



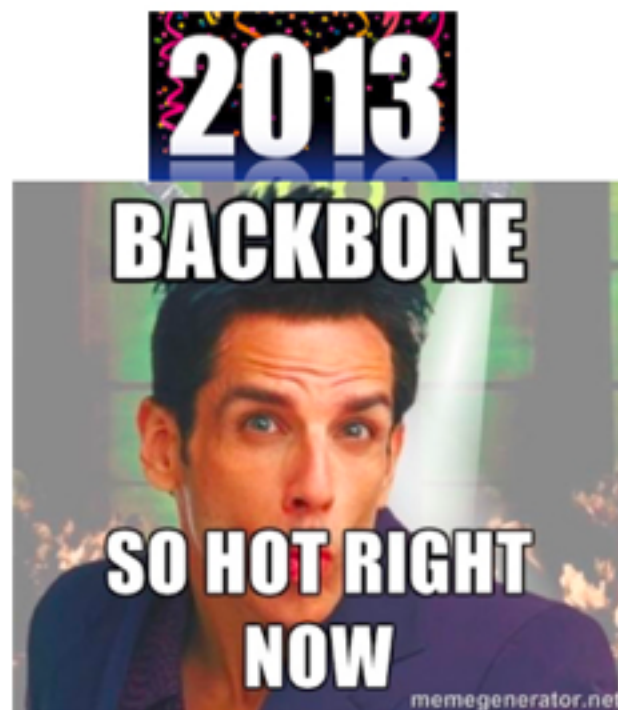
workshop for JavaEE programmer

after ~ 4 months of using :-)

```
/**  
 * @date 14/03/2018  
 * @author Martin Gabor  
 */
```

Table of contents

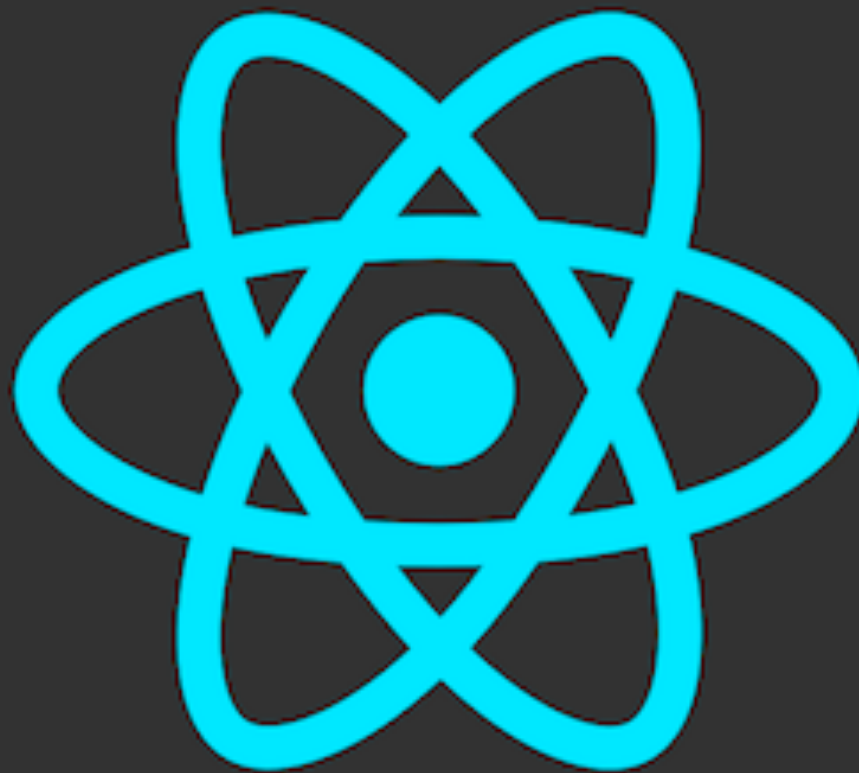
- What is React?
- Javascript evolution.
- React fundamentals.
 - Npm, webpack, babel
 - React virtual DOM.
 - JSX
 - State vs props.
- Simple application.
- Redux/Mobx libraries
- Debugging
- Logaweb react integration, shared react library
- PEP
- Links



Img: <http://blog.teamtreehouse.com/react-hype-real>

What

is



?

