



**FPT POLYTECHNIC**



**THỰC HỌC – THỰC NGHIỆP**



**Conceive Design Implement Operate**

## **LẬP TRÌNH FRONT-END FRAMEWORK 2**

---

### **TỔNG QUAN REACTJS**

- ◎ Tổng quan ReactJS
- ◎ Cài đặt ReactJS
- ◎ JSX





# PHẦN 1

# TỔNG QUAN REACTJS

- ❑ React (hay còn gọi là ReactJS) là bộ thư viện viết bằng ngôn ngữ JavaScript dùng để phát triển các ứng dụng UI (User Interface) được phát triển bởi các lập trình viên làm việc tại Facebook.
- ❑ React có thể chạy được trên cả môi trường client lẫn server.

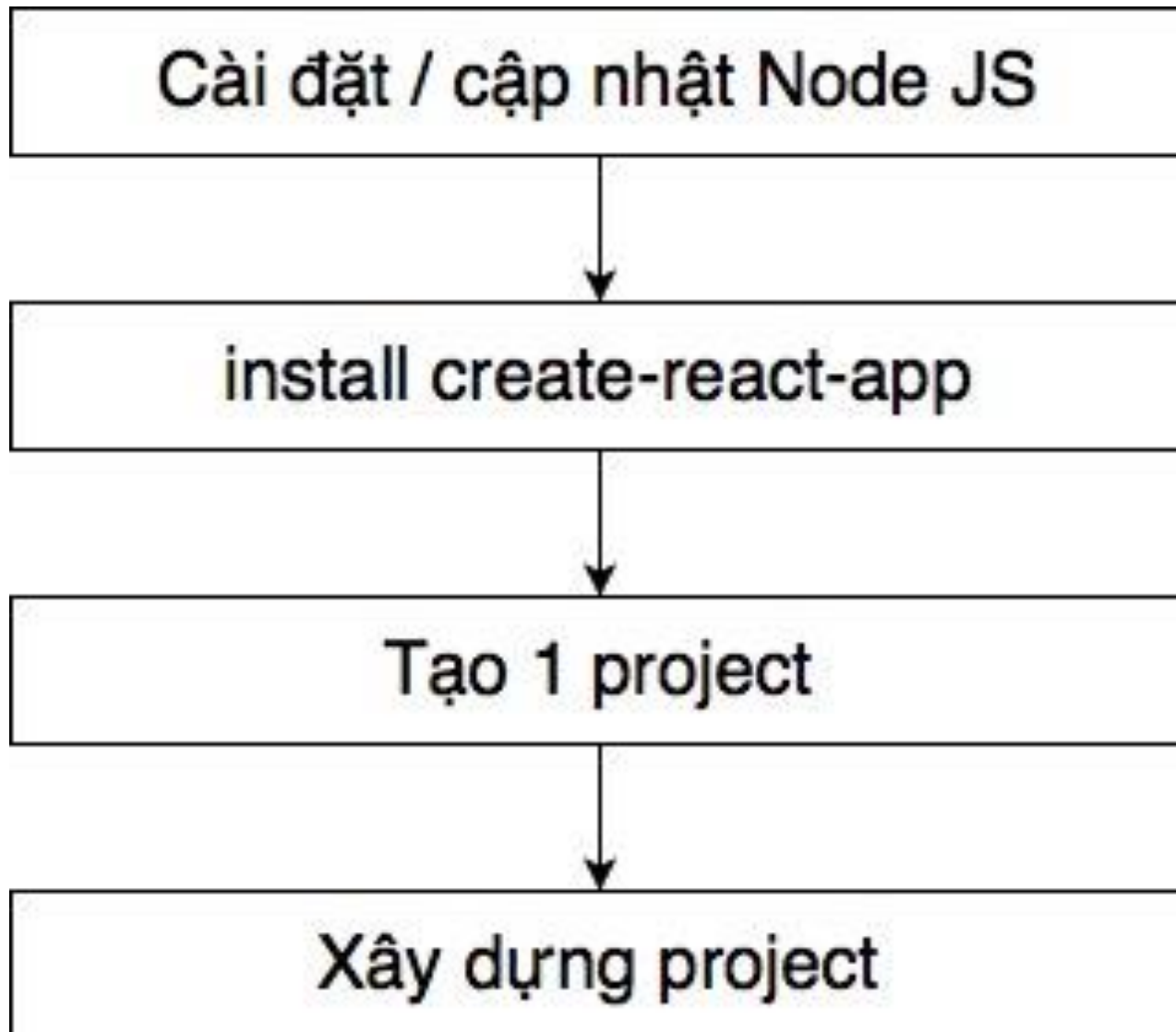
# REACT VÀ REACT NATIVE

## ❖ React (ReactJS)

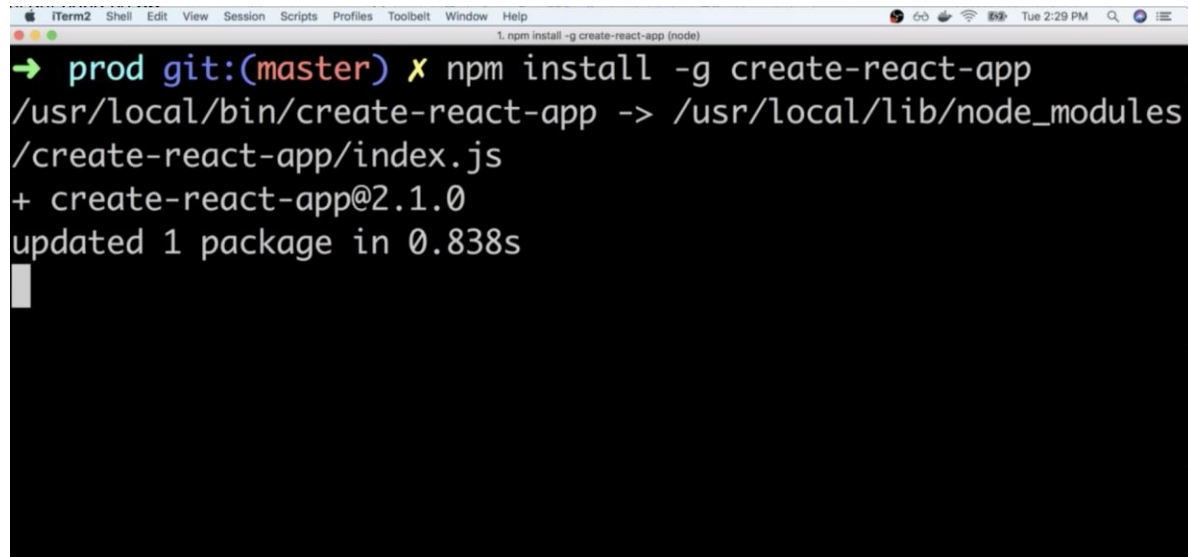
- Xây dựng các ứng dụng sử dụng Javascript (web)
- Sử dụng HTML và luồng cú pháp của nó
- Hoạt động tốt với CSS để tạo kiểu và hoạt ảnh

## ❖ React Native

- Xây dựng các ứng dụng di động đa nền tảng (android, ios)
- Sử dụng cú pháp React Native
- Sử dụng API động để tạo hoạt ảnh cho các thành phần của ứng dụng.

