

React JS

**Binding dữ liệu trong react
&&
state and updating state**

❑ Binding dữ liệu trong react

- ❑ Jsx cho phép ta lồng javascript vào HTML thông qua dấu `{ ten_bien | ten_function() }`
- ❑ *title, product name, renderProduct... jsx cho phép chúng ta có thể render biến chuỗi hàm, ... tại phần nội dung miễn kết quả trả về là 1 đoạn jsx (cách viết HTML và js kết hợp)*
- ❑ *Javascript sẽ được parse và hiển thị ra chung với html*

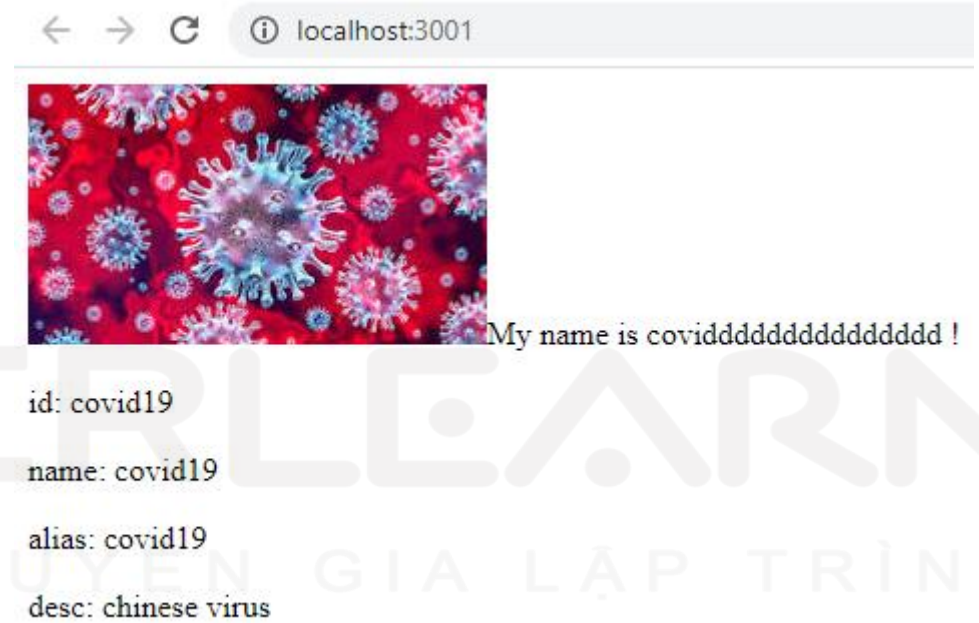
```
JS App.js  Product.jsx x
reactslide ▸ src ▸ components ▸ Product.jsx ▸ Product
1  import React, { Component } from 'react'
2
3  export default class Product extends Component {
4
5
6      render() {
7          let title = 'CYBERSOFT';
8          let productName = `<p><b>FrontEnd XXX</b></p>`;
9          return (
10
11              //Code jsx trả về
12              <div className="Product">
13                  {title}
14                  {productName}
15              </div>
16          )
17      }
18  }
```

```
JS App.js  Product.jsx x
reactslide ▸ src ▸ components ▸ Product.jsx ▸ ...
1  import React, { Component } from 'react'
2
3  export default class Product extends Component {
4
5      //Tạo ra 1 phương thức trả về 1 đoạn jsx
6      renderProduct = () => {
7          let title = 'CYBERSOFT';
8          let productName = `<p><b>FrontEnd XXX</b></p>`
9          return (
10              <div>
11                  {title}
12                  {productName}
13              </div>
14          )
15      }
16
17      render() {
18
19          return (
20
21              //Code jsx trả về
22              <div className="Product">
23                  {this.renderProduct()} /* <= Phương thức render sẽ được gọi ở đây this.tenPhuongThuc() */
24              </div>
25          )
26      }
27  }
```