Target



Lint



Compile



Test



Serve



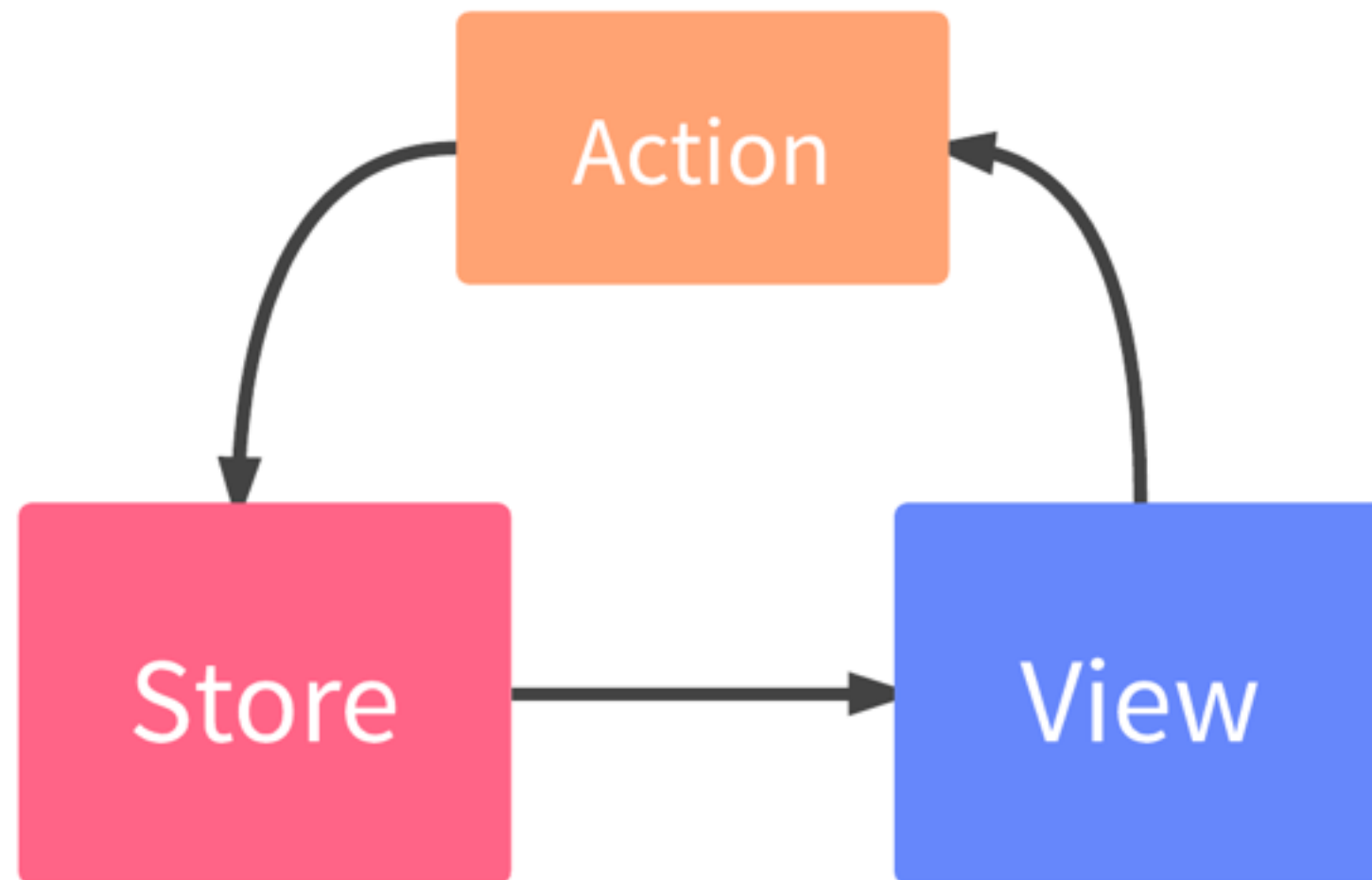
Redux



So what exactly React is?

- ☑ A javascript library for creating user interfaces.
- ☑ React does one thing and does it well.
- ☑ Declarative
- ☑ Component based.
- ☑ Renders your UI and responds to events.
- ☑ Virtual DOM

Unidirectional flow



Sample component

```
import React from 'react';

/**
 * @date 14/03/2018
 * @author Martin Gabor
 */

// component definition
class LRSampleComponent extends React.Component {

    // every component has render
    render() {

        // return JSX
        return (
            <div className="LRSampleComponent">
                <div className='LRSampleComponent-Child' />
            </div>
        );
    }
}

export default LRSampleComponent;
```


Javascript evolution

https://www.w3schools.com/js/js_versions.asp

Javascript

EcmaScript 6 quick overview

Javascript

types and standard objects

Standard types:

- number, string, boolean for true/false.
- object
- symbol for unique identifiers.