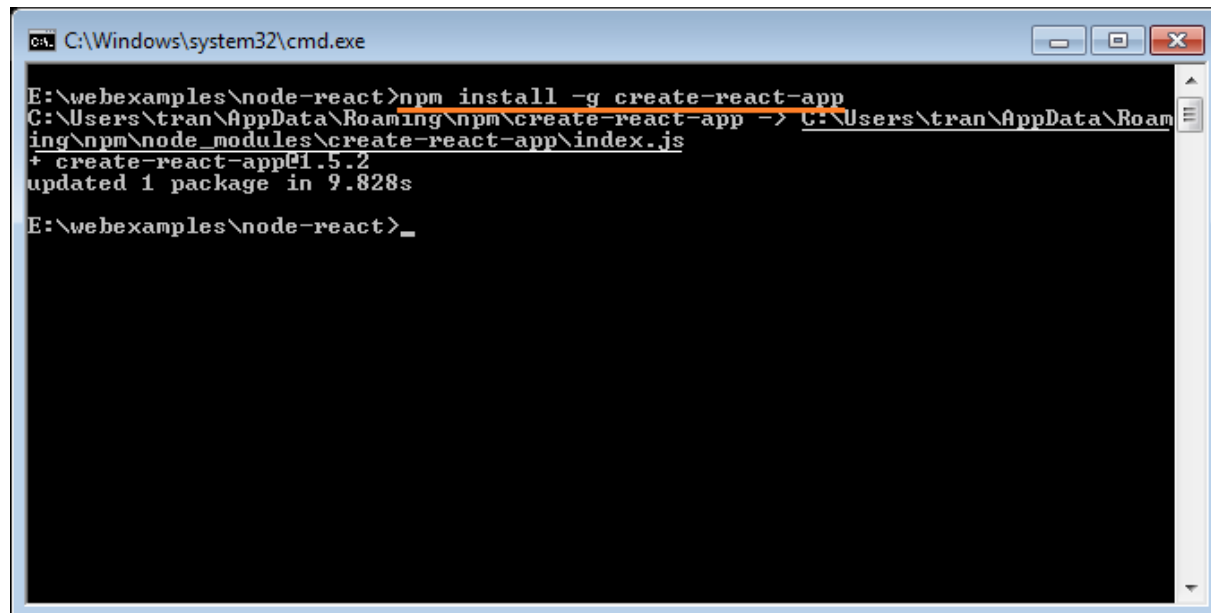


# INSTALL CREATE-REACT-APP



```

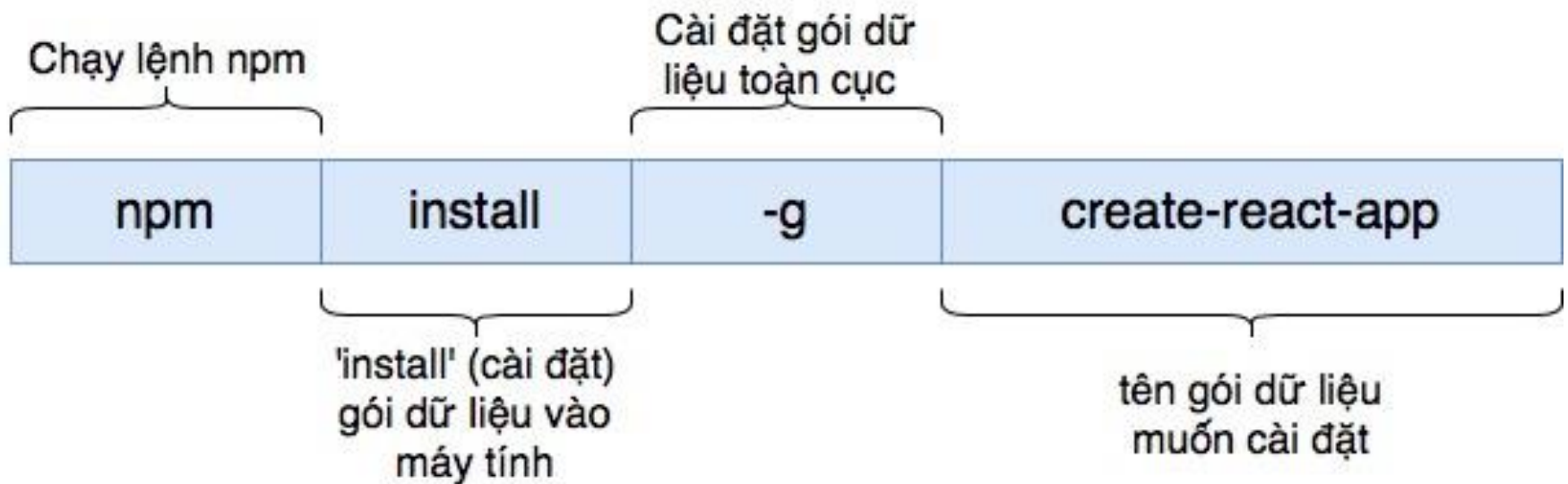
iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
1. npm install -g create-react-app (node)
→ prod git:(master) ✗ npm install -g create-react-app
/usr/local/bin/create-react-app -> /usr/local/lib/node_modules
/create-react-app/index.js
+ create-react-app@2.1.0
updated 1 package in 0.838s
  
```



```

C:\Windows\system32\cmd.exe
E:\webexamples\node-react>npm install -g create-react-app
C:\Users\tran\AppData\Roaming\npm\create-react-app -> C:\Users\tran\AppData\Roam
ing\npm\node_modules\create-react-app\index.js
+ create-react-app@1.5.2
updated 1 package in 9.828s
E:\webexamples\node-react>_
  
