

# React.JS for Beginner

---

Maiffee Ul Asad

Want to start React.js,  
but don't have that  
much time?

What about a  
kick-start?

Hope it helps!

# Intro



# What is React.JS

---

# What is React.JS

A library, not a framework



# What is React.JS

A library, not a framework

- V of MVC



# Why React.js ?

---



# Why React.js ?

Why not?

---

# Why React.js ?

Why not?

- For building UI
- Reactive
- Easy to develop
- Easy to integrate
- Widely used & huge demand



# Why React.js ?

Why not?

- For building UI
- Reactive
- Easy to develop
- Easy to integrate
- Widely used & huge demand

---

# Why React.js ?

Why not?

- For building UI
- Reactive
- Easy to develop
- Easy to integrate
- Widely used & huge demand



# Why React.js ?

Why not?

- For building UI
- Reactive
- Easy to develop
- Easy to integrate
- Widely used & huge demand



# Why React.js ?

Why not?

- For building UI
- Reactive
- Easy to develop
- Easy to integrate
- Widely used & huge demand

---

# Why React.js ?

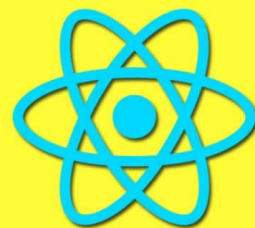
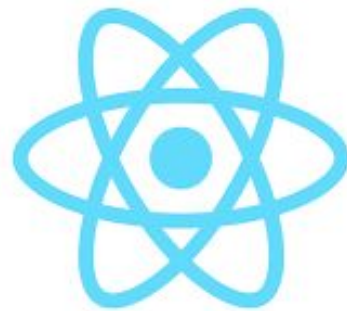
Why not?

- For building UI
- Reactive
- Easy to develop
- Easy to integrate
- Widely used & huge demand



# ES6(All we need to know)

- `let/const`
- `arrow function`
- `import/export`
- `class`
- `map`
- `promise`
- `destruction`



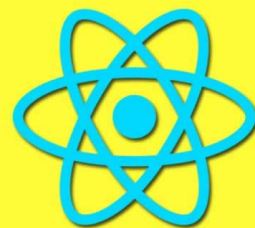
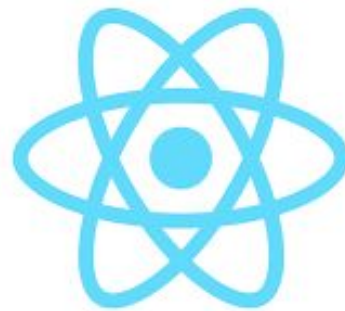
{...}

**ES6**



# ES6(All we need to know)

- `let/const`
- `arrow function`
- `import/export`
- `class`
- `map`
- `promise`
- `destruction`

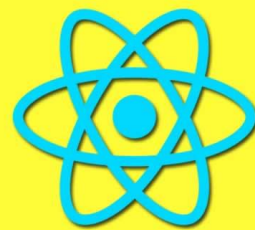
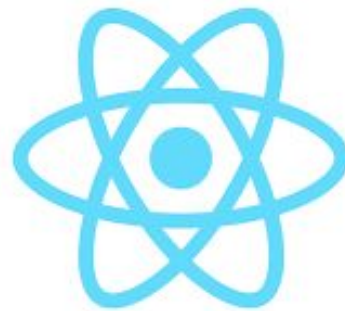


{...}

**ES6**

# ES6(All we need to know)

- `let/const`
- `arrow function`
- `import/export`
- `class`
- `map`
- `promise`
- `destruction`

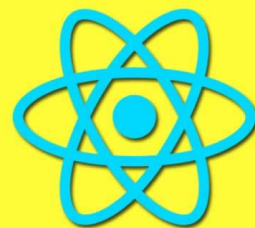
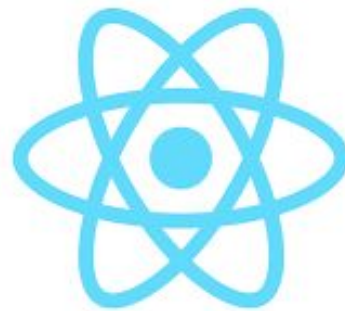