Special values:

- Infinity, NaN, undefined, null

typeof operator:

```
typeof undefined // "undefined"
```

```
typeof 0 // "number"
```

```
typeof true // "boolean"
```

```
typeof "foo" // "string"
```

```
typeof Symbol("id") // "symbol"
```

```
typeof Math // "object"
```

```
typeof null // "object" (This is error in typeof function)
```

```
typeof alert // "function" (Function is actually an object)
```

Variables

```
for(var i = 0; i < 10; i++) {  
    let scoped = i  
    const scopedConstant = i  
    console.log(i)  
}
```

```
console.log(i)  
console.log(scoped) // variable undefined  
console.log(scopedConstant) // variable undefined
```

Objects and Arrays

```
// define object person
const person = {
  name : 'Peter',
  age: 42
};

// accessing person
person.name; // "Peter"
person['age']; // 42
person.name = 'Homer';

const person2 = {
  name : 'Jan',
  age: 20
};

// define and initialize array
const personArray = [ person, person2 ];

// accessing personArray
personArray[0]['age']; // 42
```

Functions

```
// non-class function
function plus(a, b) {
    return a + b;
}

// anonymous function call
const arr = [1,2,3].map(function(item) { return item + 1 });

// arrow function call
const arr2 = [1,2,3].map(item => item + 1 );

class StringClazz {

    // class function
    concat(a, b) {
        return a + b;
    }

    // class function variable defined with arrow function
    concatArrowFn = (a, b) => {
        return a + b;
    };

    // self invoking functions
    CONSTANT = (function (a, b) { return a + b } )('Hello ', 'P&I');
    CONSTANT2 = this.concatArrowFn('Hello ', 'P&I');
}
```

Pure vs impure functions

PURE

- Given the same input, will always return the same output.
- Produces no side effects.

```
// impure addToCart mutates existing cart
const addToCart = (cart, item, quantity) => {
  cart.items.push({
    item,
    quantity
  });
  return cart;
};

function impureFunc(value){
  return Math.random() * value;
}

function pureFunc(value){
  return value * value;
}
```

SIDE EFFECTS

- Making a HTTP request
- Mutating data
- Printing to a screen or console
- DOM Query/Manipulation
- Math.random()
- Getting the current time

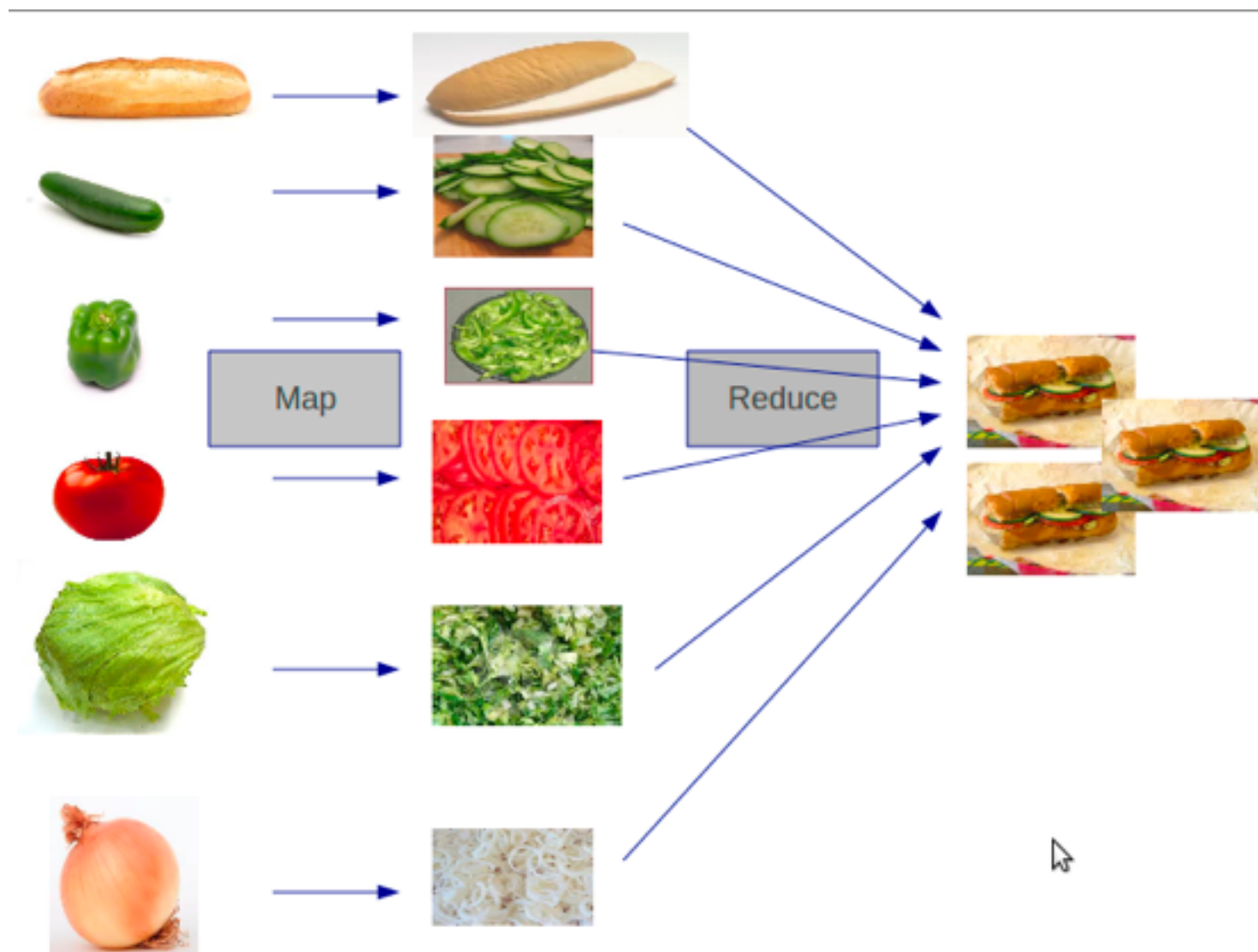
All React components must act like pure functions with respect to their props.

