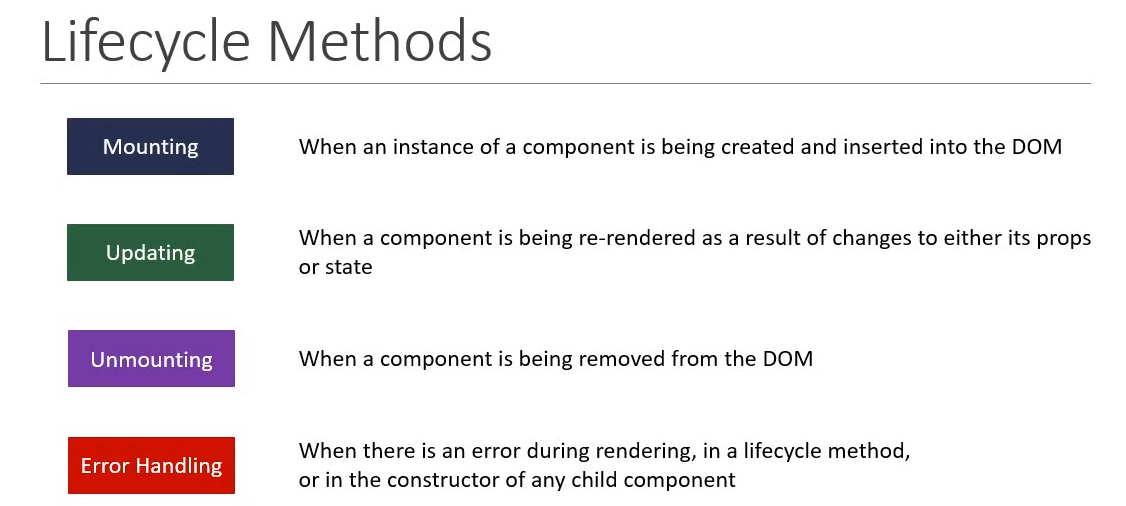
+ Css modules

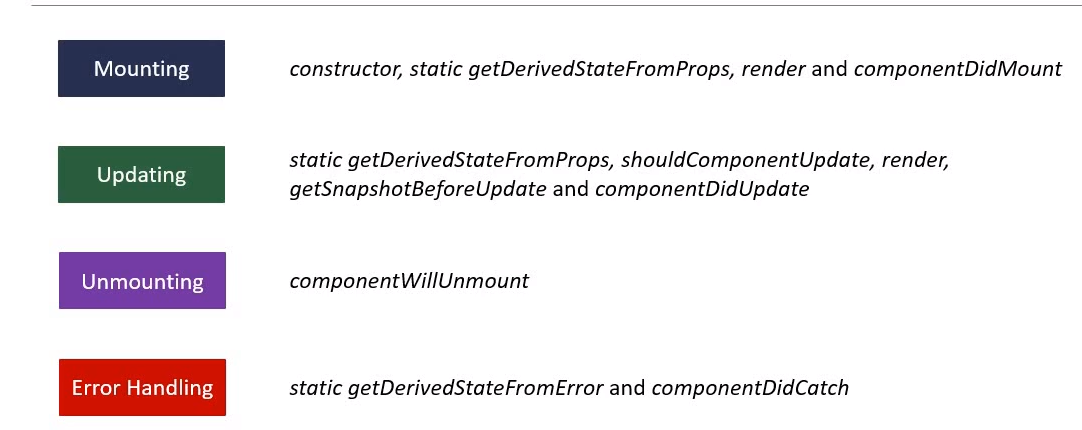
+ Css in Js libraries

**Basic Form Handling**



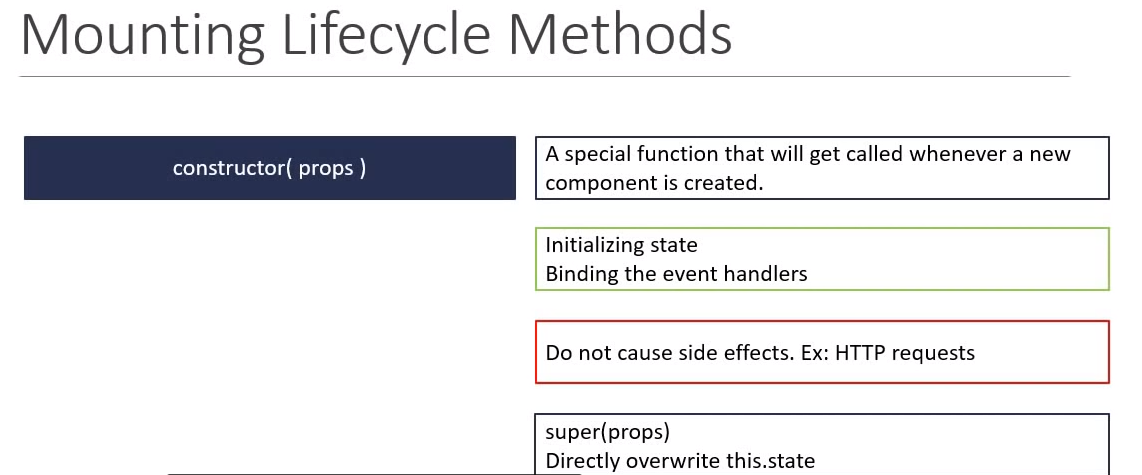
**Component Lifecycle Methods**

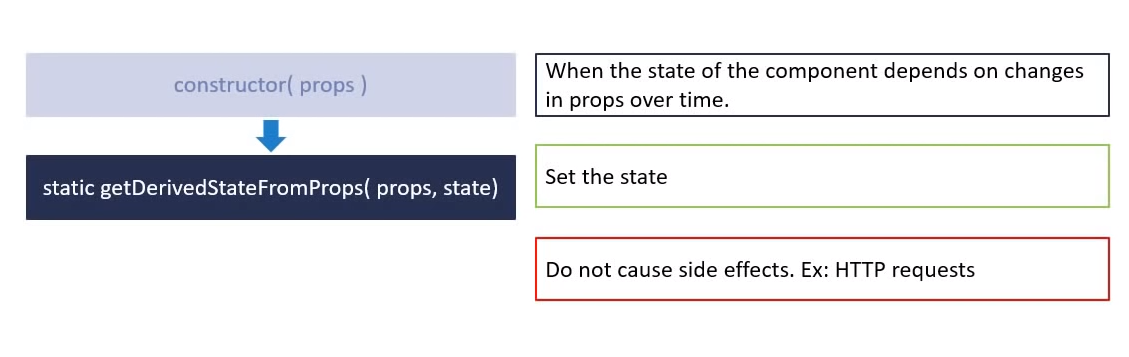


