

 무료 전자 책

배우기

React

Free unaffiliated eBook created from
Stack Overflow contributors.

#reactjs

JSX	14
JavaScript	14
.....	15
.....	15
.....	15
JSX	15
.....	15
JSX	16
JavaScript	16
.....	17
.....	17
5: React Boilerplate [+ + Webpack]	19
Examples	19
.....	19
-	21
6: React, Webpack Typescript	23
.....	23
Examples	23
webpack.config.js	23
.....	23
TS	23
tsconfig.json	23
include	24
compilerOptions.target	24
compilerOptions.jsx	24
compilerOptions.allowSyntheticDefaultImports	24
.....	24
7: React.Component React.createClass	26
.....	26
.....	26

Examples.....	26
.....	26
React.createClass ()	26
React.Component	26
PropType	26
React.createClass	27
React.Component.....	27
.....	28
React.createClass	28
React.Component	28
.....	29
React.createClass	29
React.Component	29
"this"Context.....	30
React.createClass	30
React.Component	30
1 : :.....	30
2 :	31
3 : ES6	31
ES6 / ajax "this"	32
8: ReactJS jQuery	33
Examples.....	33
ReactJS jQuery.....	33
9: React	35
.....	35
.....	35
Examples.....	35
.....	35
10: Redux	36
.....	36
.....	36

Examples.....	36
.....	36
11: Typescript ReactJS	37
Examples.....	37
Typescript ReactJS	37
Typescript Stateless React	37
.....	38
Stateless Propertyless	39
12:	40
.....	40
.....	40
Examples.....	40
.....	40
.....	40
13:	43
Examples.....	43
- HTML DOM DOM	43
diff	43
&	44
ReactJS	44
14:	46
.....	46
Examples.....	46
.....	46
.....	47
1.	48
.....	48
.....	48
.....	48
2.	48
.....	48

.....	49
.....	49
3.	49
.....	49
.....	49
.....	49
.....	49
.....	49
.....	50
.....	50
.....	51
setState	51
.....	53
-	54
.....	55
15:	57
Examples.....	57
.....	57
16:	59
.....	59
Examples.....	59
.....	59
.....	59
.....	60
17: ,	61
.....	61
Examples.....	61
"Hello world"	61
18:	66
.....	66
Examples.....	66

.....	66
19:	69
Examples.....	69
.....	69
20:	70
Examples.....	70
Routes.js ,	70
.....	71
21:	72
Examples.....	72
.....	72
22:	74
Examples.....	74
.....	74
.....	74
.....	74
.....	74
HTML	75
Transpile	75
23:	76
.....	76
Examples.....	76
.....	76
.....	76
PropType.....	77
.....	78
.....	79
Children	80
24:	81
Examples.....	81
.....	81
setState ().....	81

updater setState()	81
updater setState()	82
Object setState()	82
.....	83
,	84
25:	86
.....	86
.....	86
Examples	86
id	86
.....	87
26:	88
.....	88
Examples	88
.....	88
.....	88
.....	89
PanelGroup	89
.....	90
`PanelGroup`s	90
27:	92
Examples	92
.....	92
renderToString	92
renderToStaticMarkup	92
28:	93
Examples	93
.....	93
.....	93
.....	93
.....	93

.....	94
webpack-dev-server	94
.....	94
webpack.config.js	94
29:	96
Examples.....	96
.....	96
.....	96
30:	98
.....	98
Examples.....	98
.....	98
getDefaultProps() (ES5).....	98
getInitialState() (ES5).....	98
componentWillMount() (ES5 ES6).....	98
render() (ES5 ES6).....	99
componentDidMount() (ES5 ES6).....	99
ES6	99
getDefaultProps()	100
getInitialState()	100
.....	100
componentWillReceiveProps(nextProps).....	100
shouldComponentUpdate(nextProps, nextState).....	101
componentWillUpdate(nextProps, nextState).....	101
render().....	101
componentDidUpdate(prevProps, prevState).....	101
.....	101
componentWillUnmount().....	101
.....	103
.....	103

