React Base

Содержание

1 Знакомство

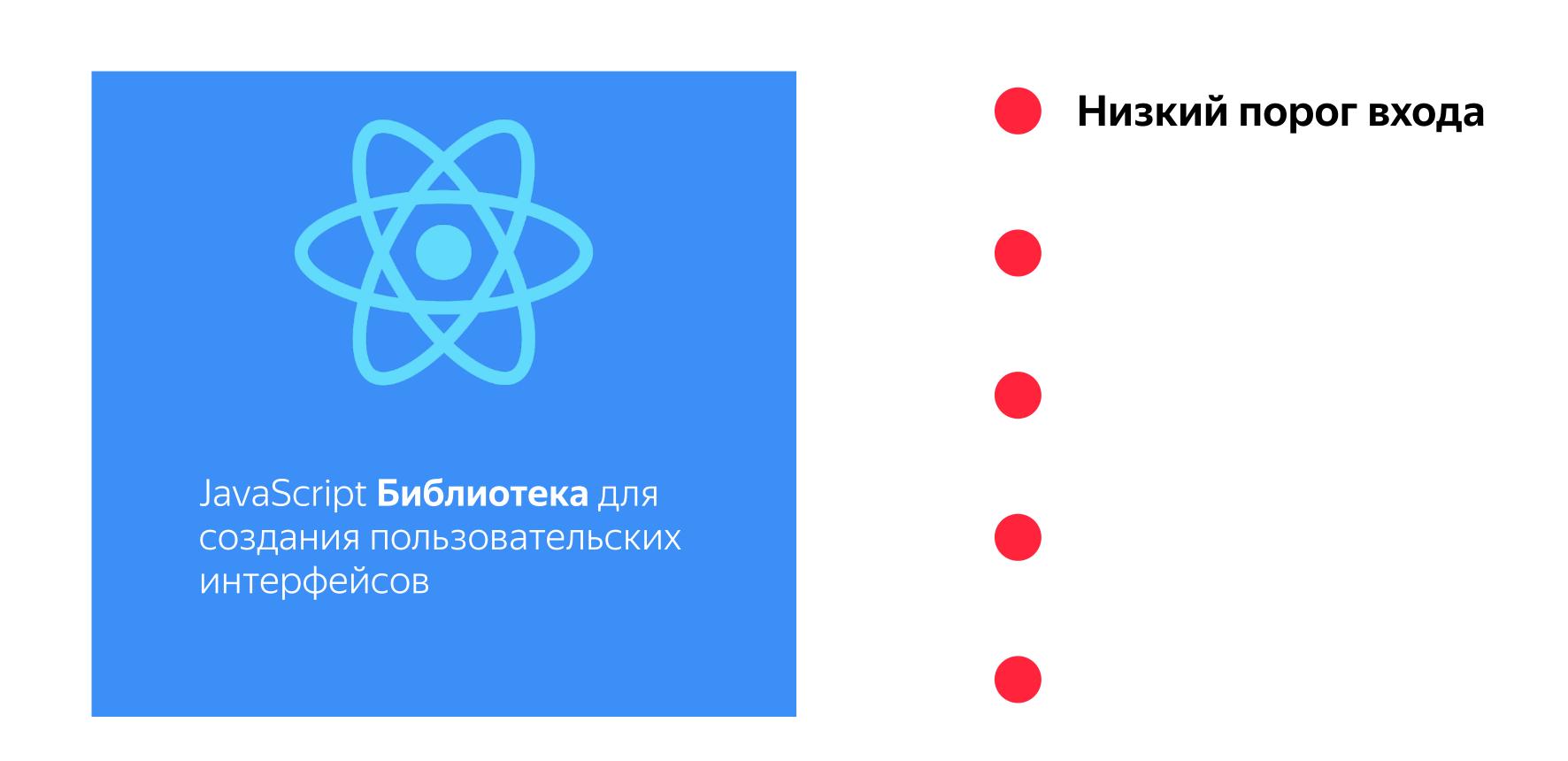
2 Основы основ

Содержание

1 Знакомство

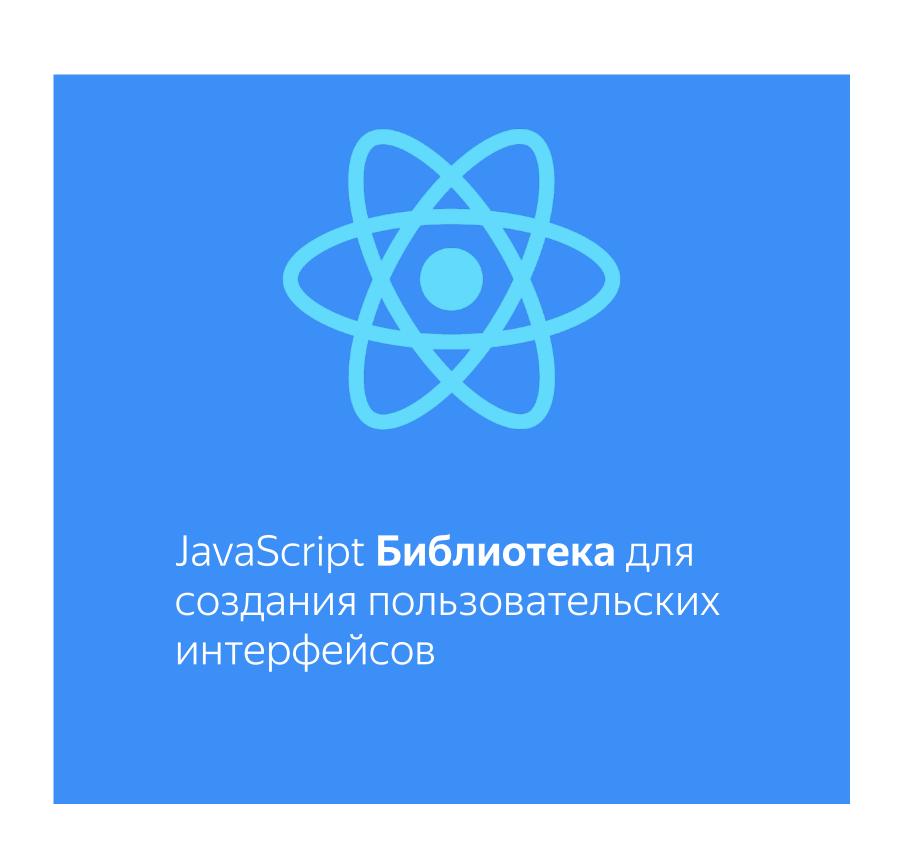
2 Основы основ







- Низкий порог входа
- Легко добавить в уже готовый проект



- Низкий порог входа
- Пегко добавить в уже готовый проект
- Декларативный подход



Императивный

Как делать ?!

Императивный

Как делать ?!

Декларативный

Что делать ?!

Императивный

Как делать ?!

Что в React?

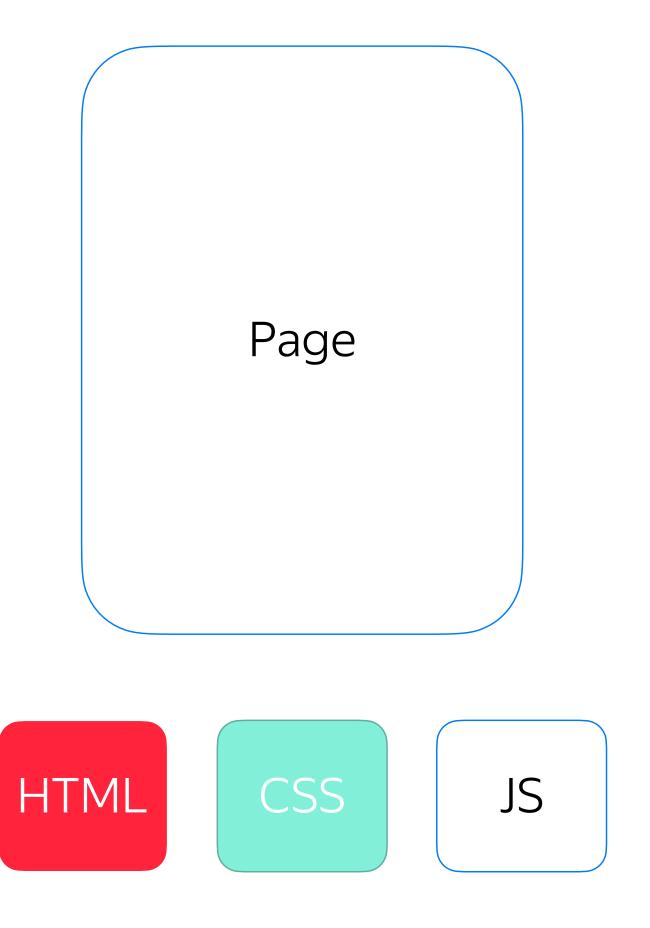
В React используется декларативный подход, что позволяет упростить процесс создания интерфейсов.

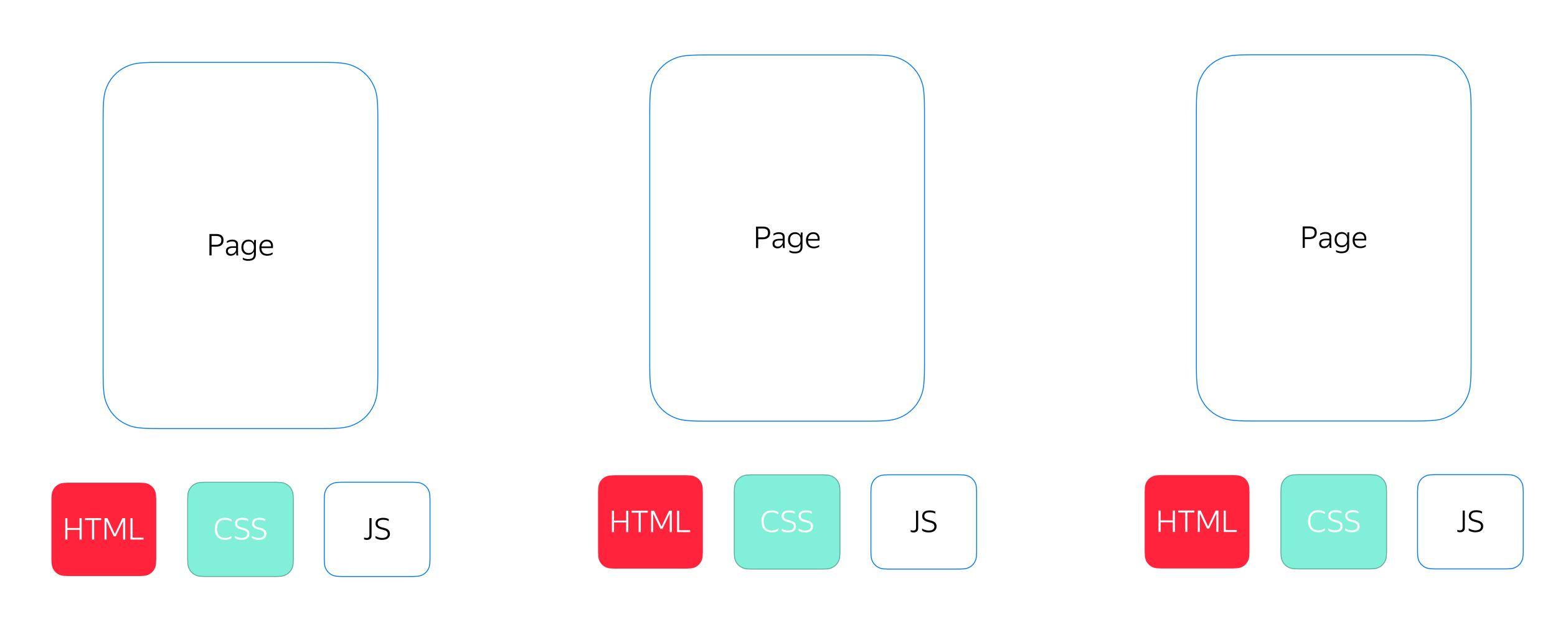
Декларативный

Что делать ?!