```

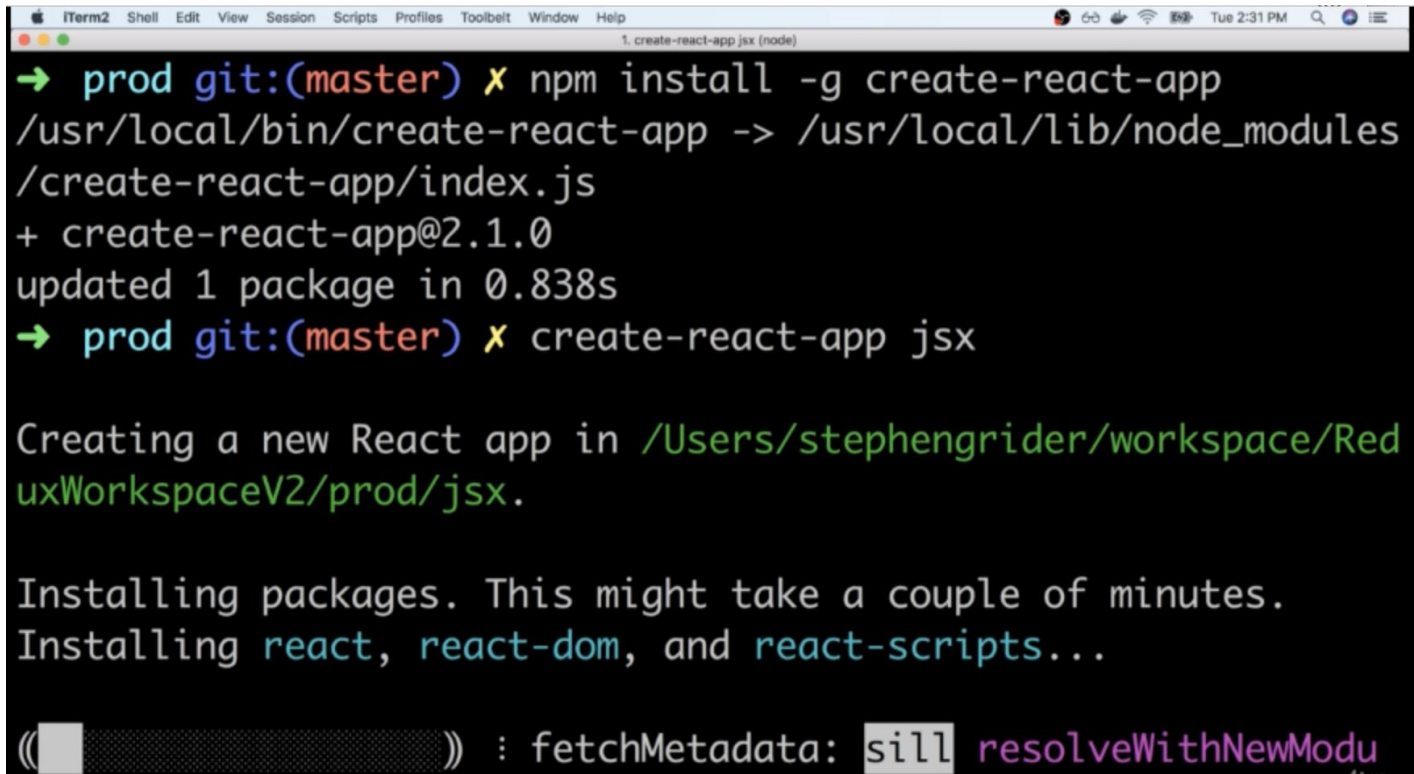
# INSTALL CREATE-REACT-APP





create-react-app

<name-of-project>



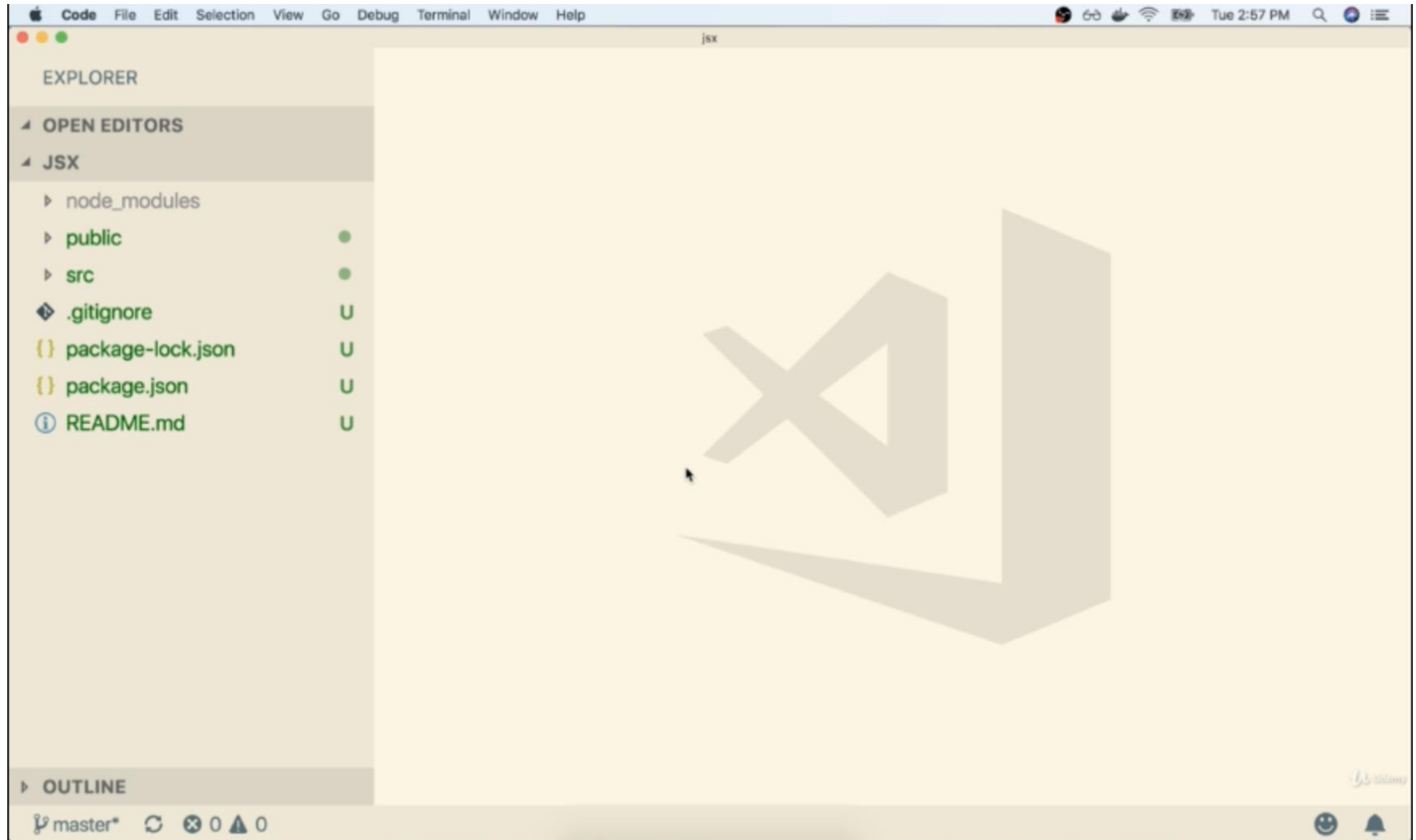
```

iTerm2  Shell  Edit  View  Session  Scripts  Profiles  Toolbelt  Window  Help
1. create-react-app jsx (node)
→ prod git:(master) ✗ npm install -g create-react-app
/usr/local/bin/create-react-app -> /usr/local/lib/node_modules
/create-react-app/index.js
+ create-react-app@2.1.0
updated 1 package in 0.838s
→ prod git:(master) ✗ create-react-app jsx

Creating a new React app in /Users/stephengrider/workspace/Red
uxWorkspaceV2/prod/jsx.

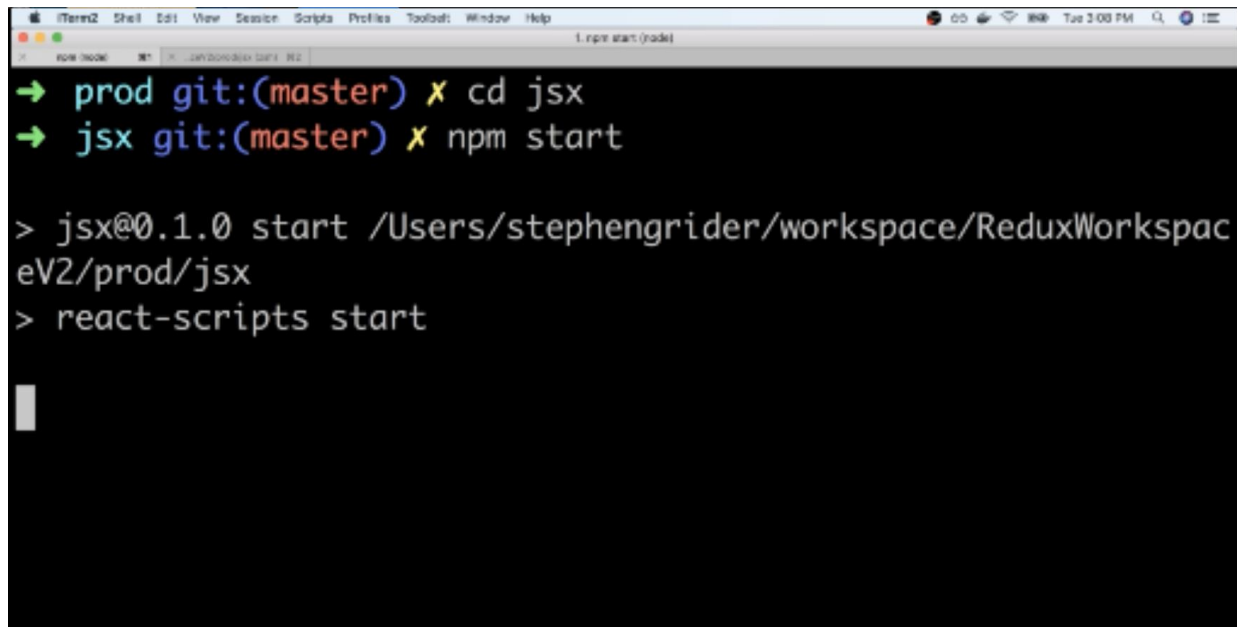
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts...

(( )) : fetchMetadata: sill resolveWithNewModu
  
```



Thư mục dự án	src	→ Thư mục chứa source code
	public	→ Thư mục chứa các static files như hình ảnh
	node_modules	→ Thư mục chứa các công cụ xây dựng (dependencies)/module của nodejs
	package.json	→ Xác định các thư viện cài đặt trong node_modules và cấu hình
	package-lock.json	→ Ghi lại các version của các gói dữ liệu được cài đặt
	README.md	→ Hướng dẫn cách sử dụng project

# START/STOP REACT APP

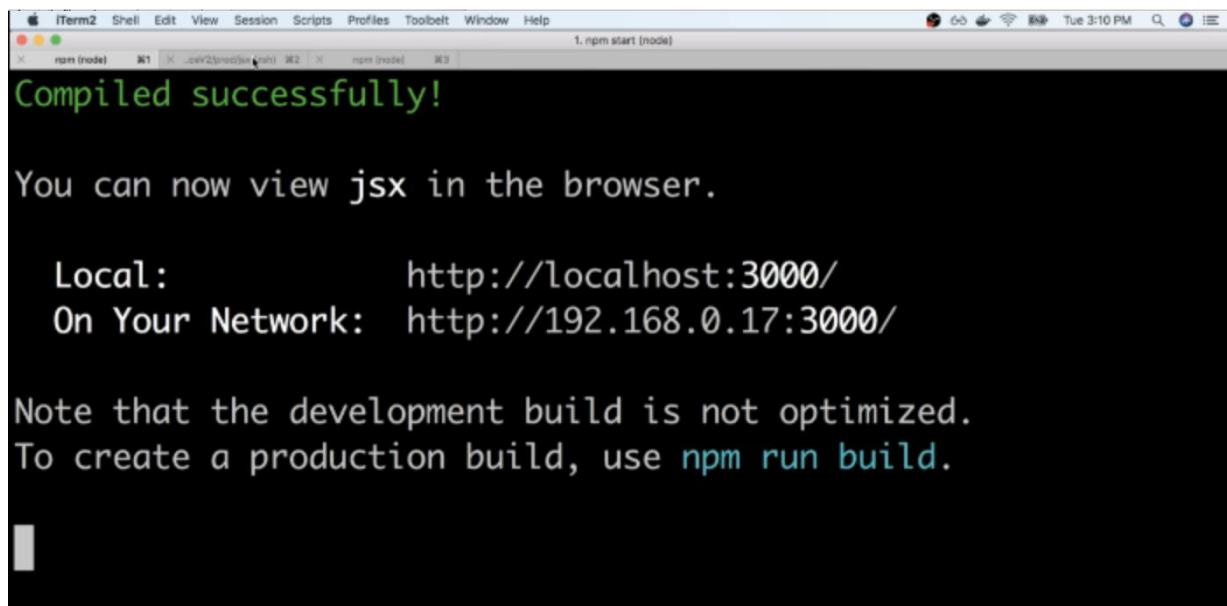


```

iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
1. npm start (node)

→ prod git:(master) ✗ cd jsx
→ jsx git:(master) ✗ npm start

> jsx@0.1.0 start /Users/stephengrider/workspace/ReduxWorkspac
eV2/prod/jsx
> react-scripts start
  
```



```

iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
1. npm start (node)

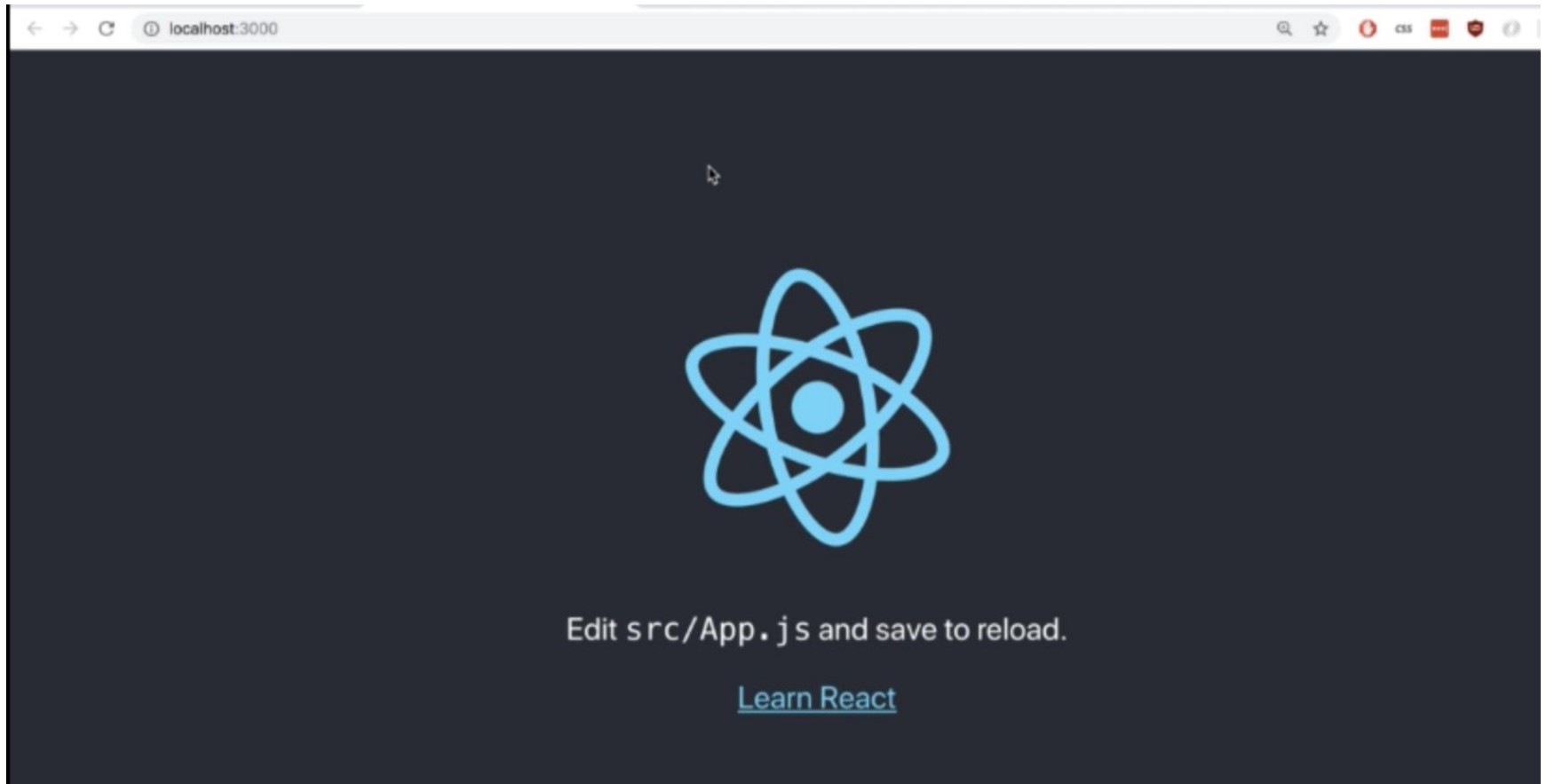
Compiled successfully!

You can now view jsx in the browser.

Local:            http://localhost:3000/
On Your Network:  http://192.168.0.17:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.
  