31:	105
.....	105
.....	105
Examples.....	105
.....	105
.....	105
.....	106

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	2013-05-29
0.4.0	2013-07-17
0.5.0	2013-10-16
0.8.0	2013-12-19
0.9.0	2014-02-20
0.10.0	2014-03-21
0.11.0	2014-07-17
0.12.0	2014-10-28
0.13.0	2015-03-10
0.14.0	2015-10-07
15.0.0	2016-04-07
15.1.0	2016-05-20
15.2.0	2016-07-01
15.2.1	2016-07-08
15.3.0	2016-07-29
15.3.1	2016-08-19
15.3.2	2016-09-19
15.4.0	2016-11-16
15.4.1	2016-11-23
15.4.2	2017-01-06
15.5.0	2017-04-07
15.6.0	2017-06-13

Examples

ReactJS HTML `react-<version>.js` JavaScript . 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>
```

React .

React [JSX](#) . JSX JavaScript XML Facebook . JSX JSX Javascript Babel `<script type="text/javascript">` `<script type="text/babel">` .

```
<!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

[npm](#) React .

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

JavaScript React .

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

[Yarn](#) . [Yan](#) React . [Yarn](#) .

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

React via npm React .

Hello World

React `React.Component` ES6 `.render` DOM `.render` JSX HTML React `.render`,
Component `.render`

```
import React from 'react'

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

export default HelloWorld
```

`props` `.render`, `.render` `onClick` `.render`, `props` `props` :

```
import React from 'react'

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

export default Hello
```

name `prop` `.render` `.render`

DOM `.ReactDOM.render` React DOM `.render`

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

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

`props` `.render` `state` `state` `.render`

Hello World `.render` `.render`

```
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

```

: .

[Codepen .](#)

JSX

React API React React DOM API React .

```

<!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

React JSX (XML Facebook) .

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

(HTML)

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

JSX <script type="text/babel"> . Babel (React) Javascript .

```
<!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 , . .

ReactJS DOM HTML DOM () . DOM DOM DOM .

React HTML . . . TabList .

React JSX . JSX JavaScript HTML . React DOM HTML .

DOM HTML DOM " ". DOM DOM . React DOM DOM , .

Hello World

. . .



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

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

statelessSum a b . statefulSum . . .

, . . .

. JavaScript . state state props Stateless Functional Components .

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 . .

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

- ES6 JSX
- Dev
- linting
- CSS
- JS, CSS
-

(npm) create-react-app .

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

.

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

.

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

create-react-app . Sass CSS eject .

```
npm run eject
```

. .

:

-
- [nwb](#)
-
- -
- [Buddō](#)
- [rwb](#)
- [quik](#)
-
-

.

```
npm run build
```

React `React` . `"Props" JSX` (`<SomeComponent someProp="some prop's value" />`) .
`SomeComponent` `this.props` . `this.props` `{someProp: "some prop's value"}` .

React `.., "props"` . `"", ()` . `""` . React .

- (**"Stateless"**)

```
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} />
  );
}
```

""" . . .

, ., . React .

[Reactable In Thinking and Reusable Components](#) .

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

2: AJAX

Examples

HTTP GET

(: REST API) . componentDidMount .

AJAX [superagent](#) .

```
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](#) .

3 React Ajax - 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

, .

[fetch](#) AJAX . [polyfill fetch](#) . (: [axios](#) , [SuperAgent](#)) .

[render](#) . [map](#) . [React](#) [key](#) ([prop](#)) .

```
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/ko/reactjs/topic/6432/ajax-->

3: Flux ReactJS

ReactJS . . .

Flux Facebook . React . Flux .

, (React) . Model-View-Controller . Flux . (Dispatcher) API . .

MVC . React . React . .

Examples

.

.

1. - type .

2. **Dispatcher** - . .

3. - . .

4. - . .

Flux , .

, .

- React () .

. .

- .

- *Dispatcher* .

- type .

() .

Flux ReactJS : <https://riptutorial.com/ko/reactjs/topic/8158/flux--reactjs->

4: JSX

JSX JavaScript XML . JSX React JSX React .

XML JSX , . . . 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);
```

JSX HTML .

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

JSX JavaScript . JSX JavaScript .

React UI .

Examples

JSX

JSX .

JavaScript

{}, JSX:

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

MyComponent 1 + 2 + 3 + 4 props.count 10.

if for JavaScript JSX.

, string literal prop. JSX.

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

HTML. JSX.

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

. .

true. JSX.

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

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

, JSX,

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

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

JSX

JSX prop: props.children.

props.children props.children . HTML . :

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

JSX MyComponent props.children <h1>Hello world!</h1> .

HTML HTML JSX .

JSX .

- .
- .
- .
- .

JSX

JSX . . .

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

JSX . JSX HTML JSX HTML.

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

React React JSX . div .

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 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>;  
};
```

. JSX .

false , null , undefined true . . JSX .