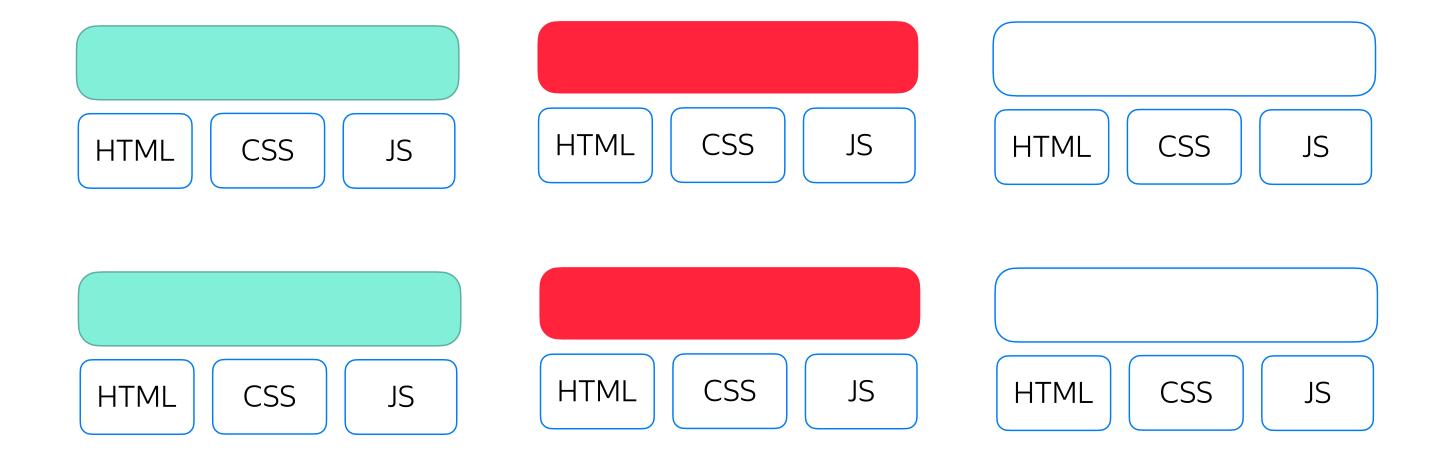


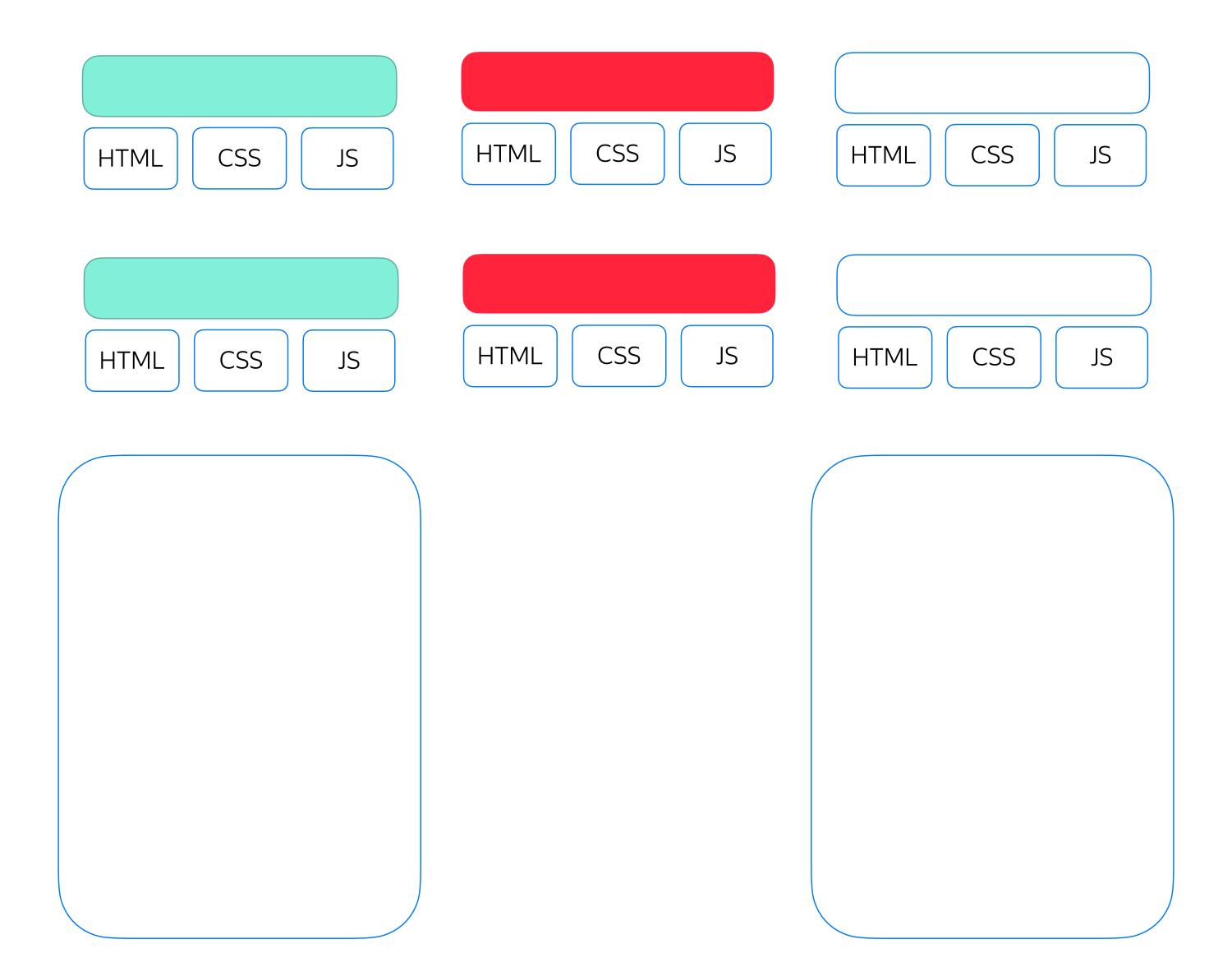
- Низкий порог входа
- Пегко добавить в уже готовый проект
- ____ Декларативный подход
- **Компонентный подход**

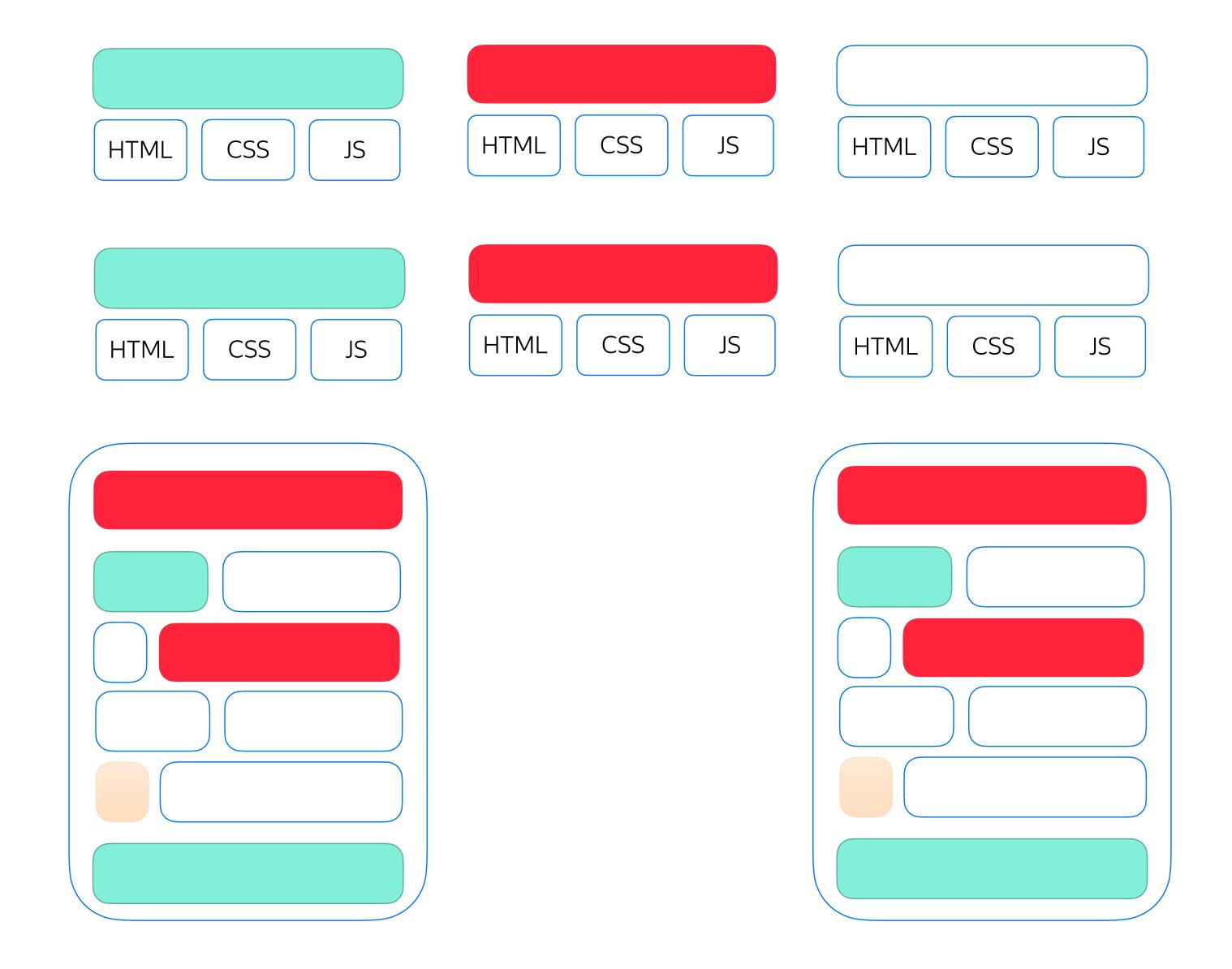


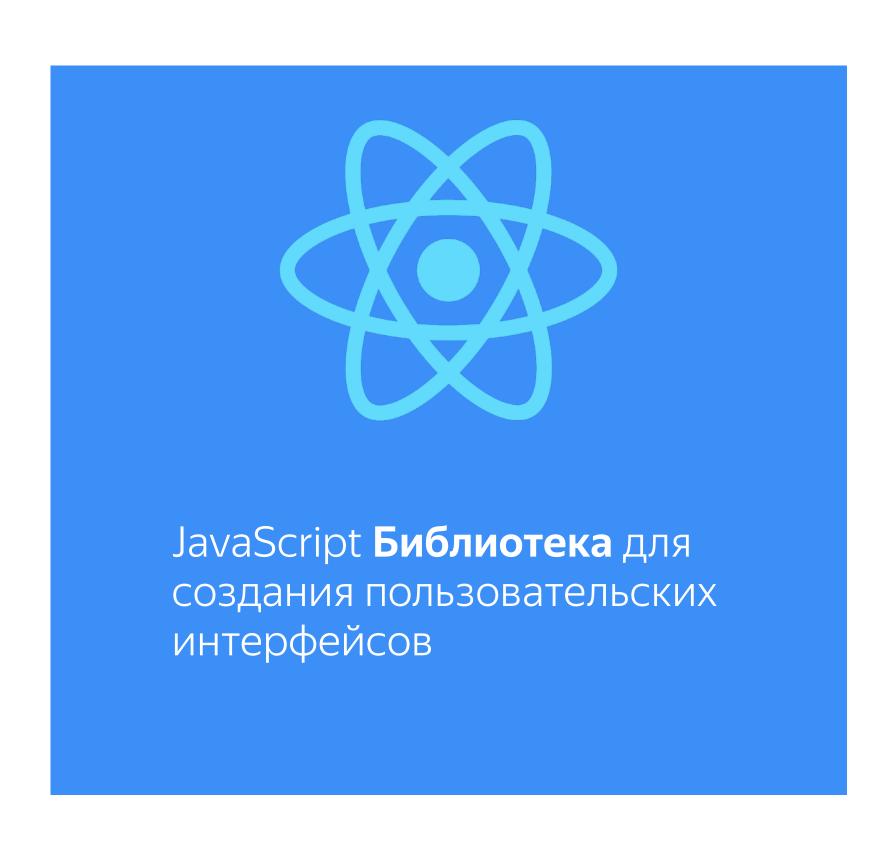






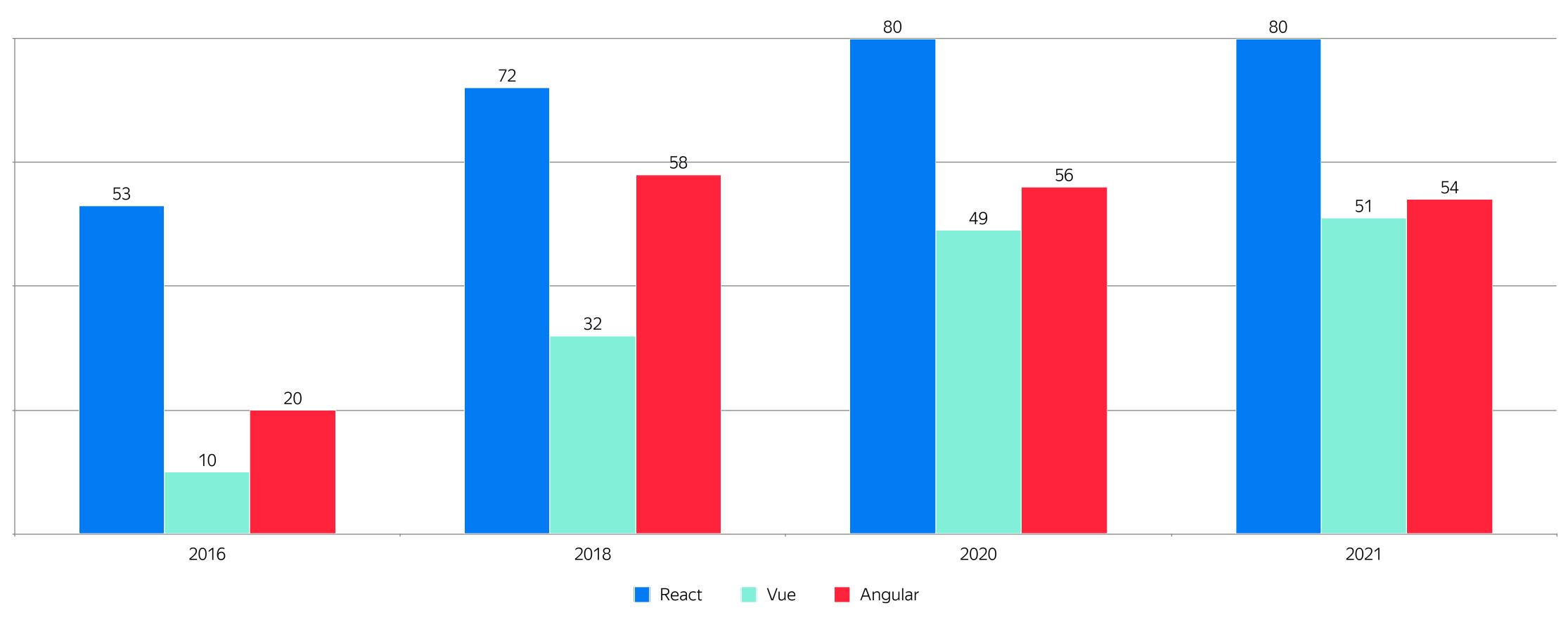






- Низкий порог входа
- Пегко добавить в уже готовый проект
- _____ Декларативный подход
- Компонентный подход
- Один из лидеров по использованию