{...}

**ES6**

# ES6(All we need to know)

- `let/const`
- `arrow function`
- `import/export`
- `class`
- `map`
- `promise`
- `destruction`

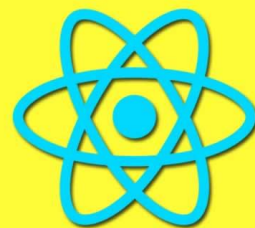
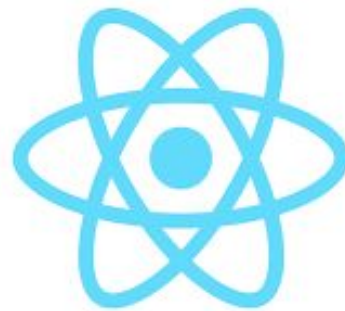


{...}

**ES6**

# ES6(All we need to know)

- `let/const`
- `arrow function`
- `import/export`
- `class`
- `map`
- `promise`
- `destruction`

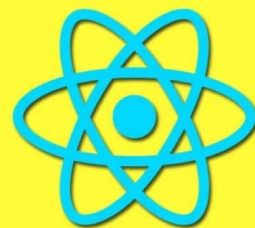
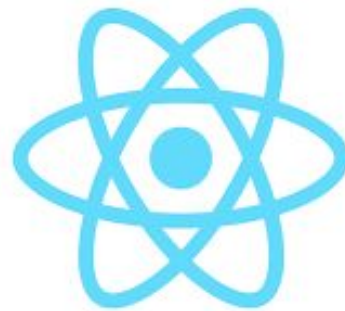


{...}

**ES6**

# ES6(All we need to know)

- `let/const`
- `arrow function`
- `import/export`
- `class`
- `map`
- `promise`
- `destruction`

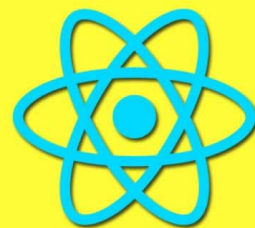
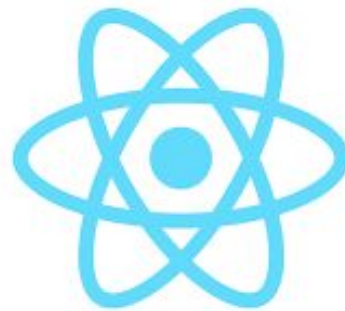


{...}

**ES6**

# ES6(All we need to know)

- `let/const`
- `arrow function`
- `import/export`
- `class`
- `map`
- `promise`
- `destruction`

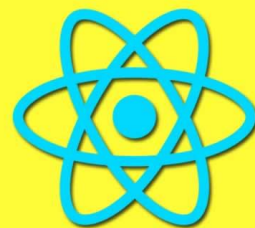
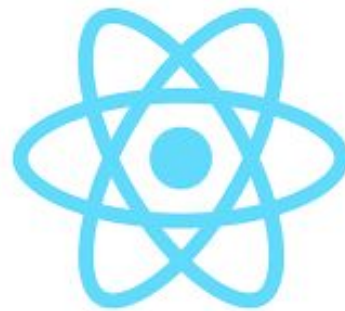


{...}

**ES6**

# ES6(All we need to know)

- `let/const`
- `arrow function`
- `import/export`
- `class`
- `map`
- `promise`
- `destruction`