JavaScript

functional programming

map, filter, reduce

<https://medium.com/jsguru/javascript-functional-programming-map-filter-and-reduce-846ff9ba492d>



Classes

```
import React from 'react';

export class LRHeader extends React.Component {

  title = "Hello P&I!";

  printTitle() {
    console.log(this.title); // doesn't know title
  }

  printArrowFunction = () => {
    console.log(this.title);
  };

  render() {
    return (
      <h1 onMouseOver={this.printArrowFunction}
        onMouseOut={this.printTitle}>
        {this.title}
      </h1>
    );
  }
}
```

Equality operators

```
if ("5" == 5)
    console.log("P&I Bratislava is the best!");

if ("5" === 5)
    console.log("P&I Greece is the best!");

if (null === undefined)
    console.log("We love JavaScript!");
```

https://www.w3schools.com/js/js_comparisons.asp

Destructuring

```
const person = {  
  name: 'Peter',  
  age: 10,  
  gender: 'Male',  
};  
  
console.log("My name is " + person.name);  
  
const {age, gender} = person;  
  
console.log("I am " + age);
```

Shorthand property names

```
const name = 'Peter', age = 10;
```

```
const personData = {  
  name: name,  
  age: age  
};
```

```
// shorthand way
```

```
const person = { name, age };
```

Spread operator

```
const person = {  
  name: 'Peter',  
  age: 42  
};
```

```
const clonedObj = { ...person, gender: 'M' };  
// clonedObj = {  
//   name: 'Peter',  
//   age: 42,  
//   gender: 'M'  
// }
```

spread in objects

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_syntax

... and other features



React environment

React world is basically similar to JavaEE

	JavaEE + GWT	Javascript + React
Packaging	Maven	NPM / yarn
Server	Wildfly/JBoss	Webpack (NodeJs)
Javascript transpiler	GWT	Babel
UI library	GWT, ...other javascript libs	React , ...other javascript libraries



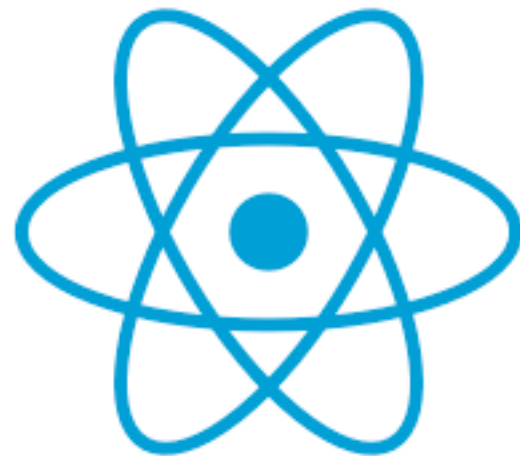
- **N**ode **P**ackage **M**anager
- package.json, package-lock.json
- javascript semantic versioning: *major.minor.patch*

```
"dependencies": {  
  "react": "^16.2.1",  
  "react-dom": "^16.2.0"  
}
```
- <https://docs.npmjs.com/getting-started/semantic-versioning>
- types: dependencies, devDependencies, peerDependencies

What about...

