

 免费电子书

学习

React

Free unaffiliated eBook created from
Stack Overflow contributors.

#reactjs

.....	1
1: React	2
.....	2
.....	2
Examples.....	3
.....	3
Hello World Component.....	4
.....	5
ReactJS.....	6
Hello World.....	6
.....	7
React App.....	7
.....	8
.....	8
.....	8
.....	8
.....	8
2: JSX	10
.....	10
Examples.....	10
JSX.....	10
JavaScript	10
.....	11
.....	11
.....	11
JSX.....	12
.....	12
JSX	12
JavaScript	12
.....	13
.....	

- 3: React Boilerplate [React + Babel + Webpack] 15**
 - Examples 15
 - 15
 - 17
- 4: ReactWebpackTypescript 19**
 - 19
 - Examples 19
 - webpack.config.js 19
 - 19
- TS 19**
 - tsconfig.json 19
 - include 20
 - compilerOptions.target 20
 - compilerOptions.jsx 20
 - compilerOptions.allowSyntheticDefaultImports 20
 - 20
- 5: React.createClass vs React.Component 22**
 - 22
 - 22
 - Examples 22
 - React 22
- React.createClass 22**
- React.Component 22**
 - PropTypes 23
- React.createClass 23**
 - React.Component 23
 - 24
- React.createClass 24**
- React.Component 25**
 - 25

React.createClass	25
React.Component	25
"'"	26
React.createClass	26
React.Component	26
1	27
2	27
3ES6	27
ES6 /"ajax""this"	28
6: React	29
.....	29
Examples	29
.....	29
.....	29
PropTypes	30
.....	31
Props.children	32
.....	33
7: Redux	34
.....	34
.....	34
Examples	34
Connect	34
8: FluxReactJS	36
.....	36
.....	36
Examples	36
.....	36
.....	36
9: React with Flow	37
.....	37
.....	

Examples.....	37
Flowprop.....	37
Flow.....	37
10: AJAX.....	38
Examples.....	38
HTTP GET.....	38
ReactAjax - VanillaJS.....	39
HTTP GET.....	39
11:	41
Examples.....	41
.....	41
setState.....	41
setState()Objectupdater.....	41
FunctionsetState()updater.....	42
ObjectsetState().....	42
.....	43
.....	44
12:	46
Examples.....	46
.....	46
13:	47
Examples.....	47
.....	47
14:	49
.....	49
Examples.....	49
.....	49
getDefaultProps() ES5.....	49
getInitialState() ES5.....	49
componentWillMount() ES5ES6.....	49

render() ES5ES6.....	50
componentDidMount() ES5ES6.....	50
ES6.....	50
getDefaultProps().....	51
getInitialState().....	51
.....	51
componentWillReceiveProps(nextProps).....	51
shouldComponentUpdate(nextProps, nextState).....	52
componentWillUpdate(nextProps, nextState).....	52
render().....	52
componentDidUpdate(prevProps, prevState).....	52
.....	52
componentWillUnmount().....	52
.....	53
.....	54
15:	56
Examples.....	56
Routes.jsRouter Link.....	56
.....	57
16:	58
Examples.....	58
.....	58
17: React	60
.....	60
.....	60
Examples.....	60
.....	60
18: webpackbabel	61
.....	61
Examples.....	61
“Hello world”.....	61

19:	66
Examples	66
.....	66
.....	66
.....	66
webpack	66
.....	67
webpack-dev-server	67
.....	67
webpack.config.js	67
20: ReactJSjQuery	69
Examples	69
ReactJSjQuery	69
21: ReactJSTypescript	71
Examples	71
ReactJSTypescript	71
.....	71
.....	72
.....	73
22:	74
Examples	74
- HTML DOMDOM	74
Reactdiff	74
.....	75
ReactJS	75
23:	77
.....	77
Examples	77
.....	77
24:	80
Examples	80
.....	

<code>renderToString</code>	80
<code>renderToStaticMarkup</code>	80
25:	81
.....	81
Examples.....	81
.....	81
.....	81
.....	82
<code>panelGroup</code>	82
.....	83
<code>`PanelGroup`s</code>	83
26:	85
.....	85
Examples.....	85
.....	85
.....	86
1.	86
.....	87
.....	87
.....	87
2.	87
.....	87
.....	87
.....	87
3.	88
.....	88
.....	88
.....	88
.....	88
.....	88
.....	

.....	89
.....	90
setState.....	90
.....	92
-	93
.....	94
27:	96
.....	96
Examples.....	96
.....	96
.....	96
.....	97
28:	98
Examples.....	98
.....	98
.....	98
29: React	100
Examples.....	100
.....	100
.....	100
webpack.....	100
babel.....	100
HTML.....	101
.....	101
30:	102
.....	102
.....	102
Examples.....	102
id.....	102
.....	103

31:	104
.....	104
.....	104
Examples.....	104
.....	104
.....	104
.....	107

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [react](#)

It is an unofficial and free React ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official React.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: React

[React](#) [JavaScript](#) ◦

[React](#) [MVC](#) [Flux](#) [Redux](#) ◦

0.3.0	2013529
0.4.0	2013717
0.5.0	20131016
0.8.0	
0.9.0	2014220
0.10.0	2014321
0.11.0	2014717
0.12.0	20141028
0.13.0	2015310
0.14.0	2015107
15.0.0	201647
15.1.0	
15.2.0	201671
15.2.1	201678
15.3.0	2016729
15.3.1	2016819
15.3.2	2016919
15.4.0	20161116
15.4.1	20161123
15.4.2	201716
15.5.0	201747

15.6.0	2017613
--------	---------

Examples

ReactJS `react-<version>.js` JavaScriptHTML。 React DOM `react-dom-<version>.js` React

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <script type="text/javascript" src="/path/to/react.js"></script>
    <script type="text/javascript" src="/path/to/react-dom.js"></script>
    <script type="text/javascript">
      // Use react JavaScript code here or in a separate file
    </script>
  </body>
</html>
```

JavaScriptReact。

ReactJSX。 JSXFacebookJavaScriptXML。 JSXBabel `<script type="text/javascript"><script type="text/babel">` JSXJavascript。

```
<!DOCTYPE html>
<html>
  <head></head>
  <body>
    <script type="text/javascript" src="/path/to/react.js"></script>
    <script type="text/javascript" src="/path/to/react-dom.js"></script>
    <script src="https://npmcdn.com/babel-core@5.8.38/browser.min.js"></script>
    <script type="text/babel">
      // Use react JSX code here or in a separate file
    </script>
  </body>
</html>
```

npm

npmReact

```
npm install --save react react-dom
```

JavaScriptReact

```
var React = require('react');
var ReactDOM = require('react-dom');
ReactDOM.render(<App />, ...);
```

FacebookYarnReact。 Yarn

```
yarn add react react-dom
```

ReactnpmReact。

Hello World Component

ReactReact.ComponentES6。 renderDOM。 renderReactJSXHTML。

```
import React from 'react'

class HelloWorld extends React.Component {
  render() {
    return <h1>Hello, World!</h1>
  }
}

export default HelloWorld
```

props。 ; - onClick。 Componentpropsprops

```
import React from 'react'

class Hello extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}!</h1>
  }
}

export default Hello
```

name prop。 。

DOMReactDOM.renderDOMReact

```
import React from 'react'
import ReactDOM from 'react-dom'
import Hello from './Hello'

ReactDOM.render(<Hello name="Billy James" />, document.getElementById('main'))
```

props。 state。 。

Hello World。

```
import React from 'react'

class Hello extends React.Component {

  constructor(props) {

    //Since we are extending the default constructor,
    //handle default activities first.
    super(props);

    //Extract the first-name from the prop
```

```
    let firstName = this.props.name.split(" ")[0];

    //In the constructor, feel free to modify the
    //state property on the current context.
    this.state = {
      name: firstName
    }

  } //Look maa, no comma required in JSX based class defs!

  render() {
    return <h1>Hello, {this.state.name}!</h1>
  }
}

export default Hello
```

o

[Codepen](#)

JSX

ReactAPIReactReact DOM API

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello React!</title>

    <!-- Include the React and ReactDOM libraries -->
    <script src="https://fb.me/react-15.2.1.js"></script>
    <script src="https://fb.me/react-dom-15.2.1.js"></script>

  </head>
  <body>
    <div id="example"></div>

    <script type="text/javascript">

      // create a React element rElement
      var rElement = React.createElement('h1', null, 'Hello, world!');

      // dElement is a DOM container
      var dElement = document.getElementById('example');

      // render the React element in the DOM container
      ReactDOM.render(rElement, dElement);

    </script>

  </body>
</html>
```

JSX

ReactJSXFacebookJavascriptJavaScriptXML

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

HTML

```
var rElement = <h1>Hello, world!</h1>;
```

JSX<script type="text/babel">◦ BabelJavascriptReact◦

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello React!</title>

    <!-- Include the React and ReactDOM libraries -->
    <script src="https://fb.me/react-15.2.1.js"></script>
    <script src="https://fb.me/react-dom-15.2.1.js"></script>
    <!-- Include the Babel library -->
    <script src="https://npmcdn.com/babel-core@5.8.38/browser.min.js"></script>

  </head>
  <body>
    <div id="example"></div>

    <script type="text/babel">

      // create a React element rElement using JSX
      var rElement = <h1>Hello, world!</h1>;

      // dElement is a DOM container
      var dElement = document.getElementById('example');

      // render the React element in the DOM container
      ReactDOM.render(rElement, dElement);

    </script>

  </body>
</html>
```

ReactJS

ReactJS ◦ Facebook◦

ReactJSDOMHTML DOM◦ DOMDOMDOM

React HTML◦ ◦ - TabList◦

ReactJSX◦ JSXHTMLJavaScript◦ ReactDOMHTML◦

“ ”DOMHTML DOM◦ DOMDOM◦ DOMDOMReact◦

Hello World

◦ ◦

```
const statelessSum = (a, b) => a + b;

let a = 0;
const statefulSum = () => a++;
```

statelessSum ◦ statefulSum ◦ ◦ ◦

◦ ◦

◦ **ReactJavaScript** ◦ Stateless Functional Components ◦ props ◦ state ◦

Stateless Functional Component

```
// In HTML
<div id="element"></div>

// In React
const MyComponent = props => {
  return <h1>Hello, {props.name}!</h1>;
};

ReactDOM.render(<MyComponent name="Arun" />, element);
// Will render <h1>Hello, Arun!</h1>
```

name prop ◦ h1 ◦ ◦ **ES6**

```
import React from 'react'

const HelloWorld = props => (
  <h1>Hello, {props.name}!</h1>
)

HelloWorld.propTypes = {
  name: React.PropTypes.string.isRequired
}

export default HelloWorld
```

React ◦ ◦

React App

[create-react-app](#) ◦ Facebook ◦ React ◦

- ES6JSX
-
- linting

- CSS
- JSCSS
- Jest

npmcreate-react-app。

```
npm install -g create-react-app
```

。

```
create-react-app my-app
```

。

```
cd my-app/  
npm start
```

create-react-app。CSSSasseject。

```
npm run eject
```

。。

React

-
- [NWB](#)
-
- [react-CLI](#)
-
- [RWB](#)
- [QUIK](#)
- [sagui](#)
-

React

```
npm run build
```

ReactReact。 “Props”JSX<SomeComponent someProp="some prop's value" />。 SomeComponent
this.props {someProp: "some prop's value"}。

React - “props”。 “”。 “”。 React

- “”

```
const FirstComponent = props => (  
  <div>{props.content}</div>  
)
```

- **React.createClass**

```
const SecondComponent = React.createClass({  
  render: function () {  
    return (  
      <div>{this.props.content}</div>  
    );  
  }  
});
```

- **ES2015**

```
class ThirdComponent extends React.Component {  
  render() {  
    return (  
      <div>{this.props.content}</div>  
    );  
  }  
}
```