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

React . JSX showHeader true a .

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

0 " " React . , . props.messages 0 .

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

&& .

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

false , true , null undefined .

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

JSX : <https://riptutorial.com/ko/reactjs/topic/8027/jsx>

5: React Boilerplate [+ + Webpack]

Examples

[. Nodejs.org](#) . . .

npm

React .

[Babel](#) [Webpack](#) .

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
```

Optional 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"  
  ]  
}
```

```
]
}
```

git repo .gitignore .

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

webpack.config.js .

```
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')
    }]
  }
};
```

npm start sever.js .

```
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/');
});
```

React 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 node server.js .

-

. ReactJs + Webpack + Babel .

. . : .

GUI . npm init package.json , git , , .json .

Express npm install express @ 4 - save . . . package.js . package.json . 4 .

package.json "dependencies": { "express": "^4.13.4", }, "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'); api .

var app = express (); . app.use (); . app.use (express.static ('public'))); . app.listen (port,
function () {}) 3000 . express .

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 ('Express server is port : 3000'); * .

<http://localhost:3000> `hello World` .

`app.jsx` . JSX XML . JSX React JSX React . `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 \[+ + Webpack \]](https://riptutorial.com/ko/reactjs/topic/5969/react-boilerplate---plus--plus-webpack-) : <https://riptutorial.com/ko/reactjs/topic/5969/react-boilerplate---plus--plus-webpack->

6: React, Webpack Typescript

(:VS) .

React ReactDOM Intellisense . . .

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

Intellisense .

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-loader .
```

TS

```
.ts .tsx resolve . webpack .
```

tsconfig.json

tsconfig .

```
{
  "include": [
    "src/*"
  ]
}
```

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

.

include

```
. src src/* .
```

compilerOptions.target

ES5 .

compilerOptions.jsx

```
true TypeScript <div /> React.createElement("div") tsx .
```

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);
```

React TypeScript **React DefinitelyTyped** (`npm install --save @types/react`) .

:

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

AppProps AppState ().

React, Webpack Typescript : <https://riptutorial.com/ko/reactjs/topic/9590/react--webpack--typescript->

7: React.createClass React.createClass

- 1 : React.createClass ({})
- 2 : MyComponent React.Component {} .

React.createClass React.createClass v16 . . .

Examples

React.createClass ()

React const , render .

```
import React from 'react';

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

export default MyComponent;
```

React.Component

React.createClass ES6 .

```
import React from 'react';

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

export default MyComponent;
```

ES6 . React React.createClass **MyComponent** React.Component . React .

: ES6 ES5 Babel .

PropTypes

React.createClass

propTypes Object. getDefaultProps Object .

```
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 createClass Object **MyComponent** .

getDefaultProps "get" defaultProps Object . Object . React . JavaScript.

```
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;
```

propTypes defaultProps . 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

Object getInitialState .

```
import React from 'react';

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

export default MyComponent;
```

React.Component

getInitialState . API "React API" .

```
import React from 'react';

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

export default MyComponent;
```

mixins React.createClass .

React.createClass

mixins mixins . mixins mixins . .

```
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

React mixins ES6 React . React ES6 . .

"this"Context

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 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:

constructor this.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;

```

3: ES6

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.Component **React.createClass** : <https://riptutorial.com/ko/reactjs/topic/6371/react-component---react-createclass>

8: ReactJS jQuery

Examples

ReactJS jQuery

, jquery . dom findDOMNode . React .

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

'handleToggle' . " div .

```
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>
```

div onClick 'handleToggle' .

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

```
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>
  );
}
}
}

```

! React jQuery .

ReactJS jQuery : <https://riptutorial.com/ko/reactjs/topic/6009/reactjs-jquery->

9: React

React key . .

React React . React . React .

. <https://facebook.github.io/react/docs/lists-and-keys.html>

. <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/ko/reactjs/topic/9665/react----->

10: Redux

Redux " " . React Redux .

React , . Redux () . . Redux .

Examples

createStore Redux .

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

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 }
}
```

```
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/ko/reactjs/topic/10856/redux->

11: Typescript ReactJS

Examples

Typescript ReactJS

ReactJS Typescript `. 'jsx' 'tsx'`.

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

Typescript () .

1) React.createClass ES6 . Class :

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

2) :

```
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 .