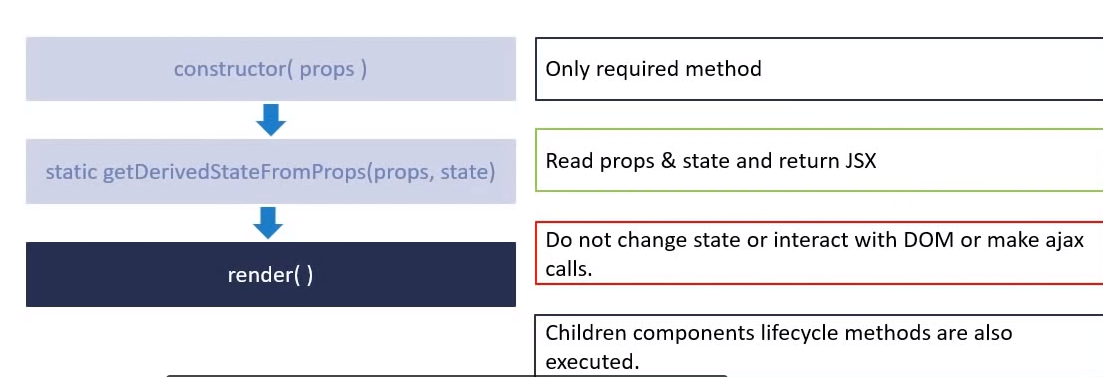


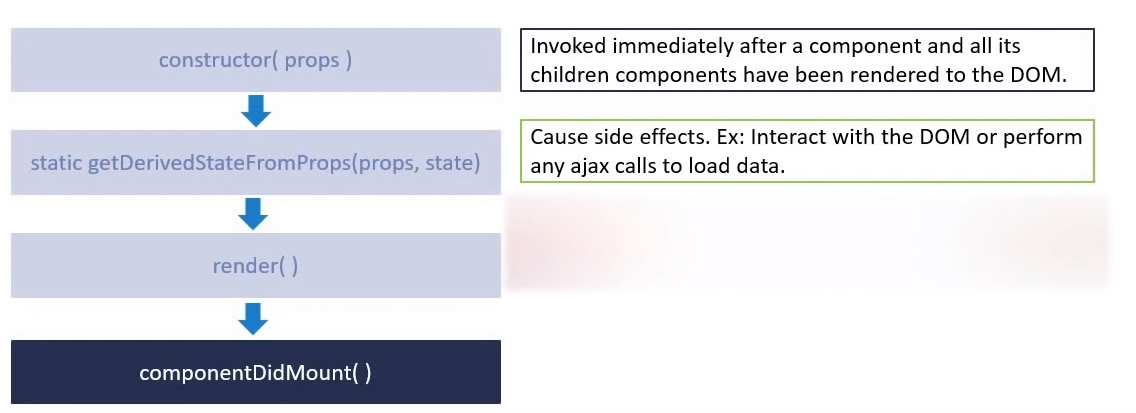
**Mounting Lifecycle Methods**

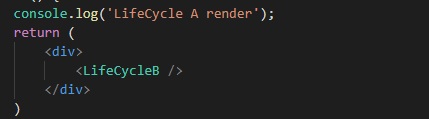
Never make http request from within a constructor