```
const ParentComponent = function (props) {  
  const someText = "FooBar";  
  return (  
    <FirstComponent content={someText} />  
    <SecondComponent content={someText} />  
    <ThirdComponent content={someText} />  
  );  
}
```

-
- “” ◦ ◦
- propReactnewprops◦
-

React <https://riptutorial.com/zh-CN/reactjs/topic/797/react>

2: JSX

JSX JavaScriptXML。 JSXReactJSXReact。

XMLJSX。 。 JavaScript。

JSX`React.createElement(component, props, ...children)`。

JSX

```
class HelloMessage extends React.Component {
  render() {
    return <div>Hello {this.props.name}</div>;
  }
}

ReactDOM.render(<HelloMessage name="Kalo" />, mountNode);
```

JavaScript

```
class HelloMessage extends React.Component {
  render() {
    return React.createElement(
      "div",
      null,
      "Hello ",
      this.props.name
    );
  }
}

ReactDOM.render(React.createElement(HelloMessage, { name: "Kalo" }), mountNode);
```

JSXHTML

```
const element = <h1>Hello, world!</h1>;
```

JSXJavaScript。 JSXJavaScript。

ReactUI。

Examples

JSX

JSX。

JavaScript

JavaScriptprop{}◦ JSX

```
<MyComponent count={1 + 2 + 3 + 4} />
```

MyComponent props.count10 1 + 2 + 3 + 4◦

forJavaScriptJSX◦

string literalprop◦ JSX

```
<MyComponent message="hello world" />
```

```
<MyComponent message={'hello world'} />
```

HTML-unescaped◦ JSX

```
<MyComponent message="&lt;3" />
```

```
<MyComponent message={'<3'} />
```

◦ ◦

prop true◦ JSX

```
<MyTextBox autocomplete />
```

```
<MyTextBox autocomplete={true} />
```

React ES6{foo}{foo: foo}{foo: true}◦ HTML◦

JSX...◦

```
function Case1() {  
  return <Greeting firstName="Kaloyab" lastName="Kosev" />;  
}
```

```
function Case2() {  
  const person = {firstName: 'Kaloyan', lastName: 'Kosev'};  
  return <Greeting {...person} />;  
}
```

JSX

JSXprop props.children ◦

props.children◦ HTML◦

```
<MyComponent>
  <h1>Hello world!</h1>
</MyComponent>
```

JSXMyComponentprops.children<h1>Hello world!</h1> ◦

HTML HTMLJSX◦

JSX

- ;
- ;
- ;
- ◦

JSX

JSX◦

```
<MyContainer>
  <MyFirstComponent />
  <MySecondComponent />
</MyContainer>
```

JSX◦ JSXHTMLJSXHTML

```
<div>
  <h2>Here is a list</h2>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
  </ul>
</div>
```

ReactReactJSX◦ div◦

JavaScript

JavaScript {} ◦

```
<MyComponent>foo</MyComponent>

<MyComponent>{'foo'}</MyComponent>
```

JSX ◦ HTML

```
const Item = ({ message }) => (
  <li>{ message }</li>
);

const TodoList = () => {
  const todos = ['finish doc', 'submit review', 'wait stackoverflow review'];
  return (
    <ul>
      { todos.map(message => (<Item key={message} message={message} />)) }
    </ul>
  );
};
```

JavaScript ◦

JSX JavaScript React ◦ props.children React ◦ props.children

```
const ListOfTenThings = () => (
  <Repeat numTimes={10}>
    {(index) => <div key={index}>This is item {index} in the list</div>}
  </Repeat>
);

// Calls the children callback numTimes to produce a repeated component
const Repeat = ({ numTimes, children }) => {
  let items = [];
  for (let i = 0; i < numTimes; i++) {
    items.push(children(i));
  }
  return <div>{items}</div>;
};
```

React ◦ JSX ◦

false null undefined true ◦ ◦ JSX

```
<MyComponent />

<MyComponent></MyComponent>

<MyComponent>{false}</MyComponent>
```

```
<MyComponent>{null}</MyComponent>
```

```
<MyComponent>{true}</MyComponent>
```

React^o JSXif showHeadertrue

```
<div>
  {showHeader && <Header />}
  <Content />
</div>
```

“”^oReact^o props.messages0

```
<div>
  {props.messages.length &&
    <MessageList messages={props.messages} />
  }
</div>
```

&&boolean

```
<div>
  {props.messages.length > 0 &&
    <MessageList messages={props.messages} />
  }
</div>
```

false true nullundefined

```
<div>
  My JavaScript variable is {String(myVariable)}.
</div>
```

JSX <https://riptutorial.com/zh-CN/reactjs/topic/8027/jsx>

3: React Boilerplate [React + Babel + Webpack]

Examples

Node Package Manager。 [Nodejs.org](https://nodejs.org)。 ◦

npm。 ◦ ◦

BabelWebpackReact。

◦ [BabelWebpack](#)。

npm init。 ◦ -

```
npm install react react-dom --save
```

Dev Dependecies

```
npm install babel-core babel-loader babel-preset-es2015 babel-preset-react babel-preset-stage-0 webpack webpack-dev-server react-hot-loader --save-dev
```

Dev

```
npm install eslint eslint-plugin-react babel-eslint --save-dev
```

package.json

.babelrc

```
{
  "presets": ["es2015", "stage-0", "react"]
}
```

.eslintrc

```
{
  "ecmaFeatures": {
    "jsx": true,
    "modules": true
  },
  "env": {
    "browser": true,
    "node": true
  },
  "parser": "babel-eslint",
  "rules": {
    "quotes": [2, "single"],
    "strict": [2, "never"],
  },
  "plugins": [
```

```
  "react"
]
}
```

.gitignore^o

```
node_modules
npm-debug.log
.DS_Store
dist
```

webpack.config.js^o

```
var path = require('path');
var webpack = require('webpack');

module.exports = {
  devtool: 'eval',
  entry: [
    'webpack-dev-server/client?http://localhost:3000',
    'webpack/hot/only-dev-server',
    './src/index'
  ],
  output: {
    path: path.join(__dirname, 'dist'),
    filename: 'bundle.js',
    publicPath: '/static/'
  },
  plugins: [
    new webpack.HotModuleReplacementPlugin()
  ],
  module: {
    loaders: [{
      test: /\.js$/,
      loaders: ['react-hot', 'babel'],
      include: path.join(__dirname, 'src')
    }]
  }
};
```

sever.js^o npm start

```
var webpack = require('webpack');
var WebpackDevServer = require('webpack-dev-server');
var config = require('./webpack.config');

new WebpackDevServer(webpack(config), {
  publicPath: config.output.publicPath,
  hot: true,
  historyApiFallback: true
}).listen(3000, 'localhost', function (err, result) {
  if (err) {
    return console.log(err);
  }

  console.log('Serving your awesome project at http://localhost:3000/');
});
```

src/app.js

```
import React, { Component } from 'react';

export default class App extends Component {
  render() {
    return (
      <h1>Hello, world.</h1>
    );
  }
}
```

package.json start node server.js npm start

◦ ReactJs + Webpack + Babel

◦ ◦ ◦

GUI ◦ npm init package.json git ◦ ◦ ◦ yes ◦ package.json

npm install express @ 4 --save ◦ ◦ package.js ◦ package.json ◦ 4.package.json "dependencies":
{ "express": "^4.13.4", },

node_modules ◦ server.js ◦ ◦

```
var express = require('express');
// Create our app
var app = express();

app.use(express.static('public'));

app.listen(3000, function () {
  console.log('Express server is using port:3000');
});
```

var express = require('express'); app

var app = express; app.use; app.use express.static "/Web app.listen port function {} 3000 Web
◦ ◦

public index.html ◦ index.html ◦ Express ◦ index.html/html

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8"/>
</head>

<body>
  <h1>hello World</h1>
</body>

</html>
```

`node server.js` ◦ `* console.log'Express3000'; *` ◦

<http://localhost3000> `hello World` ◦

`app.jsx` ◦ JSXJavaScriptXML ◦ ReactJSXJSXReact ◦ `app.jsx`

```
ReactDOM.render(  
  <h1>Hello World!!!</h1>,  
  document.getElementById('app')  
)
```

`index.html`

```
<!DOCTYPE html>  
<html>  
  
<head>  
  <meta charset="UTF-8"/>  
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.23/  
/browser.min.js"></script>  
  <script src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.7/react.js">  
</script>  
  <script src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.7/react-dom.js"> </script>  
</head>  
  
<body>  
  <div id="app"></div>  
  
  <script type="text/babel" src="app.jsx"></script>  
</body>  
  
</html>
```

◦

[React Boilerplate \[React + Babel + Webpack\] https://riptutorial.com/zh-CN/reactjs/topic/5969/react-boilerplate--react-plus-babel-plus-webpack-](https://riptutorial.com/zh-CN/reactjs/topic/5969/react-boilerplate--react-plus-babel-plus-webpack-)

4: ReactWebpackTypescript

VS.

ReactReactDOMIntellisense.

```
npm install --save @types/react @types/react-dom
```

。

Examples

webpack.config.js

```
module.exports = {
  entry: './src/index',
  output: {
    path: __dirname + '/build',
    filename: 'bundle.js'
  },
  module: {
    rules: [{
      test: /\.tsx?$/,
      loader: 'ts-loader',
      exclude: /node_modules/
    }]
  },
  resolve: {
    extensions: ['.ts', '.tsx']
  }
};
```

entry output **webpack**

.ts.tsx **ts-loader**

TS

resolve.ts.tsx **webpack**

tsconfig.json

tsconfig.

```
{
  "include": [
```

```
    "src/*"
  ],
  "compilerOptions": {
    "target": "es5",
    "jsx": "react",
    "allowSyntheticDefaultImports": true
  }
}
```

include

◦ src/* src◦

compilerOptions.target

ES5

compilerOptions.jsx

trueTypeScripttsx<div />React.createElement("div") ◦

compilerOptions.allowSyntheticDefaultImports

ES6

```
import * as React from 'react'
const { Component } = React
```

```
import React, { Component } from 'react'
```

React◦

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';

interface AppProps {
  name: string;
}
interface AppState {
  words: string[];
}

class App extends Component<AppProps, AppState> {
  constructor() {
    super();
    this.state = {
      words: ['foo', 'bar']
    };
  }

  render() {
    const { name } = this.props;
    return (<h1>Hello {name}!</h1>);
  }
}
```

```
    }  
  }  
  
  const root = document.getElementById('root');  
  ReactDOM.render(<App name="Foo Bar" />, root);
```

TypeScriptReactReact DefinitelyTyped `npm install --save @types/react` ◦

```
class App extends Component<AppProps, AppState> { }
```

AppPropsAppState**propsstate**◦

ReactWebpackTypescript <https://riptutorial.com/zh-CN/reactjs/topic/9590/react-webpacktypescript>

5: React.createClass vs React.Component

- 1 React.createClass {}
- 2 MyComponent React.Component {}

React.createClass React.createClass v16 ◦ ◦ ◦

Examples

React

◦

React.createClass

React const render ◦

```
import React from 'react';

const MyComponent = React.createClass({
  render() {
    return (
      <div></div>
    );
  }
});

export default MyComponent;
```

React.Component

React.createClassES6 ◦

```
import React from 'react';

class MyComponent extends React.Component {
  render() {
    return (
      <div></div>
    );
  }
}

export default MyComponent;
```

ES6 ◦ React MyComponent React.Component React.createClass ◦ ReactJavaScript ◦

PropTypes

。

React.createClass

propTypesObjectprop。 getDefaultPropsObjectprops。