Typescript Stateless React

JavaScript .

```
import React from 'react'

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

React.SFC Typescript .

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

class GreeterProps {
  name: string
}

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

React.SFC React.StatelessComponent .

npm . npm init

npm

npm React .

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

Yarn . Yan React . Yarn .

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

React via 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 , html / xml javascript typescript . typescript (tsconfig.json) JSX .

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

typescript JSX javascript .

typescript JSX .

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

Stateless Propertyless

.

```
import * as React from 'react';  
  
const Greeter = () => <span>Hello, World!</span>
```

typescript this.props . .

```
import * as React from 'react';  
  
const Greeter: React.SFC<{}> = props => () => <span>Hello, World!</span>
```

props.children props.children props.children .

stateless property-less . Container , NavTop NavBottom Page .

```
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 .

Typescript ReactJS : <https://riptutorial.com/ko/reactjs/topic/1419/typescript-reactjs->

12:

("HOC") . . .

HOC javascript .

HOC 3 . 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);
```

.

render () HOC .

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);

```

.
: <https://riptutorial.com/ko/reactjs/topic/9819/-->

13:

Examples

- HTML DOM DOM

HTML DOM .

. . (: JavaScript) HTML .

HTML DOM HTML , , , .

DOM .

DOM

HTML DOM HTML DOM DOM .

DOM DOM .

?

React Virtual DOM . React

1. DOM

2. DOM (HTML DOM) DOM

3. 2. DOM DOM (HTML DOM) .

- - React Diff .

4. HTML DOM

- DOM HTML DOM DOM

5. HTML DOM HTML DOM HTML DOM .

: DOM JavaScript Virtual DOM .

diff

$O(n^3)$. n . - $O(n)$

1. .

2. .

React 3 .

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

[React Official Guide](#) [Benchling Engg](#) .

: <https://riptutorial.com/ko/reactjs/topic/6875/>

14:

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>
```

scripts / 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>
    );
  }
});

```

AB BC .

- UI
- .
- .
- .
- BC .
- BC

2.

()

```

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

```

AB ABC . .

-
-
-

- .
-
- C B.
- A C .

B this.props.children C , B . B B # 3 .

3.

()

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

A B B A . . .

-
-
-
- .
-
- B .
- B .

3 . # 1 # 2 # 3 .

.

```
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')
  );

```

().

ES6 Stateless Functional Components .

JSX HTML . . .

2 :

- 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;

```

'stateless' 'stateful' 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>
  );
}
}

```

`PureComponent` Component `shouldComponentUpdate()` . . . 'Pure' .

(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 . **get** setState .

```

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>;
  }
}

```

.Component this.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 .

. cancel this.setState 'stateSetter' .

```

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>
  }
}

```

setState cancelled .

React ..

JSX

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

MyComponent userID props .

```

// 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 someObject` `prop userID .userID MyComponent` `React` `. React` , .

`defaultProps` .

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

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

object array functions functions . .

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

`this.props.someObject` `undefined` `someKey` `.defaultProps` .

-

- (h3). `. React` `if` `.()`

```
// 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>
        </Button>
      </div>;
    }

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

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

```

clickableTitle . show Form Button . React .

{renderableTitle} . 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')
)
```

[: https://riptutorial.com/ko/reactjs/topic/1185/-](https://riptutorial.com/ko/reactjs/topic/1185/)

15:

Examples

Redux React Redux . React React Dom

- **UserInputForm** : . props inputChange () .
- **UserDashboard** : UserInputForm inputChange UserInputForm , UserInputForm return
 - **UserDashboardConnected** : ReactRedux connect UserDashboard 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/ko/reactjs/topic/6137/--->

16:

React 3 .

- 1:
- 2:
- 3: ()

Examples

React , . . .

. message .

```
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 child 5 .

() () .

.
.

```
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 button onClick .

- (:) .

2 : , /// .

3 . .

.

- 1 : Event Emitter / Target / Dispatcher : .

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

- 2 : / : .

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

- 3 : : Event Emitter / Target / Dispatcher . . .

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

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

17: ,

" " . () Yeoman.io . .

Mocha Jasmine Karma . ESLint linting. webpack-dev-server Ctrl + S, F5 . dev .

!

webpack.config.js __dirname /src . Windows . .

webpack.config.js

webpack . index.jsx require('./index.html') require('./index.html') src/ src/ .

webpack . index.jsx .

webpack . .

devServer

webpack-dev-server . contentBase dist/ folder . port . open webpack-dev-server .