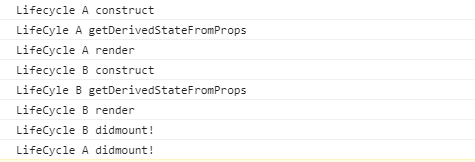




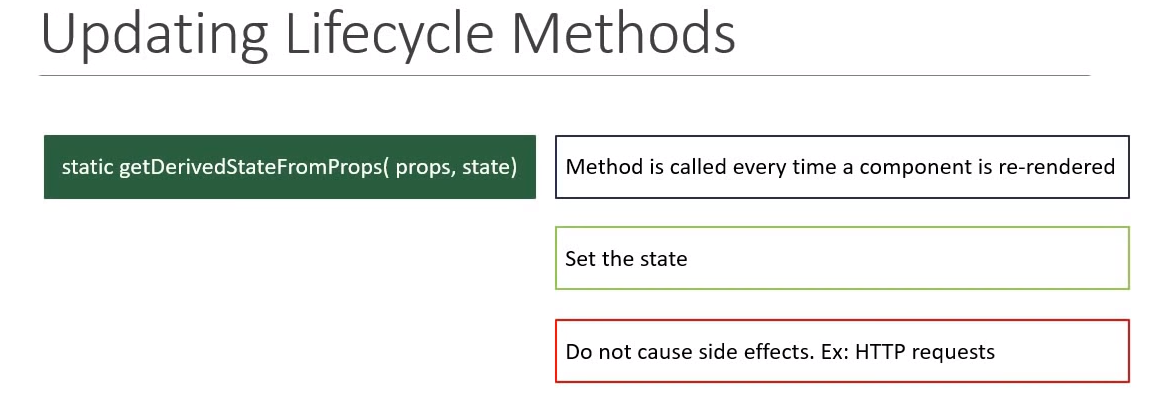


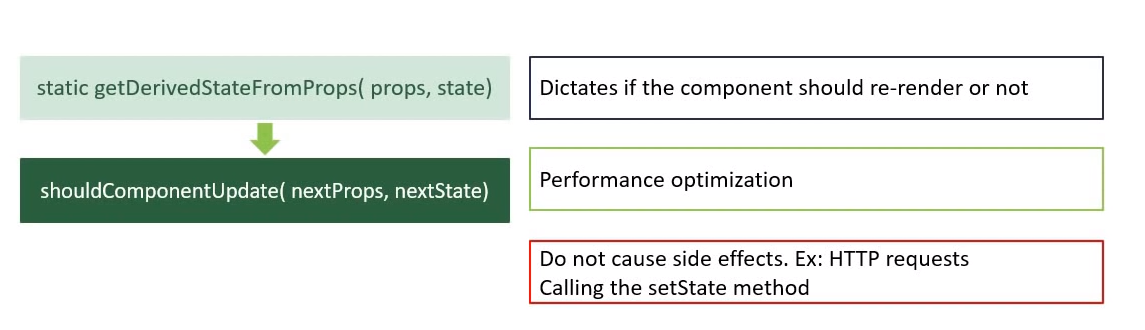


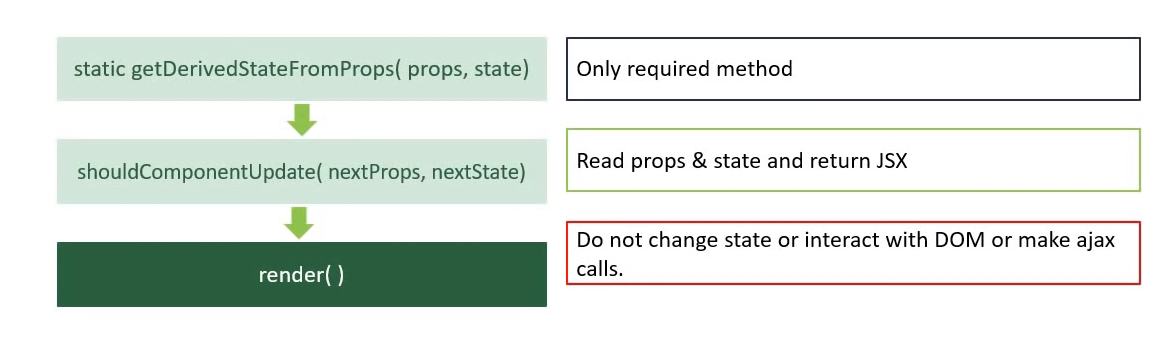


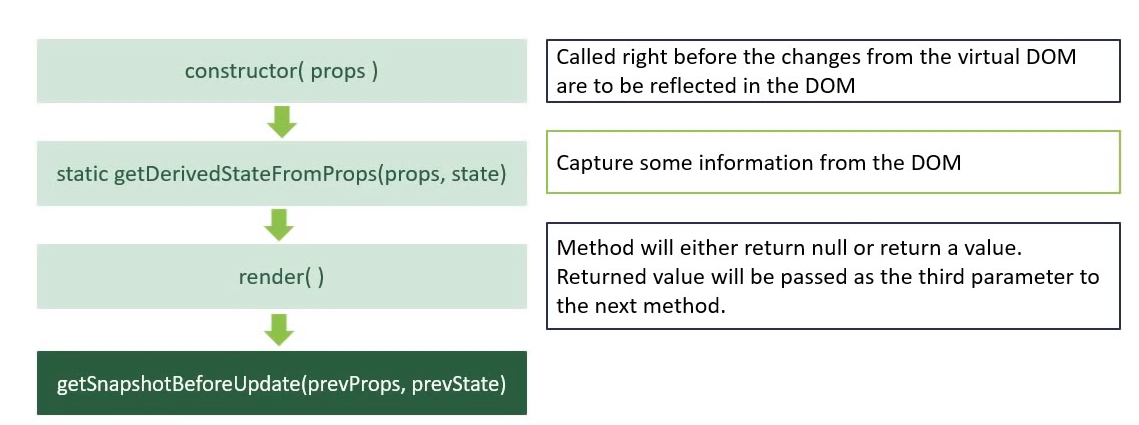


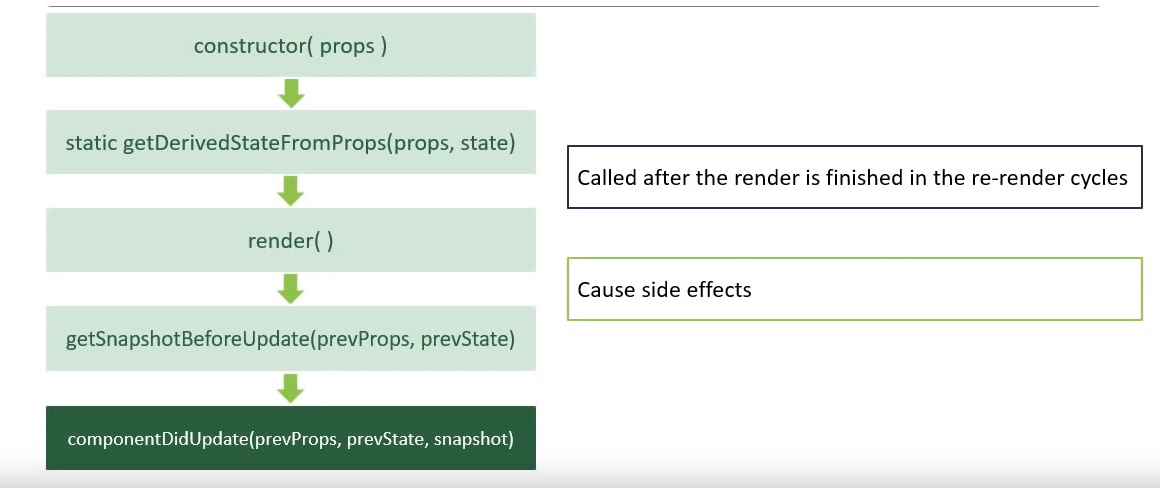
**Updating Lifecycle Methods**

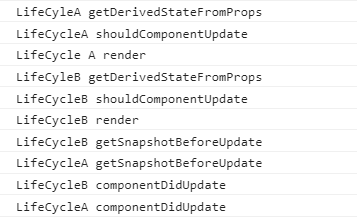