```
import React from 'react';

const MyComponent = React.createClass({
  propTypes: {
    name: React.PropTypes.string,
    position: React.PropTypes.number
  },
  getDefaultProps() {
    return {
      name: 'Home',
      position: 1
    };
  },
  render() {
    return (
      <div></div>
    );
  }
});

export default MyComponent;
```

React.Component

propTypes**MyComponent**createClassObject。

getDefaultPropsdefaultPropsObject“get”Object。 ReactJavaScript。

```
import React from 'react';

class MyComponent extends React.Component {
  constructor(props) {
    super(props);
  }
  render() {
    return (
      <div></div>
    );
  }
}

MyComponent.propTypes = {
  name: React.PropTypes.string,
```

```

    position: React.PropTypes.number
  };
  MyComponent.defaultProps = {
    name: 'Home',
    position: 1
  };

  export default MyComponent;

```

propTypesdefaultProps° ES7

```

import React from 'react';

class MyComponent extends React.Component {
  static propTypes = {
    name: React.PropTypes.string,
    position: React.PropTypes.number
  };
  static defaultProps = {
    name: 'Home',
    position: 1
  };
  constructor(props) {
    super(props);
  }
  render() {
    return (
      <div></div>
    );
  }
}

export default MyComponent;

```

°

React.createClass

getInitialState getInitialStateObject°

```

import React from 'react';

const MyComponent = React.createClass({
  getInitialState () {
    return {
      activePage: 1
    };
  },
  render() {
    return (
      <div></div>
    );
  }
});

```

```
export default MyComponent;
```

React.Component

getInitialState ◦ “React API” JavaScript ◦

```
import React from 'react';

class MyComponent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      activePage: 1
    };
  }
  render() {
    return (
      <div></div>
    );
  }
}

export default MyComponent;
```

React.createClass ◦ mixins ◦

React.createClass

mixins ◦ mixins ◦ mixin ◦ ◦

```
import React from 'react';

var MyMixin = {
  doSomething() {

  }
};

const MyComponent = React.createClass({
  mixins: [MyMixin],
  handleClick() {
    this.doSomething(); // invoke mixin's method
  },
  render() {
    return (
      <button onClick={this.handleClick}>Do Something</button>
    );
  }
});

export default MyComponent;
```

React.Component

ES6 React mixins ReactES6

“”

React.createClass this ES6

React.createClass

this.handleClick onClick React handleClick

```
import React from 'react';

const MyComponent = React.createClass({
  handleClick() {
    console.log(this); // the React Component instance
  },
  render() {
    return (
      <div onClick={this.handleClick}></div>
    );
  }
});

export default MyComponent;
```

React.Component

ES6 this null React

```
import React from 'react';

class MyComponent extends React.Component {
  constructor(props) {
    super(props);
  }
  handleClick() {
    console.log(this); // null
  }
  render() {
    return (
      <div onClick={this.handleClick}></div>
    );
  }
}

export default MyComponent;
```

this

1

```
import React from 'react';

class MyComponent extends React.Component {
  constructor(props) {
    super(props);
  }
  handleClick() {
    console.log(this); // the React Component instance
  }
  render() {
    return (
      <div onClick={this.handleClick.bind(this)}></div>
    );
  }
}

export default MyComponent;
```

2

constructorthis.handleClick ◦ JSX

```
import React from 'react';

class MyComponent extends React.Component {
  constructor(props) {
    super(props);
    this.handleClick = this.handleClick.bind(this);
  }
  handleClick() {
    console.log(this); // the React Component instance
  }
  render() {
    return (
      <div onClick={this.handleClick}></div>
    );
  }
}

export default MyComponent;
```

3ES6

ES6

```
import React from 'react';

class MyComponent extends React.Component {
  constructor(props) {
```

```

    super(props);
  }
  handleClick = () => {
    console.log(this); // the React Component instance
  }
  render() {
    return (
      <div onClick={this.handleClick}></div>
    );
  }
}

export default MyComponent;

```

ES6 /“ajax”“this”

```

import React from 'react';

class SearchEs6 extends React.Component{
  constructor(props) {
    super(props);
    this.state = {
      searchResults: []
    };
  }

  showResults(response){
    this.setState({
      searchResults: response.results
    })
  }

  search(url){
    $.ajax({
      type: "GET",
      dataType: 'jsonp',
      url: url,
      success: (data) => {
        this.showResults(data);
      },
      error: (xhr, status, err) => {
        console.error(url, status, err.toString());
      }
    });
  }

  render() {
    return (
      <div>
        <SearchBox search={this.search.bind(this)} />
        <Results searchResults={this.state.searchResults} />
      </div>
    );
  }
}

```

[React.createClass vs React.Component](https://riptutorial.com/zh-CN/reactjs/topic/6371/react-createclass-vsreact-component) <https://riptutorial.com/zh-CN/reactjs/topic/6371/react-createclass-vsreact-component>

6: React

React 15.5PropTypesnpm'prop-types'PropTypesimport。 <https://facebook.github.io/react/blog/2017/04/07/react-v15.5.0.html>

Examples

props°

props

1. °
 2. °
-

```
class Parent extends React.Component {
  doSomething() {
    console.log("Parent component");
  }
  render() {
    return <div>
      <Child
        text="This is the child number 1"
        title="Title 1"
        onClick={this.doSomething} />
      <Child
        text="This is the child number 2"
        title="Title 2"
        onClick={this.doSomething} />
    </div>
  }
}

class Child extends React.Component {
  render() {
    return <div>
      <h1>{this.props.title}</h1>
      <h2>{this.props.text}</h2>
    </div>
  }
}
```

props°

defaultPropsprops° defaultProps°

ES5

```
var MyClass = React.createClass({
  getDefaultProps: function() {
    return {
```

```
    randomObject: {},
    ...
  };
}
```

ES6

```
class MyClass extends React.Component {...}

MyClass.defaultProps = {
  randomObject: {},
  ...
}
```

ES7

```
class MyClass extends React.Component {
  static defaultProps = {
    randomObject: {},
    ...
  };
}
```

`getDefaultProps()` `defaultPropsthis.props.randomObject`

PropTypes

`propTypesprops` `propTypesReact`

`propTypespropTypes` -

```
optionalArray: React.PropTypes.array,
optionalBool: React.PropTypes.bool,
optionalFunc: React.PropTypes.func,
optionalNumber: React.PropTypes.number,
optionalObject: React.PropTypes.object,
optionalString: React.PropTypes.string,
optionalSymbol: React.PropTypes.symbol
```

`isRequiredpropTypesprop` `propTypes`

ES5

```
var MyClass = React.createClass({
  propTypes: {
    randomObject: React.PropTypes.object,
    callback: React.PropTypes.func.isRequired,
    ...
  }
});
```



```
}
```

ES6

```
class MyClass extends React.Component {...}

MyClass.propTypes = {
  randomObject: React.PropTypes.object,
  callback: React.PropTypes.func.isRequired,
  ...
};
```

ES7

```
class MyClass extends React.Component {
  static propTypes = {
    randomObject: React.PropTypes.object,
    callback: React.PropTypes.func.isRequired,
    ...
  };
}
```

PropTypes

```
...
  randomObject: React.PropTypes.shape({
    id: React.PropTypes.number.isRequired,
    text: React.PropTypes.string,
  }).isRequired,
...

```

```
...
  arrayOfObjects: React.PropTypes.arrayOf(React.PropTypes.shape({
    id: React.PropTypes.number.isRequired,
    text: React.PropTypes.string,
  })).isRequired,
...

```

```
var component = <Component foo={this.props.x} bar={this.props.y} />;
```

prop...ES6。。

```
var component = <Component {...props} />;
```

props。

。

```
var props = { foo: 'default' };
```

```
var component = <Component {...props} foo={'override'} />;
console.log(component.props.foo); // 'override'
```

◦

◦

```
const { foo, bar, other } = this.props // { foo: 'foo', bar: 'bar', other: 'other' };
var component = <Component {...{foo, bar}} />;
const { foo, bar } = component.props
console.log(foo, bar); // 'foo bar'
```

Props.children

“”props.children ◦ prop“”JSXDOM

```
var SomeComponent = function () {
  return (
    <article className="textBox">
      <header>{this.props.heading}</header>
      <div className="paragraphs">
        {this.props.children}
      </div>
    </article>
  );
}
```

```
var ParentComponent = function () {
  return (
    <SomeComponent heading="Amazing Article Box" >
      <p className="first"> Lots of content </p>
      <p> Or not </p>
    </SomeComponent>
  );
}
```

Props.children ◦ props.children [React.Children](#) ◦ <section>

```
var SomeComponent = function () {
  return (
    <article className="textBox">
      <header>{this.props.heading}</header>
      <div className="paragraphs">
        {React.Children.map(this.props.children, function (child) {
          return (
            <section className={child.props.className}>
              React.cloneElement(child)
            </section>
          );
        })}
      </div>
    </article>
  );
}
```

React.cloneElement<p>props - props◦ ◦

React key◦

◦ **React Children.map util**

```
React.Children.map(this.props.children, (child) => {  
  if (child.type === MyComponentType) {  
    ...  
  }  
});
```

type◦

React <https://riptutorial.com/zh-CN/reactjs/topic/2749/react>

7: Redux

Redux。 ReactRedux。

React。 Redux。 。 Redux。

Examples

Connect

createStore Redux。

```
import { createStore } from 'redux'
import todoApp from './reducers'
let store = createStore(todoApp, { initialStateVariable: "derp"})
```

connect Redux。

```
import { connect } from 'react-redux'

const VisibleTodoList = connect(
  mapStateToProps,
  mapDispatchToProps
)(TodoList)

export default VisibleTodoList
```

Redux。

```
/*
 * action types
 */

export const ADD_TODO = 'ADD_TODO'

export function addTodo(text) {
  return { type: ADD_TODO, text }
}
```

reducer。

```
function todoApp(state = initialState, action) {
  switch (action.type) {
    case SET_VISIBILITY_FILTER:
      return Object.assign({}, state, {
        visibilityFilter: action.filter
      })
    default:
      return state
  }
}
```

Redux <https://riptutorial.com/zh-CN/reactjs/topic/10856/redux>

8: FluxReactJS

ReactJSFlux

FluxFacebookWeb React Flux

Flux React Model-View-Controller Flux - - - API Flux

FluxMVC React React

Examples

◦

Flux

1. - type

2. **Dispatcher** - ◦ ◦

3. - ◦ ◦

4. - React ◦ ◦

Flux

◦

- React ◦

- Controller-views

- ◦

- *Dispatcher*

- type

◦

FluxReactJS <https://riptutorial.com/zh-CN/reactjs/topic/8158/fluxreactjs>

9: React with Flow

FlowReact。

|

Examples

Flowprop

```
type Props = {
  posts: Array<Article>,
  dispatch: Function,
  children: ReactElement
}

const AppContainer =
  ({ posts, dispatch, children }: Props) => (
    <div className="main-app">
      <Header {...{ posts, dispatch }} />
      {children}
    </div>
  )
```

Flow

```
import React, { Component } from 'react';

type Props = {
  posts: Array<Article>,
  dispatch: Function,
  children: ReactElement
}

class Posts extends Component {
  props: Props;

  render () {
    // rest of the code goes here
  }
}
```

React with Flow <https://riptutorial.com/zh-CN/reactjs/topic/7918/react-with-flow>

10: AJAX

Examples

HTTP GET

REST API ◦ [componentDidMount](#) ◦

superagentAJAX

```
import React from 'react'
import request from 'superagent'

class App extends React.Component {
  constructor () {
    super()
    this.state = {}
  }
  componentDidMount () {
    request
      .get('/search')
      .query({ query: 'Manny' })
      .query({ range: '1..5' })
      .query({ order: 'desc' })
      .set('API-Key', 'foobar')
      .set('Accept', 'application/json')
      .end((err, resp) => {
        if (!err) {
          this.setState({someData: resp.text})
        }
      })
  },
  render() {
    return (
      <div>{this.state.someData || 'waiting for response...'}</div>
    )
  }
}

