

React

- `npx create-react-app <smth>`
- `cd <smth>`
- `npm start`

Структура проекта

- actions
- **components**
- **constants**
- epics
- **routes**
- resources
- resolvers
- **index.js**

Разбивайте страницу
на компоненты!

Состояние компонента

```
class App extends Component {  
  constructor(props) {  
    super(props);  
    this.state = {} //this.setState()!  
  }  
  render() {  
    return <smth>;  
  }  
}  
  
export default App;
```

Stateless компонент

```
(export) const App = ({param1, param2}) => {  
  const smth = <myComponent param2>  
  return { param1 && smth }  
}
```

```
(export) const App = ({param1, param2}) => (  
  <div>  
    {param1}  
  </div>  
)
```

Стрелочные функции

`{param1, param2, ..., paramN} => {statements}`

`{param1, param2, ..., paramN} => expression`

`{param1, param2, ..., paramN} => {return expression;}`

`(singleParam) => {statements}`

`singleParam => {statements}`

`() => {statements}`

`() => expression`

- Зачем нужна типизация?
- Насколько уровней вложенности делать components?
- Зачем нужна нормализация?
 - Где делать нормализацию данных?

В следующих сериях

- routes: что это, как установить и чего бояться
- функции, методы, onClick и прочее

Дома:

- Перенести верстку на React
 - Структура директории
 - Codestyle