React usage by StateOfJS





- Низкий порог входа
- Пегко добавить в уже готовый проект
- ____ Декларативный подход
- Компонентный подход
- Один из лидеров по использованию

Содержание

1 Знакомство

2 Основы основ

Как начать использовать?

```
<script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>
<script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js" crossorigin></script>
```

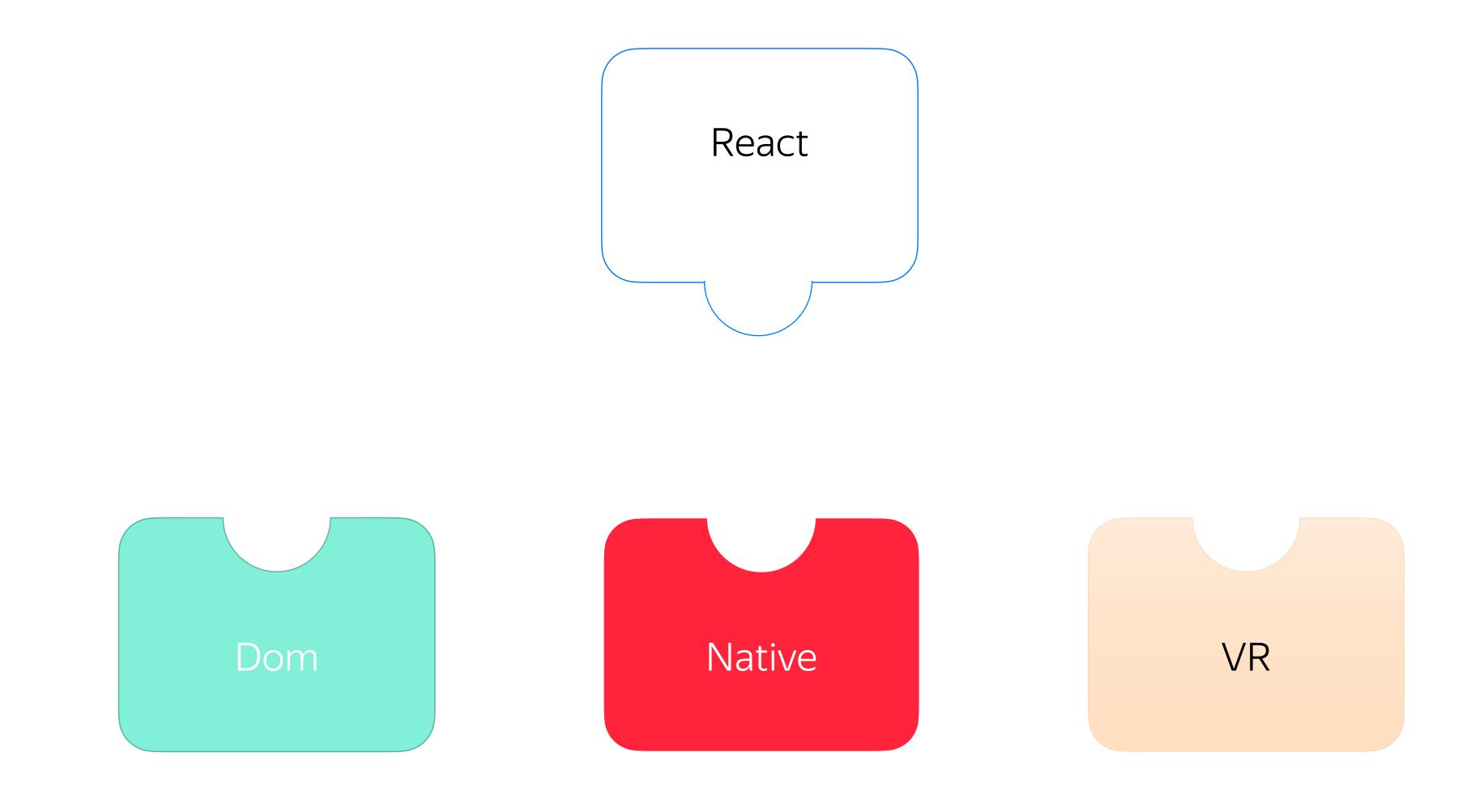
Апочему либы две???

React
Dom

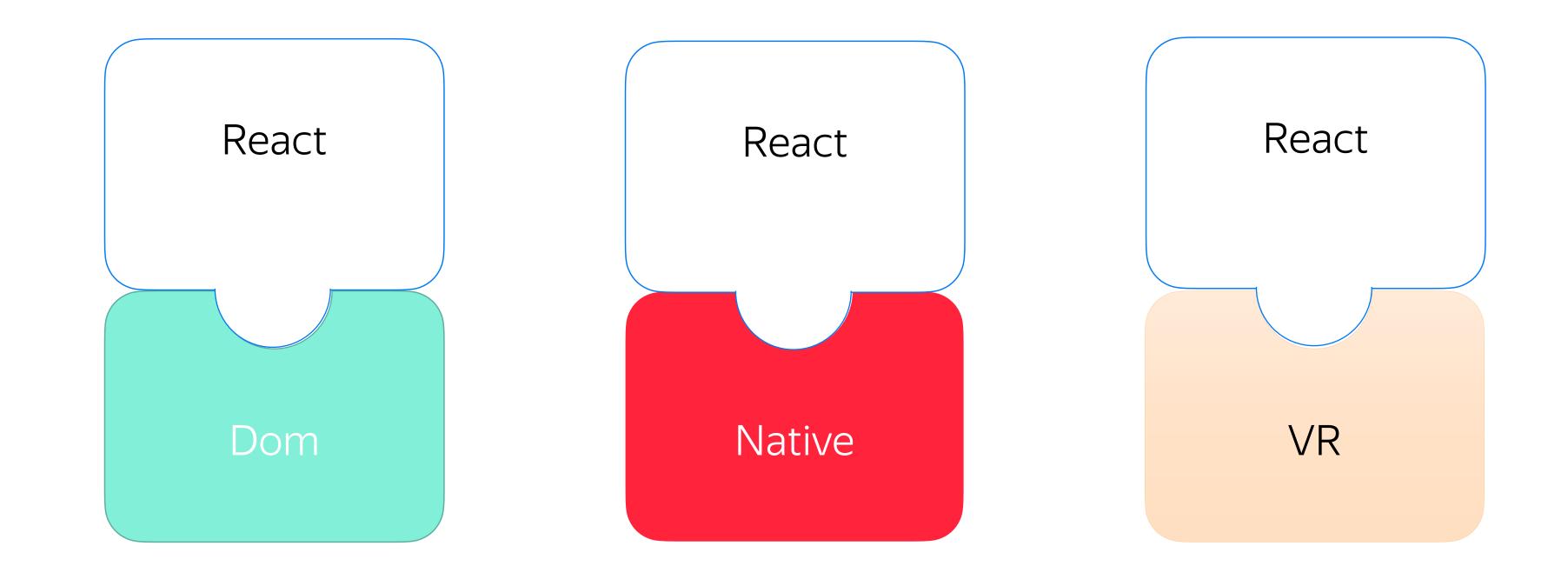
React Native

React VR

Апочему либы две???



Так удобнее

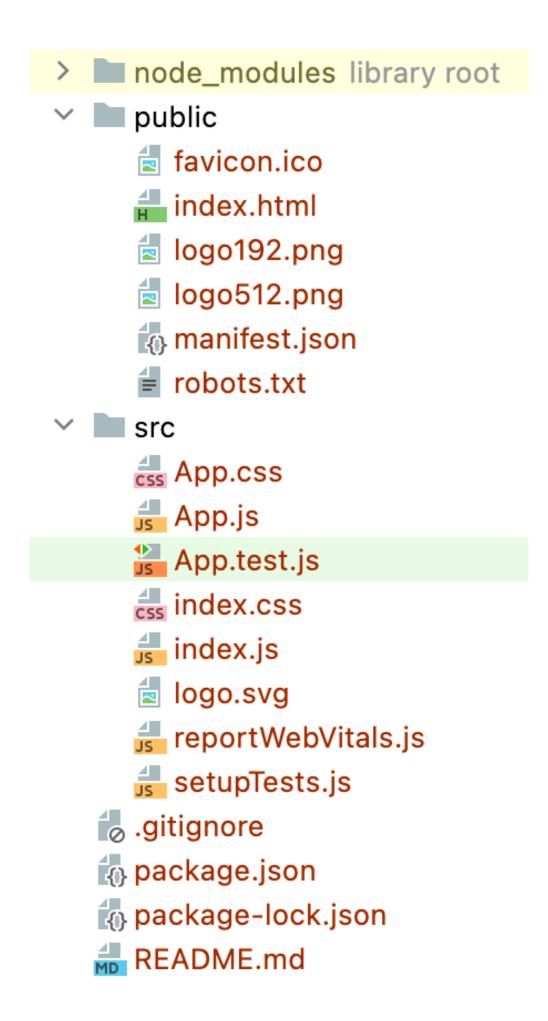


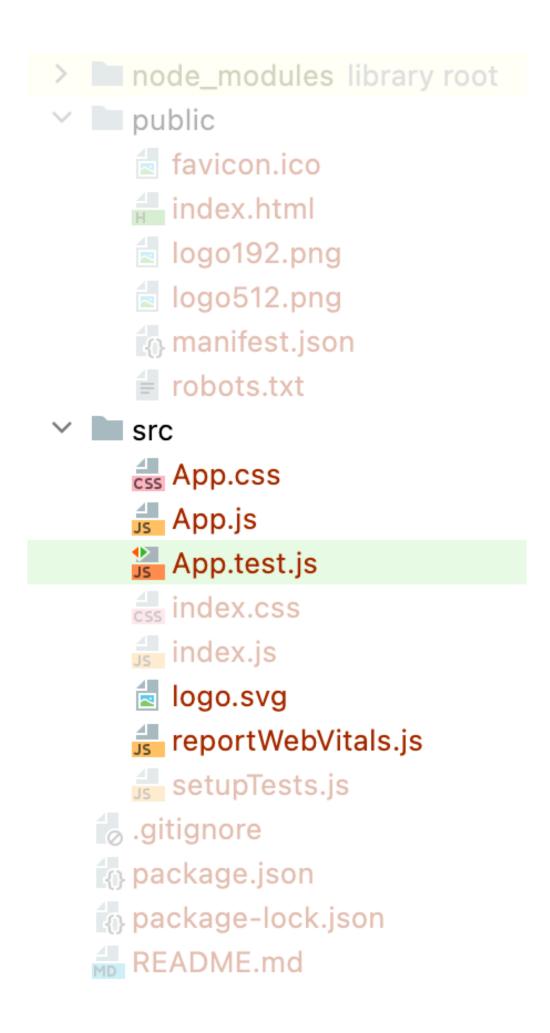
Как начать использовать?

```
<script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>
<script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js" crossorigin></script>
```

npx create-react-app my-app

Что мы в итоге получили?

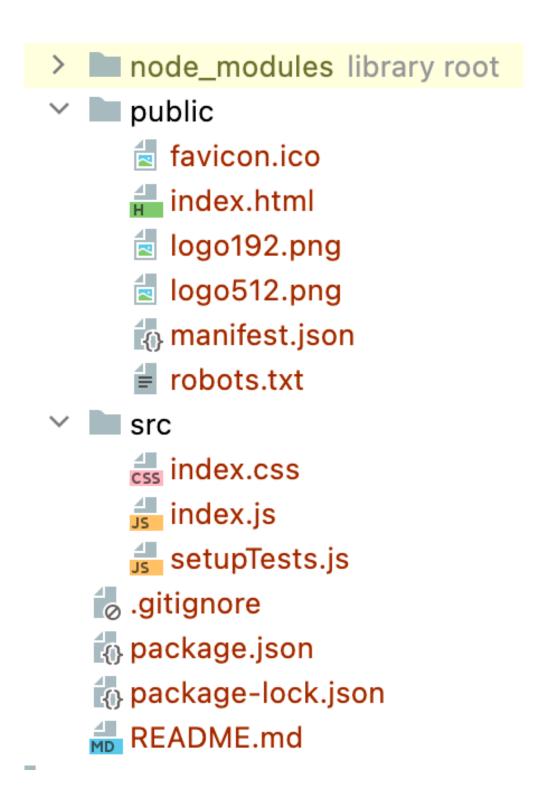




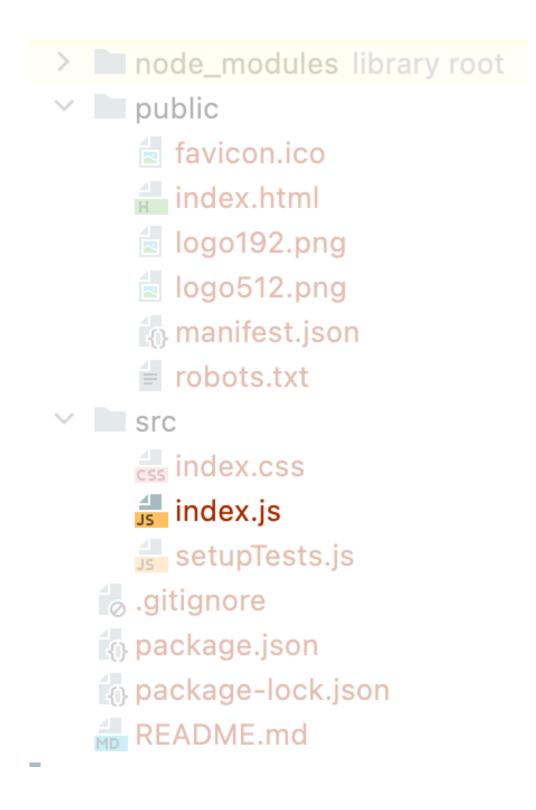
README.md

Стало Было > node_modules library root > node_modules library root ✓ ■ public ✓ ■ public favicon.ico favicon.ico index.html index.html logo192.png logo192.png logo512.png logo512.png manifest.json manifest.json robots.txt robots.txt ✓ src ✓ src App.css index.css 📇 App.js 🚛 index.js App.test.js setupTests.js index.css agitignore 🚚 index.js package.json logo.svg package-lock.json reportWebVitals.js README.md setupTests.js **a** .gitignore package.json package-lock.json