React.render(<App />, document.getElementById('root'))
```

`request.end()` ◦ `.set()` ◦

`.query()` **GET** ◦ `/search?query=Manny&range=1..5&order=desc` ◦

POST

```
request.post('/user')
  .set('Content-Type', 'application/json')
  .send('{"name":"tj","pet":"tobi"}')
  .end(callback)
```

[Superagent](#) ◦

ReactAjax - VanillaJS

IE9 +

```
import React from 'react'

class App extends React.Component {
  constructor () {
    super()
    this.state = {someData: null}
  }
  componentDidMount () {
    var request = new XMLHttpRequest();
    request.open('GET', '/my/url', true);

    request.onload = () => {
      if (request.status >= 200 && request.status < 400) {
        // Success!
        this.setState({someData: request.responseText})
      } else {
        // We reached our target server, but it returned an error
        // Possibly handle the error by changing your state.
      }
    };

    request.onerror = () => {
      // There was a connection error of some sort.
      // Possibly handle the error by changing your state.
    };

    request.send();
  },
  render() {
    return (
      <div>{this.state.someData || 'waiting for response...'}</div>
    )
  }
}

React.render(<App />, document.getElementById('root'))
```

HTTP GET

◦

[fetchAJAX](#)◦ [fetch polyfill](#)◦ [axios](#) [SuperAgent Javascript](#)◦

[render](#)◦ [map](#)◦ [key](#) [propReact](#)◦

```
import React from 'react';

class Users extends React.Component {
  constructor() {
    super();
    this.state = { users: [] };
  }
}
```

```
componentDidMount() {
  fetch('/api/users')
    .then(response => response.json())
    .then(json => this.setState({ users: json.data }));
}

render() {
  return (
    <div>
      <h1>Users</h1>
      {
        this.state.users.length == 0
        ? 'Loading users...'
        : this.state.users.map(user => (
          <figure key={user.id}>
            <img src={user.avatar} />
            <figcaption>
              {user.name}
            </figcaption>
          </figure>
        ))
      }
    </div>
  );
}
}

ReactDOM.render(<Users />, document.getElementById('root'));
```

[JSBin](#)。

[AJAX](https://riptutorial.com/zh-CN/reactjs/topic/6432/ajax) <https://riptutorial.com/zh-CN/reactjs/topic/6432/ajax>

11:

Examples

React JavaScript

constructor render

```
class ExampleComponent extends React.Component {
  constructor(props) {
    super(props);

    // Set-up our initial state
    this.state = {
      greeting: 'Hiya Buddy!'
    };
  }

  render() {
    // We can access the greeting property through this.state
    return (
      <div>{this.state.greeting}</div>
    );
  }
}
```

setState

ReactUI setState() . .

1. updater .
2. callback (optional) setState() . React setState() setState() .

setState updater prevState props

setState() **Object** updater

```
//
// An example ES6 style component, updating the state on a simple button click.
// Also demonstrates where the state can be set directly and where setState should be used.
//
class Greeting extends React.Component {
  constructor(props) {
    super(props);
    this.click = this.click.bind(this);
    // Set initial state (ONLY ALLOWED IN CONSTRUCTOR)
    this.state = {
      greeting: 'Hello!'
    };
  }
}
```

```

click(e) {
  this.setState({
    greeting: 'Hello World!'
  });
}
render() {
  return(
    <div>
      <p>{this.state.greeting}</p>
      <button onClick={this.click}>Click me</button>
    </div>
  );
}
}

```

Function `setState()` updater

```

//
// This is most often used when you want to check or make use
// of previous state before updating any values.
//
this.setState(function(previousState, currentProps) {
  return {
    counter: previousState.counter + 1
  };
});

```

`setState()` **React**

```

this.setState({ counter: this.state.counter + 1 });
this.setState({ counter: this.state.counter + 1 });
this.setState({ counter: this.state.counter + 1 });

```

React `Object.assign()` 13

◦ ◦

```

// Outside of component class, potentially in another file/module

function incrementCounter(previousState, currentProps) {
  return {
    counter: previousState.counter + 1
  };
}

// Within component

this.setState(incrementCounter);

```

Object `setState()`

```
//
// 'Hi There' will be logged to the console after setState completes
//

this.setState({ name: 'John Doe' }, console.log('Hi there'));
```

propsstate ◦ ◦

```
export default class MyComponent extends React.Component {
  constructor() {
    super();

    this.state = {
      url: ''
    }

    this.onChange = this.onChange.bind(this);
  }

  onChange(e) {
    this.setState({
      url: this.props.url + '/days=?' + e.target.value
    });
  }

  componentWillMount() {
    this.setState({url: this.props.url});
  }

  render() {
    return (
      <div>
        <input defaultValue={2} onChange={this.onChange} />

        URL: {this.state.url}
      </div>
    )
  }
}
```

prop urlstate ◦ stateprops

```
export default class MyComponent extends React.Component {
  constructor() {
    super();

    this.state = {
      days: ''
    }

    this.onChange = this.onChange.bind(this);
  }

  onChange(e) {
    this.setState({
      days: e.target.value
    });
  }
}
```

```

render() {
  return (
    <div>
      <input defaultValue={2} onChange={this.onChange} />

      URL: {this.props.url + '/days?=' + this.state.days}
    </div>
  )
}
}

```

React - ◦ props◦

MyComponent“url”◦ MyComponentstate.url◦ ◦ - urlMyComponentprop◦

“”React◦ ◦ setState◦ render◦ dom◦

```

import React from 'react';
import {render} from 'react-dom';

class ManagedControlDemo extends React.Component {

  constructor(props) {
    super(props);
    this.state = {message: ""};
  }

  handleChange(e) {
    this.setState({message: e.target.value});
  }

  render() {
    return (
      <div>
        <legend>Type something here</legend>
        <input
          onChange={this.handleChange.bind(this)}
          value={this.state.message}
          autoFocus />
        <h1>{this.state.message}</h1>
      </div>
    );
  }
}

render(<ManagedControlDemo/>, document.querySelector('#app'));

```

◦

- handleChange
- setState
- render

DOM

1. - divh1
2. h1
- 3.
4. DOM

<https://riptutorial.com/zh-CN/reactjs/topic/1816/>

12:

Examples

React;

-
- [JS.coach](#)

<https://riptutorial.com/zh-CN/reactjs/topic/6595/>

13:

Examples

◦

```
class CustomForm extends React.Component {
  constructor() {
    super();
    this.state = {
      person: {
        firstName: '',
        lastName: ''
      }
    }
  }

  handleChange(event) {
    let person = this.state.person;
    person[event.target.name] = event.target.value;
    this.setState({person});
  }

  render() {
    return (
      <form>
        <input
          type="text"
          name="firstName"
          value={this.state.firstName}
          onChange={this.handleChange.bind(this)} />

        <input
          type="text"
          name="lastName"
          value={this.state.lastName}
          onChange={this.handleChange.bind(this)} />
      </form>
    )
  }
}
```

}

◦ `2person` ◦ `handleChange` ◦ `setState()` ◦

`setState()` **React** ◦

`person` ◦ ◦

```
handleChange(event) {
  let person = this.state.person;
  person[event.target.name] = event.target.value;
  this.setState({person});
}
```

```
person[event.target.name]person.firstName || person.lastNameperson.firstName || person.lastName
```

- ◦ **noonChange**◦

<https://riptutorial.com/zh-CN/reactjs/topic/8047/>

14:

- MountingUpdatingUnmounting◦

Examples

React

- `React.createClass` ES55
- `class Component extends React.Component` ES63

`getDefaultProps()` ES5

◦

Prop◦

`this.props.nameBob`

```
getDefaultProps() {
  return {
    initialCount: 0,
    name: 'Bob'
  };
}
```

`getInitialState()` ES5

◦

`getInitialState()` React◦ `Reactthis.state` ◦

`this.state.count` **intialized** `this.props.initialCount`

```
getInitialState() {
  return {
    count : this.props.initialCount
  };
}
```

`componentWillMount()` ES5ES6

◦

DOM。

```
componentWillMount() {  
  ...  
}
```

render() ES5ES6

。

render() state props。 DOM <p />。 null undefined。

。

```
render() {  
  return (  
    <div>  
      Hello, {this.props.name}!  
    </div>  
  );  
}
```

componentDidMount() ES5ES6

。

DOM refs。

-
-
-
- DOM

```
componentDidMount() {  
  ...  
}
```

ES6

ES6 getDefaultProps() getInitialState()。

defaultProps。 React。

```
class MyReactClass extends React.Component {  
  constructor(props) {  
    super(props);  
  }  
}
```

```

    this.state = {
      count: this.props.initialCount
    };
  }

  upCount() {
    this.setState((prevState) => ({
      count: prevState.count + 1
    }));
  }

  render() {
    return (
      <div>
        Hello, {this.props.name}!<br />
        You clicked the button {this.state.count} times.<br />
        <button onClick={this.upCount}>Click here!</button>
      </div>
    );
  }
}

MyReactClass.defaultProps = {
  name: 'Bob',
  initialCount: 0
};

```

getDefaultProps()

defaultProps **props**

```

MyReactClass.defaultProps = {
  name: 'Bob',
  initialCount: 0
};

```

getInitialState()

this.state

```

constructor(props) {
  super(props);

  this.state = {
    count: this.props.initialCount
  };
}

```

componentWillReceiveProps (nextProps)

◦

React ◦ *this.props* ◦ *nextProps* ◦

◦

```
componentWillReceiveProps (nextProps) {  
  if (nextProps.initialCount && nextProps.initialCount > this.state.count) {  
    this.setState({  
      count : nextProps.initialCount  
    });  
  }  
}
```

shouldComponentUpdate (nextProps, nextState)

on ◦

// **React** ◦ true ◦

◦

◦

false ◦

```
componentShouldUpdate (nextProps, nextState) {  
  return this.props.name !== nextProps.name ||  
    this.state.count !== nextState.count;  
}
```

componentWillUpdate (nextProps, nextState)

componentWillMount () ◦ **DOM** ◦

Athis.setState ◦

```
componentWillUpdate (nextProps, nextState) {}
```

render ()

◦

componentDidUpdate (prevProps, prevState)

componentDidMount () **DOM DOM** ◦

```
componentDidUpdate (prevProps, prevState) {}
```

componentWillUnmount ()

DOM ◦

- ◦
- ◦
- ◦
- **redux**◦

```
componentWillUnmount(){
  ...
}
```

componentWillUnMount

```
import React, { Component } from 'react';

export default class SideMenu extends Component {

  constructor(props) {
    super(props);
    this.state = {
      ...
    };
    this.openMenu = this.openMenu.bind(this);
    this.closeMenu = this.closeMenu.bind(this);
  }

  componentDidMount() {
    document.addEventListener("click", this.closeMenu);
  }

  componentWillUnmount() {
    document.removeEventListener("click", this.closeMenu);
  }

  openMenu() {
    ...
  }

  closeMenu() {
    ...
  }

  render() {
    return (
      <div>
        <a
          href      = "javascript:void(0)"
          className = "closebtn"
          onClick   = {this.closeMenu}
        >
          x
        </a>
        <div>
          Some other structure
        </div>
      </div>
    );
  }
}
```

ReactPresentationalContainer。

- 。。。。 Redux 。

```
// Container component
import React, { Component } from 'react';
import Api from 'path/to/api';

class CommentsListContainer extends Component {
  constructor() {
    super();
    // Set initial state
    this.state = { comments: [] }
  }

  componentDidMount() {
    // Make API call and update state with returned comments
    Api.getComments().then(comments => this.setState({ comments }));
  }

  render() {
    // Pass our state comments to the presentational component
    return (
      <CommentsList comments={this.state.comments} />;
    );
  }
}