```

# START/STOP REACT APP



Stop react app

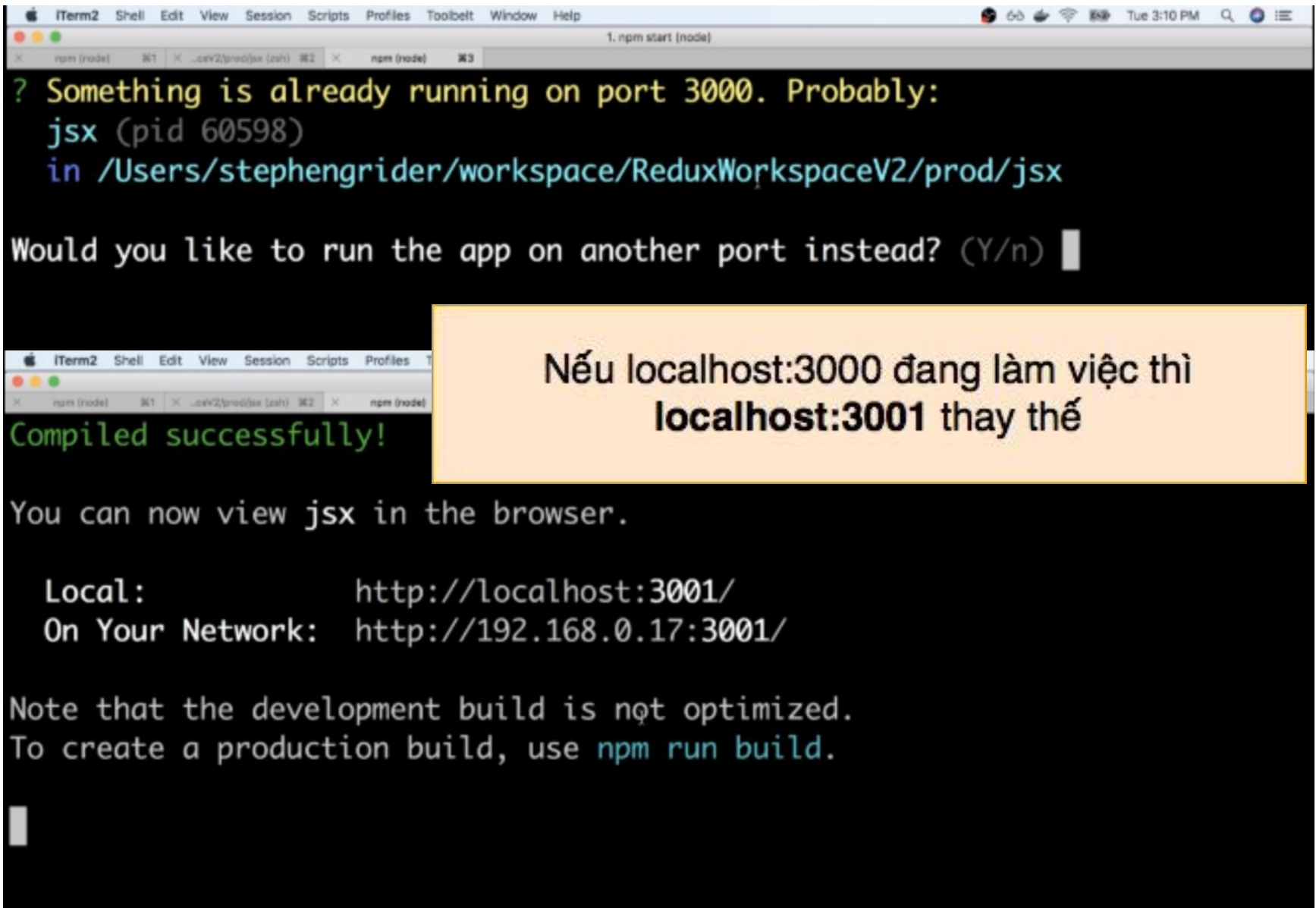
Nhấn **Ctrl + C** tại terminal

Start react app

Lệnh **npm start** trong thư mục project

Lần đầu tiên app bắt đầu, link đường dẫn là  
**localhost:3000**  
*localhost:8080 không hoạt động*

# START/STOP REACT APP



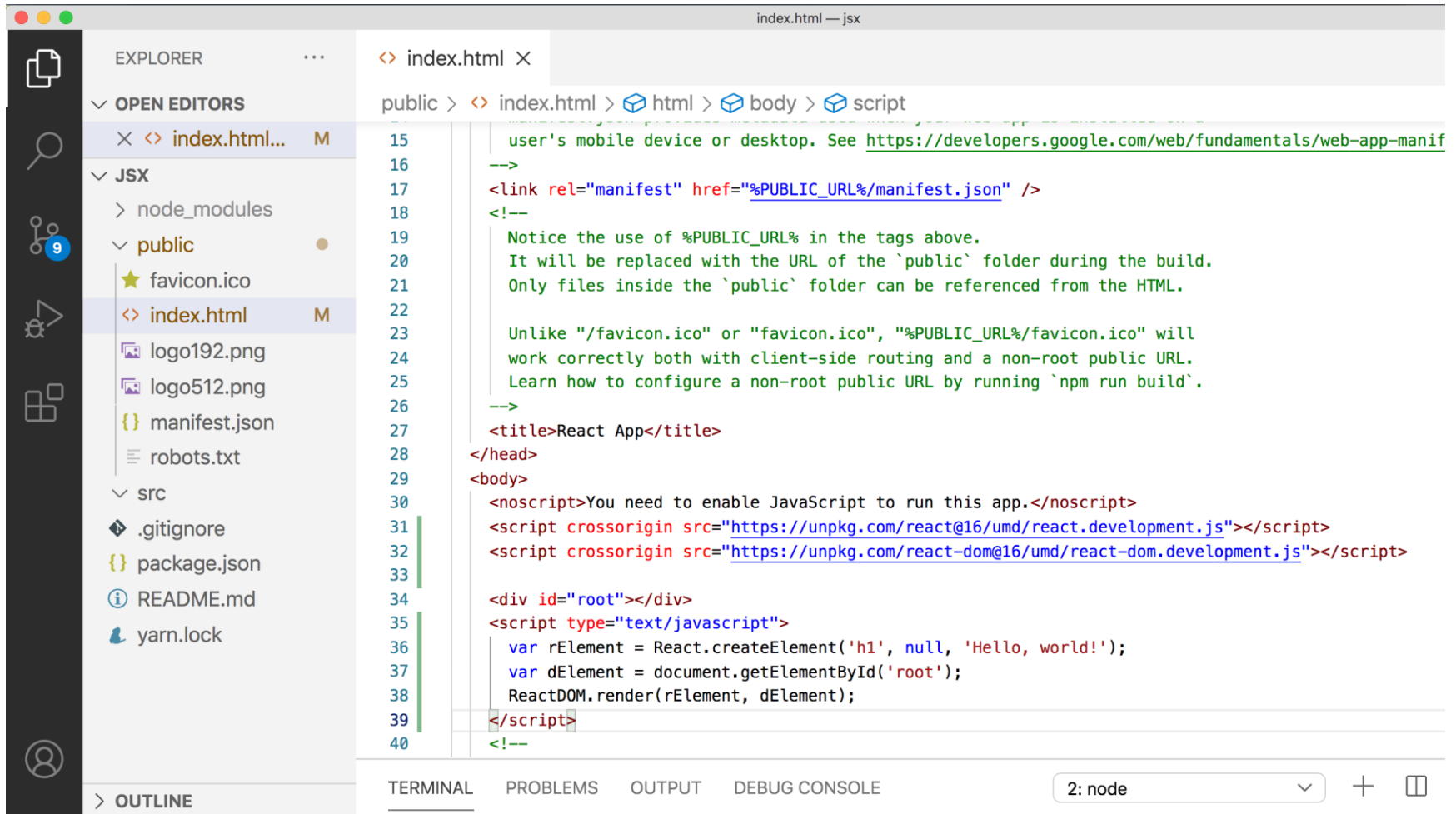
```
1. npm start (node)
? Something is already running on port 3000. Probably:
  jsx (pid 60598)
  in /Users/stephengrider/workspace/ReduxWorkspaceV2/prod/jsx
Would you like to run the app on another port instead? (Y/n)
Compiled successfully!
You can now view jsx in the browser.

Local:      http://localhost:3001/
On Your Network: http://192.168.0.17:3001/

Note that the development build is not optimized.
To create a production build, use npm run build.
```

