[Lesson 3]

[DAN.IT EDUCATION]

[Что мы рассмотрим:]

- propTypes и defaultProps
- Жизненный цикл компонентов
- React Router v4, основы роутинга
- this.props.match, this.props.location, this.props.history

[DAN.IT EDUCATION]

defaultProps

```
// Определение значений по умолчанию для props:
Greeting.defaultProps = {
  name: 'Stranger'
};

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

[defaultProps]

```
class Greeting extends Component {
     static defaultProps = {
    name: 'Stranger'
  };

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

[DAN.IT] EDUCATION

[Prop-types]

```
npm i prop-types -S
```

```
import React from 'react';
import PropTypes from 'prop-types';

class MyComponent extends React.Component {
  render() {
    // ... do things with the props
  }
}

MyComponent.propTypes = {
     ...
}
```

[DAN.IT] EDUCATION

[Prop-types]

```
import React from 'react';
import PropTypes from 'prop-types';

class MyComponent extends React.Component {
  static propTypes = {
     ...
  }

  render() {
    // ... do things with the props
  }
}
```

[Prop-types]

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

// Anything that can be rendered: numbers, strings, elements or an array
// (or fragment) containing these types.
optionalNode: PropTypes.node,

// A React element.
optionalElement: PropTypes.element,

// You can also declare that a prop is an instance of a class. This uses
// JS's instanceof operator.
optionalMessage: PropTypes.instanceOf(Message),

// You can ensure that your prop is limited to specific values by treating
// it as an enum.
optionalEnum: PropTypes.oneOf(['News', 'Photos']),
```

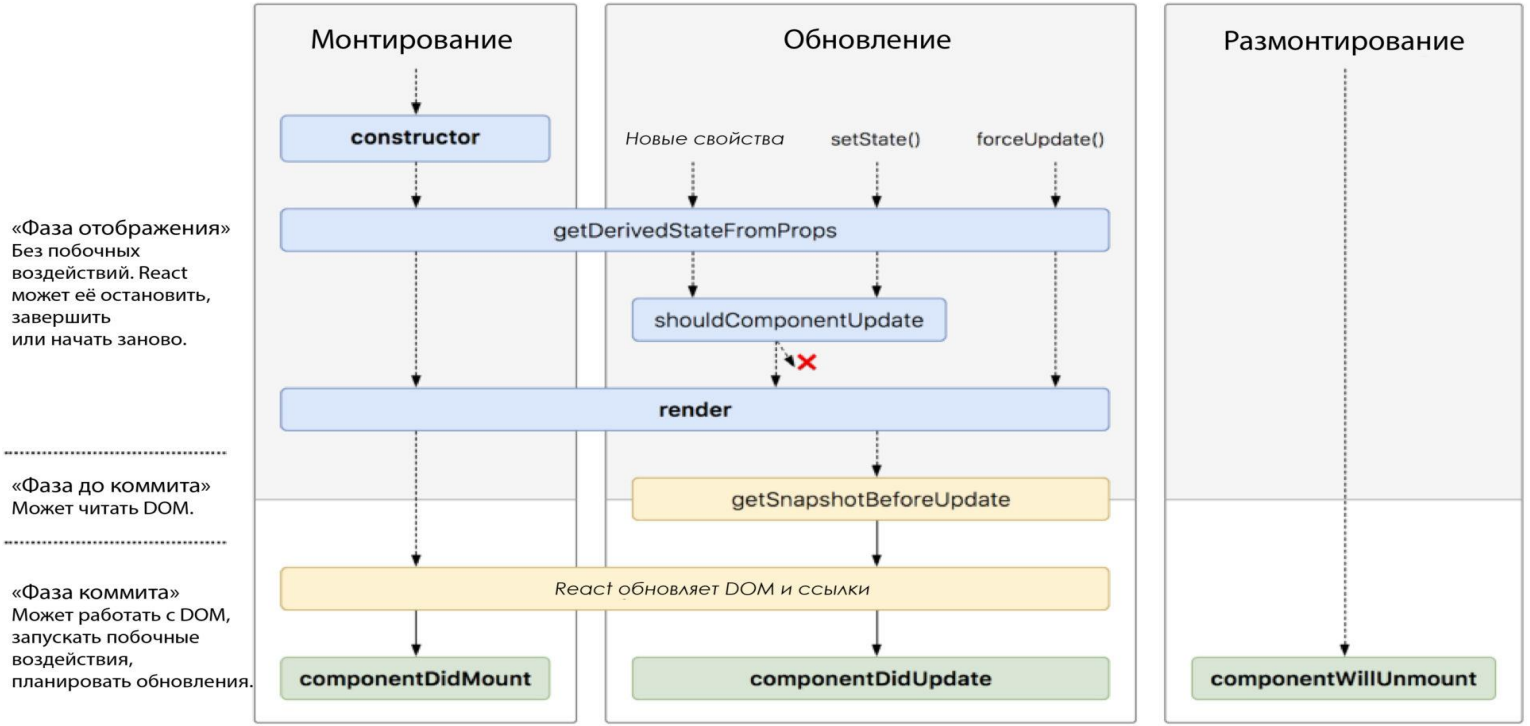
[DAN.IT] EDUCATION

[Prop-types]