// Presentational Component
const CommentsList = ({ comments }) => (
  <div>
    {comments.map(comment => (
      <div>{comment}</div>
    ))}
  </div>
);

CommentsList.propTypes = {
  comments: React.PropTypes.arrayOf(React.PropTypes.string)
}
```

。

componentWillMountcomponentWillReceiveProps

1. getDefaultProps
2. getInitialState
3. componentWillMount
- 4.
5. componentDidMount

1. shouldComponentUpdate
2. componentWillUpdate
- 3.
4. componentDidUpdate

1. componentWillReceiveProps

2. shouldComponentUpdate
3. componentWillUpdate
- 4.
5. componentDidUpdate

1. componentWillUnmount

<https://riptutorial.com/zh-CN/reactjs/topic/2750/>

15:

Examples

Routes.js Router Link

◦

```
import React from 'react';
import { Route, IndexRoute } from 'react-router';
import New from './containers/new-post';
import Show from './containers/show';

import Index from './containers/home';
import App from './components/app';

export default(
  <Route path="/" component={App}>
    <IndexRoute component={Index} />
    <Route path="posts/new" component={New} />
    <Route path="posts/:id" component={Show} />

  </Route>
);
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import { Router, browserHistory } from 'react-router';
// import the routes component we created in routes.js
import routes from './routes';

// entry point
ReactDOM.render(
  <Router history={browserHistory} routes={routes} />
  , document.getElementById('main'));
```

Link[◦](#) **LinkReact RouterReact Routerroutes.js**

```
import React from 'react';
import { Link } from 'react-router';

export default function PostButton(props) {
  return (
    <Link to={`posts/${props.postId}`}>
      <div className="post-button" >
        {props.title}
        <span>{props.tags}</span>
      </div>
    </Link>
  );
}
```

```
);  
}
```

```
import React from 'react';  
import { Route, IndexRoute } from 'react-router';  
  
import Index from './containers/home';  
import App from './components/app';  
  
//for single Component lazy load use this  
const ContactComponent = () => {  
  return {  
    getComponent: (location, callback)=> {  
      require.ensure([], require => {  
        callback(null, require('./components/Contact')["default"]);  
      }, 'Contact');  
    }  
  }  
};  
  
//for multiple componnets  
const groupedComponents = (pageName) => {  
  return {  
    getComponent: (location, callback)=> {  
      require.ensure([], require => {  
        switch(pageName){  
          case 'about' :  
            callback(null, require( "./components/about" )["default"]);  
            break ;  
          case 'tos' :  
            callback(null, require( "./components/tos" )["default"]);  
            break ;  
        }  
      }, "groupedComponents");  
    }  
  }  
};  
export default(  
  <Route path="/" component={App}>  
    <IndexRoute component={Index} />  
    <Route path="/contact" {...ContactComponent()} />  
    <Route path="/about" {...groupedComponents('about')} />  
    <Route path="/tos" {...groupedComponents('tos')} />  
  </Route>  
)  
);
```

<https://riptutorial.com/zh-CN/reactjs/topic/6096/>

16:

Examples

ReduxReact Redux React Dom

- **UserInputForm** props inputChange
- **UserDashboard** UserInputForm inputChange UserInputForm UserInputForm
 - **UserDashboardConnected** ReactRedux connect UserDashboard
- UserDashboardConnected

```
const UserInputForm = (props) => {

  let handleSubmit = (e) => {
    e.preventDefault();
  }

  return(
    <form action="" onSubmit={handleSubmit}>
      <label htmlFor="name">Please enter your name</label>
      <br />
      <input type="text" id="name" defaultValue={props.data.name || ''} onChange={
props.inputChange } />
    </form>
  )
}

const UserDashboard = (props) => {

  let inputChangeHandler = (event) => {
    props.updateName(event.target.value);
  }

  return(
    <div>
      <h1>Hi { props.user.name || 'User' }</h1>
      <UserInputForm data={props.user} inputChange={inputChangeHandler} />
    </div>
  )
}

const mapStateToProps = (state) => {
  return {
    user: state
  };
}

const mapDispatchToProps = (dispatch) => {
  return {
    updateName: (data) => dispatch( Action.updateName(data) ),
  }
}
```

```

    };
};

const { connect, Provider } = ReactRedux;
const UserDashboardConnected = connect(
  mapStateToProps,
  mapDispatchToProps
)(UserDashboard);

const App = (props) => {
  return(
    <div>
      <h1>Communication between Stateless Functional Components</h1>
      <UserDashboardConnected />
    </div>
  )
}

const user = (state={name: 'John'}, action) => {
  switch (action.type) {
    case 'UPDATE_NAME':
      return Object.assign( {}, state, {name: action.payload} );

    default:
      return state;
  }
};

const { createStore } = Redux;
const store = createStore(user);
const Action = {
  updateName: (data) => {
    return { type : 'UPDATE_NAME', payload: data }
  },
}

ReactDOM.render(
  <Provider store={ store }>
    <App />
  </Provider>,
  document.getElementById('application')
);

```

[JS Bin URL](#)

<https://riptutorial.com/zh-CN/reactjs/topic/6137/>

17: React

React_{key} ◦ ◦

ReactReact ◦ React ◦ React

[https //facebook.github.io/react/docs/lists-and-keys.html](https://facebook.github.io/react/docs/lists-and-keys.html)

[https //facebook.github.io/react/docs/reconciliation.html#recurring-on-children](https://facebook.github.io/react/docs/reconciliation.html#recurring-on-children)

Examples

React

```
function SomeComponent (props) {  
  
  const ITEMS = ['cat', 'dog', 'rat']  
  function getItemsList () {  
    return ITEMS.map (item => <li key={item}>{item}</li>);  
  }  
  
  return (  
    <ul>  
      {getItemsList ()}  
    </ul>  
  );  
}
```

```
<ul>  
  <li key='cat'>cat</li>  
  <li key='dog'>dog</li>  
  <li key='rat'>rat</li>  
</ul>
```

React <https://riptutorial.com/zh-CN/reactjs/topic/9665/react>

18: webpackbabel

“”◦ Yeoman.io◦ ◦

Karma with MochaJasmine◦ ESLint◦ webpack-dev-serverCtrl + SF5◦ ◦

```
webpack.config.jscontext__dirname/srcwindows◦ ◦
```

webpack.config.js

```
webpack◦ require('./index.html') index.jsxsrc/◦
```

```
webpack◦ index.jsx◦
```

```
webpack◦ javascript◦
```

devServer

```
webpack-dev-server◦ contentBaserootdist/◦ port◦ openwebpack-dev-server◦
```

>

```
webpack◦ testwebpack◦ loaderloaderswebpack◦ javascriptqueryhtml◦ ◦
```

Examples

“Hello world”◦

1Node.js.

Node.js◦ Node.js SO

2

```
npm init
```

SO◦

3npm

```
npm install --save react react-dom
```

dev

```
npm install --save-dev babel-core babel-preset-react babel-preset-es2015 webpack babel-loader  
css-loader style-loader file-loader image-webpack-loader
```

webpackwebpack-dev-server。

```
npm install --global webpack webpack-dev-server
```

3.babelrc

babel。 .babelrc

```
{
  "presets": ["react", "es2015"]
}
```

4

```
|- node_modules
|- src/
  |- components/
  |- images/
  |- styles/
  |- index.html
  |- index.jsx
|- .babelrc
|- package.json
```

node_modules .babelrcpackage.json。

5Hello World

SRC // HelloWorldComponent.jsx

```
import React, { Component } from 'react';

class HelloWorldComponent extends Component {
  constructor(props) {
    super(props);
    this.state = {name: 'Student'};
    this.handleChange = this.handleChange.bind(this);
  }

  handleChange(e) {
    this.setState({name: e.target.value});
  }

  render() {
    return (
      <div>
        <div className="image-container">
          
        </div>
        <div className="form">
          <input type="text" onChange={this.handleChange} />
        </div>
        My name is {this.state.name} and I'm a clever cloggs because I built a React build
pipeline
      </div>
    );
  }
}
```



```
        </div>
      </div>
    );
  }
}

export default HelloWorldComponent;
```

SRC // myImage.gif

◦ HelloWorldComponent.jsx ◦ webpack.config.js ◦ test ◦

SRC // Styles.css

```
.form {
  margin: 25px;
  padding: 25px;
  border: 1px solid #ddd;
  background-color: #eaeaea;
  border-radius: 10px;
}

.form div {
  padding-top: 25px;
}

.image-container {
  display: flex;
  justify-content: center;
}
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Learning to build a react pipeline</title>
</head>
<body>
  <div id="content"></div>
  <script src="app.js"></script>
</body>
</html>
```

index.jsx

```
import React from 'react';
import { render } from 'react-dom';
import HelloWorldComponent from './components/HelloWorldComponent.jsx';

require('./images/myImage.gif');
require('./styles/styles.css');
require('./index.html');

render(<HelloWorldComponent />, document.getElementById('content'));
```

6webpack

webpack.config.js

webpack.config.js

```
var path = require('path');

var config = {
  context: path.resolve(__dirname + '/src'),
  entry: './index.jsx',
  output: {
    filename: 'app.js',
    path: path.resolve(__dirname + '/dist'),
  },
  devServer: {
    contentBase: path.join(__dirname + '/dist'),
    port: 3000,
    open: true,
  },
  module: {
    loaders: [
      {
        test: /\.js$/,
        exclude: /node_modules/,
        loader: 'babel-loader'
      },
      {
        test: /\.css$/,
        loader: "style!css"
      },
      {
        test: /\.gif$/,
        loaders: [
          'file?name=[path][name].[ext]',
          'image-webpack',
        ]
      },
      { test: /\.html$/,
        loader: "file?name=[path][name].[ext]"
      }
    ],
  },
};

module.exports = config;
```

7npm

package.jsonJSONscripts。

```
"scripts": {
  "start": "webpack-dev-server",
  "build": "webpack",
  "test": "echo \"Error: no test specified\" && exit 1"
},
```

test。

8

```
npm run build
```

dist/◦

```
npm start
```

webpack devWeb◦

[webpackbabel](https://riptutorial.com/zh-CN/reactjs/topic/6294/webpack-babel) <https://riptutorial.com/zh-CN/reactjs/topic/6294/webpack-babel>

19:

Examples

src/ out/ ◦

```
example/  
|-- src/  
|   |-- index.js  
|   `-- ...  
|-- out/  
`-- package.json
```

npm babel React es5 ◦

```
$npm install --save-dev babel-core babel-loader babel-preset-es2015 babel-preset-react
```

npm babel webpack ◦ es6 babel JSX es6 ◦ [Babel](#)

```
$npm i -D webpack
```

webpack ◦ **-D** --save-dev

webpack webpack-dev-server

```
$npm i -D react react-dom
```

webpack

webpack.config.js webpack

webpack.config.js

```
var path = require('path');  
  
module.exports = {  
  entry: './src/index.js',  
  output: {  
    path: path.resolve(__dirname, 'out'),  
    filename: 'bundle.js'  
  },  
  module: {  
    loaders: [  
      {  
        test: /\.js$/,  
        exclude: /(node_modules)/,  
        loader: 'babel-loader',
```

```
    query: {
      presets: ['es2015', 'react']
    }
  }
]
}
};
```

`webpackindex.jssrc /out.bundle.js`

`modulewebpack babel-loader excludewebpacknode_modules querywebpackbabel`

`src/index.js`

SRC / index.js

```
'use strict'

import React from 'react'
import { render } from 'react-dom'

const App = () => {
  return <h1>Hello world!</h1>
}

render(
  <App />,
  document.getElementById('app')
)
```

`<h1>Hello world!</h1>'app'html`

`$. /node_modules/.bin/webpack .webpack-gwebpack$webpack`

`out/bundle.js`

webpack-dev-server

`webpackbabel$npm i -g webpack-dev-serverhttp`

webpack.config.js