>

webpack . test webpack . . loader loaders webpack . query . HTML

. .

Examples

"Hello world" .

1 : Node.js

Node.js . Node.js SO .

2 :

:

```
npm init
```

SO .

3 : npm

.

```
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
```

webpack webpack-dev-server

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

3 : .babelrc .

. .babelrc .

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

4 :

.

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

: node_modules , .babelrc package.json .

5 : Hello World

.

src / components / 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 / images / myImage.gif

```

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

```

src / styles / 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'));

```

6 :

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;

```

7 : npm

package.json JSON . . .

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

test . . .

8 :

.

```
npm run build
```

dist/ .

:

```
npm start
```

webpack dev .

, : <https://riptutorial.com/ko/reactjs/topic/6294/----->

18:

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) `Stateful React` .

React v14 . JavaScript .

:

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

props `React` `React` . `JavaScript` .

UI . "" `Flux / Redux` . `Stateless` .

:

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

: `React` `UI` `React` **`Stateless`** .

.
.

`index.js` .

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

import HomePage from './homepage'

ReactDOM.render(
  <HomePage/>,
  document.getElementById('app')
);
```

`HomePage`

UI ., UI React . state List () 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>
);
}
}

```

, 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/ko/reactjs/topic/6588/--->

19:

Examples

;

-
- [JS.coach](#)

: [https://riptutorial.com/ko/reactjs/topic/6595/-](https://riptutorial.com/ko/reactjs/topic/6595/)

20:

Examples

Routes.js ,

. .

```
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'));
```

<a> Link . React Router React Router . routes.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/ko/reactjs/topic/6096/>

21:

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>
    )
  }
}
```

}
person . 2 person . 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.lastName person[event.target.name]
person.firstName || person.lastName . . . . . onChange .
```

: [https://riptutorial.com/ko/reactjs/topic/8047/-](https://riptutorial.com/ko/reactjs/topic/8047/)

22:

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.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"}
    ]
  }
}
```

.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>
```

Transpile

webpack .

```
$ webpack
```

build .

HTML

: <https://riptutorial.com/ko/reactjs/topic/7480/-->

23:

: React 15.5 PropTypes npm 'prop-types' PropTypes import .
<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 .

defaultProps props . 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() defaultProps this.props.randomObject .

PropTypes

propTypes props props . propTypes React .

propTypes propTypes propTypes propTypes

```

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

```

isRequired propTypes . 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} />;
```

... ES6 . .

```
var component = <Component {...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 . "" JSX DOM .

```
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 [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>
  );
}
```

```
}
```

Reps.cloneElement <p> . . .

React [Re レン](#) React key .

Children

. children React Children.map util :

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

child type .

: [https://riptutorial.com/ko/reactjs/topic/2749/-](https://riptutorial.com/ko/reactjs/topic/2749/)

24:

Examples

React . JavaScript . .

constructor render 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 ()

React UI setState() . .

1. updater : - .
2. callback (optional) : setState() . setState() **React** setState() setState() .

:

setState - prevState props updater .

updater **setState ()**

```
//
// 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>
      );
    }
  }
}

```

updater **setState()**

```

//
// 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 });

```

Object.assign() **React** **3 1** .

```

// 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'));
```

props state . . . :

```
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>  
    )  
  }  
}
```

url state . state props .

```
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 . . .

MyComponent 'url' . MyComponent state.url . . . url MyComponent .

,

""" React . . . setState dom render .

```

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. - div, , , h1
2. h1
- 3.
4. DOM ?

.

: [https://riptutorial.com/ko/reactjs/topic/1816/-](https://riptutorial.com/ko/reactjs/topic/1816/)

25:

DOM .

li li key . ID .

. React Docs :

. . .

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

Examples

id

.

id . id .

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

todo-\${todo.id} .

```
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/ko/reactjs/topic/9805/>

26:

React . " ". appretiated.

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 '';
  }
}
```

- .
- title title (, title);
- css (PANEL.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()
      )
    );
  }
}
```

Tab panelClass .

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();
}
}

```

PanelGroup panels

- **name** - .
- **class** .

activeTab .

DOM active (,) .

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/ko/reactjs/topic/8112/--](https://riptutorial.com/ko/reactjs/topic/8112/--)

27:

Examples

(renderToString renderToStaticMarkup .

renderToString

React HTML . HTML data-react- React .