```
class MyComponent extends Component {
  render() {
  // Если в props.children будет более одного элемента возникнет предупреждение.
    const children = this.props.children;
    return (
      <div>
        {children}
      </div>
    );
  }
}

MyComponent.propTypes = {
  children: React.PropTypes.element.isRequired
};
```

Жизненный цикл компонентов



[React Router v4]

```
npm i react-router-dom --S
```

```
import { BrowserRouter } from 'react-router-dom';

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

[DAN.IT] EDUCATION

[React Router v4]

```
import { BrowserRouter } from 'react-router-dom';

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

[React Router v4]

```
import { Route } from 'react-router-dom’;

...

<Route path="/" component={Body} />
<Route path="/profile" component={Profile} />
```

[React Router v4]

```
import { Route } from 'react-router-dom’;
...

<Route path="/" component={Body} /><Route path='/page' render={(props) => (
  <Page {...props} data={extraProps}/>
)}/>
<Route path='/page' children={(props) => (
  props.match
    ? <Page {...props}/>
    : <EmptyPage {...props}/>
)}/>
```

this.props.match

url — сопоставляемая часть текущего location.pathname

path — путь в компоненте Route

isExact — path в Route === location.pathname

params — объект содержит значения из path которые возвращает модуль path-to-regexp

[this.props.history]

Каждый Router создает объект this.props.history который хранит путь к текущему this.props.location и перерисовывает интерфейс сайта когда происходят какие то изменения пути.

```
action: "PUSH"
block: function block()
createHref: function createHref()
go: function go()
goBack: function goBack()
goForward: function goForward()
length: 2
listen: function listen()
location: Object { pathname: "/signin", search: "", key: "1ffgxt", … }
push: function push()
replace: function replace()
<prototype>: Object { … }
```

[DAN.IT] EDUCATION

React Router v4

```
import { Switch, Route } from 'react-router-dom'

<Switch>
 <Route exact path='/' component={Home}/>
 <Route path='/roster' component={Roster}/>
 <Route path='/schedule' component={Schedule}/>
</Switch>
```

[React Router v4]

```
import { Switch, Route } from 'react-router-dom'

<Switch>
 <Route exact path='/' component={Home}/>
 <Route path='/contacts' component={ContactsBook}/>
 <Route path='/contacts/:id' component={ContactInfo}/>
</Switch>
```

[React Router v4]

```
import { Switch, Route } from 'react-router-dom'

const Contacts = () => (
 <div>
  <h2>Adress book</h2>
  <Switch>
   <Route exact path='/contacts' component={ContactsList}/>
   <Route path='/contacts/:id' component={ContactFullInfo}/>
  </Switch>
 </div>
)
```

[DAN.IT] EDUCATION

[React Router v4]

```
import { NavLink } from 'react-router-dom’

...

<NavLink to="/" activeClassName="active">Dashboard</NavLink>

...
```