Откроем index.js



Откроем index.js



Откроем index.js

```
> node_modules library root
public
    favicon.ico
    index.html
    logo192.png
    logo512.png
    manifest.json
    robots.txt
✓ src
    index.css
    🚛 index.js
    setupTests.js
  agitignore
  package.json
  package-lock.json
  README.md
```

```
> node_modules library root
public
    favicon.ico
    index.html
    logo192.png
    logo512.png
    manifest.json
    robots.txt
∨ src
    index.css
    index.js
    setupTests.js
  .gitignore
  package.json
  package-lock.json
  README.md
```

```
> node_modules library root
public
    favicon.ico
    index.html
    logo192.png
    logo512.png
    manifest.json
    robots.txt
∨ src
    index.css
    index.js
    setupTests.js
  .gitignore
  package.json
  package-lock.json
  README.md
```

Было Стало

import React from 'react';

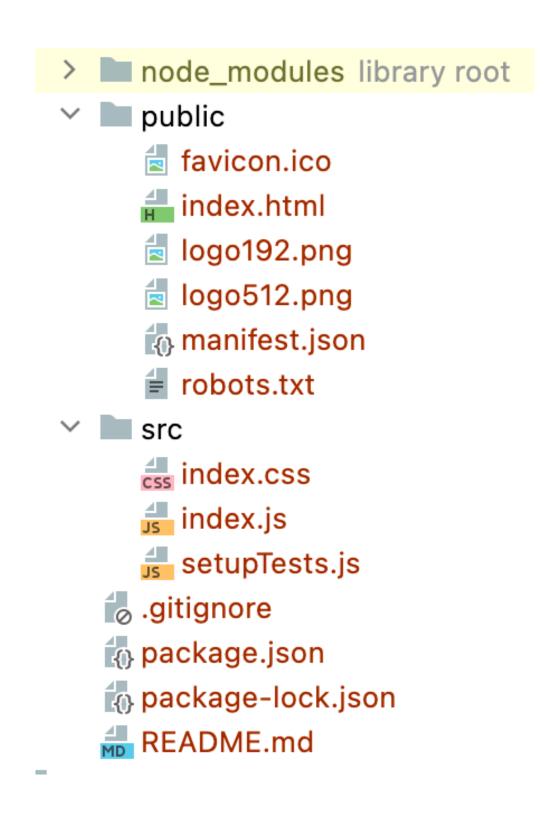
import './index.css';

root.render();

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
const root = ReactDOM.createRoot(document.getElementById( elementId: 'root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

```
import ReactDOM from 'react-dom/client';
const root = ReactDOM.createRoot(document.getElementById( elementId: 'root'));
```

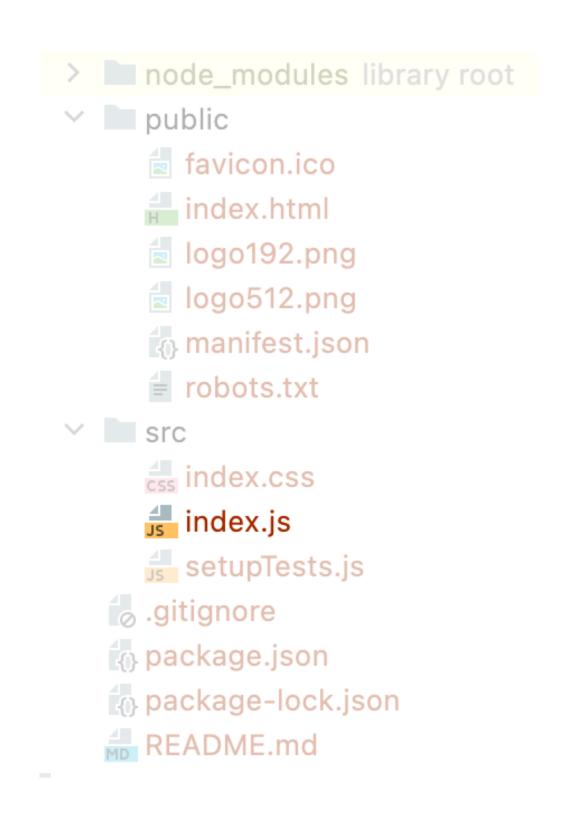
С чистого листа



Как запустить?

yarn/npm/pnpm start

Откроем index.js



```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';

const root = ReactDOM.createRoot(document.getElementById(elementId: 'root'));
root.render();
```

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';

const root = ReactDOM.createRoot(document.getElementById(elementId: 'root'));
root.render();
```

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';

const root = ReactDOM.createRoot(document.getElementById(elementId: 'root'));
root.render();
```

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';

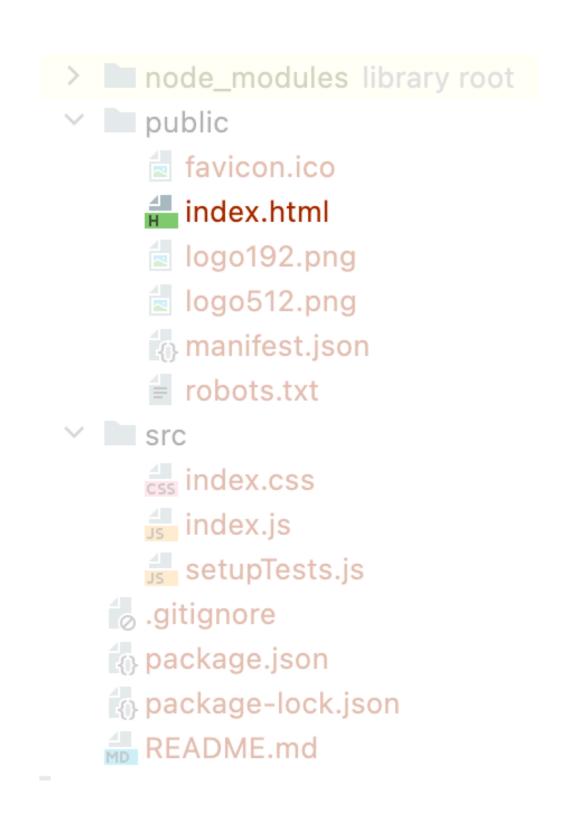
const root = ReactDOM.createRoot(document.getElementById(elementId: 'root'));
root.render();
```

А где этот root???

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';

const root = ReactDOM.createRoot(document.getElementById(elementId: 'root'));
root.render();
```

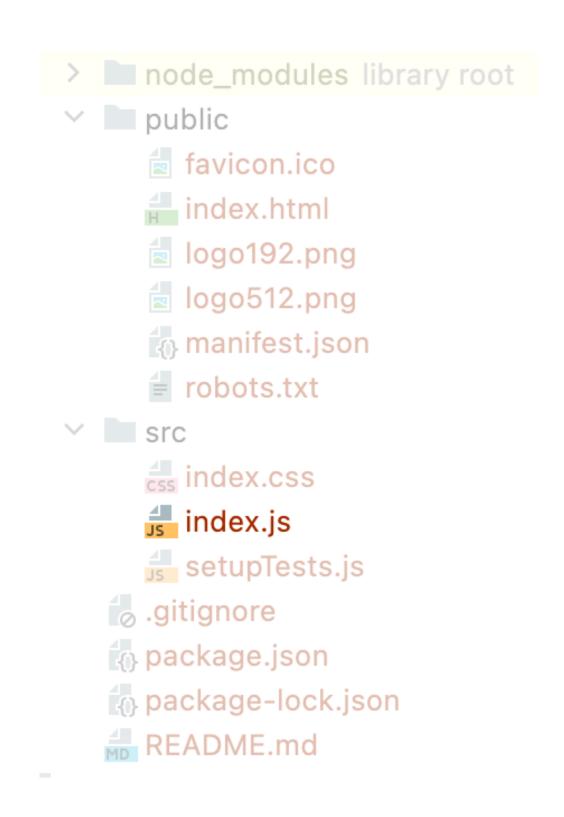
Откроем index.html



index.html

Тот самый root!

Возвращаемся в index.js



Теперь мы знаем где искать root

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';

const root = ReactDOM.createRoot(document.getElementById(elementId: 'root'));
root.render();
```

Создаем корень

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';

const root = ReactDOM.createRoot(document.getElementById(elementId: 'root'));
root.render();
```

Рендерим

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';

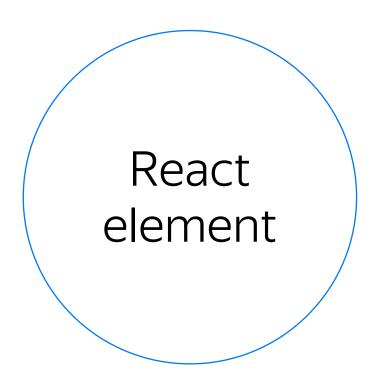
const root = ReactDOM.createRoot(document.getElementById(elementId: 'root'));
root.render();
```

А что дальше?

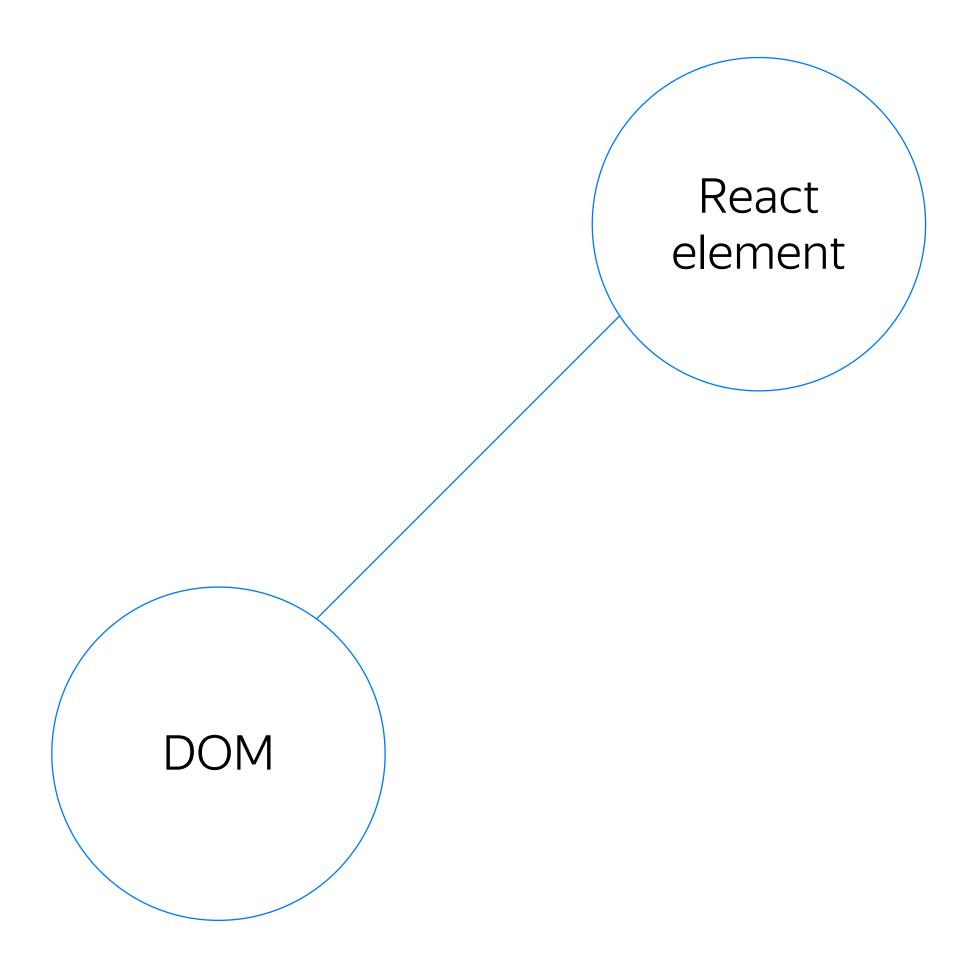
```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';