{...}

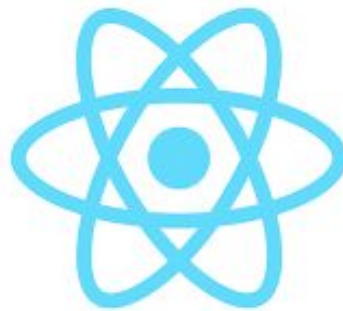
**ES6**

# JSX(All we need to know)

- js + xml
- `<something></something>`
- `<something/>`
- They are actually object 🤔
- Under the hood

```
import * as serviceWorker from './serviceWorker'
import { render } from 'react-dom'
import './index.css'
import App from './App'
import * as routes from './routes'

ReactDOM.render((
  <BrowserRouter>
    <Switch>
      <Route path="/login" component={Login} />
      <ProtectedRoute exact={true} path="/" component={App} />
      <ProtectedRoute path="/settings" component={Settings} />
      <ProtectedRoute path="/about" component={About} />
    </Switch>
  </BrowserRouter>
), document.getElementById('root'))
```



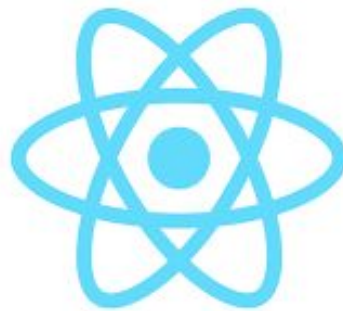


# JSX(All we need to know)

- js + xml
- `<something></something>`
- `<something/>`
- They are actually object 🤔
- Under the hood

```
import * as serviceWorker from './serviceWorker'
import { render } from 'react-dom'
import './index.css'

ReactDOM.render((
  <BrowserRouter>
    <Switch>
      <Route path="/login" component={Login} />
      <ProtectedRoute exact={true} path="/" component={Home} />
      <ProtectedRoute path="/settings" component={Settings} />
      <ProtectedRoute path="/about" component={About} />
    </Switch>
  </BrowserRouter>
), document.getElementById('root'))
```

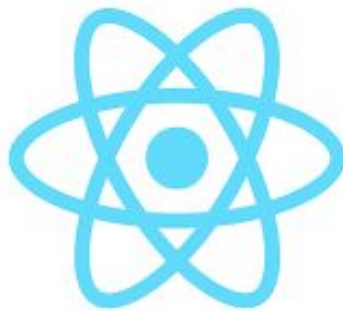


# JSX(All we need to know)

- js + xml
- `<something></something>`
- `<something/>`
- They are actually object 😊
- Under the hood

```
import * as serviceWorker from './serviceWorker'
import { render } from 'react-dom'
import './index.css'

ReactDOM.render((
  <BrowserRouter>
    <Switch>
      <Route path="/login" component={Login} />
      <ProtectedRoute exact={true} path="/" component={Home} />
      <ProtectedRoute path="/settings" component={Settings} />
      <ProtectedRoute path="/about" component={About} />
    </Switch>
  </BrowserRouter>
), document.getElementById('root'))
```

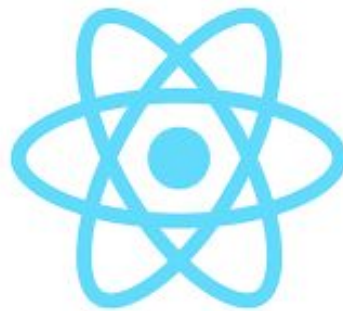


# JSX(All we need to know)

- `js + xml`
- `<something></something>`
- `<something/>`
- They are actually object 🤔
- Under the hood

```
import * as serviceWorker from './serviceWorker';

ReactDOM.render((
  <BrowserRouter>
    <Switch>
      <Route path="/login" component={Login} />
      <ProtectedRoute exact={true} path="/settings" component={Settings} />
      <ProtectedRoute path="/dashboard" component={Dashboard} />
    </Switch>
  </BrowserRouter>
), document.getElementById('root'));
```