Nếu localhost:3000 đang làm việc thì  
**localhost:3001** thay thế





index.html — jsx

public > <> index.html > html > body > script

```

15   user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manif
16   -->
17   <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
18   <!--
19   Notice the use of %PUBLIC_URL% in the tags above.
20   It will be replaced with the URL of the `public` folder during the build.
21   Only files inside the `public` folder can be referenced from the HTML.
22
23   Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
24   work correctly both with client-side routing and a non-root public URL.
25   Learn how to configure a non-root public URL by running `npm run build`.
26   -->
27   <title>React App</title>
28 </head>
29 <body>
30   <noscript>You need to enable JavaScript to run this app.</noscript>
31   <script crossorigin src="https://unpkg.com/react@16/umd/react.development.js"></script>
32   <script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
33
34   <div id="root"></div>
35   <script type="text/javascript">
36     var rElement = React.createElement('h1', null, 'Hello, world!');
37     var dElement = document.getElementById('root');
38     ReactDOM.render(rElement, dElement);
39   </script>
40   <!--
  
```

2: node



## ❖ HTML

```
<!doctype html>
  <html lang="en">
    <head></head>
    <body></body>
</html>
```

## ❖ Tạo HTML element

```
const newElement =
document.createElement("h1");
newElement.innerHTML = "Hello,
world!";
```

## ❖ Đưa h1 vào div có id="root"

```
const divElement = document.
.querySelector('#root');
divElement.appendChild(newEleme
nt);
```

## ❖ React & ReactDOM

```
<script crossorigin
src="https://unpkg.com/react@16
/umd/react.development.js"></sc
ript>
```

```
<script crossorigin
src="https://unpkg.com/react-
dom@16/umd/react-
dom.development.js"></script>
```

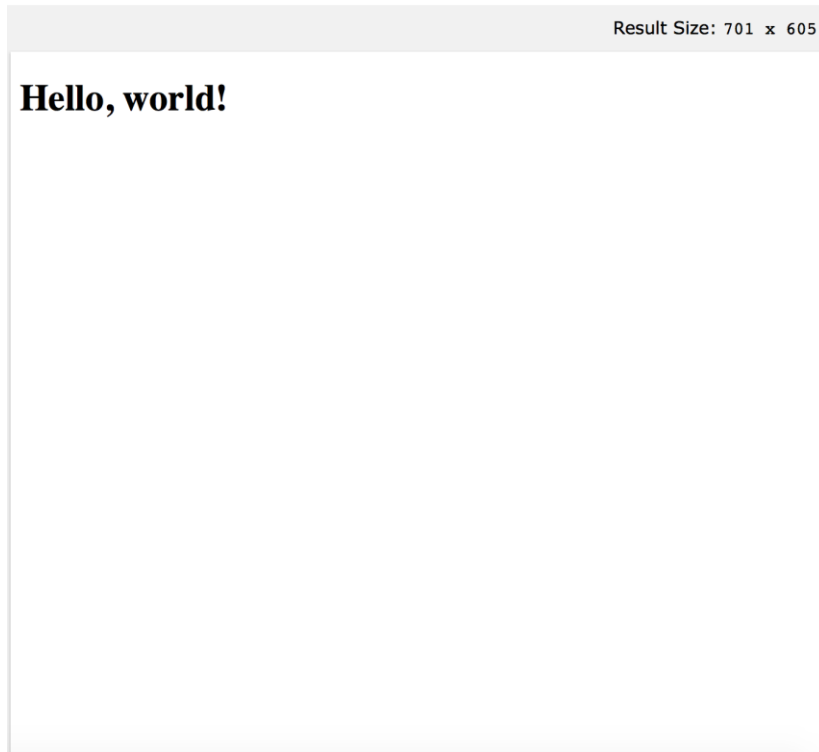
## ❖ Tạo element

```
var newElement =
React.createElement('h1', null,
'Hello, world!');
```

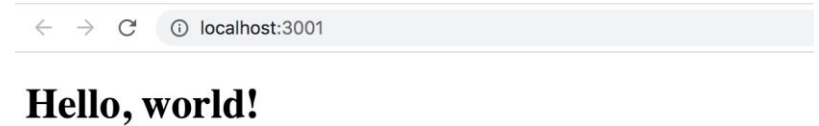
## ❖ Đưa h1 vào div có id="root"

```
ReactDOM.render(newElement,
document.querySelector('#root'
));
```

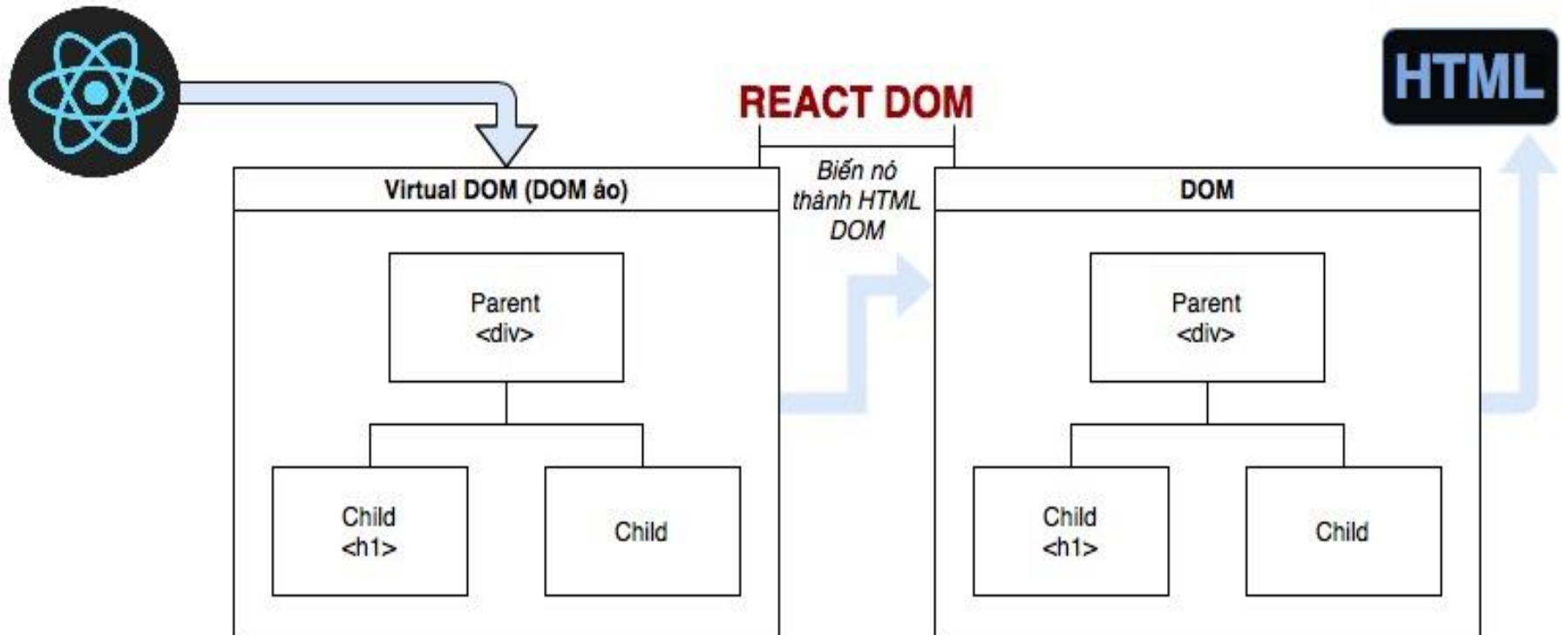
## ❖ Kết quả HTML DOM



## ❖ Kết quả với react & reactdom



# REACT & REACTDOM



## React

Làm thế nào để làm việc với component  
(createElement,...)