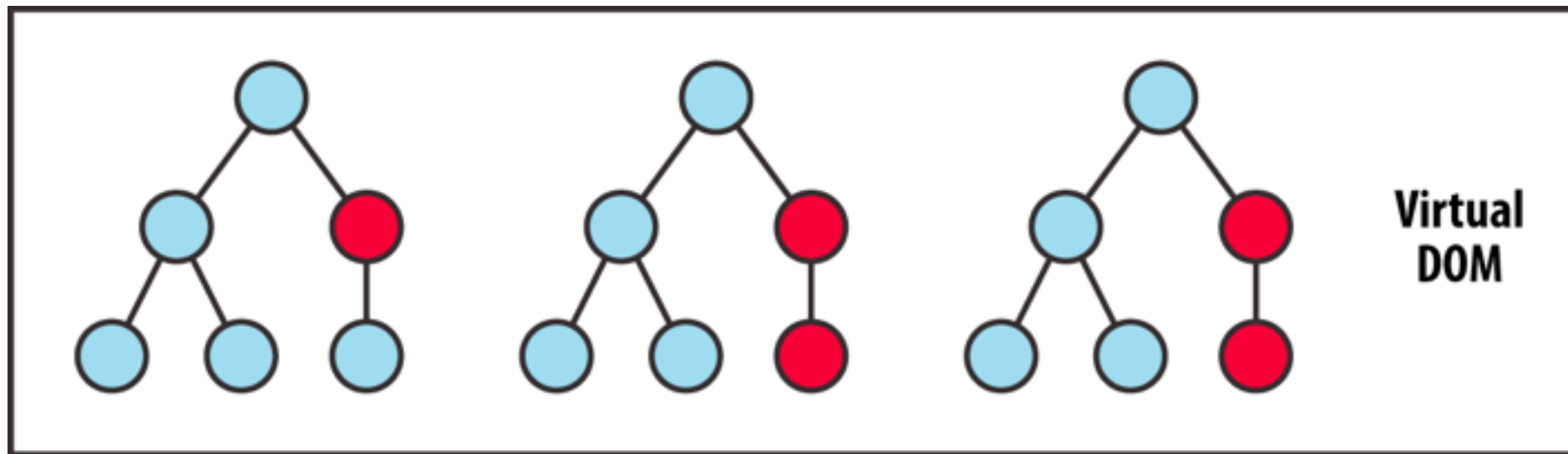
webpack and **babel**?

Create React App

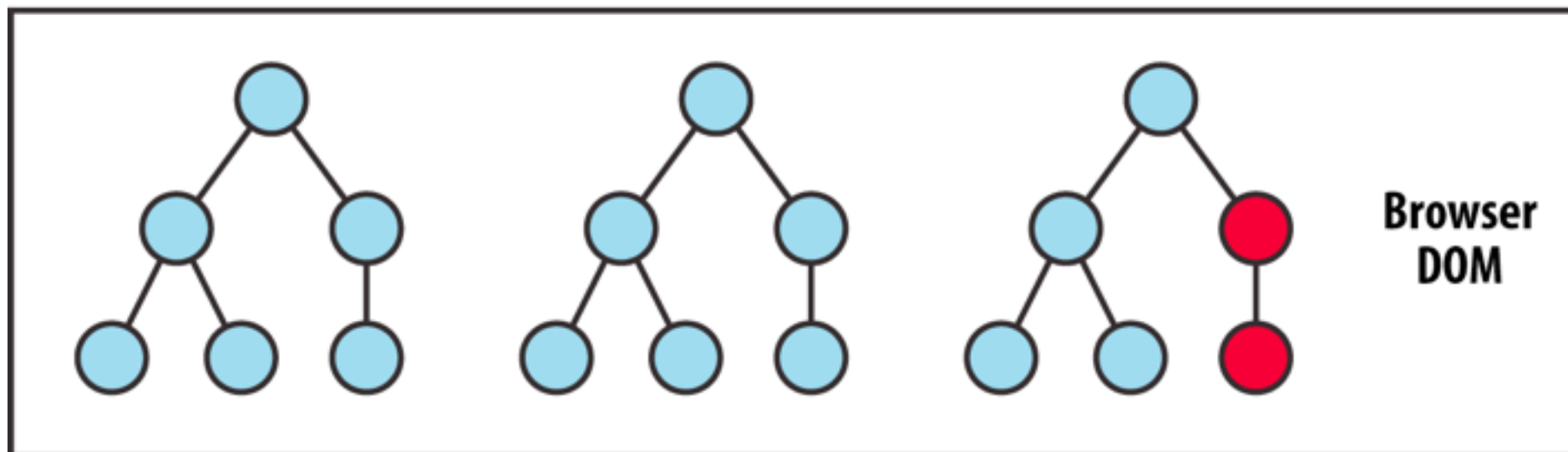


Official. No Setup. Minimal.