✓ React còn cho phép chúng ta sử dụng gọi hàm để binding dữ liệu `{tenham()}`

➔ Lưu ý: Kết quả của hàm trả về bắt buộc là chuỗi hoặc 1 đoạn jsx (React component)

```
3 export default class BindingDemoRCC extends Component {
4   //Xây dựng phương thức binding dữ liệu là component
5   renderImage = () => {
6     //Nội dung trả về 1 component
7     //(Lưu ý là đối tượng <img> jsx gần giống html do react định nghĩa không phải html)
8     return 
9   }
10  //Xây dựng phương thức binding dữ liệu là chuỗi
11  renderText = () => {
12    return 'My name is coviddddddddddddd !'
13  }
14  renderMultiComponent = () => {
15    //Kết hợp xử lý binding đối tượng tại hàm và trả về 1 component
16    const virus = {
17      id: 'covid19',
18      name: 'corona',
19      alias: 'SARS-CoV-2',
20      desc: 'chinese virus'
21    }
22    return <div className="container">
23      <p>id: {virus.id}</p>
24      <p>name: {virus.id}</p>
25      <p>alias: {virus.id}</p>
26      <p>desc: {virus.desc}</p>
27    </div>
28  }
29  render() {
30    return (
31      <div>
32        /*Gọi hàm */
33        {this.renderImage()}
34        {this.renderText()}
35        {this.renderMultiComponent()}
36      </div>
37    )
38  }
39 }
40
```



Kết quả hiển thị

React class component (Ứng với react functional component cũng tương tự). Đề cập slide bên dưới !

❑ Xử lý sự kiện trong react (Handle Event React)

- ❑ Các sự kiện *onClick*, *onChange*, *onSubmit* ... trong *javascript* đều có thể sử dụng trong react. Tuy nhiên sẽ có khác biệt về cú pháp => Tham khảo ví dụ bên dưới
- ❑ Cú pháp: *suKien*=*{callbackfunction}* => sự kiện là các sự kiện nêu trên, *callback function* là 1 *function* để xử lý cho sự kiện đó lưu ý: *callback function* gán vào không có 2 dấu ().

React class component

```
1 import React, { Component } from 'react'
2
3 export default class EventDemoRCC extends Component {
4   //Tạo ra 1 PHƯƠNG THỨC (Để truy xuất đến PHƯƠNG THỨC trong CLASS ta dùng từ khóa this.tenPhuongThuc)
5   handleClick = () => {
6     alert("Hello Khái !!!")
7   }
8
9   render() {
10     return (
11       <div>
12         <button onClick={this.handleClick}>Click me</button> <br />
13       </div>
14     )
15   }
16 }
```

React functional component

```
1 import React from 'react'
2
3 export default function EventDemoRFC() {
4
5   /*Khác với react class react functional vì chỉ là 1 hàm nên bên trong hàm
6    object hay function đều được khai báo dưới dạng biến là let var hay const
7    */
8   const handleClick = () => {
9     alert("Hello Khái !!!")
10   }
11
12   return (
13     <div>
14
15       /*
16        Khi gọi hàm không cần con trỏ this bởi vì đơn giản component này không phải là
17        lớp đối tượng mà ta sử dụng nó như là 1 function bình thường thôi
18        (Bên trong function thì ta khai báo và sử dụng biến bình thường thôi hehe khai báo this chỉ cho rườm rà nè !)
19        Tuy nhiên vẫn sử dụng được this nhé (nhưng không cần thiết)
20        */
21       <button onClick={handleClick}>Click me</button>
22     </div>
23   )
24 }
```

Truyền tham số qua sự kiện (Passing Arguments to Event Handlers)

```
1 import React, { Component } from 'react'
2
3 export default class EventDemoRCC extends Component {
4   //Tạo ra 1 PHƯƠNG THỨC (Để truy xuất đến PHƯƠNG THỨC trong CLASS ta dùng từ khóa this.tenPhuongThuc)
5   handleClick = () => {
6     alert("Hello Khái !!!")
7   }
8
9   //Đối với sự kiện có tham số truyền vào có 2 cách để gán cho sự kiện (ở đây là onClick)
10  //Cách 1 dùng phương thức .bind(this, tenHam) để xử lý khi người dùng click mới gọi hàm này
11  showMessage = (message,tagButton) => {
12    console.log(tagButton.target) //tagButton.target: đại diện cho đối tượng button => tag button là this
13    alert(message); //message: là chuỗi 'cyberlearn chào covid !'
14  }
15
16  //Cách 2 dùng arrowFunction khi người dùng click mới gọi hàm này
17  //Cách này đơn giản hơn dùng khai báo 1 function nặc danh khi click vào sẽ gọi function đó (bên trong function khi thực hiện sẽ gọi hàm này)
18  reply = (content) => {
19    alert(content);
20  }
21  render() {
22    return (
23      <div>
24        <button onClick={this.handleClick}>Click me</button> <br />
25
26        /* Khi sử dụng bind truyền
27        tham số 1: this là chính đối tượng button (mỗi thẻ component là 1 đối tượng)
28        tham số 2: giá trị khi hàm đó được click sẽ gọi hàm đó xử lý
29        */
30        <button name="btnShowMessage" onClick={this.showMessage.bind(this, 'cyberlearn chào covid !')}>Show Message</button>
31
32        <button name="btnReplyMessage" onClick={() => { this.reply('next !') }}>Reply message!</button>
33      </div>
34    )
35  }
36 }
37
```