Gọi là 'reconciler'

## ReactDOM

Thực hiện hướng dẫn về những gì muốn  
hiển thị và biến nó thành HTML

Gọi là 'render'

demo



## PHẦN 2

## JSX

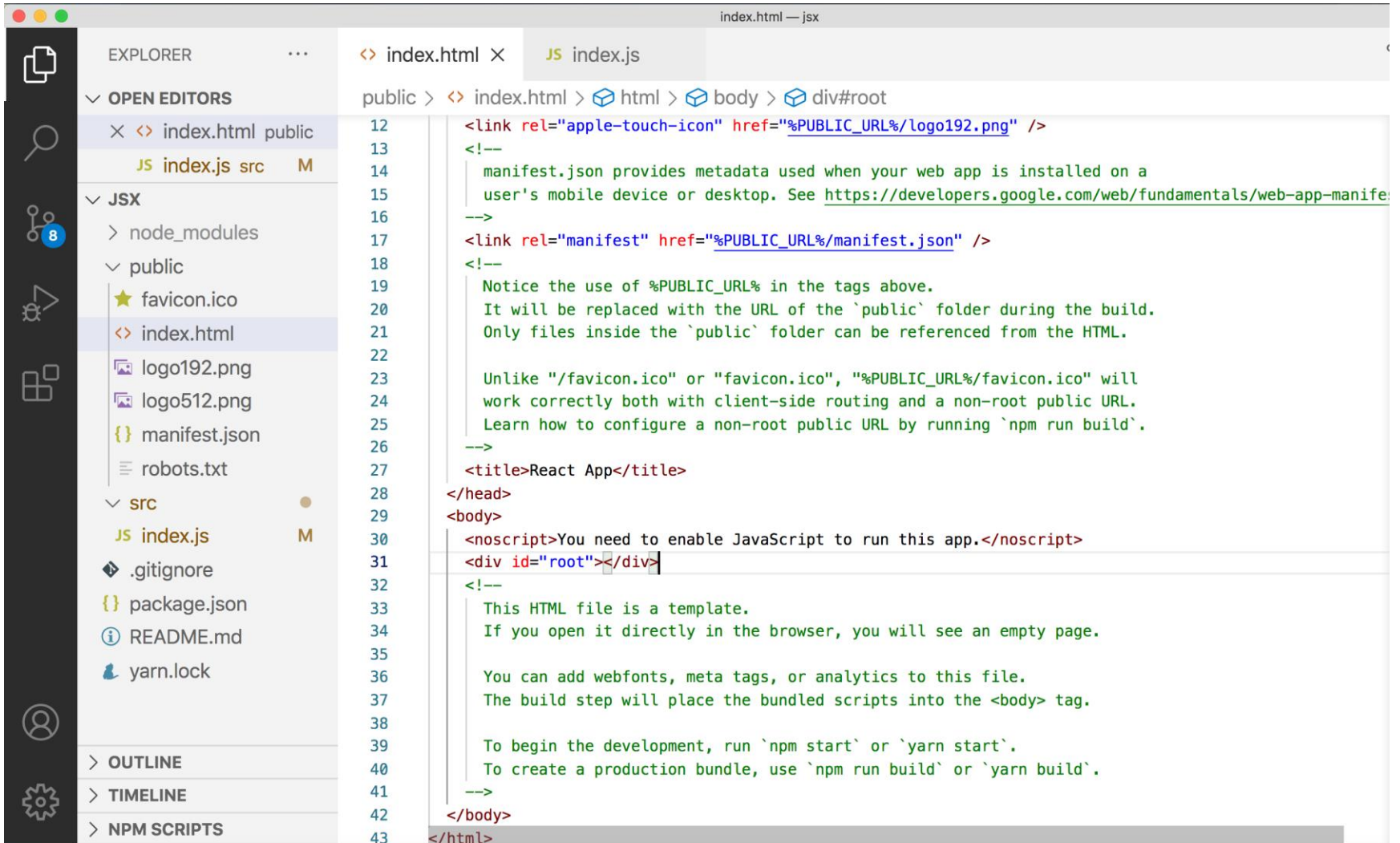
The screenshot shows a VS Code editor with a React application. The Explorer sidebar on the left shows the file structure with 'index.html' selected. The main editor area displays the HTML code for index.html. A blue arrow points from the code in the main editor to a callout box at the top right.

**Callout Box Content:**

```
<!-- Don't use this in production: -->
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
<div id="root"></div>
<script type="text/babel">
  var rElement = <h1>Hello, world!</h1>;
  var dElement = document.querySelector('#root');
  ReactDOM.render(rElement, dElement)
</script>
```

**Main Editor Content (index.html):**

```
20 It will be replaced with the URL of the `public` folder during the build.
21 Only files inside the `public` folder can be referenced from the HTML.
22
23 Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
24 work correctly both with client-side routing and a non-root public URL.
25 Learn how to configure a non-root public URL by running `npm run build`.
26
27 -->
28 <title>React App</title>
29 </head>
30 <body>
31   <noscript>You need to enable JavaScript to run this app.</noscript>
32   <script crossorigin src="https://unpkg.com/react@16/umd/react.development.js"></script>
33   <script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
34   <div id="root"></div>
35   <script type="text/javascript">
36     var rElement = React.createElement('h1', null, 'Hello, world!');
37     var dElement = document.getElementById('root');
38     ReactDOM.render(rElement, dElement);
39   </script>
40   <!--
```

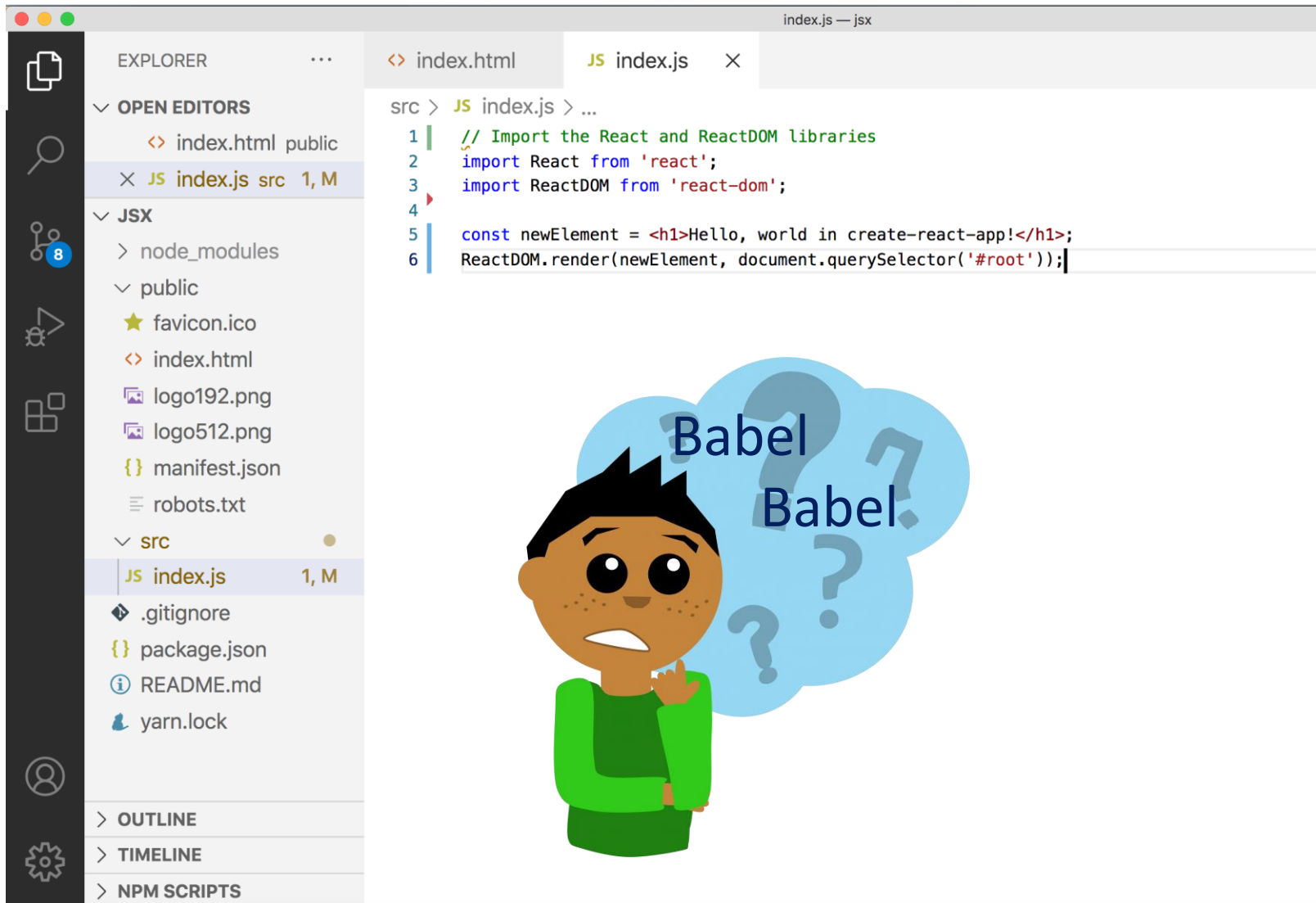


```

index.html — jsx
public > <> index.html > html > body > div#root
12  <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13  <!--
14  |   manifest.json provides metadata used when your web app is installed on a
15  |   user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manife
16  |   -->
17  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
18  <!--
19  |   Notice the use of %PUBLIC_URL% in the tags above.
20  |   It will be replaced with the URL of the `public` folder during the build.
21  |   Only files inside the `public` folder can be referenced from the HTML.
22  |
23  |   Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
24  |   work correctly both with client-side routing and a non-root public URL.
25  |   Learn how to configure a non-root public URL by running `npm run build`.
26  |   -->
27  <title>React App</title>
28  </head>
29  <body>
30  |   <noscript>You need to enable JavaScript to run this app.</noscript>
31  |   <div id="root"></div>
32  |   <!--
33  |   |   This HTML file is a template.
34  |   |   If you open it directly in the browser, you will see an empty page.
35  |   |
36  |   |   You can add webfonts, meta tags, or analytics to this file.
37  |   |   The build step will place the bundled scripts into the <body> tag.
38  |   |
39  |   |   To begin the development, run `npm start` or `yarn start`.
40  |   |   To create a production bundle, use `npm run build` or `yarn build`.
41  |   |   -->
42  |   </body>
43  </html>
  