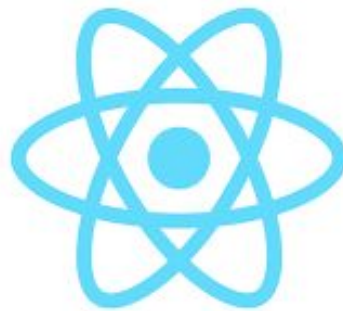


# JSX(All we need to know)

- js + xml
- `<something></something>`
- `<something/>`
- They are actually object 🤔
- Under the hood

```
import * as serviceWorker from './serviceWorker'
import { render } from 'react-dom'
import { BrowserRouter } from 'react-router-dom'
import { Switch, Route, ProtectedRoute } from 'react-router-dom'

ReactDOM.render((
  <BrowserRouter>
    <Switch>
      <Route path="/login" component={Login} />
      <ProtectedRoute exact={true} path="/settings" component={Settings} />
      <ProtectedRoute path="/settings" component={Settings} />
    </Switch>
  </BrowserRouter>
), document.getElementById('root'))
```



# State

---

# What is 'state'

Just some observable properties  
in a component

- Think like - they are just member of class, specifically field
- And they are reactive



# Props

---

# What is 'props'

Something got passed to a component are props

- Just think we have a parent component which is calling some API, or somehow it has some data. And it will render some child component based on those data. To render those data, we need to pass those data to child first. We will be doing this with props.
-



Communication

# Parent to Child

**In one word `props`**

**Just pass whatever, and receive using  
props**

- `<Child name={“babu”}/>`
- `this.props.name`

# Child to Parent

**In a few words `callback` as props**

**Instead of passing a variable, we will be passing an callback function**

- `<Child name={xyz=>{}}>`
- `this.props.name('shona')`

# Child to Child

**First pass to parent, then send it down to it's child**

**Instead of passing a variable, we will be passing an callback function**

- `<Child name={xyz}=>{} defaultName={'babu'}/>`
- `this.props.name('shona')`

# Anywhere?

## Callback Hell

```
1 function hell(win) {  
2   // for listener purpose  
3   return function() {  
4     loadLink(win, REMOTE_SRC+'/assets/css/style.css', function() {  
5       loadLink(win, REMOTE_SRC+'/lib/async.js', function() {  
6         loadLink(win, REMOTE_SRC+'/lib/easyXDM.js', function() {  
7           loadLink(win, REMOTE_SRC+'/lib/json2.js', function() {  
8             loadLink(win, REMOTE_SRC+'/lib/underscore.min.js', function() {  
9               loadLink(win, REMOTE_SRC+'/lib/backbone.min.js', function() {  
10                loadLink(win, REMOTE_SRC+'/dev/base_dev.js', function() {  
11                 loadLink(win, REMOTE_SRC+'/assets/js/deps.js', function() {  
12                  loadLink(win, REMOTE_SRC+'/src/' + win.loader_path + '/loader.js', function() {  
13                   async.eachSeries(SERIALS, function(src, callback) {  
14                     loadScript(win, BASE_URL+src, callback);  
15                   });  
16                 });  
17               });  
18             });  
19           });  
20         });  
21       });  
22     });  
23   });  
24 });  
25 }  
26 }
```



