

React.memo

Chúng ta dùng React.memo khi không muốn component bị re-render mỗi khi component cha re-render

Các tác nhân làm component re-render: cập nhật state, cập nhật prop, component cha re-render

React.memo là một HOC, vậy nên chỉ cần bao bọc component là được

```
const newComponent = React.memo(Component);
```

React.memo chỉ tác động đến prop, không ảnh hưởng đến state

Cơ chế memo là cơ chế dùng bộ nhớ (RAM) để lưu trữ, vì thế khi dùng những thứ liên quan đến memo nghĩa là chúng ta đang đánh đổi giữa tốn nhiều bộ nhớ hơn để tăng tốc performance.

Vậy nên đừng lạm dụng quá nhé!

Cùng tới với demo

đầu tiên là viết 1 component đơn giản

```
function Title(props: TitleProps) {  
  console.log('render title');  
  return <span>hi</span>;  
}  
export default Title;
```

và ta có 1 component cha

```
function TaskInput(props: TaskInputProps & typeof injectedProps) {  
  return {  
    <div>  
      <Title />  
      <input />  
    </div>  
  }  
}
```

và mọi người thấy thì chả có gì đặc biệt, nhưng mà trên thực tế thì lúc ta nhập input thì component Title lại re-render, vậy làm sao để nó không re-render khi component cha re-render

Giải pháp ở đây là dùng React.memo, nó là 1 HOC, vậy nên chỉ cần bao bọc component là được. Và nó cũng nhận 1 props là `areEqual` để so sánh 2 props trước và sau khi re-render, nếu 2 props giống nhau thì nó sẽ không re-render

```
function equal(prevProps: TitleProps, nextProps: TitleProps) {  
  return prevProps.name === nextProps.name;  
}
```

Và đó cũng là cách mà React.memo hoạt động. Tiếp đến là trên component cha, ta sẽ viết 1 const và truyền nó vào component Title

```
const address = {  
  street: '10 Tran Hung Dao'  
}  
function TaskInput(props: TaskInputProps & typeof injectedProps) {  
  return {  
    <div>  
      <Title address={address} />  
      <input />  
    </div>  
  }  
}
```

► Và bạn có đoán được kết quả không ???

đầu tiên là dùng equal, trong đó ta sẽ so sánh 2 object address.street, nếu giống nhau thì nó sẽ trả về true, và nó sẽ không re-render với status là false

```
function equal(prevProp: TitleProps, nextProps: TitleProps) {  
  if (prevProp.address?.street === nextProps.address?.street) {  
    return true;  
  }  
  return false;  
}
```

Cách thứ 2, dùng useMemo. Với 1 dependency array rỗng, nó sẽ chỉ tạo ra 1 object address duy nhất, và nó sẽ không thay đổi khi component cha re-render

```
const address = useMemo(() => {  
  return {  
    street: '10 Tran Hung Dao',  
  };  
}, []);
```