Render với điều kiện if (Rendering conditions)

- ❑ Kết hợp if else và hàm để xác định nội dung cần hiển thị trong react.
- ❑ Bằng cách xây dựng 1 hàm bên trong sử dụng lệnh if else để tùy biến nội dung jsx được render ra giao diện

```
1  import React, { Component } from 'react'
2
3  export default class Rendering_Conditions extends Component {
4
5      //Thuộc tính
6      login = true;
7      userName = 'Trương Tấn Khải';
8
9      //Phương thức gọi render nội dung
10     renderContent = () => {
11         /*Để render kết quả tùy chọn bởi 1 điều kiện nào đó ta viết hàm render và thực thi logic tại hàm
12         => Kết quả miễn là trả về js là được => Bên dưới chỉ việc gọi hàm này khi cần render nội dung
13         */
14         if(this.login)
15         {
16             return <h1>Hello {this.userName}</h1>
17         }
18         return <button> Đăng nhập </button>
19     }
20
21
22     render() {
23         return (
24             <div>
25                 {this.renderContent()}
26             </div>
27         )
28     }
29 }
30
```

❑ Render và rerender với state (state and updating state)

- ❑ State là một thuộc tính mặc định của class kế thừa từ class component để quản lý trạng thái của component
- ❑ Mỗi khi state thay đổi, thì hàm render sẽ được gọi chạy lại. Lưu ý: Muốn component render lại ta phải thay đổi state thông qua phương thức setState chứ không được gán trực tiếp.
- ❑ Phương thức setState là 1 phương thức bất đồng bộ, có 2 tham số

+ Tham số 1: giá trị state mới

+ Tham số 2: callback thực thi ngay sau khi state thay đổi

Lưu ý:

Không được set lại giá trị state theo cách này: ~~this.state.thuocTinh = [giá trị]~~

Ta set giá trị của state thông qua phương thức setState:

```
this.setState({  
  thuocTinh: [giá trị mới]  
})
```

```

1  import React, { Component } from 'react'
2
3  export default class StateDemo extends Component {
4
5      userName = 'Trương Tấn Khải';
6
7      //Thuộc tính state : là thuộc tính có sẵn từ class Component
8      state = {
9          login:false //Định nghĩa các biến làm thay đổi giao diện tại đây (Là thuộc tính của state)
10     }
11
12     //Phương thức login
13     login = () => {
14         /*Phương thức this.setState(): phương thức làm thay đổi giá trị thuộc tính trong state
15         và đồng thời làm hàm render gọi lại
16         */
17         this.setState({
18             login:true
19         })
20         /*Lưu ý: Không được gán this.state.login = false => gán như vậy hàm render sẽ không được gọi
21         => Giao diện không được render lại
22         */
23     }
24
25     render() {
26         return (
27             <div>
28                 {this.state.login ? <b>Hello {this.userName}</b> : <button onClick={this.login}>Login</button>}
29             </div>
30         )
31     }
32 }
33

```

➔ *Vậy khi nào nên cần dùng state ?*

- ☐ Khi ta cần thực thi 1 sự kiện như là nhấn nút button hay gõ phím ... làm cho các giá trị trên giao diện thay đổi thì ta đặt giá trị thuộc tính đó trong state.