```
var path = require('path');

module.exports = {
  entry: './src/index.js',
  output: {
    path: path.resolve(__dirname, 'out'),
    publicPath: '/public/',
    filename: 'bundle.js'
  }
}
```

```

},
module: {
  loaders: [
    {
      test: /\.js$/,
      exclude: /(node_modules)/,
      loader: 'babel',
      query: {
        presets: ['es2015', 'react']
      }
    }
  ]
},
devServer: {
  contentBase: path.resolve(__dirname, 'public'),
  hot: true
}
};

```

- output.publicPath [Webpack](#)
- devServer
 - contentBaseindex.html
 - hot[webpack-dev-server](#)

index.html◦

index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>React Sandbox</title>
  </head>
  <body>

    <div id="app" />

    <script src="public/bundle.js"></script>
  </body>
</html>

```

\$webpack-dev-server8080http<h1>Hello world!</h1> ◦

<https://riptutorial.com/zh-CN/reactjs/topic/6441/>

20: ReactJSjQuery

Examples

ReactJSjQuery

jquery. findDOMNodedom. React.

```
import React from 'react';
import { findDOMNode } from 'react-dom';
import $ from 'jquery';
```

'handleToggle'. 'toggle'ondiv.

```
handleToggle = () => {
  const el = findDOMNode(this.refs.toggle);
  $(el).slideToggle();
};
```

'toggle'

```
<ul className="profile-info additional-profile-info-list" ref="toggle">
  <li>
    <span className="info-email">Office Email</span> me@shuvohabib.com
  </li>
</ul>
```

onClick'handleToggle'div.

```
<div className="ellipsis-click" onClick={this.handleToggle}>
  <i className="fa-ellipsis-h"/>
</div>
```

o

```
import React from 'react';
import { findDOMNode } from 'react-dom';
import $ from 'jquery';

export default class FullDesc extends React.Component {
  constructor() {
    super();
  }

  handleToggle = () => {
    const el = findDOMNode(this.refs.toggle);
    $(el).slideToggle();
  };

  render() {
```

```
return (
  <div className="long-desc">
    <ul className="profile-info">
      <li>
        <span className="info-title">User Name : </span> Shuvo Habib
      </li>
    </ul>

    <ul className="profile-info additional-profile-info-list" ref="toggle">
      <li>
        <span className="info-email">Office Email</span> me@shuvohabib.com
      </li>
    </ul>

    <div className="ellipsis-click" onClick={this.handleToggle}>
      <i className="fa-ellipsis-h"/>
    </div>
  </div>
);
}
```

ReactjQuery

ReactJSjQuery <https://riptutorial.com/zh-CN/reactjs/topic/6009/reactjsjquery>

21: ReactJSTypescript

Examples

ReactJSTypescript

facebookTypescriptReactJS。 'jsx' 'tsx'

```
//helloMessage.tsx:
var HelloMessage = React.createClass({
  render: function() {
    return <div>Hello {this.props.name}</div>;
  }
});
ReactDOM.render(<HelloMessage name="John" />, mountNode);
```

Typescript

1React.createClassES6

```
//helloMessage.tsx:
class HelloMessage extends React.Component {
  render() {
    return <div>Hello {this.props.name}</div>;
  }
}
ReactDOM.render(<HelloMessage name="John" />, mountNode);
```

2PropsState

```
interface IHelloMessageProps {
  name:string;
}

interface IHelloMessageState {
  //empty in our case
}

class HelloMessage extends React.Component<IHelloMessageProps, IHelloMessageState> {
  constructor(){
    super();
  }
  render() {
    return <div>Hello {this.props.name}</div>;
  }
}
ReactDOM.render(<HelloMessage name="Sebastian" />, mountNode);
```

Typescript。 ◦

ReactJavaScript

```
import React from 'react'

const HelloWorld = (props) => (
  <h1>Hello, {props.name}!</h1>
);
```

React.SFC **Typescript** React.SFC

```
import * as React from 'react';

class GreeterProps {
  name: string
}

const Greeter : React.SFC<GreeterProps> = props =>
  <h1>Hello, {props.name}!</h1>;
```

React.SFC React.StatelessComponent

reacttypescript npm npm init

npm

npm React

```
npm install --save react react-dom
```

Facebook **Yarn** React Yarn

```
yarn add react react-dom
```

React npm React

Typescript 2.0+

typescript npm yarn

```
npm install --save-dev @types/react @types/react-dom
```

```
yarn add --dev @types/react @types/react-dom
```

Typescript

tsd

```
tsd install react react-dom --save
```

Typescript

JSX javascript html / xml typescript tsconfig.json tsconfig.json **JSX**

```
"compilerOptions": {
  "jsx": "react"
},
```

typescriptJSXjavascript。

JSXjavascript

```
"compilerOptions": {
  "jsx": "preserve"
},
```

```
import * as React from 'react';

const Greeter = () => <span>Hello, World!</span>
```

this.propsthis.props。

```
import * as React from 'react';

const Greeter: React.SFC<{}> = props => () => <span>Hello, World!</span>
```

props.childrenprops.children。

◦ **exemplary**PageContainer NavTopNavBottom

```
import * as React from 'react';

const Page: React.SFC<{}> = props => () =>
  <Container>
    <NavTop />
    {props.children}
    <NavBottom />
  </Container>

const LoginPage: React.SFC<{}> = props => () =>
  <Page>
    Login Pass: <input type="password" />
  </Page>
```

Page。

ReactJSTypescript <https://riptutorial.com/zh-CN/reactjs/topic/1419/reactjstypescript>

22:

Examples

- HTML DOM

HTML DOM

- ◦ JavaScriptHTML
- HTML DOMHTML

DOM

DOM

ReactHTML DOMVirtual DOMHTML DOM

Virtual DOM

ReactVirtual DOM ◦ React

1. *Virtual DOM*
2. Virtual DOMHTML DOMVirtual DOM
3. 2.DOMHTML DOMDOM
 - - ReactDiff
4. *HTML DOM*
 - *Virtual DOMHTML DOM*
5. *HTML DOMHTML DOM HTML DOM*

Virtual DOMVirtual DOMJavaScript

Reactdiff

On ^ 3n ◦ React - On

1. tw ◦
2. ◦

React3

1. ◦

- `<div>...</div>...`

2.

- `<div key="1">...</div><div key="2">...</div>`

[] React ◦ DOM ◦

DOM ◦ React ◦

React - DOM ◦

1. ◦

2. shouldComponentUpdate

```
shouldComponentUpdate: function(nextProps, nextState) {  
  return nextProps.id !== this.props.id;  
}
```

ReactJS

◦ React ◦ ReactJS ◦ react-addons-perf

```
import Perf from 'react-addons-perf' // ES6  
var Perf = require('react-addons-perf') // ES5 with npm  
var Perf = React.addons.Perf; // ES5 with react-with-addons.js
```

Perf

- Perf.printInclusive
- Perf.printExclusive
- Perf.printWasted
- Perf.printOperations
- Perf.printDOM

Perf.printWasted() ◦

(index)	Owner > component	Waste
0	"Todos > TodoItem"	102.7

Total time: 132.71 ms

[ReactBenchling Engg ◦](#)

<https://riptutorial.com/zh-CN/reactjs/topic/6875/>

23:

React props ◦ propTypes defaultProps ◦

<https://facebook.github.io/react/docs/reusable-components.html#stateless-functions> ◦

Examples

UI ◦ React; ◦

React v14 React.Component ES6 React.createClass ES5 React React.createClass ◦

React v14 ◦ JavaScript ◦

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

React props React ◦ JavaScript ◦

UI; "Flux / Redux" ◦

- 1.
2. this
- 3.
- 4.
- 5.

UI React React ◦

◦

index.js

```
import React from 'react';  
import ReactDOM from 'react-dom';  
  
import HomePage from './homepage'  
  
ReactDOM.render(  
  <HomePage />,  
  document.getElementById('app')  
);
```

HomePage UI ◦ React UI ◦ stateList UI ◦

homepage.js

```
import React from 'react'  
import {Component} from 'react';
```

```

import List from './list';

export default class Temp extends Component{

  constructor(props) {
    super();
    this.state={users:[], showSearchResult: false, searchResult: []};
  }

  registerClick(){
    let users = this.state.users.slice();
    if(users.indexOf(this.refs.mail_id.value) == -1){
      users.push(this.refs.mail_id.value);
      this.refs.mail_id.value = '';
      this.setState({users});
    }else{
      alert('user already registered');
    }
  }

  searchClick(){
    let users = this.state.users;
    let index = users.indexOf(this.refs.search.value);
    if(index >= 0){
      this.setState({searchResult: users[index], showSearchResult: true});
    }else{
      alert('no user found with this mail id');
    }
  }

  hideSearchResult(){
    this.setState({showSearchResult: false});
  }

  render() {
    return (
      <div>
        <input placeholder='email-id' ref='mail_id' />
        <input type='submit' value='Click here to register'
onClick={this.registerClick.bind(this)} />
        <input style={{marginLeft: '100px'}} placeholder='search' ref='search' />
        <input type='submit' value='Click here to register'
onClick={this.searchClick.bind(this)} />
        {this.state.showSearchResult ?
          <div>
            Search Result:
            <List users={this.state.searchResult} />
            <p onClick={this.hideSearchResult.bind(this)}>Close this</p>
          </div>
          :
          <div>
            Registered users:
            <br />
            {this.state.users.length ?
              <List users={this.state.users} />
              :
              "no user is registered"
            }
          </div>
        }
      </div>
    )
  }
}

```



```
        </div>
      );
    }
  }
}
```

List ◦

list.js

```
import React from 'react';
var colors = ['#6A1B9A', '#76FF03', '#4527A0'];

var List = (props) => {
  return (
    <div>
      {
        props.users.map((user, i)=>{
          return (
            <div key={i} style={{color: colors[i%3]}}>
              {user}
            </div>
          );
        })
      }
    </div>
  );
}

export default List;
```

<https://facebook.github.io/react/docs/components-and-props.html>

<https://riptutorial.com/zh-CN/reactjs/topic/6588/>

24:

Examples

renderToStringrenderToStaticMarkup ◦

renderToString

ReactDOM.renderToStaticMarkup ◦

```
import { renderToString } from "react-dom/server";
renderToString(<App />);
```

renderToStaticMarkup

ReactDOM.renderToStaticMarkup ◦

```
import { renderToStaticMarkup } from "react-dom/server";
renderToStaticMarkup(<App />);
```

<https://riptutorial.com/zh-CN/reactjs/topic/7478/>

25:

React “ ”。

Examples

```
import React from 'react';

class Pane extends React.Component {
  constructor(props) {
    super(props);
  }

  render() {
    return React.createElement(
      'section', this.props
    );
  }
}
```

```
import React from 'react';

class Panel extends React.Component {
  constructor(props) {
    super(props);
  }

  render(...elements) {
    var props = Object.assign({
      className: this.props.active ? 'active' : '',
      tabIndex: -1
    }, this.props);

    var css = this.css();
    if (css !== '') {
      elements.unshift(React.createElement(
        'style', null,
        css
      ));
    }

    return React.createElement(
      'div', props,
      ...elements
    );
  }

  static title() {
    return '';
  }

  static css() {
    return '';
  }
}
```

•

- `panel` `title` `title` `title` ;
- `css` `PANEL` `.css` °

```
import React from 'react';

class Tab extends React.Component {
  constructor(props) {
    super(props);
  }

  render() {
    var props = Object.assign({
      className: this.props.active ? 'active' : ''
    }, this.props);
    return React.createElement(
      'li', props,
      React.createElement(
        'span', props,
        props.panelClass.title()
      )
    );
  }
}
```