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

renderToStaticMarkup

React HTML data-react- .

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

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

28:

Examples

```
src/ out/ . .
```

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

npm , React es5 babel .

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

npack webpack babel . JSX es6 es6 .([Babel](#) .)

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

webpack .(**i -D --save-dev**)

webpack (webpack-dev-server) ,

.

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

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']
  }
}
]
}
};

```

webpack index.js (src /) out bundle.js .

```

module webpack      .( babel-loader ) exclude regex webpack node_modules      . , query
babel  webpack      .

```

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' id html , .

```

$./node_modules/.bin/webpack . webpack (-g webpack $webpack )

```

transpilation out/bundle.js .

webpack-dev-server

webpack , \$npm i -g webpack-dev-server \$npm i -g webpack-dev-server http .

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`
 - `contentBase` (`:index.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-server 8080 http <h1>Hello world!</h1>` .

[: https://riptutorial.com/ko/reactjs/topic/6441/](https://riptutorial.com/ko/reactjs/topic/6441/)

29:

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 . value defaultValue

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

30:

. Mounting, Updating Unmounting .

Examples

React .

- `React.createClass` (ES5) 5 .
- `class Component extends React.Component` (ES6) 3 .

`getDefaultProps()` (ES5)

.

Prop .

`this.props.name` Bob .

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

`getInitialState()` (ES5)

.

`getInitialState()` React . React `this.state` .

`this.state.count` `this.props.initialCount` .

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

`componentWillMount()` (ES5 ES6)

.

DOM .

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

render() (ES5 ES6)

render() . DOM (:<p />) . null undefined .

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

componentDidMount() (ES5 ES6)

refs DOM .

-
-
-
- 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 .

```

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)

/ / **React** . true .

false .

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

componentWillUpdate (nextProps, nextState)

componentWillMount () . **DOM** .

! \ : this.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>
    );
  }
}
```



```
}
```

React , Presentational Container 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)
}
```

(componentWillMount, componentWillReceiveProps) , .

:

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/ko/reactjs/topic/2750/--->

31:

React

Examples

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

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

```
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
  }
}
```

[: https://riptutorial.com/ko/reactjs/topic/7918/---](https://riptutorial.com/ko/reactjs/topic/7918/---)

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	AJAX	adamboro , Fabian Schultz , Jason Bourne , lifeiscontent , McGrady , Sunny R Gupta
3	Flux ReactJS	vintproykt
4	JSX	Kaloyan Kosev , Ming Soon
5	React Boilerplate [+ + Webpack]	Mihir , parlad neupane , Tien Do
6	React, Webpack Typescript	Aron
7	React.Component React.createClass	Kaloyan Kosev , leonardoborges , Michael Peyper , pwolaq , Qianyue , sqzaman
8	ReactJS jQuery	Kousha , Shuvo Habib
9	React	Sammy I.
10	Redux	Jim
11	Typescript ReactJS	Everettss , John Ruddell , kevgathuku , Leone , Rajab Shakirov
12		Dennis Stücken
13		Aditya Singh , Iustoykov , thibmaek
14		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
15		Random User
16		David , Kaloyan Kosev
17	,	Bart Riordan , Tien Do , Zac Braddy
18		Adam , Mark Lapierre , Mayank Shukla , Valter Júnior
19		brillout
20		abhirathore2006 , Robeen
21		promisified
22		ghostffcode , Tien Do
23		Ahmad , Anuj , Danillo Corvalan , Everettss , Faktor 10 , Fellow Stranger , hansn , Ilya Lyamkin , Jack7 , Jagadish Upadhyay , JimmyLv , MaxPRafferty , QoP , Sergii Bishyr , vintproykt , WitVault , zbynour
24		Alex Young , Alexander , Brad Colthurst , Everettss , Kousha , Kyle Richardson , QoP , skav , Timo
25		Dennis Stücken , thibmaek
26		vintproykt
27		Adrián Daraš , MauroPorrasP
28		Rene R , Ruairi O'Brien
29		Everettss , Henrik Karlsson , ivarni , Timo
30		Alex Young , Alexg2195 , Anuj , Ashari , Everettss , F. Kauder , irrigator , John Ruddell , QoP , Salman Saleem , Saravana , Siddharth , skav , Timo , ultrasamad , Vivian , WitVault
31		JimmyLv , lifeiscontent , Rifat , Rory O'Kane