❑ Sử dụng css trong react (Styling with Stylesheets - styling with module - working with inline style)

✓ Css nhúng (styling with stylesheets)

- Để nhúng css vào component ta dùng cú pháp import → đường dẫn đến file .css từ component đó.
- Lưu ý: css đó sẽ áp dụng cho toàn ứng dụng

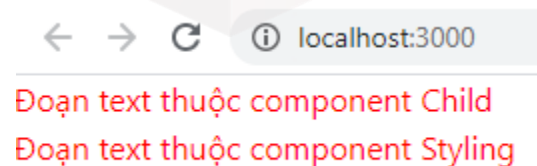


```
Styling.css
1 .txt
2   color: red
3

Styling.js
1 import React, { Component } from 'react'
2
3 //Nhúng file css
4 import './Styling.css';
5 import Child from './Child';
6 export default class Styling extends Component {
7   render() {
8     return (
9       <div>
10        <Child />
11        <p className="txt">Đoạn text thuộc component Styling</p>
12      </div>
13    )
14  }
15 }

Child.js
1 import React, { Component } from 'react'
2
3 export default class Child extends Component {
4   render() {
5     return (
6       <div className="txt">
7        Đoạn text thuộc component Child
8      </div>
9    )
10  }
11 }
```

■ Kết quả:



✓ styling with module

- Là phương pháp đóng gói css theo component, sử dụng dưới dạng import đối tượng giống như component.
- Lưu ý: Khi áp dụng css dưới dạng styling module này, css chỉ được áp dụng cho tại component import các component khác không bị ảnh hưởng.

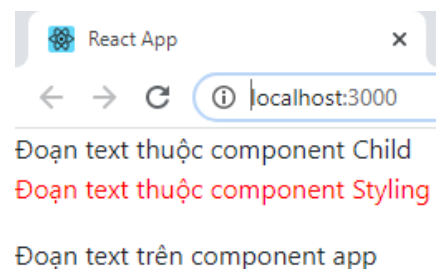
css cho phần text

Khi import import dạng đối tượng tương tự như component

Khi cần nhúng css ta sử dụng tên đối tượng .class
=> Lưu ý: Khi nhúng kiểu này css chỉ áp dụng cho component này, các component khác không bị ảnh hưởng

css txt không bị ảnh hưởng khi ta sử dụng import dưới dạng file css.module

■ Kết quả:



✓ working with inline style

```
JS Styling.js x
src > Styling > JS Styling.js > ...
1  import React, { Component } from 'react' 8.3K (gzipped: 3.3K)
2  export default class Styling extends Component {
3      render() {
4
5          const obStyle = {
6              color:'green',
7              backgroundColor:'black',
8              padding:'15px',
9          }
10
11         return (
12             <div>
13                 /*Cách 1: Truyền object Style */
14                 <span style={obStyle}>Đoạn text thuộc component Styling</span>
15
16                 <br />
17                 <br />
18                 /*Cách 2: Truyền trực tiếp object {} */
19                 <span style={{color:'pink',background:'black'}}>Đoạn text thuộc component Styling</span>
20
21             </div>
22         )
23     }
24 }
25
```

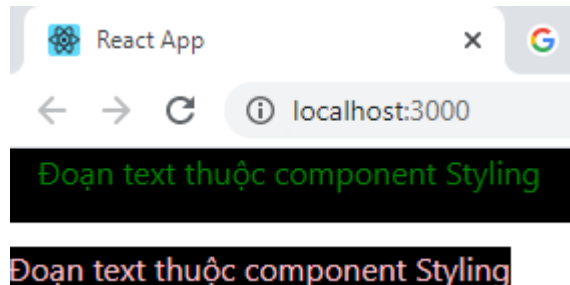
- React cho phép chúng ta style trực tiếp trên thẻ - tuy nhiên với thuộc tính css tương tự dom.style trong js
- Lưu ý: Truyền thông qua style đối với các thuộc tính css có dấu - thì chuyển sang ký tự thứ đầu tiên từ thứ 2 viết hoa.
- Ví dụ:

background-color → backgroundColor

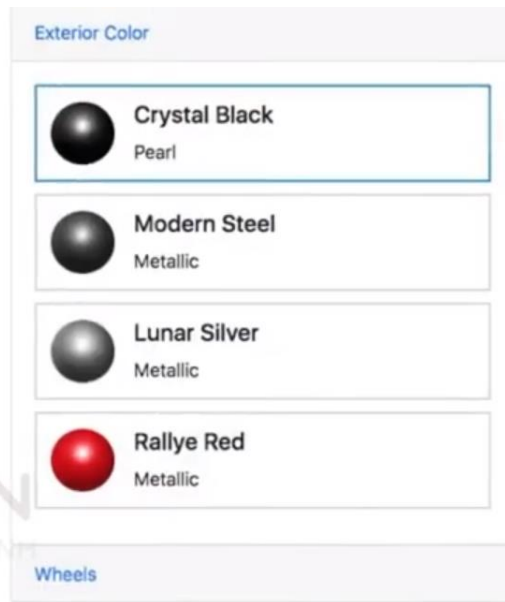
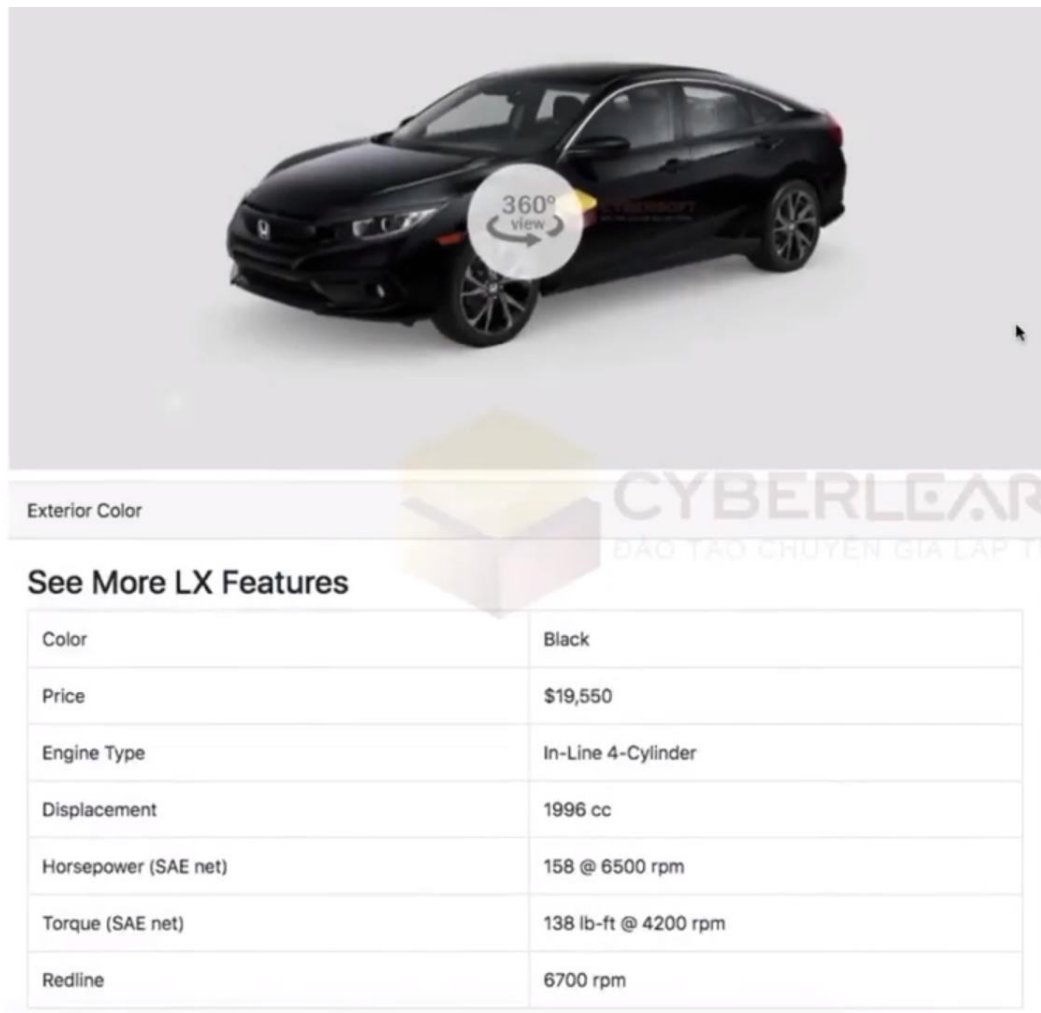
padding-top → paddingTop

border-radius → borderRadius

- Kết quả:



THỰC HÀNH



Yêu cầu:

Tạo 1 thư mục bài tập bên trong tạo 1 component BaiTapChonXe.js

Sử dụng state lưu trữ thuộc tính img của hình

Tạo các button chọn màu bên phải tương ứng xử lý state thay đổi

Download resource tại đây

<https://drive.google.com/open?id=1usu44aCL3o-Lnh9hbtxYVwOZMS6cN3Dm>