```



# JSX TRONG CREATE-REACT-APP

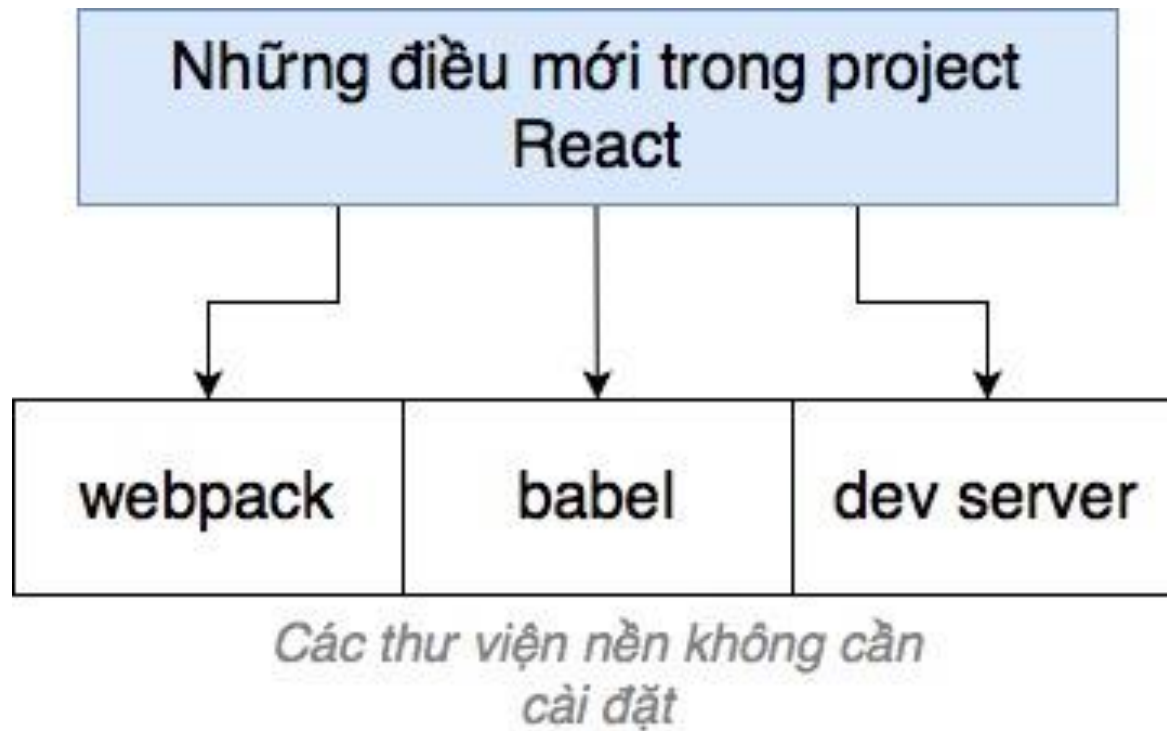


The image shows a VS Code editor window with a project structure on the left and a code editor on the right. The project structure includes a `public` folder with `index.html` and a `src` folder with `index.js`. The code editor shows the following JSX code in `index.js`:

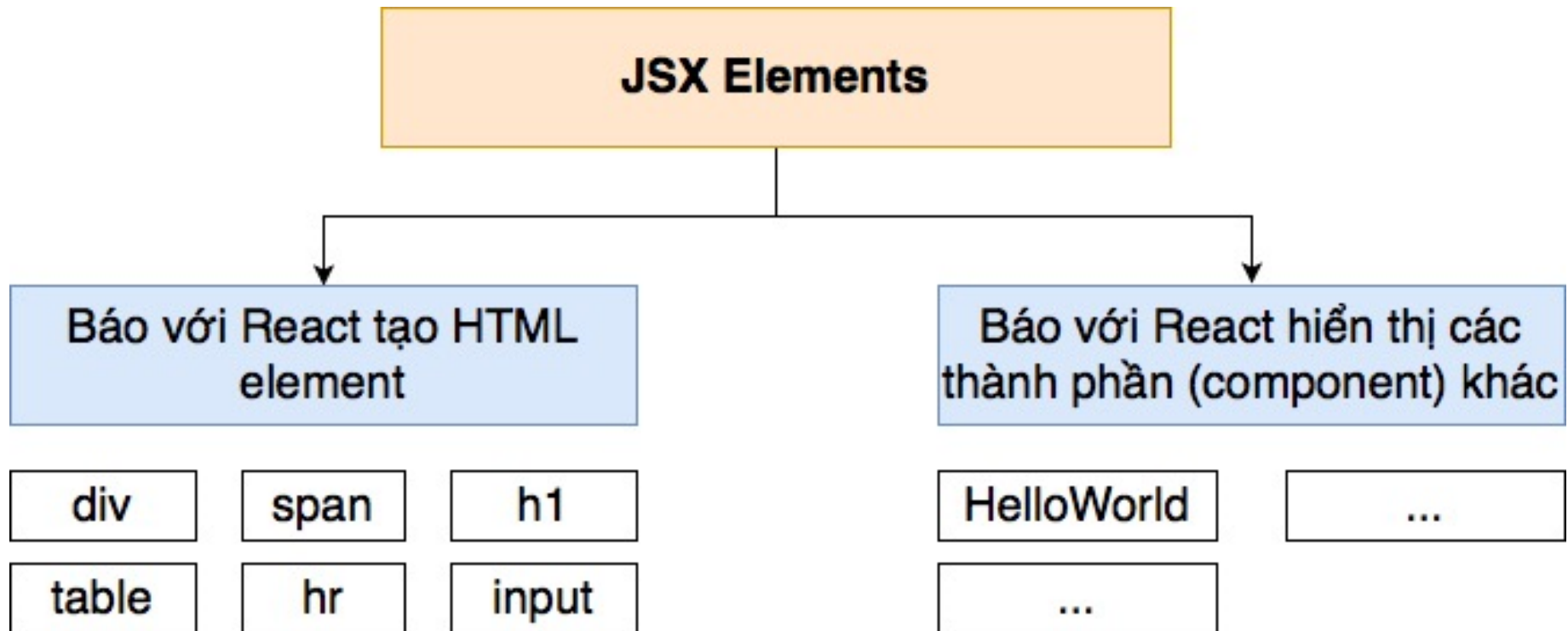
```
src > JS index.js > ...
1 // Import the React and ReactDOM libraries
2 import React from 'react';
3 import ReactDOM from 'react-dom';
4
5 const newElement = <h1>Hello, world in create-react-app!</h1>;
6 ReactDOM.render(newElement, document.querySelector('#root'));
```

Below the code editor, there is a cartoon illustration of a boy with a confused expression, with a thought bubble above him containing the word "Babel" and several question marks, indicating a common point of confusion for developers learning about JSX.

❑ Vì sao phải dùng create-react-app?



- ❑ JSX (JavaScript XML) là cú pháp mở rộng cho Javascript viết theo kiểu XML
- ❑ Sử dụng JSX để React để mô tả giao diện người dùng (UI)
- ❑ JSX cho phép kết nối HTML và JS trong cùng 1 source



**Component:** là một khối code độc lập, có thể tái sử dụng, chia UI thành nhiều phần nhỏ

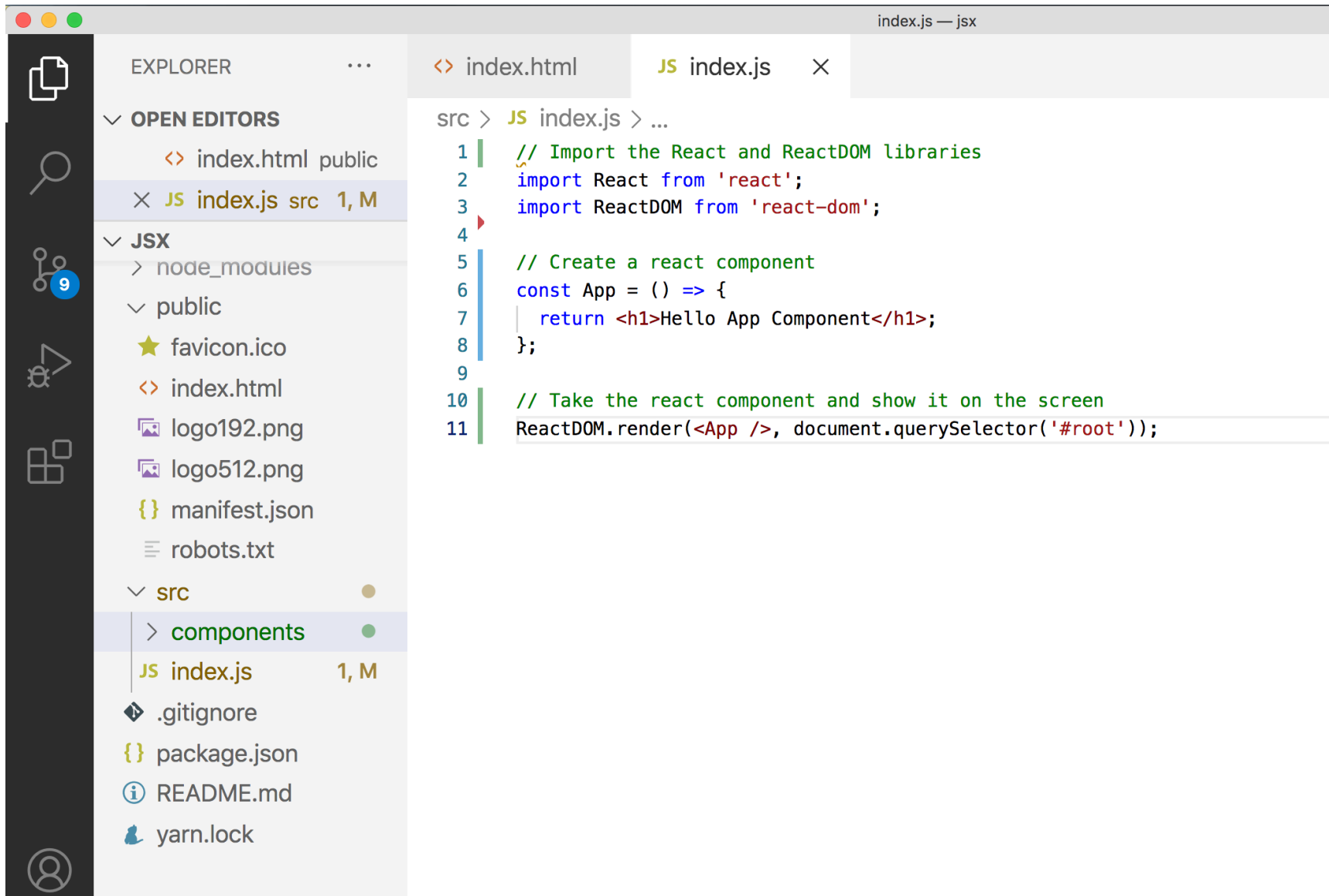
The screenshot shows a VS Code editor with two files open: `index.html` and `index.js`. The `index.html` file contains a `<div id='root'>` tag. The `index.js` file contains the following code:

```
src > JS index.js > ...
1 // Import the React and ReactDOM libraries
2 import React from 'react';
3 import ReactDOM from 'react-dom';
4
5 // Create a react component
6 const HelloWorld = () => {
7   return <h1>Hello world Component</h1>;
8 };
9
10 // Take the react component and show it on the screen
11 ReactDOM.render(<HelloWorld />, document.querySelector('#root'));
12
13
14
```