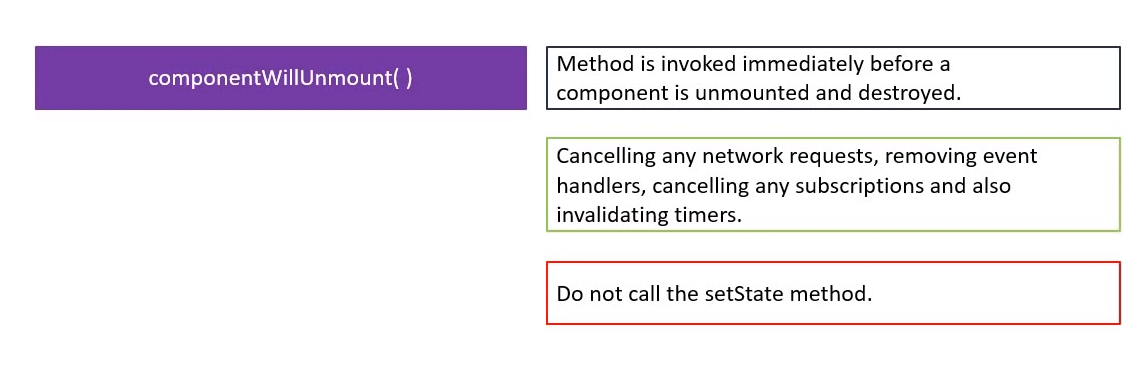




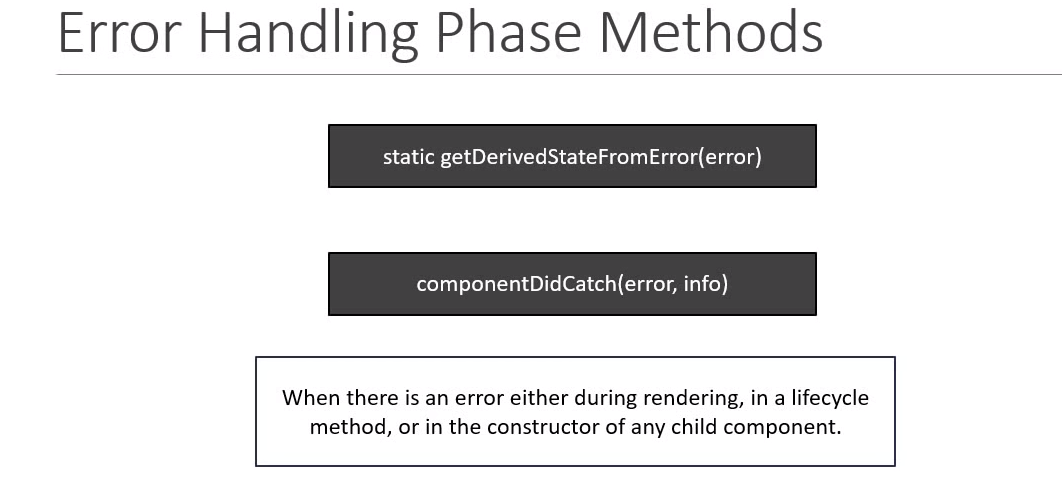




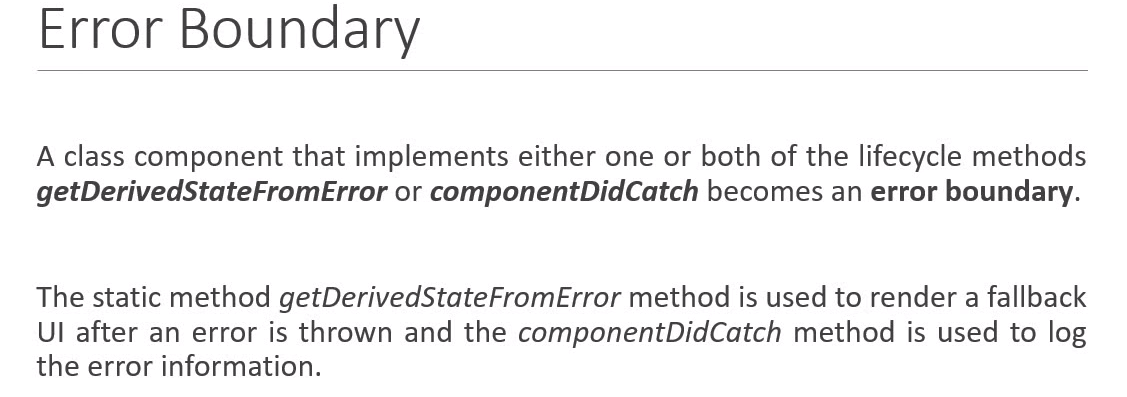
**Unmounting Phase Method**



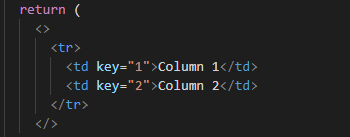
**Error Handing Phase Method**



**Error Boundary**



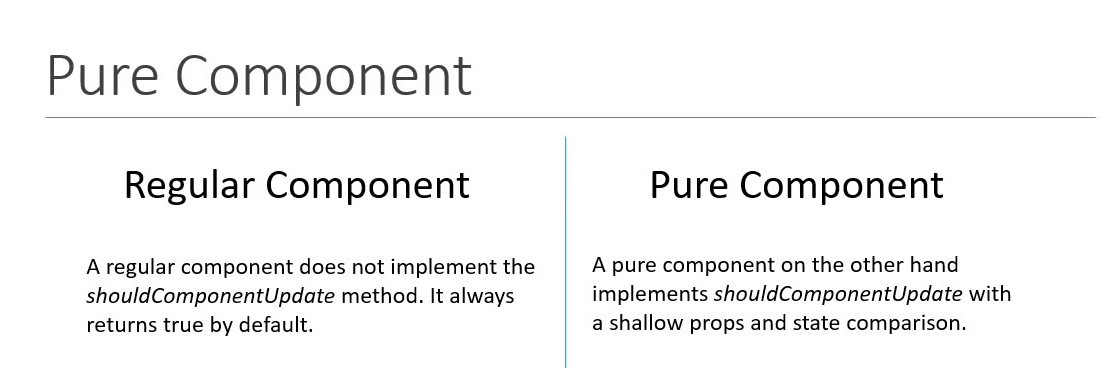