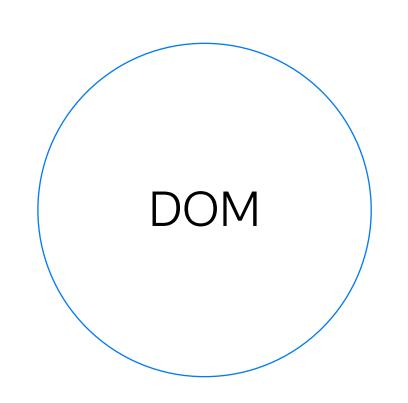
const root = ReactDOM.createRoot(document.getElementById(elementId: 'root'));
root.render();
```

Строительные блоки

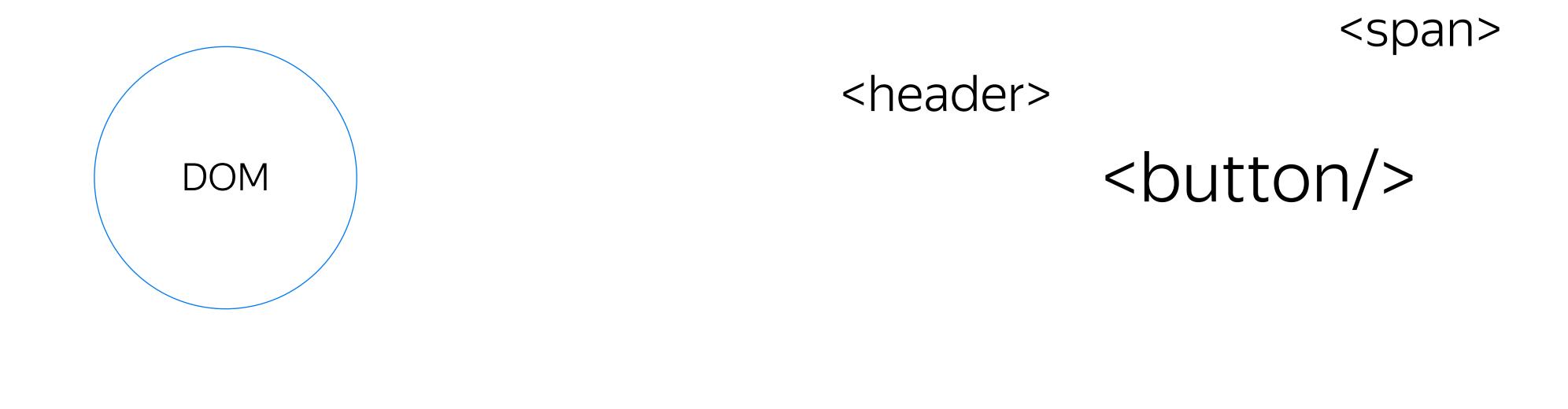


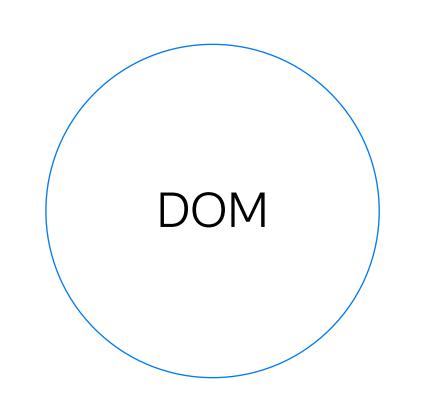
Строительные блоки

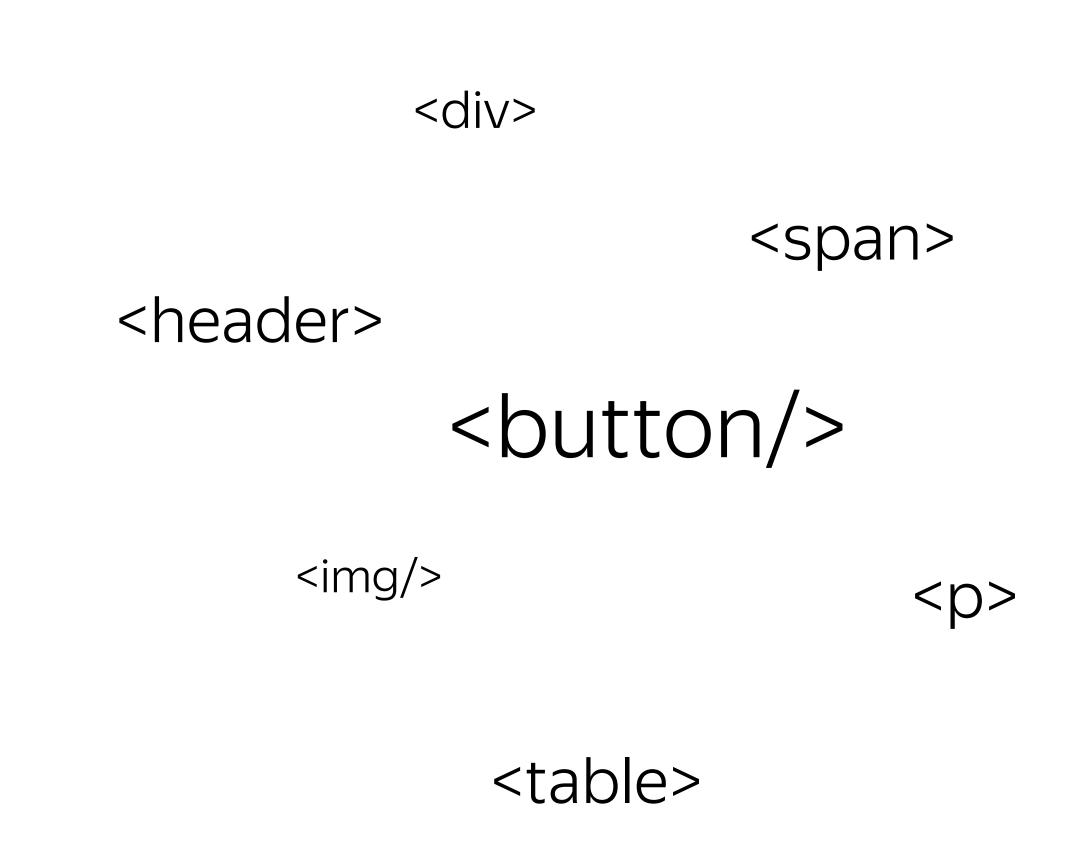


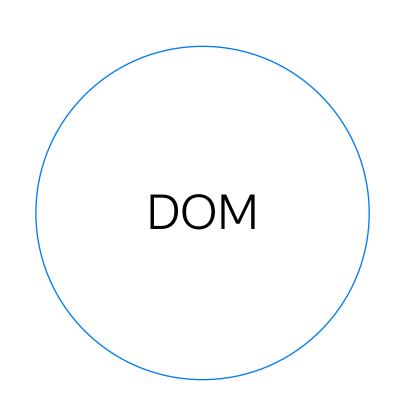


<button/>

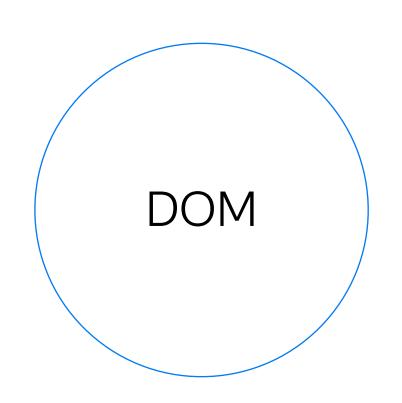




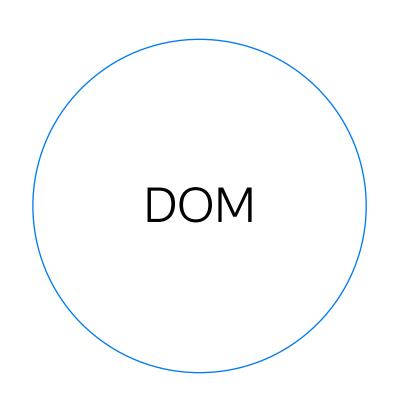


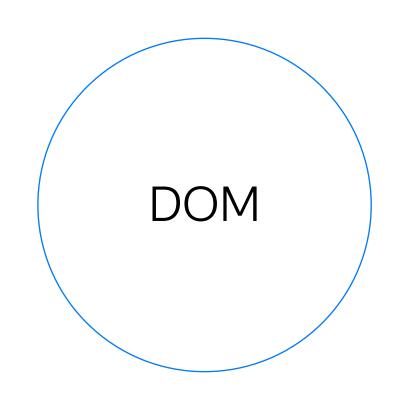


React.createElement('span')

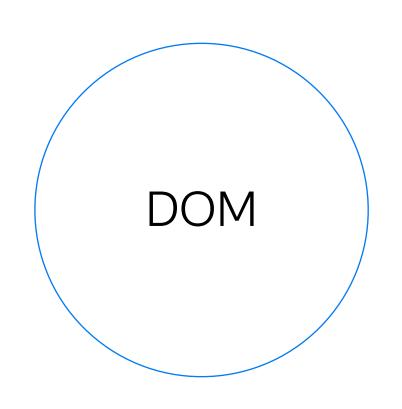


```
React.createElement(
  'span',
  { children: 'Hello Students!' },
)
```

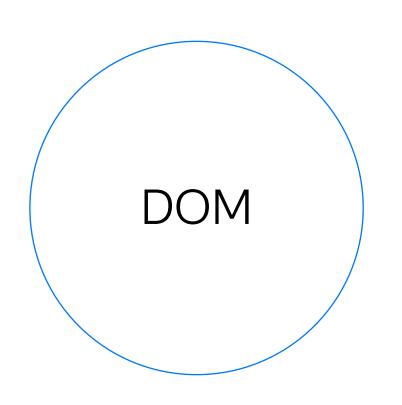




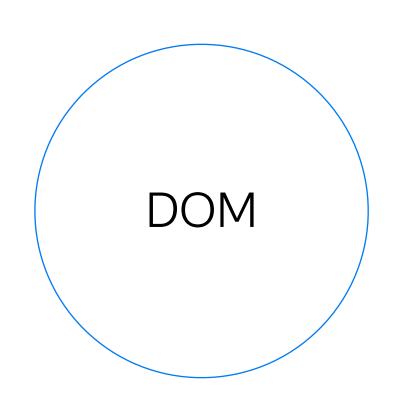
Hello Students!



```
<div id="root">
    <span>Hello Students!</span>
</div>
```

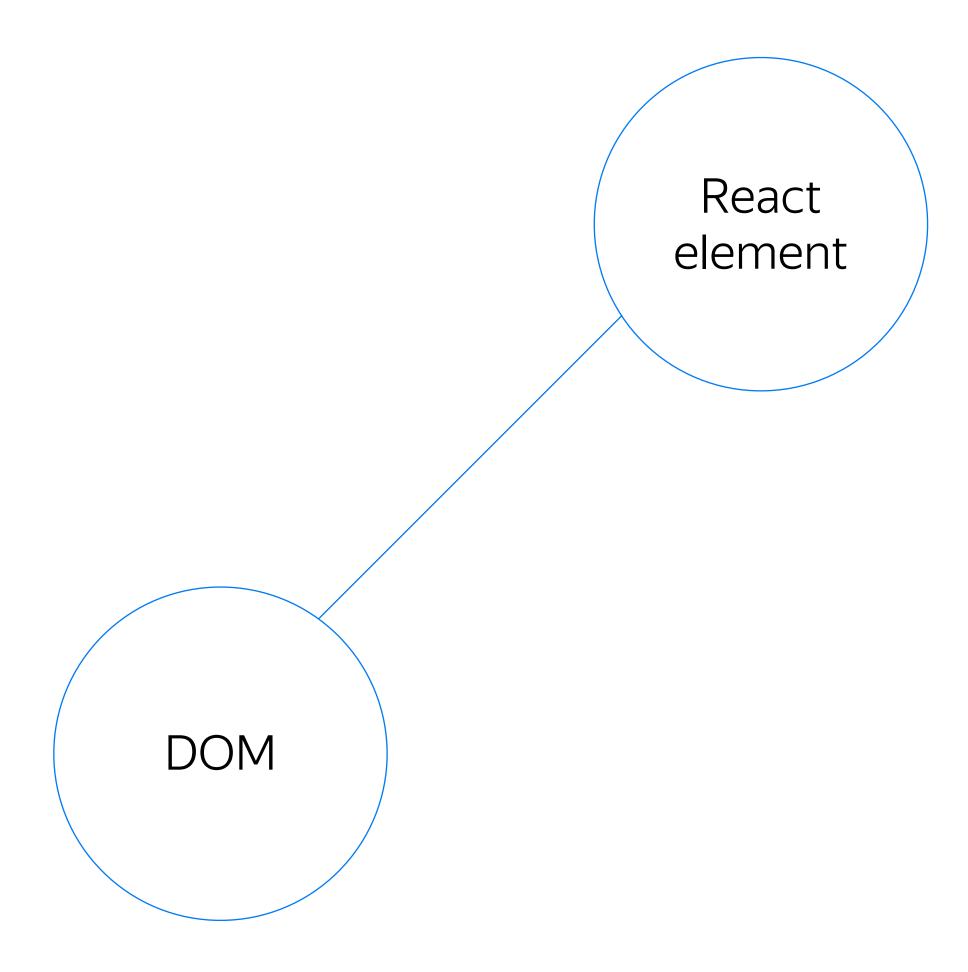