Virtual DOM



State Change → Compute Diff → Re-render



JSX

```
const element = (  
  <h1 className="greeting">  
    Hello, world!  
  </h1>  
);
```

Babel transpiles it into:

```
const element = React.createElement(  
  'h1',  
  {className: 'greeting'},  
  'Hello, world!',  
);
```

React state vs props

<http://lucybain.com/blog/2016/react-state-vs-props/>

Props

(= properties)

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

```
const element = <Welcome name="P&I" />;
```

- ☑ are used to pass data down
- ☑ are immutable

State

(stateful component)

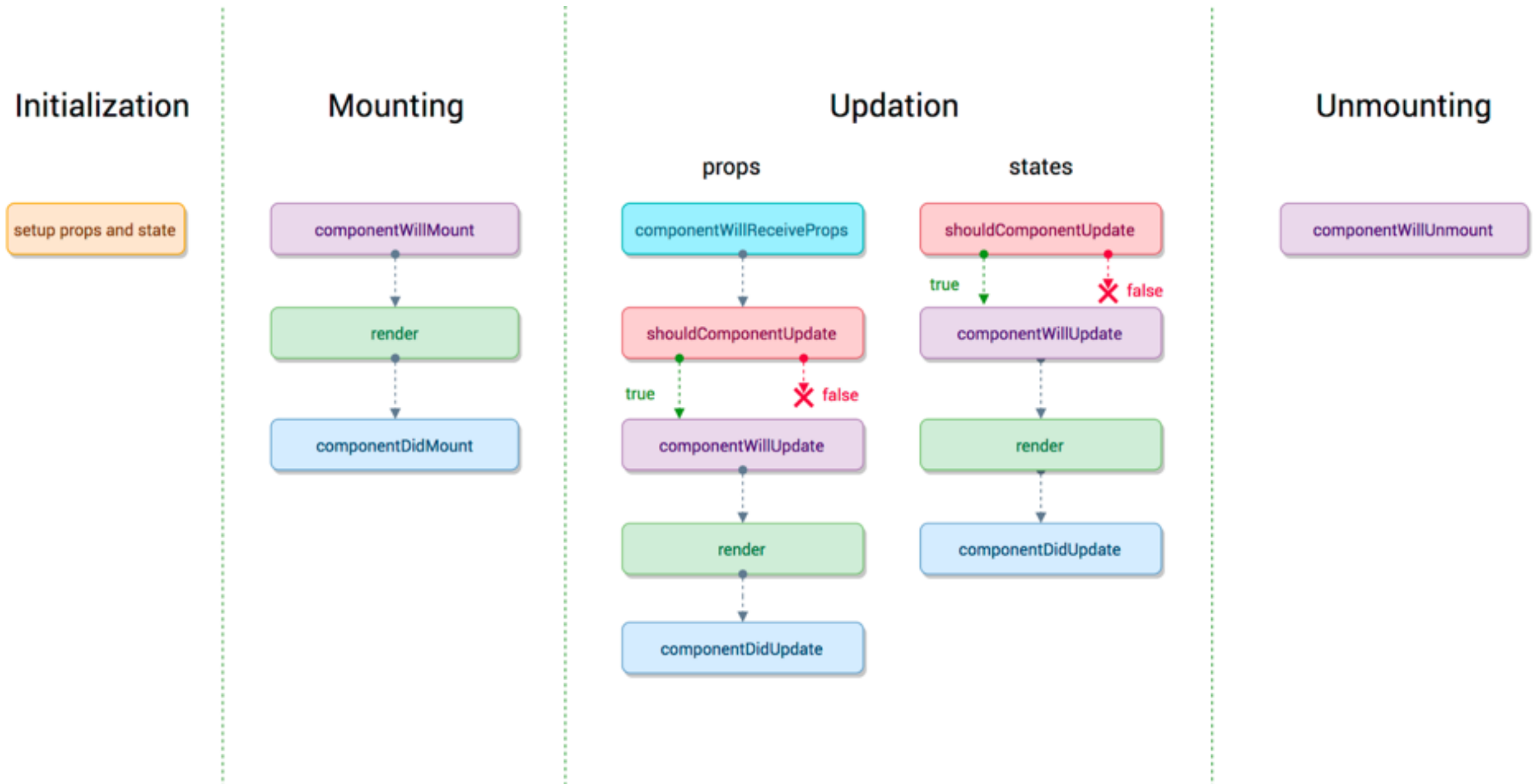
```
class Button extends React.Component {  
  state = {  
    count: 0,  
  };  
  
  updateCount = () => {  
    this.setState({ count: this.state.count + 1 });  
  };  
  
  render() {  
    return (  
      <button onClick={this.updateCount}>  
        Clicked {this.state.count} times  
      </button>  
    );  
  }  
}
```