TabpanelClass °

panelGroup

```
import React from 'react';
import Tab from './Tab.js';

class PanelGroup extends React.Component {
  constructor(props) {
    super(props);
    this.setState({
      panels: props.panels
    });
  }

  render() {
    this.tabSet = [];
    this.panelSet = [];
    for (let panelData of this.state.panels) {
      var tabIsActive = this.state.activeTab == panelData.name;
      this.tabSet.push(React.createElement(
        Tab, {
          name: panelData.name,
          active: tabIsActive,
          panelClass: panelData.class,
          onMouseDown: () => this.openTab(panelData.name)
        }
      ));
      this.panelSet.push(React.createElement(
        panelData.class, {
          id: panelData.name,
          active: tabIsActive,
```

```

        ref: tabIsActive ? 'activePanel' : null
      }
    ));
  }
  return React.createElement(
    'div', { className: 'PanelGroup' },
    React.createElement(
      'nav', null,
      React.createElement(
        'ul', null,
        ...this.tabSet
      )
    ),
    ...this.panelSet
  );
}

openTab(name) {
  this.setState({ activeTab: name });
  this.findDOMNode(this.refs.activePanel).focus();
}
}

```

PanelGrouppanels°

- **name** - ;
- **class** - panel°

activeTab°

tabDOMactive°

`PanelGroup`s

```

import React from 'react';
import Pane from './components/Pane.js';
import Panel from './components/Panel.js';
import PanelGroup from './components/PanelGroup.js';

class MainView extends React.Component {
  constructor(props) {
    super(props);
  }

  render() {
    return React.createElement(
      'main', null,
      React.createElement(
        Pane, { id: 'common' },
        React.createElement(
          PanelGroup, {
            panels: [
              {
                name: 'console',
                panelClass: ConsolePanel
              },
              {

```

```

        name: 'figures',
        panelClass: FiguresPanel
    }
    ],
    activeTab: 'console'
}
)
),
React.createElement(
  Pane, { id: 'side' },
  React.createElement(
    PanelGroup, {
      panels: [
        {
          name: 'properties',
          panelClass: PropertiesPanel
        }
      ],
      activeTab: 'properties'
    }
  )
)
);
}
}

class ConsolePanel extends Panel {
  constructor(props) {
    super(props);
  }

  static title() {
    return 'Console';
  }
}

class FiguresPanel extends Panel {
  constructor(props) {
    super(props);
  }

  static title() {
    return 'Figures';
  }
}

class PropertiesPanel extends Panel {
  constructor(props) {
    super(props);
  }

  static title() {
    return 'Properties';
  }
}

```

<https://riptutorial.com/zh-CN/reactjs/topic/8112/>

26:

React.createClass React.createClass v16 ◦ ◦ ◦

Examples

HTML

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>React Tutorial</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/15.2.1/react.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/15.2.1/react-dom.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-
core/5.8.34/browser.min.js"></script>
  </head>
  <body>
    <div id="content"></div>
    <script type="text/babel" src="scripts/example.js"></script>
  </body>
</html>
```

/example.js

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';

class FirstComponent extends Component {
  render() {
    return (
      <div className="firstComponent">
        Hello, world! I am a FirstComponent.
      </div>
    );
  }
}

ReactDOM.render(
  <FirstComponent />, // Note that this is the same as the variable you stored above
  document.getElementById('content')
);
```

div#content

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>React Tutorial</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/15.2.1/react.js"></script>
```

```

    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/15.2.1/react-dom.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-
core/5.8.34/browser.min.js"></script>
</head>
<body>
  <div id="content">
    <div className="firstComponent">
      Hello, world! I am a FirstComponent.
    </div>
  </div>
  <script type="text/babel" src="scripts/example.js"></script>
</body>
</html>

```

ReactJS。

```

var React = require('react');
var createReactClass = require('create-react-class');

var CommentList = reactCreateClass({
  render: function() {
    return (
      <div className="commentList">
        Hello, world! I am a CommentList.
      </div>
    );
  }
});

var CommentForm = reactCreateClass({
  render: function() {
    return (
      <div className="commentForm">
        Hello, world! I am a CommentForm.
      </div>
    );
  }
});

```

```

var React = require('react');
var createReactClass = require('create-react-class');

var CommentBox = reactCreateClass({
  render: function() {
    return (
      <div className="commentBox">
        <h1>Comments</h1>
        <CommentList /> // Which was defined above and can be reused
        <CommentForm /> // Same here
      </div>
    );
  }
});

```

。

1.

```
var CommentList = reactCreateClass({
  render: function() {
    return (
      <div className="commentList">
        <ListTitle/>
        Hello, world! I am a CommentList.
      </div>
    );
  }
});
```

ABBC。

- UI
-
-
-
- BC
- BC

2.

```
var CommentBox = reactCreateClass({
  render: function() {
    return (
      <div className="commentBox">
        <h1>Comments</h1>
        <CommentList>
          <ListTitle/> // child
        </CommentList>
        <CommentForm />
      </div>
    );
  }
});
```

ABABC。 。

-
-
-
-
-
- BC
- AC

`this.props.children` `CB` `BB3`

3.

```
var CommentBox = react.createClass({
  render: function() {
    return (
      <div className="commentBox">
        <h1>Comments</h1>
        <CommentList title={ListTitle}/> //prop
        <CommentForm />
      </div>
    );
  }
});
```

`ABBA` `.`

-
-
-

-
-

- B
- B

`3` `123`

```
import React, { Component } from 'react';
import { render } from 'react-dom';

class FirstComponent extends Component {
  render() {
    return (
      <div>
        Hello, {this.props.name}! I am a FirstComponent.
      </div>
    );
  }
}

render(
  <FirstComponent name={ 'User' } />,
  document.getElementById('content')
);
```

`React`

`ES6`

HTMLJSX。。

- 1.
2. JSX

```
// When using JSX inside a module you must import React
import React from 'react';
import PropTypes from 'prop-types';

const FirstComponent = props => (
  <div>
    Hello, {props.name}! I am a FirstComponent.
  </div>
);

//arrow components also may have props validation
FirstComponent.propTypes = {
  name: PropTypes.string.isRequired,
}

// To use FirstComponent in another file it must be exposed through an export call:
export default FirstComponent;
```

“”“”
setState。 constructor

```
import React, { Component } from 'react';

class SecondComponent extends Component {
  constructor(props) {
    super(props);

    this.state = {
      toggle: true
    };

    // This is to bind context when passing onClick as a callback
    this.onClick = this.onClick.bind(this);
  }

  onClick() {
    this.setState((prevState, props) => ({
      toggle: !prevState.toggle
    }));
  }

  render() {
    return (
      <div onClick={this.onClick}>
        Hello, {this.props.name}! I am a SecondComponent.
        <br />
        Toggle is: {this.state.toggle}
      </div>
    );
  }
}
```

HOC◦

```
import React, { Component } from 'react';

const PrintHello = ComposedComponent => class extends Component {
  onClick() {
    console.log('hello');
  }

  /* The higher order component takes another component as a parameter
  and then renders it with additional props */
  render() {
    return <ComposedComponent {...this.props } onClick={this.onClick} />
  }
}

const FirstComponent = props => (
  <div onClick={ props.onClick }>
    Hello, {props.name}! I am a FirstComponent.
  </div>
);

const ExtendedComponent = PrintHello(FirstComponent);
```

◦

setState

setState◦ getsetState

```
class MyClass extends React.Component {
  constructor() {
    super();

    this.state = {
      user: {}
    };
  }

  componentDidMount() {
    this.fetchUser();
  }

  fetchUser() {
    $.get('/api/users/self')
      .then((user) => {
        this.setState({user: user});
      });
  }

  render() {
    return <h1>{this.state.user}</h1>;
  }
}
```

- Componentthis.setState◦ setState◦

XHR

```
class MyClass extends React.Component {
  constructor() {
    super();

    this.state = {
      user: {},
      xhr: null
    };
  }

  componentWillUnmount() {
    let xhr = this.state.xhr;

    // Cancel the xhr request, so the callback is never called
    if (xhr && xhr.readyState !== 4) {
      xhr.abort();
    }
  }

  componentDidMount() {
    this.fetchUser();
  }

  fetchUser() {
    let xhr = $.get('/api/users/self')
      .then((user) => {
        this.setState({user: user});
      });

    this.setState({xhr: xhr});
  }
}
```

- componentWillUnmount - XHR◦
- 'stateSetter'cancelthis.setState

```
function stateSetter(context) {
  var cancelled = false;
  return {
    cancel: function () {
      cancelled = true;
    },
    setState(newState) {
      if (!cancelled) {
        context.setState(newState);
      }
    }
  }
}

class Component extends React.Component {
  constructor(props) {
    super(props);
  }
}
```

```

    this.setter = stateSetter(this);
    this.state = {
      user: 'loading'
    };
  }
  componentWillUnmount() {
    this.setter.cancel();
  }
  componentDidMount() {
    this.fetchUser();
  }
  fetchUser() {
    $.get('/api/users/self')
      .then((user) => {
        this.setter.setState({user: user});
      });
  }
  render() {
    return <h1>{this.state.user}</h1>
  }
}

```

cancelledsetState

React -

JSXprops

```
<MyComponent userID={123} />
```

MyComponentpropsuserID

```

// The render function inside MyComponent
render() {
  return (
    <span>The user's ID is {this.props.userID}</span>
  )
}

```

props

```

// defined at the bottom of MyComponent
MyComponent.propTypes = {
  someObject: React.PropTypes.object,
  userID: React.PropTypes.number.isRequired,
  title: React.PropTypes.string
};

MyComponent.defaultProps = {
  someObject: {},
  title: 'My Default Title'
}

```

prop someObjectprop userID MyComponentuserID Reactprop React

defaultProps

```
const { title = 'My Default Title' } = this.props;
console.log(title);
```

```
console.log(this.props.title);
```

object arrayfunctions°

```
if (this.props.someObject.someKey)
```

```
this.props.someObjectundefined someKey° defaultProps°
```

-

- h3° Reactif else°

```
// I have used ReactBootstrap elements. But the code works with regular html elements also
var Button = ReactBootstrap.Button;
var Form = ReactBootstrap.Form;
var FormGroup = ReactBootstrap.FormGroup;
var FormControl = ReactBootstrap.FormControl;

var Comment = reactCreateClass({
  getInitialState: function() {
    return {show: false, newTitle: ''};
  },

  handleTitleSubmit: function() {
    //code to handle input box submit - for example, issue an ajax request to change name in
    database
  },

  handleTitleChange: function(e) {
    //code to change the name in form input box. newTitle is initialized as empty string. We
    need to update it with the string currently entered by user in the form
    this.setState({newTitle: e.target.value});
  },

  changeComponent: function() {
    // this toggles the show variable which is used for dynamic UI
    this.setState({show: !this.state.show});
  },

  render: function() {

    var clickableTitle;

    if(this.state.show) {
      clickableTitle = <Form inline onSubmit={this.handleTitleSubmit}>
        <FormGroup controlId="formInlineTitle">
          <FormControl type="text" onChange={this.handleTitleChange}>
        </FormGroup>
      </Form>;
    } else {
      clickableTitle = <div>
        <Button bsStyle="link" onClick={this.changeComponent}>
          <h3> Default Text </h3>
        </div>
    }
  }
});
```

```

        </Button>
      </div>;
    }

    return (
      <div className="comment">
        {clickableTitle}
      </div>
    );
  }
});

ReactDOM.render(
  <Comment />, document.getElementById('content')
);

```

clickableTitle。 **show** FormButton。 React。

render{clickableTitle}。 clickableTitle。 'this.state.show'。

```

const languages = [
  'JavaScript',
  'Python',
  'Java',
  'Elm',
  'TypeScript',
  'C#',
  'F#'
]

```

```

// one liner
const Language = ({language}) => <li>{language}</li>

Language.propTypes = {
  message: React.PropTypes.string.isRequired
}

```

```

/**
 * If there are more than one line.
 * Please notice that round brackets are optional here,
 * However it's better to use them for readability
 */
const LanguagesList = ({languages}) => {
  <ul>
    {languages.map(language => <Language language={language} />)}
  </ul>
}