```
<div id="root">
    <span>Hello Students!</span>
</div>
```

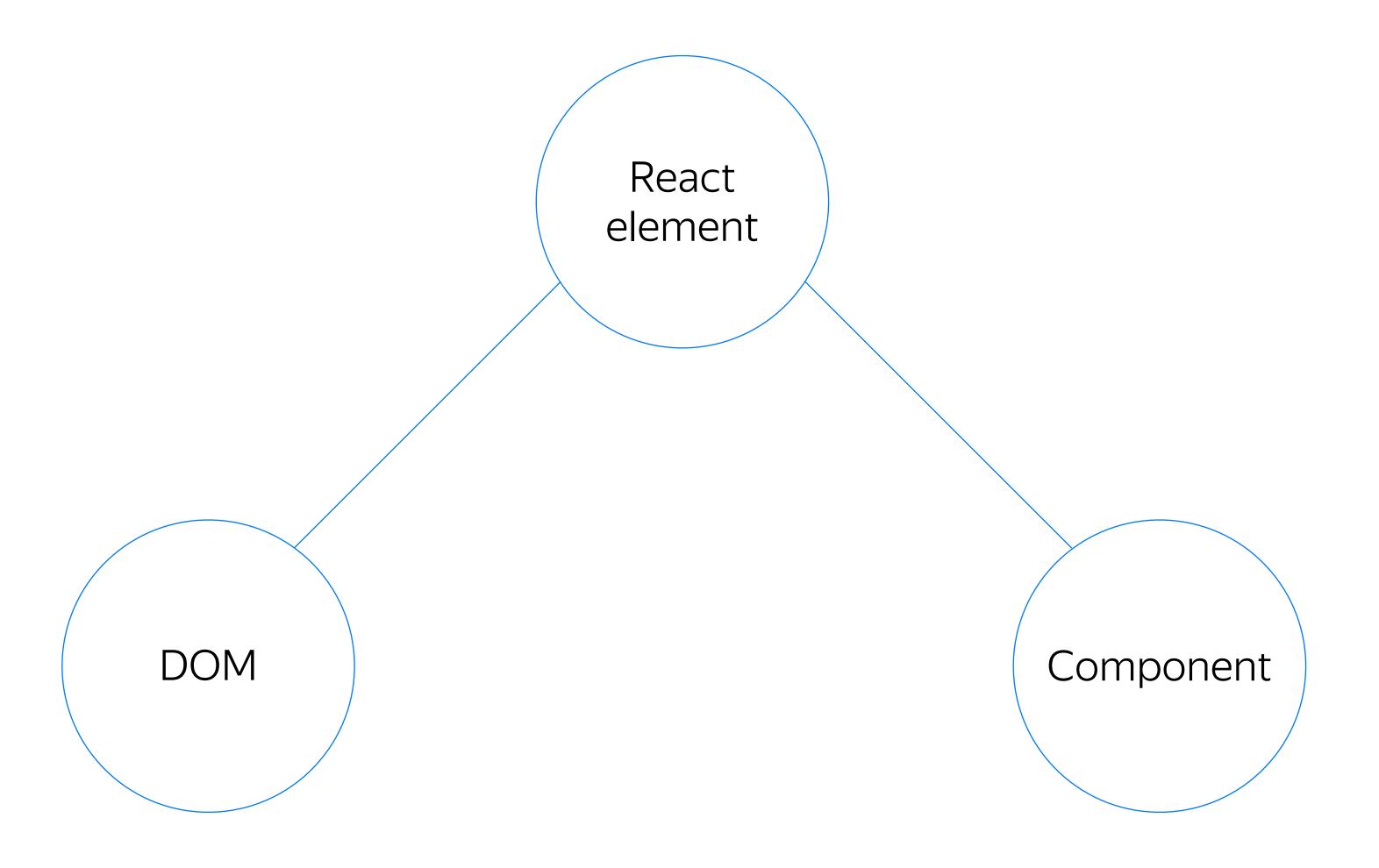


```
<div id="root">
    <span>Hello Students!</span>
</div>
```

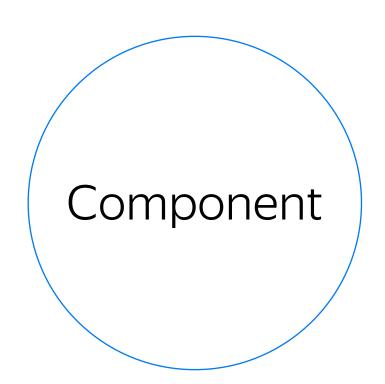
Строительные блоки



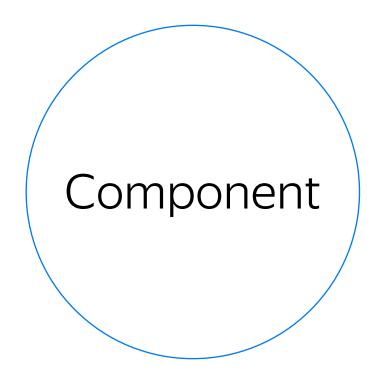
Строительные блоки



Component



Component



ElementDOM

ElementComponent

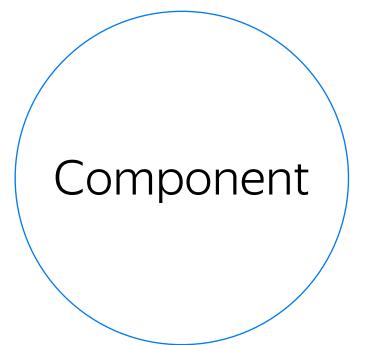
ElementDOM

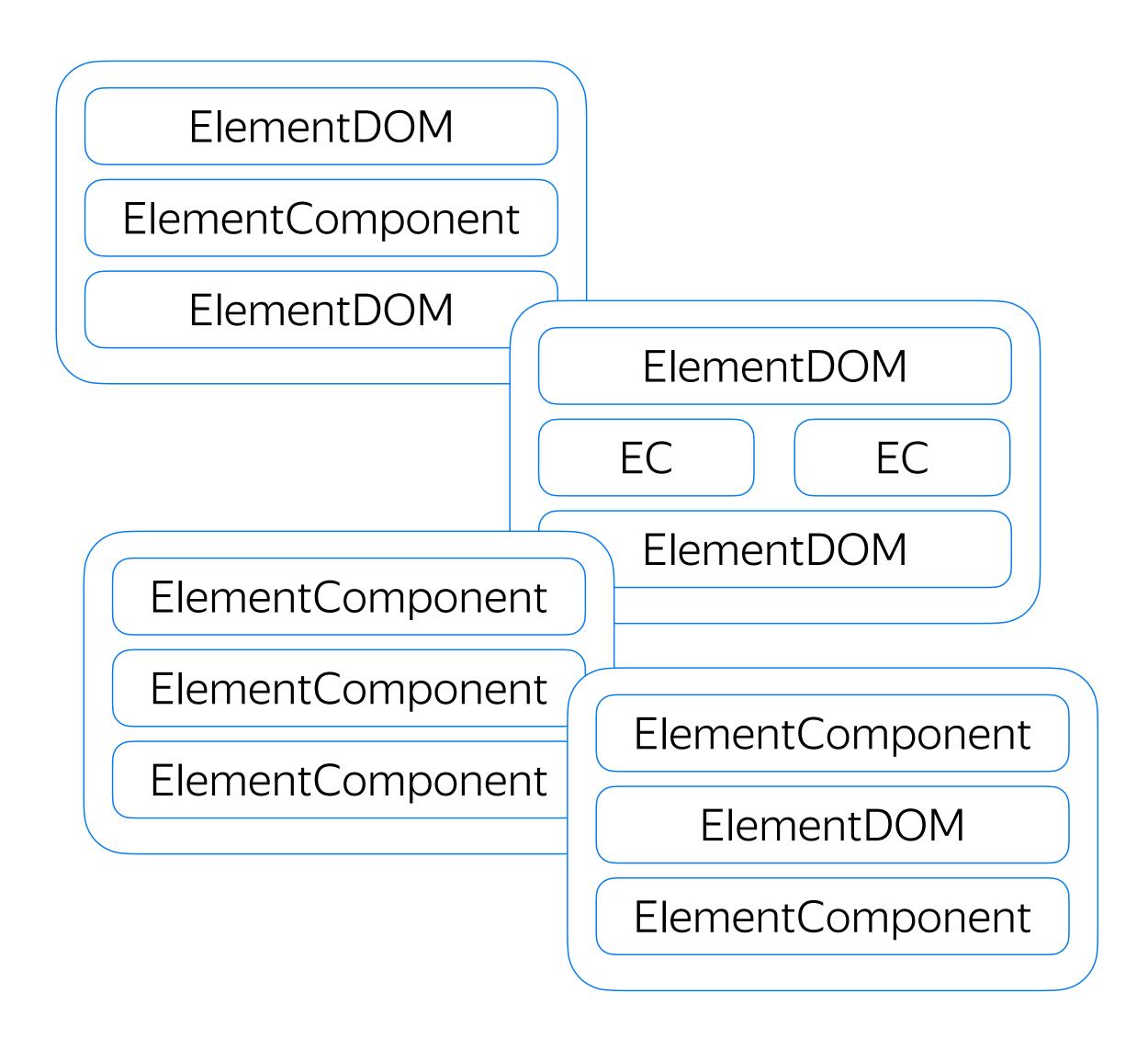
ElementComponent

ElementDOM

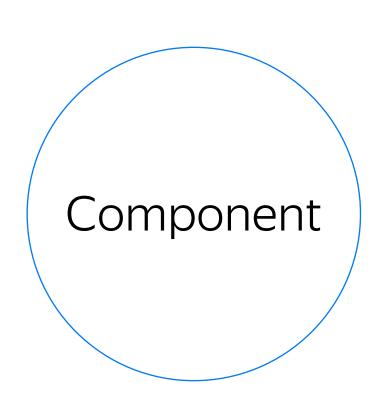
ElementComponent

Component

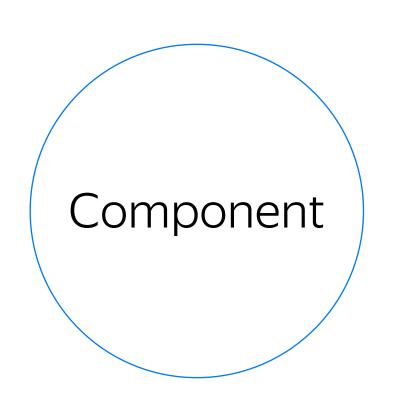




Class Component

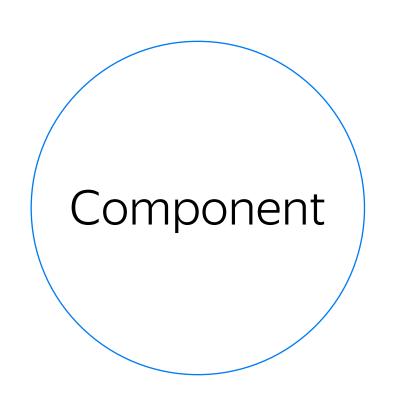


Function Component

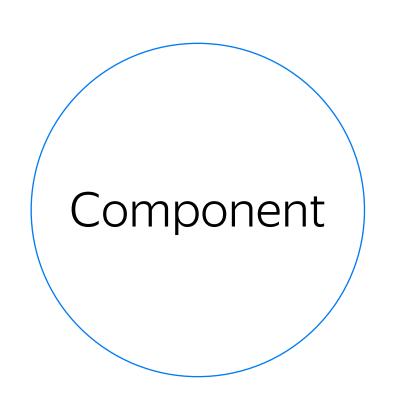