# Anywhere? Solution

- context
- store
- 3rd party libraries

# Lifecycle

# Lifecycle methods

- `componentDidMount`
- `componentDidUpdate`
- `componentWillUnmount`



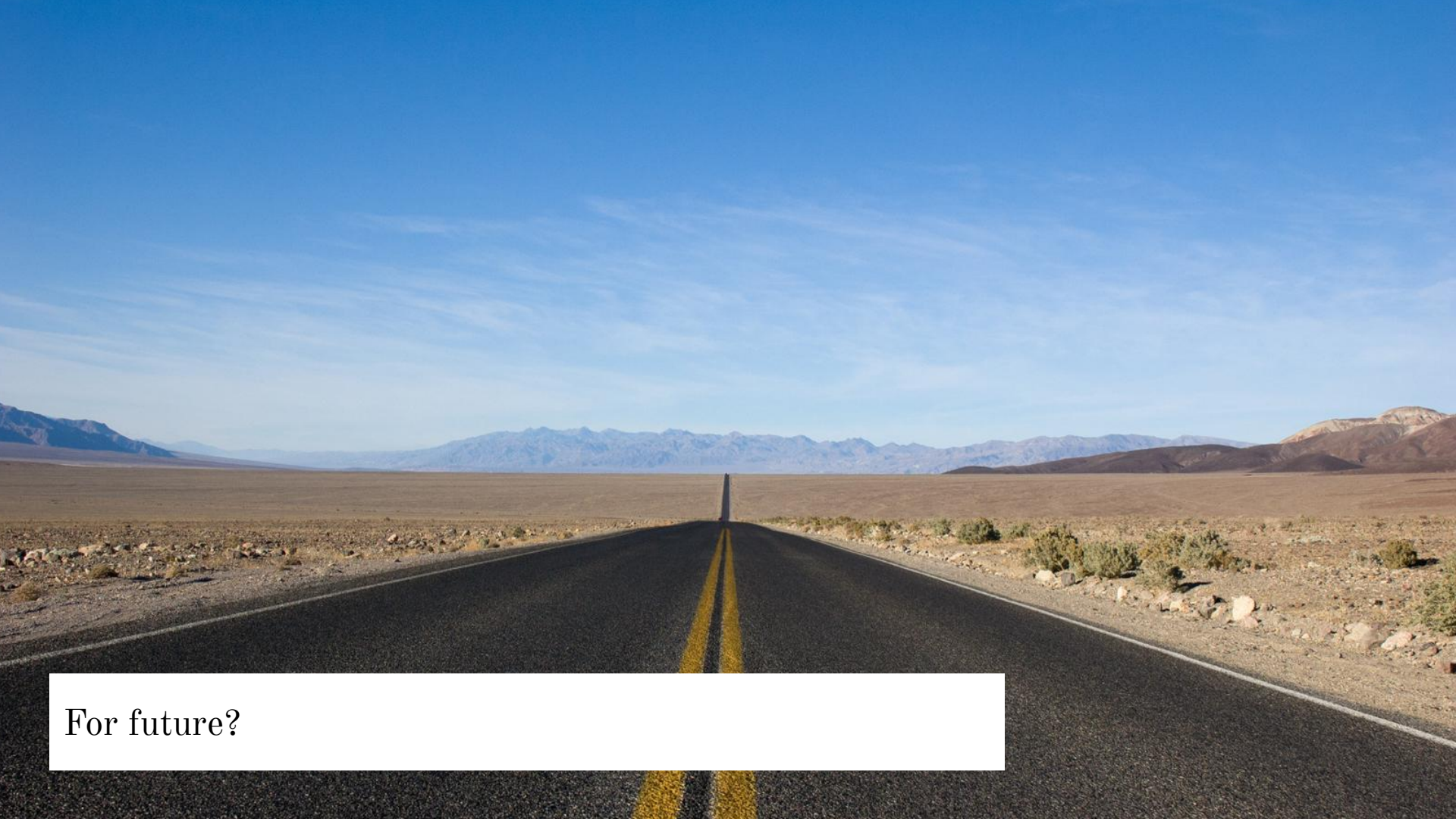
# Project Structure

A suggested structure? No, not really


# But a suggested one for Beginners

```
tree -I node_modules
├── LICENSE
├── package.json
├── public
│   ├── favicon.ico
│   ├── index.html
│   ├── logo192.png
│   ├── logo512.png
│   ├── manifest.json
│   └── robots.txt
├── README.md
├── src
│   ├── index.js
│   ├── network
│   │   └── http.js
│   ├── presentation
│   │   └── component
│   │       ├── auth
│   │       │   └── HOCAuth.js
│   │       ├── home
│   │       │   ├── HomeComponent.js
│   │       │   └── styles.module.css
│   │       └── login
│   │           ├── LoginComponent.js
│   │           ├── login.svg
│   │           └── styles.module.css
│   └── util
│       └── constant.js
└── yarn.lock

9 directories, 19 files
```



For future?



HOC & TypeScript & Redux & Deep Dive





These will be published, when they will be published.

# Tada

Keep me in your prayer.

And I post randomly on social networks.

YouTube tutorials are "videos of, by and for the noob"

Peace out.