LanguagesList.propTypes = {
  languages: React.PropTypes.array.isRequired
}

```

```

/**
 * This syntax is used if there are more work beside just JSX presentation

```



```
* For instance some data manipulations needs to be done.
* Please notice that round brackets after return are required,
* Otherwise return will return nothing (undefined)
*/
const LanguageSection = ({header, languages}) => {
  // do some work
  const formattedLanguages = languages.map(language => language.toUpperCase())
  return (
    <fieldset>
      <legend>{header}</legend>
      <LanguagesList languages={formattedLanguages} />
    </fieldset>
  )
}

LanguageSection.propTypes = {
  header: React.PropTypes.string.isRequired,
  languages: React.PropTypes.array.isRequired
}
```

```
ReactDOM.render (
  <LanguageSection
    header="Languages"
    languages={languages} />,
  document.getElementById('app')
)
```

o

<https://riptutorial.com/zh-CN/reactjs/topic/1185/>

27:

React3

- 1
- 2
- 3

Examples

React - ◦

- messageprop◦

```
import React from 'react';

class Parent extends React.Component {
  render() {
    const variable = 5;
    return (
      <div>
        <Child message="message for child" />
        <Child message={variable} />
      </div>
    );
  }
}

class Child extends React.Component {
  render() {
    return <h1>{this.props.message}</h1>
  }
}

export default Parent;
```

<Parent /><Child />message for child5 ◦

◦

prop ◦

ChildChildParent◦

```
import React from 'react';

class Parent extends React.Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };

    this.outputEvent = this.outputEvent.bind(this);
  }
}
```

```

    }
    outputEvent(event) {
      // the event context comes from the Child
      this.setState({ count: this.state.count++ });
    }

    render() {
      const variable = 5;
      return (
        <div>
          Count: { this.state.count }
          <Child clickHandler={this.outputEvent} />
        </div>
      );
    }
  }
}

class Child extends React.Component {
  render() {
    return (
      <button onClick={this.props.clickHandler}>
        Add One More
      </button>
    );
  }
}

export default Parent;

```

Parent outputEvent Child onClick

◦

2 **////** ◦

3 ◦ ◦

• 1 **//** ◦

- otherObject.addEventListener('click', () => { alert('click!'); });
- this.dispatchEvent('click');

• 2 **/** ◦

- globalBroadcaster.subscribe('click', () => { alert('click!'); });
- globalBroadcaster.publish('click');

• 3 **//** ◦ ◦ ◦

- otherObject.clicked.add(() => { alert('click'); });
- this.clicked.dispatch();

<https://riptutorial.com/zh-CN/reactjs/topic/6567/>

28:

Examples

value° React° value°

```
class Form extends React.Component {
  constructor(props) {
    super(props);

    this.onChange = this.onChange.bind(this);

    this.state = {
      name: ''
    };
  }

  onChange(e) {
    this.setState({
      name: e.target.value
    });
  }

  render() {
    return (
      <div>
        <label for='name-input'>Name: </label>
        <input
          id='name-input'
          onChange={this.onChange}
          value={this.state.name} />
      </div>
    )
  }
}
```

value onChange°

° °

value° °

```
class Form extends React.Component {
  constructor(props) {
    super(props);

    this.onChange = this.onChange.bind(this);

    this.state = {
      name: 'John'
    };
  }

  onChange(e) {
```

```
    this.setState({
      name: e.target.value
    });
  }

  render() {
    return (
      <div>
        <label for='name-input'>Name: </label>
        <input
          id='name-input'
          onChange={this.onChange}
          defaultValue={this.state.name} />
        </div>
      )
    }
  }
}
```

onChange° valuedefaultValue° ° ;°

<https://riptutorial.com/zh-CN/reactjs/topic/2884/>

29: React

Examples

src / index.jsx

```
import React from 'react';
import ReactDOM from 'react-dom';

class ToDo extends React.Component {
  render() {
    return (<div>I am working</div>);
  }
}

ReactDOM.render(<ToDo />, document.getElementById('App'));
```

```
# install react and react-dom
$ npm i react react-dom --save

# install webpack for bundling
$ npm i webpack -g

# install babel for module loading, bundling and transpiling
$ npm i babel-core babel-loader --save

# install babel presets for react and es6
$ npm i babel-preset-react babel-preset-es2015 --save
```

webpack

webpack.config.js

webpack.config.js

```
module.exports = {
  entry: __dirname + "/src/index.jsx",
  devtool: "source-map",
  output: {
    path: __dirname + "/build",
    filename: "bundle.js"
  },
  module: {
    loaders: [
      {test: /\.jsx?$/, exclude: /node_modules/, loader: "babel-loader"}
    ]
  }
}
```

babel

.babelrc

.babelrc

```
{
  "presets": ["es2015", "react"]
}
```

HTML

html

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <div id="App"></div>
    <script src="build/bundle.js" charset="utf-8"></script>
  </body>
</html>
```

webpack

```
$ webpack
```

build

HTML

React <https://riptutorial.com/zh-CN/reactjs/topic/7480/react>

30:

ReactDOM。

lilikey。id。

◦ React Docs

◦ ◦

<https://medium.com/@robinpokorny/index-as-a-key-is-an-anti-pattern-e0349aece318>

Examples

id

◦

textid。id

```
todos = [  
  {  
    id: 1,  
    text: 'value 1'  
  },  
  {  
    id: 2,  
    text: 'value 2'  
  },  
  {  
    id: 3,  
    text: 'value 3'  
  },  
  {  
    id: 4,  
    text: 'value 4'  
  },  
];
```

key`todo-${todo.id}`react

```
render() {  
  const { todos } = this.props;  
  return (  
    <ul>  
      { todos.map((todo) =>  
        <li key={ `todo-${todo.id}` }>  
          { todo.text }  
        </li>  
      ) }  
    </ul>  
  );  
}
```



```
}
```

ID

```
render() {  
  const { todos } = this.props;  
  return (  
    <ul>  
      { todos.map((todo, index) =>  
        <li key={ `todo-${index}` }>  
          { todo.text }  
        </li>  
      ) }  
    </ul>  
  );  
}
```

<https://riptutorial.com/zh-CN/reactjs/topic/9805/>

31:

“HOC”。

HOC javascript。

HOC。 Redux。

Examples

console.log

hocLogger.js

```
export default function hocLogger(Component) {
  return class extends React.Component {
    componentDidMount() {
      console.log('Hey, we are mounted!');
    }
    render() {
      return <Component {...this.props} />;
    }
  }
}
```

HOC

MyLoggedComponent.js

```
import React from "react";
import {hocLogger} from "./hocLogger";

export class MyLoggedComponent extends React.Component {
  render() {
    return (
      <div>
        This component get's logged to console on each mount.
      </div>
    );
  }
}

// Now wrap MyLoggedComponent with the hocLogger function
export default hocLogger(MyLoggedComponent);
```

。

HOCrender

AuthenticatedComponent.js

```

import React from "react";

export function requireAuthentication(Component) {
  return class AuthenticatedComponent extends React.Component {

    /**
     * Check if the user is authenticated, this.props.isAuthenticated
     * has to be set from your application logic (or use react-redux to retrieve it from
global state).
     */
    isAuthenticated() {
      return this.props.isAuthenticated;
    }

    /**
     * Render
     */
    render() {
      const loginErrorMessage = (
        <div>
          Please <a href="/login">login</a> in order to view this part of the
application.
        </div>
      );

      return (
        <div>
          { this.isAuthenticated === true ? <Component {...this.props} /> :
loginErrorMessage }
        </div>
      );
    }
  };
}

export default requireAuthentication;

```

MyPrivateComponent.js

```

import React from "react";
import {requireAuthentication} from "../AuthenticatedComponent";

export class MyPrivateComponent extends React.Component {
  /**
   * Render
   */
  render() {
    return (
      <div>
        My secret search, that is only viewable by authenticated users.
      </div>
    );
  }
}

// Now wrap MyPrivateComponent with the requireAuthentication function
export default requireAuthentication(MyPrivateComponent);

```

o

S. No		Contributors
1	React	Adam , Adrián Daraš , Alex , Alex Young , Anuj , Bart Riordan , Cassidy , Community , Daksh Gupta , Dave Kaye , diabolicfreak , DMan , Donald , Everettss , Gianluca Esposito , himanshuITian , hyde , Ilya Lyamkin , Inanc Gumus , ivarni , jengeb , jolyonruss , Jon Chan , JordanHendrix , juandemarco , Kaloyan Kosev , Konstantin Grushetsky , Maksim , Marty , MaxPRafferty , Md. Nahiduzzaman Rose , Md.Sifatul Islam , Ming Soon , MMachinegun , Nick Bartlett , orvi , paqash , Prakash , rossipedia , Shabin Hashim , Simplans , Sunny R Gupta , TheShadowbyte , Timo , Tushar Khanna , user2314737
2	JSX	Kaloyan Kosev , Ming Soon
3	React Boilerplate [React + Babel + Webpack]	Mihir , parlad neupane , Tien Do
4	ReactWebpack Typescript	Aron
5	React.createClass vs React.Component	Kaloyan Kosev , leonardoborges , Michael Peyper , pwolaq , Qianyue , sqzaman
6	React	Ahmad , Anuj , Danillo Corvalan , Everettss , Faktor 10 , Fellow Stranger , hansn , Ilya Lyamkin , Jack7 , Jagadish Upadhyay , JimmyLv , MaxPRafferty , QoP , Sergii Bishyr , vintproykt , WitVault , zbynour
7	Redux	Jim
8	FluxReactJS	vintproykt
9	React with Flow	JimmyLv , lifeiscontent , Rifat , Rory O'Kane
10	AJAX	adamboro , Fabian Schultz , Jason Bourne , lifeiscontent , McGrady , Sunny R Gupta
11		Alex Young , Alexander , Brad Colthurst , Everettss , Kousha , Kyle Richardson , QoP , skav , Timo
12		brillout
13		promisified
14		Alex Young , Alexg2195 , Anuj , Ashari , Everettss , F. Kauder ,

		irrigator , John Ruddell , QoP , Salman Saleem , Saravana , Siddharth , skav , Timo , ultrasamad , Vivian , WitVault
15		abhirathore2006 , Robeen
16		Random User
17	React	Sammy I.
18	webpackbabel	Bart Riordan , Tien Do , Zac Braddy
19		Rene R , Ruairi O'Brien
20	ReactJSjQuery	Kousha , Shuvo Habib
21	ReactJSTypescript	Everettss , John Ruddell , kevgathuku , Leone , Rajab Shakirov
22		Aditya Singh , Iustoykov , thibmaek
23		Adam , Mark Lapierre , Mayank Shukla , Valter Júnior
24		Adrián Daraš , MauroPorrasP
25		vintproykt
26		akashrajkn , Anuj , Bart Riordan , Bond , Brandon Roberts , Denis Ivanov , Diego V , DMan , Evan Hammer , Everettss , goldbullet , GordyD , hmnzr , Ilya Lyamkin , ivarni , Jagadish Upadhyay , jbmartinez , John Ruddell , jolyonruss , Jon Chan , jonathangoodman , JordanHendrix , justabuzz , k170 , Kousha , Kyle Richardson , m_callens , Maayan Glikser , Michael Peyper , Paul Graffam , philpee2 , QoP , Radu Brehar , Sai Vikas , sjmarshy , Timo , Vlad Bezden , WooCaSh , Zakaria Ridouh , zurfyx
27		David , Kaloyan Kosev
28		Everettss , Henrik Karlsson , ivarni , Timo
29	React	ghostffcode , Tien Do
30		Dennis Stücken , thibmaek
31		Dennis Stücken