```
function FirstComponent() {
    //...
    return ...
}
```

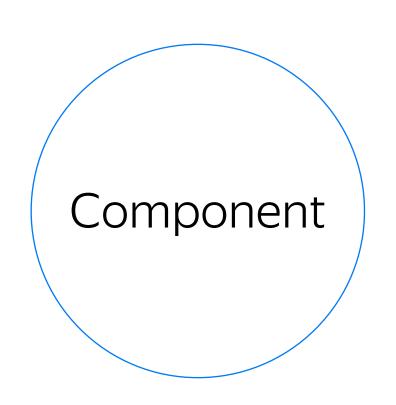
Function Component



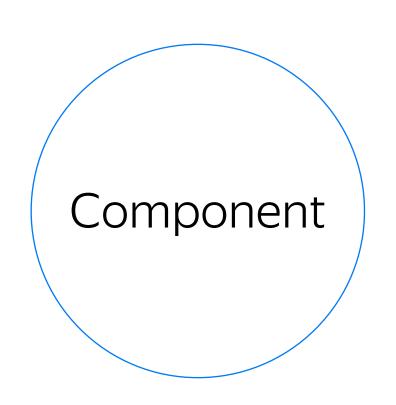
```
const FirstComponent = () => {
    //...
    return ...
}
```



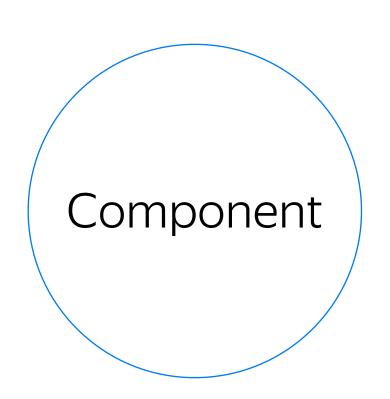
```
const FirstComponent = () => {
    //...
    return ...
}
```



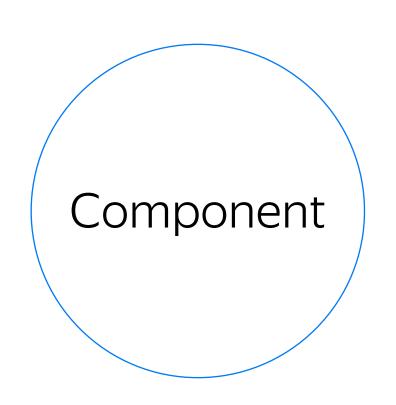
```
const FirstComponent = () => {
    //...
    return ...
}
```



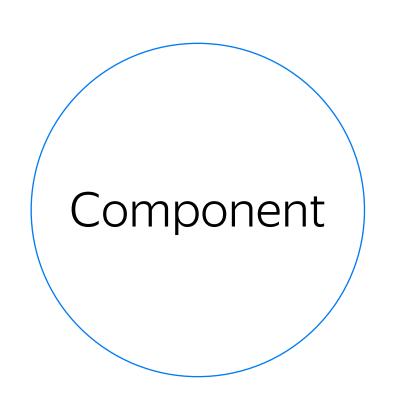
```
const firstComponent = () => {
    //...
    return ...
}
```



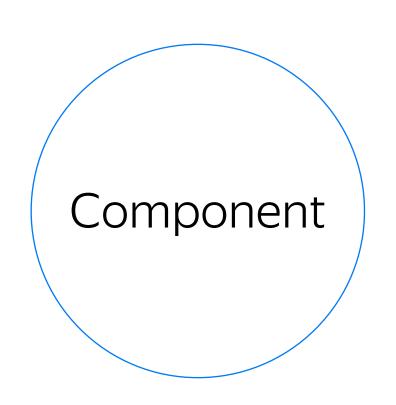
```
const firstComponent = () => {
    //...
    return ...
}
```



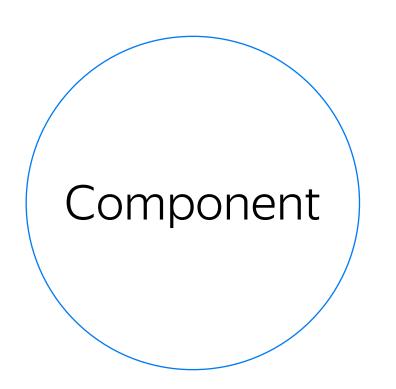
```
const FirstComponent = () => {
    //...
    return ...
}
```