- ✓ should not be accessed from child components
- ✓ are mutable

Let's go practicing

git clone <https://github.com/m4g4/p-i-react-training-2018.git>

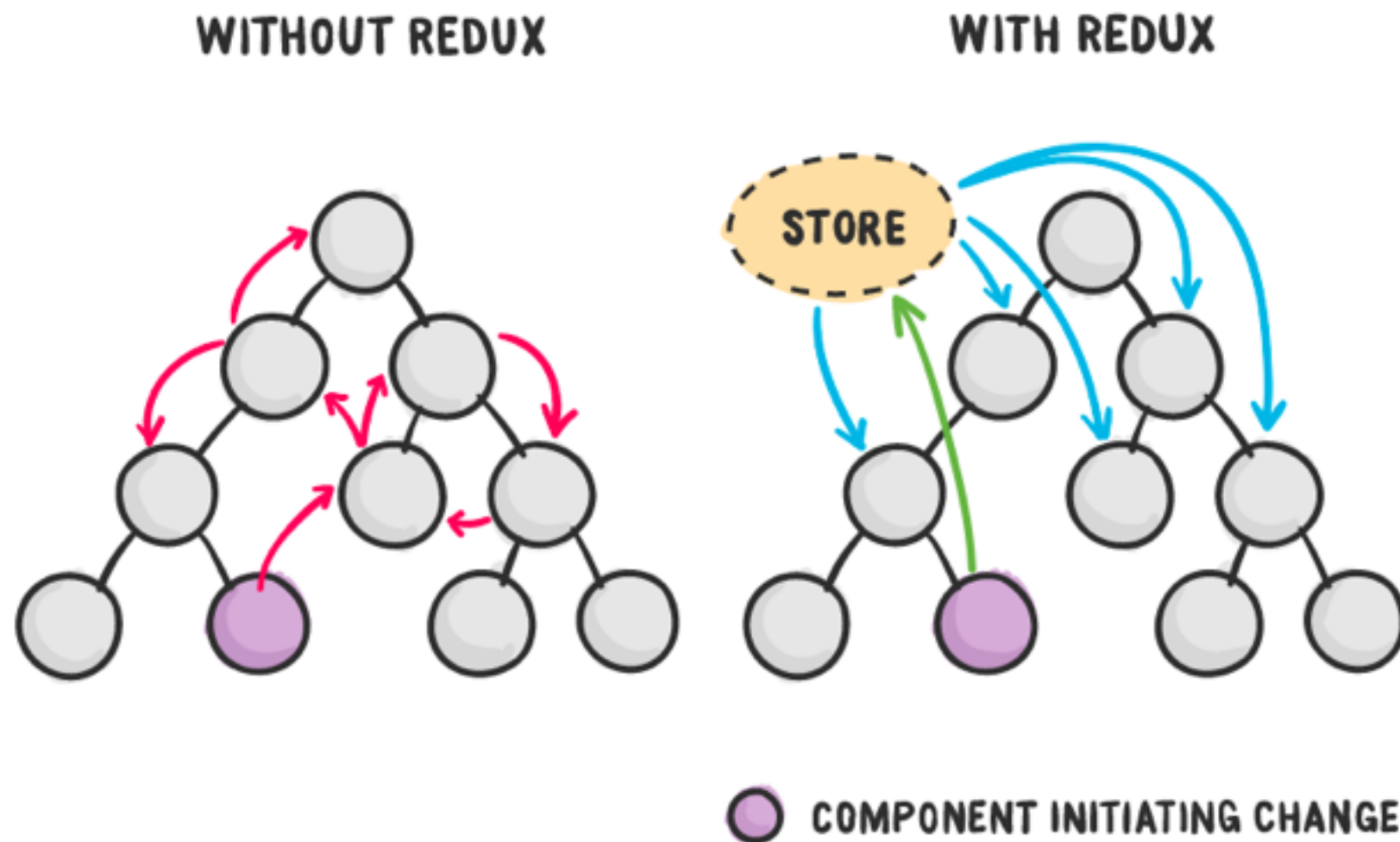
The full lifecycle of React component



img: <https://hackernoon.com/reactjs-component-lifecycle-methods-a-deep-dive-38275d9d13c0>

Mobx / Redux libraries

Single Source of Truth



Redux lifecycle

Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->
Views -> Action Creators -> Dispatcher -> Stores ->



Code linting

analysing source code to flag programming errors, bugs, stylistic errors, and suspicious constructs



<https://eslint.org/docs/rules/>



<https://flow.org/>

```
// @flow
function square(n: number): number {
    return n * n;
}


square("2"); // Error!
```

Debugging



- Chrome plugins
 - **JetBrains IDE Support** for debugging in WebStorm.
 - **React Developer Tools** for inspecting state, props and changes.
 - **MobX Developer Tools** for inspecting events in MobX.

LogaWeb React integration ...