Two arrows originate from the text below: one red arrow points from the text "HelloWorld component" to the `<HelloWorld />` tag in the `ReactDOM.render` call on line 11, and one black arrow points from the text "ReactDOM.render được sử dụng để render nội dung của HelloWorld component ra thẻ <div id='root'> của trang index.html" to the `<div id='root'>` tag in the `index.html` file.

HelloWorld component

ReactDOM.render được sử dụng để render nội dung của **HelloWorld** component ra thẻ `<div id='root'>` của trang index.html



The image shows a screenshot of the Visual Studio Code (VS Code) editor interface. On the left is the Explorer sidebar, which displays the project's file structure. The 'src' directory is expanded, showing a 'components' subdirectory. Inside 'components', the file 'index.js' is selected. The main editor area on the right displays the content of 'index.js', which is a React component. The code includes imports for React and ReactDOM, a functional component definition 'App', and a call to ReactDOM.render to mount the component on the page.

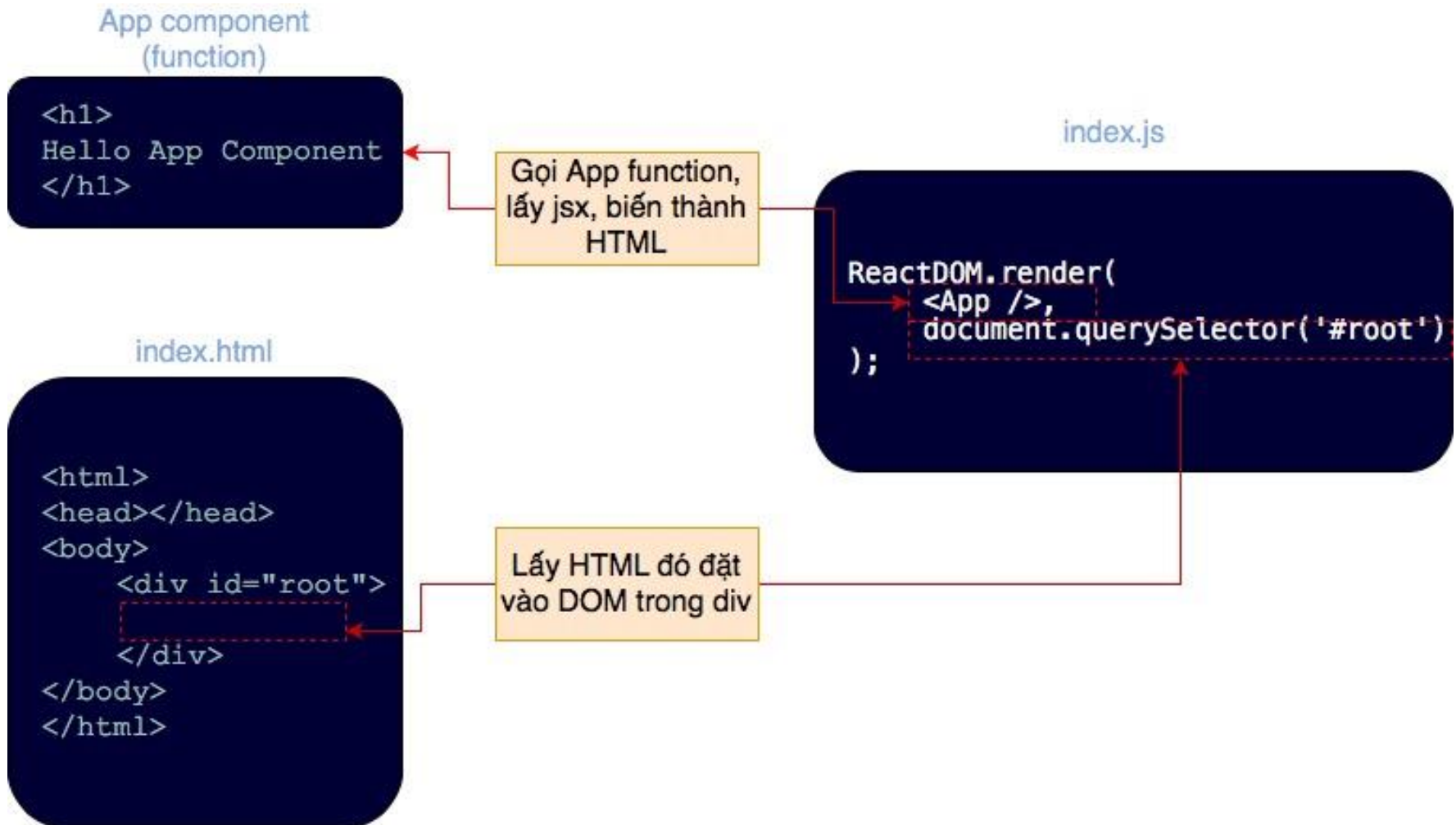
**EXPLORER**

- OPEN EDITORS
  - index.html public
  - JS index.js src 1, M
- JSX
  - node\_modules
  - public
    - favicon.ico
    - index.html
    - logo192.png
    - logo512.png
    - manifest.json
    - robots.txt
  - src
    - components
      - index.js 1, M
    - .gitignore
    - package.json
    - README.md
    - yarn.lock

**index.js — jsx**

```

src > JS index.js > ...
1 | // Import the React and ReactDOM libraries
2 | import React from 'react';
3 | import ReactDOM from 'react-dom';
4 |
5 | // Create a react component
6 | const App = () => {
7 |   return <h1>Hello App Component</h1>;
8 | };
9 |
10 | // Take the react component and show it on the screen
11 | ReactDOM.render(<App />, document.querySelector('#root'));
```



```
<div>
  <label for="name">Enter name:
</label>
  <input id="name" type="text" />
  <button>Submit</button>
</div>
```

Enter name:

```
// Create a react component
```

```
const App = () => {
  return <div>
    <label for="name">Enter name:</label>
    <input id="name" type="text" />
    <button>Submit</button>
  </div>;
};
```

← → ↻ ⓘ localhost:3001

Enter name:



# ĐỔI STYLE TRONG HTML > JSX

```
<div>
  <label for="name">Enter name:
</label>
  <input id="name" type="text" />
  <button style="background-
color:blue; color:
white;">Submit</button>
</div>
```

Enter name:

```
// Create a react component
```

```
const App = () => {
  return <div>
    <label for="name">
      Enter name:
    </label>
    <input id="name" type="text" />
    <button style={{ backgroundColor:'blue', color:'white' }}>
      Submit
    </button>
  </div>;
};
```

← → ↻ ⓘ localhost:3001

Enter name:

# INLINE STYLE TRONG HTML > JSX

HTML

```
<div style="background-color: blue"></div>
```



JSX

```
<div style={{ backgroundColor: 'blue' }}></div>
```

HTML (attribute)

```
for="label"
```



JSX

```
htmlFor="label"
```

```
<div>
  <label for="name"
class="label">
    Enter name:
  </label>
  <input id="name" type="text" />
  <button style=" background-
color:blue;color:white">
    Submit
  </button>
</div>
```

Enter name:

Submit

```
// Create a react component
```

```
const App = () => {
  return <div>
    <label for="name" className="label">
      Enter name:
    </label>
    <input id="name" type="text" />
    <button style={{ backgroundColor:'blue', color:'white' }}>
      Submit
    </button>
  </div>;
};
```

```
// Create a react component
const App = () => {
  const buttonText = 'Click on me';
  const labelText = 'Enter name: ';

  return (
    <div>
      <label className="label" for="name">
        {labelText}
      </label>
      <input id="name" type="text" />
      <button style={{ backgroundColor: 'blue', color: 'white' }}>
        {buttonText}
      </button>
    </div>
  );
};
```

# THAM CHIẾU BIẾN JS TRONG JSX

```
const buttonText = 'Click me';
```

```
{buttonText}
```

```
const buttonText = 123456;
```

```
{buttonText}
```

```
const buttonText = ['Hi', 'There'];
```

```
{buttonText}
```

```
const buttonText = [10, 20];
```

```
{buttonText}
```

```
const buttonText = { text: 'Click me'};
```

```
{buttonText.text}
```

demo

- ☑ Tổng quan ReactJS
- ☑ Cài đặt ReactJS
- ☑ JSX



thank  
you!