**Fragments**

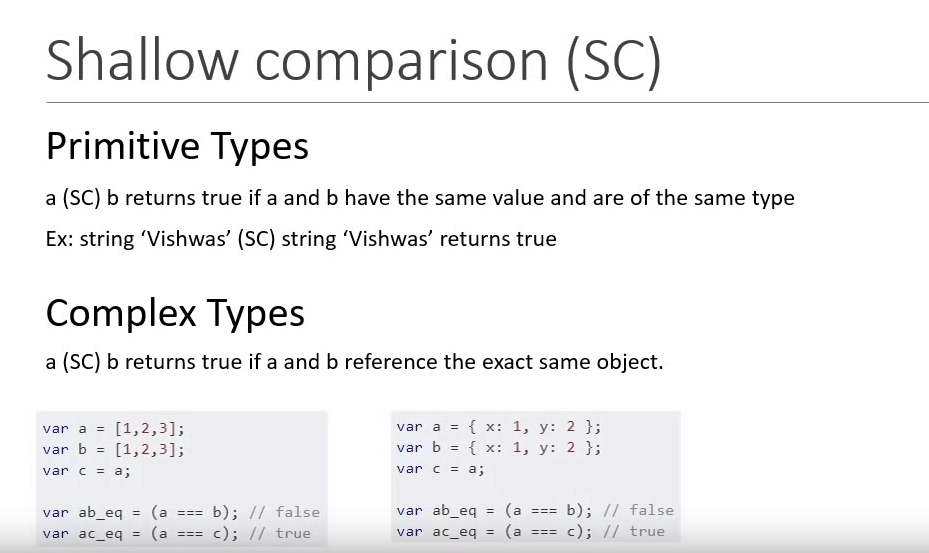


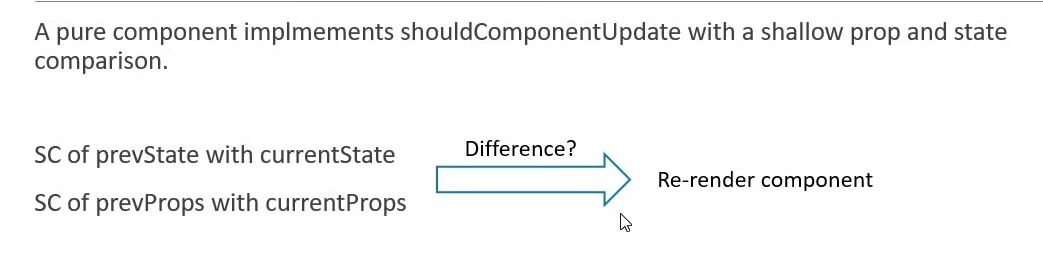
**Pure Component**

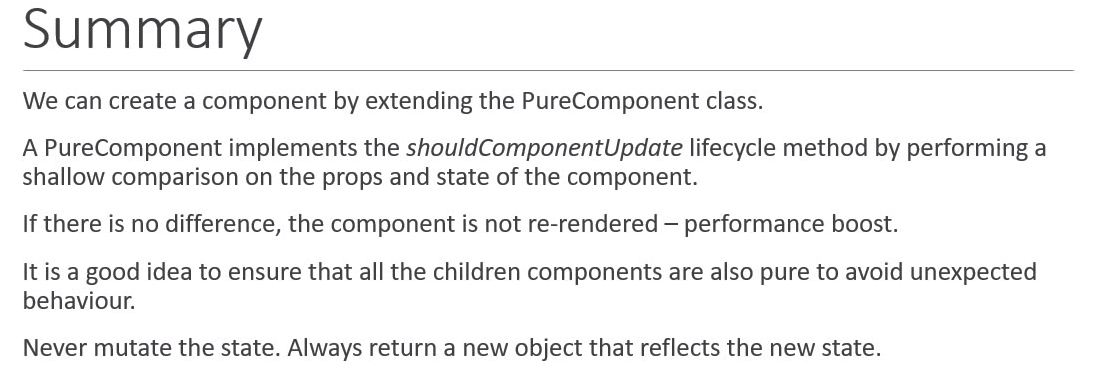
Giống Component ngoại trừ việc nó xử lí shouldComponent.

Khi state hoặc props thay đổi thì sẽ làm một shallow