JS  GWT

- **JSNI**

- <http://www.gwtproject.org/doc/latest/DevGuideCodingBasicsJSNI.html>
- will be removed with GWT 3

- **JS Interop**

- <http://www.gwtproject.org/doc/latest/DevGuideCodingBasicsJsInterop.html>

- **We have a GWT component**

- `com.piag.loga2.platform.ui.gwt.components.client.react.ReactWidget`

integration...

pom.xml (lweb)

```
<plugin>
  <artifactId>maven-war-plugin</artifactId>
  <configuration>
    <!-- exclude client package at runtime (has references to other client
         packages which are not included at runtime) -->
    <packagingExcludes>WEB-INF/classes/com/piag/**/client/**</packagingExcludes>

    <webResources>
      <resource>
        <directory>../../modules/logaweb/lwreact/src/main/react/build/static/js/</directory>
        <includes>
          <include>lweb.main.js</include>
        </includes>
        <targetPath>LWebMain</targetPath>
      </resource>
      <resource>
        <directory>../../modules/logaweb/lwreact/src/main/react/build/static/css/</directory>
        <includes>
          <include>lweb.main.css</include>
        </includes>
        <targetPath>LWebMain/css</targetPath>
      </resource>
    </webResources>
  </configuration>
</plugin>
```


... integration ...

main.jsp

```
<script>
    var L2MainContext = '<%=layoutSession.getMainWebContextLinkWithoutDefaultParams("/LWebMain") %>';
    var mainContext = '<%=WebContextLoader.getMainWebContext()%>';
    var webstylesContext = '<%=WebContextLoader.getWebStyleContext()%>';
    var tenant = '<%= tenant == null ? "" : tenant %>';
    var logoutLink = '<%= layoutSession.getLogoutLink()%>' + ((tenant == '') ? '' : '&dbname=' + tenant);
    var maskId = '<%= maskId == null ? "" : maskId %>';
    var hideMenu = '<%= hideMenu == null ? "" : hideMenu %>';
    var pp = '<%= System.getProperty("devel.payload.plain") %>';
    var logoutDisabled = '<%= com.piag.loga2.app.web.server.auth.sso.SSOResolver.isLogoutDisabled(request) %>';
</script>

<link rel="shortcut icon" href="<%=layoutSession.getMainWebContextLinkWithoutDefaultParams("/LWebMain/images/favicon.ico")%>" type="image/x-ico">

<script type="text/javascript" src="<%= layoutSession.getMainWebContextLinkWithoutDefaultParams("/LWebMain/LWebMain.nocache.js")%>"></script>
<%= GwtJavaScriptsLoader.getReactApp(layoutSession.getMainWebContextLinkWithoutDefaultParams("/LWebMain/"), "lweb.main") %>
</head>

<body class="page page-default">

</body>
</html>
```


... integration

GWTJavascriptLoader.java

```
public static String getReactApp(String baseUrl, String module, int port) {
    StringBuilder result = new StringBuilder();
    if (Boolean.valueOf(System.getProperty(WEBPACK_DEV_SYSTEM_PROPERTY))) {
        result.append("<script>__REACT_DEVTOOLS_GLOBAL_HOOK__ = parent.__REACT_DEVTOOLS_GLOBAL_HOOK__</script>\n");
        String webpackDevServer = SharedConstants.REACT_DEV_URL_NO_PORT + ":" + port + "/";
        result.append(getScriptTag(webpackDevServer, scriptPath: "webpack-dev-server.js", extraAttr: ""));
        result.append(getScriptTag(webpackDevServer, scriptPath: "static/js/bundle.js", extraAttr: ""));
        result.append("<script>window." + WEBPACK_DEV_SYSTEM_PROPERTY + "=true;</script>\n");
    } else {
        result.append(getScriptTag(Strings.nullToEmpty(baseUrl), scriptPath: module + ".js", extraAttr: ""));
        result.append(getCssLinkTag(Strings.nullToEmpty(baseUrl), scriptPath: "css/" + module + ".css"));
    }
    return result.toString();
}
```

DEV MODE

Loads sources from NodeJs server

PRODUCTION MODE

Loads js/css from Wildfly server

react-shared-lib

- P&I react shared library
- similar to sgwt-components in GWT
- For sharing components such as ComboBox, DatePicker or Table.

```
"dependencies": {  
  "react-shared-lib": "file:../../../../../platform/ui/react-shared-lib/build"  
},
```

- Confluence page:

<http://192.168.118.50/confluence/display/LII/React+shared+library>

React other topics

List and keys in React

<https://reactjs.org/docs/lists-and-keys.html>

...

What next?

Useful links

- Javascript:
 - https://www.youtube.com/playlist?list=PLoYCgNOIyGACDQLaThEEKBAIgs4OIUGif&src_vid=MhkGQAoc7bc
- React:
 - <https://www.youtube.com/watch?v=MhkGQAoc7bc&list=PLoYCgNOIyGABj2GQSIDRjgvXtqfDxKm5b>
 - mainly first 5 lectures, #8, #14
 - Best practices: <https://engineering.musefind.com/our-best-practices-for-writing-react-components-dec3eb5c3fc8>
- Redux:
 - Very good basics: <https://egghead.io/courses/getting-started-with-redux>
- Flow: <https://flow.org/en/>