comparison(so sánh) trên cả prop và state, component sẽ ko so sánh => render bất cứ khi nào shouldComponentUpdate đc gọi.









**Memo**

Tránh việc rerender nhiều ảnh hưởng tới performance.

**Refs**

Cho phép truy cập tới Dom Node hoặc React Element được tạo ra trong method render.

**Refs with class Component**

**Forwarding Refs**

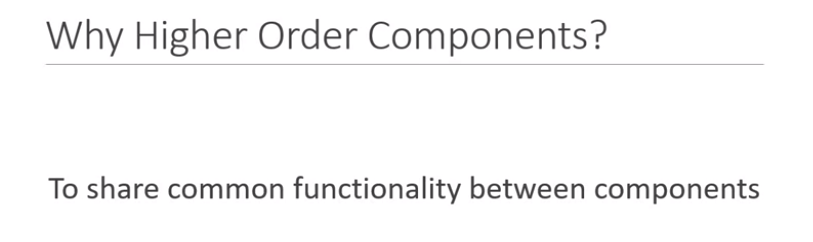
**Portals**

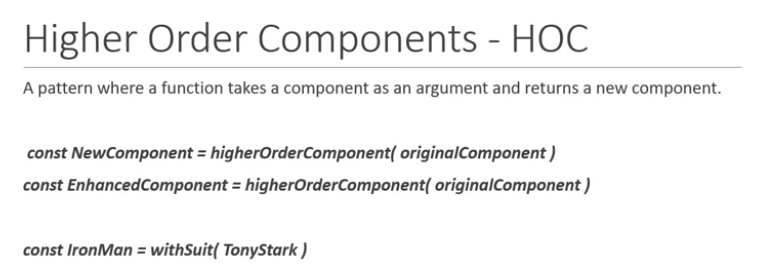
Cho phép render lại 1 phần html độc lập với component tree.

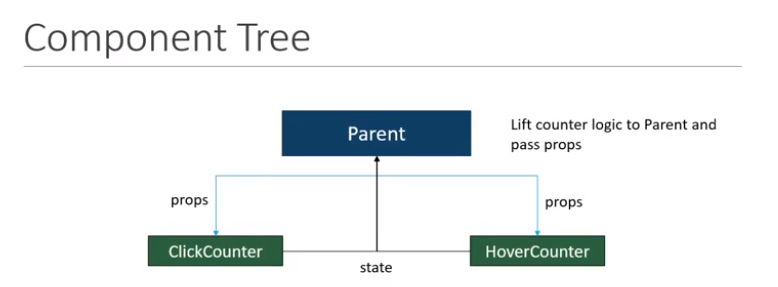
<https://viblo.asia/p/tim-hieu-va-su-dung-react-portal-gDVK2pWnlLj>

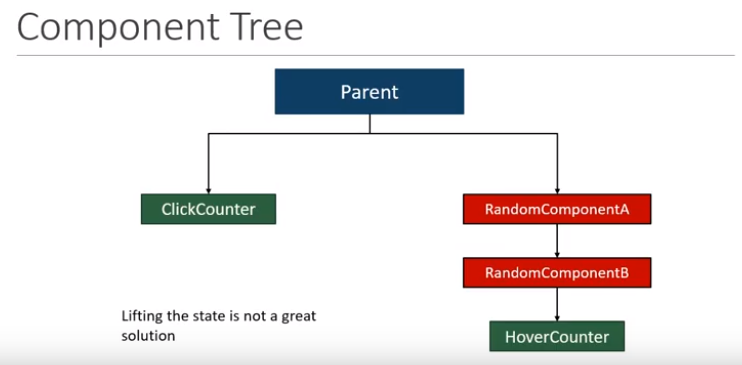
Code: <https://codesandbox.io/s/00254q4n6p>

**Higher Order Component**



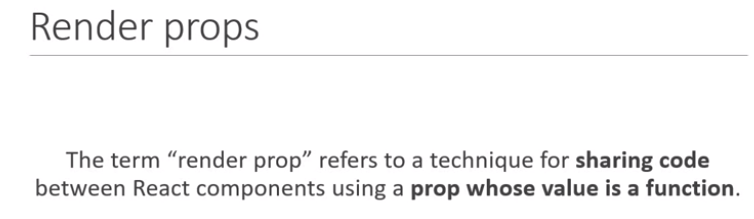






**Render props**

Share function between components => pattern: render props



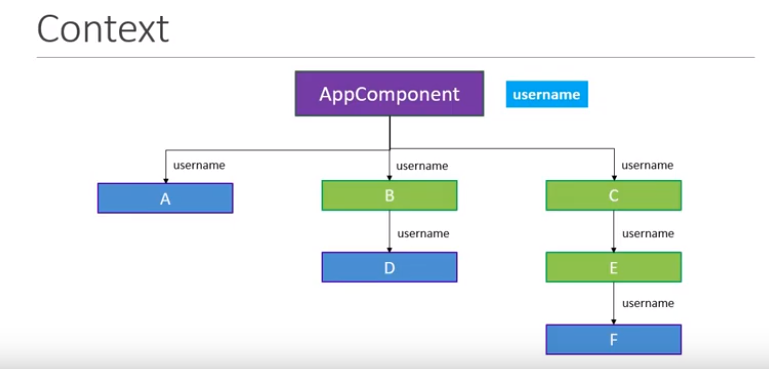
**Context**

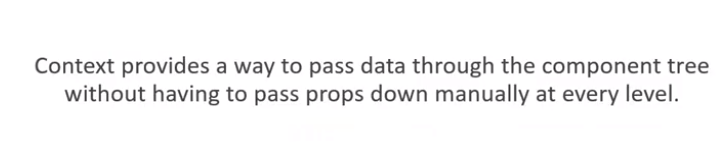
Truyền dữ liệu tới tất cả component con (ko cần dùng props)

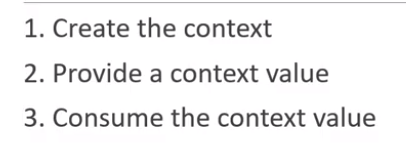
Chia sẻ dữ liệu global: current authentication user, theme (value=”dark), language.

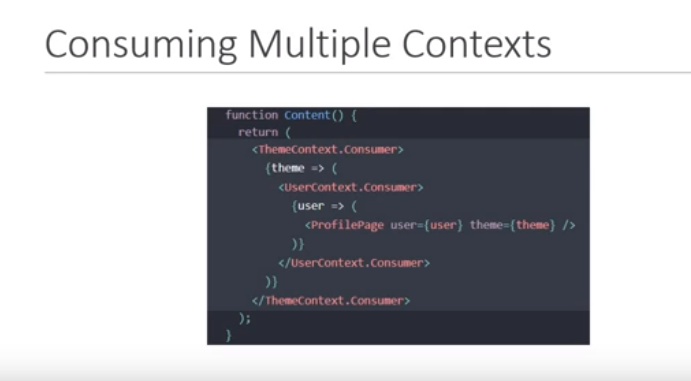
Ít áp dụng vì việc tái sử dụng khó khan.

Link: <https://viblo.asia/p/context-api-trong-react-XL6lAovJKek>









**React and HTTP**