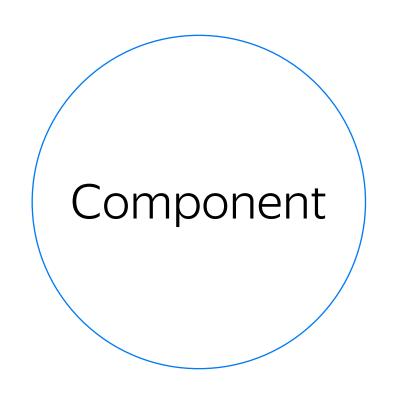
```
const FirstComponent = () => {
    //...
    return ...
}
```



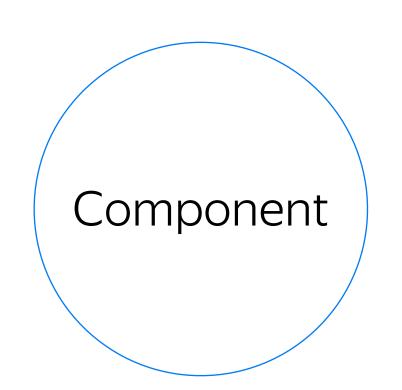
```
const FirstComponent = () => {
    //...
    return ...
}
```

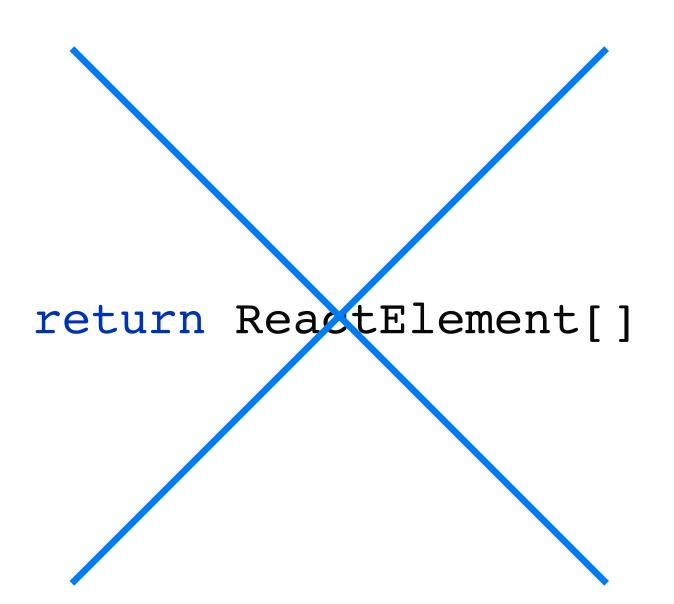


return ReactElement | null

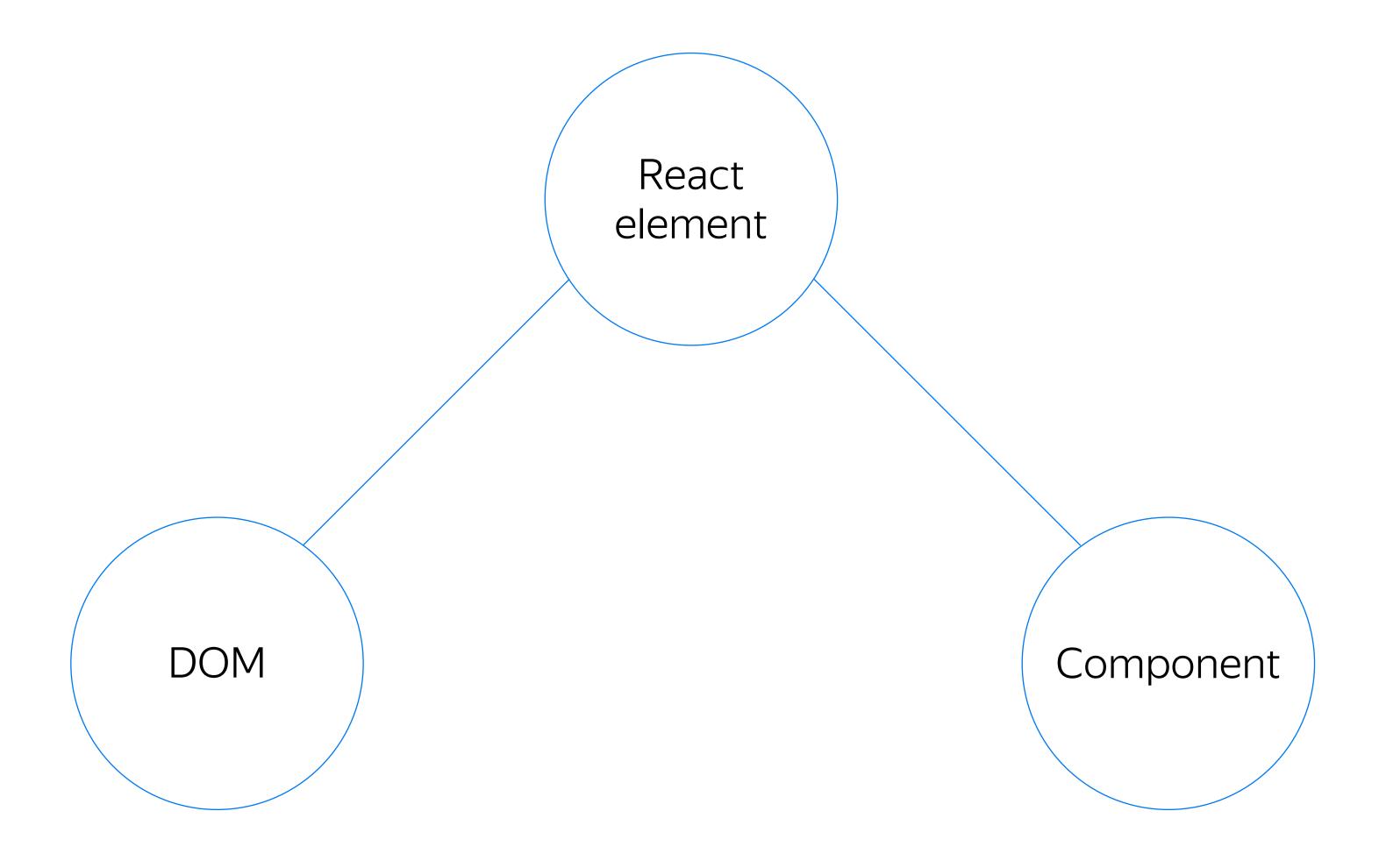


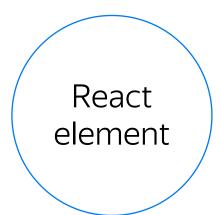
return ReactElement[]

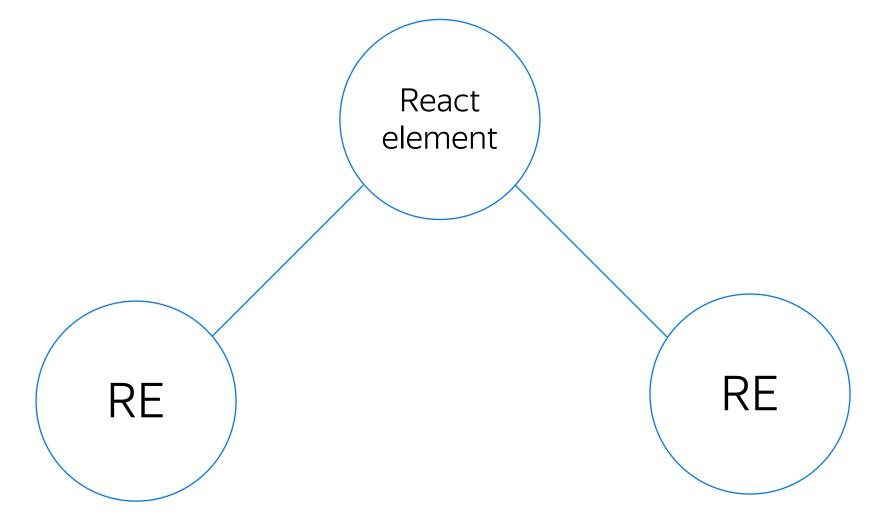


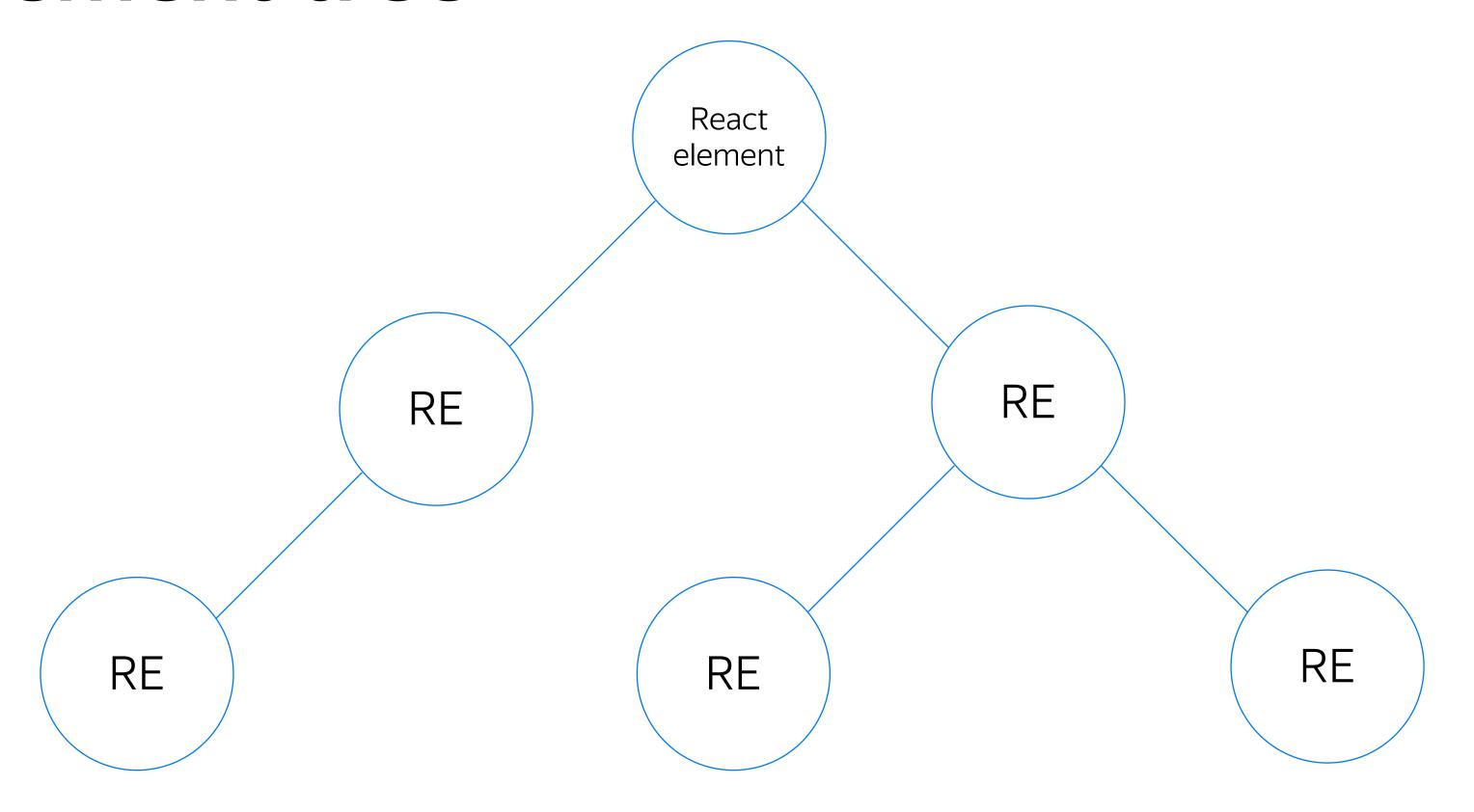


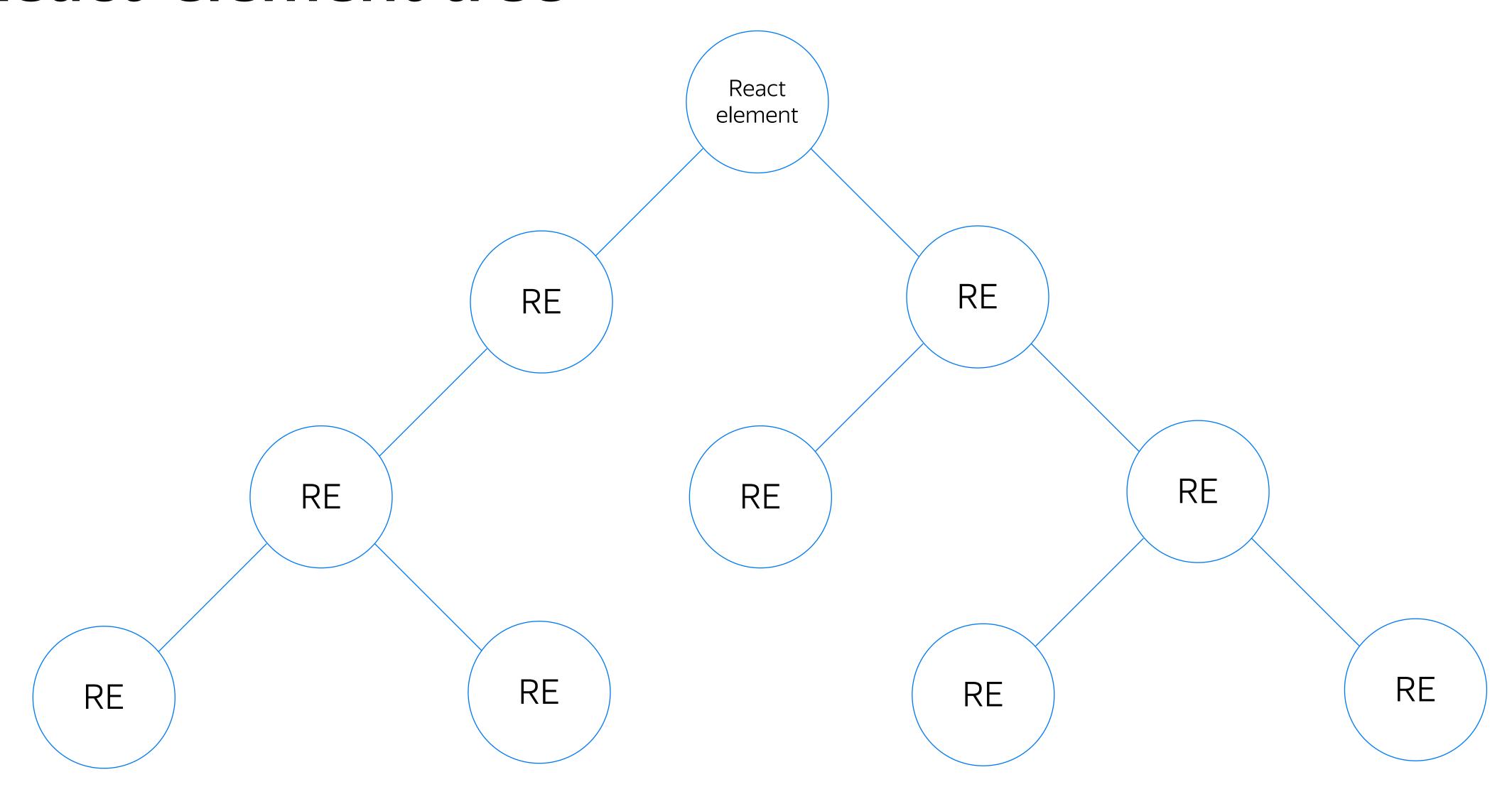
Строительные блоки

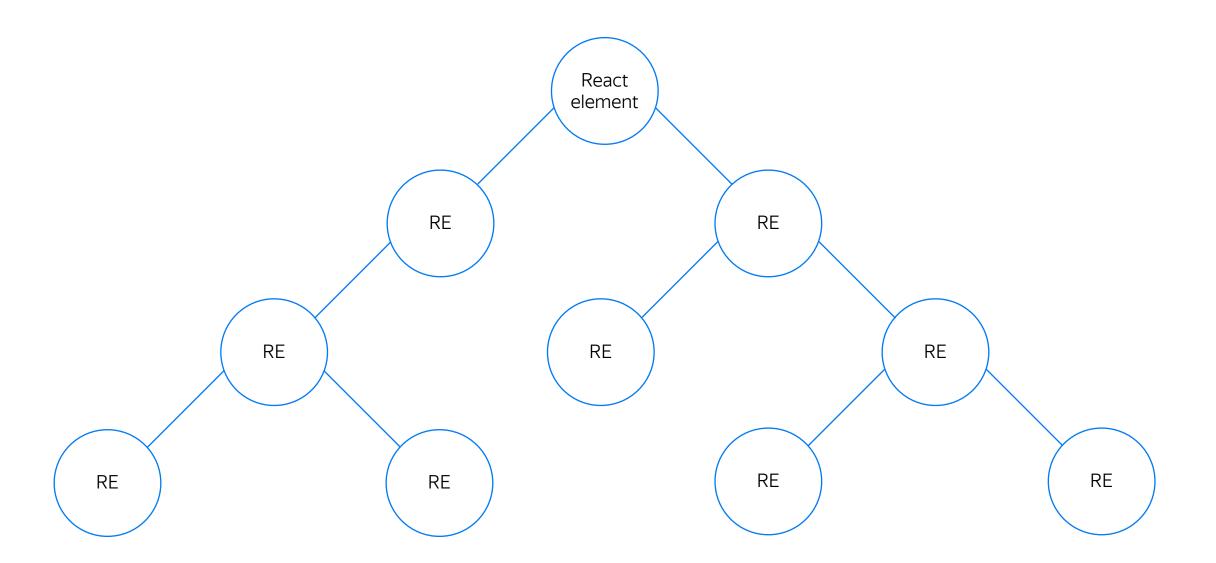




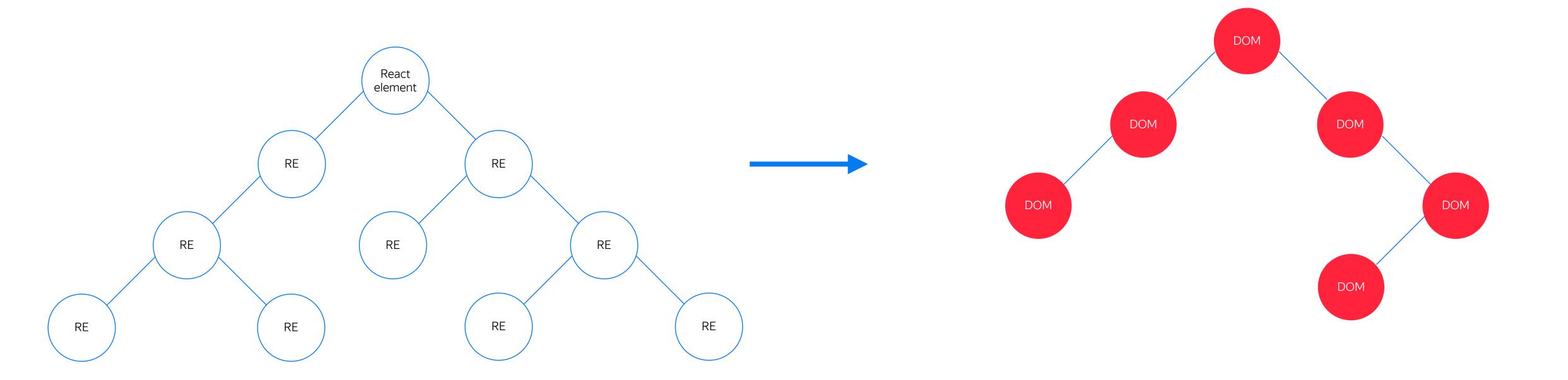


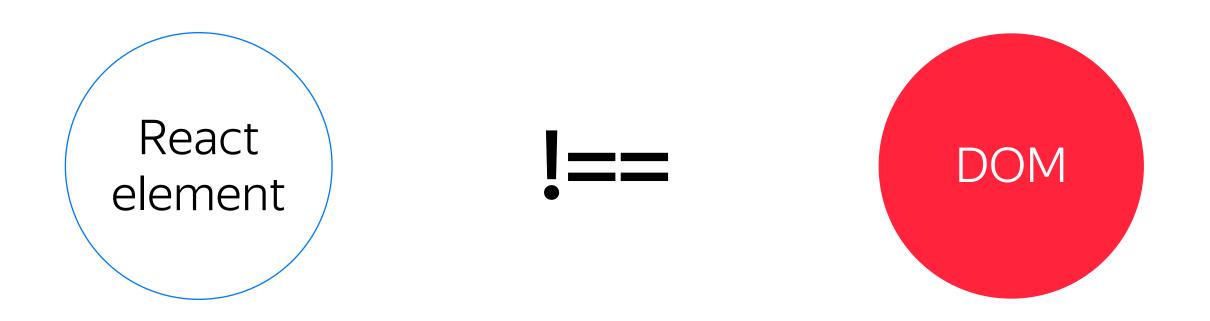


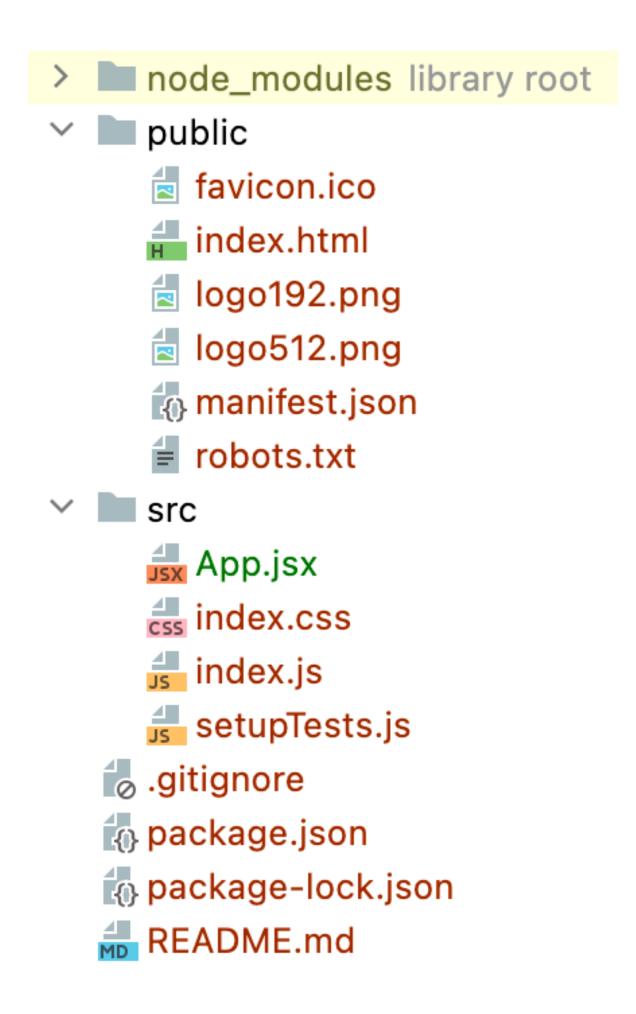


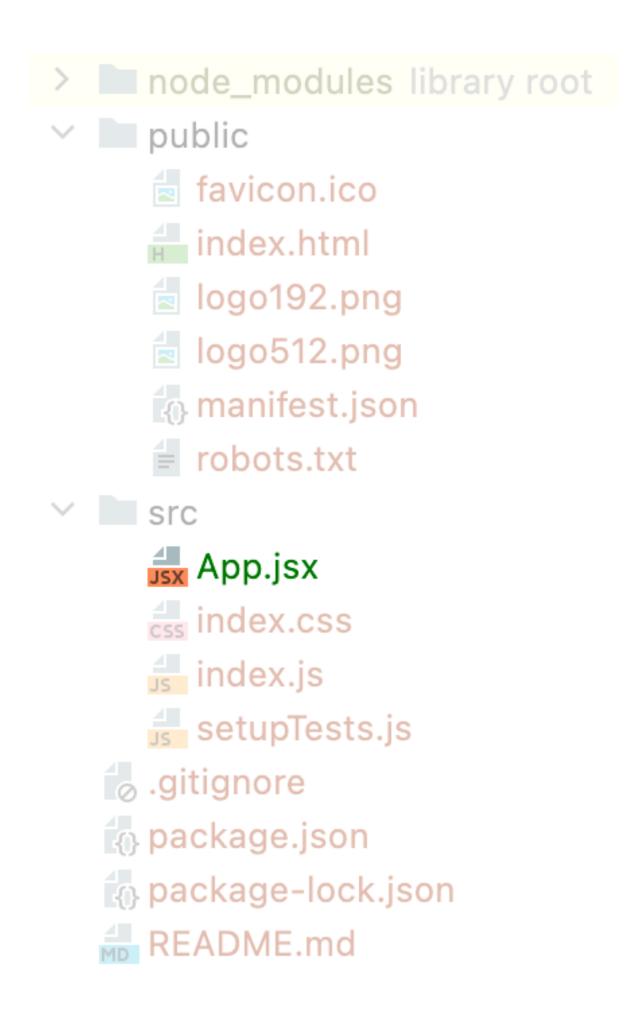


React-element tree !== DOM tree





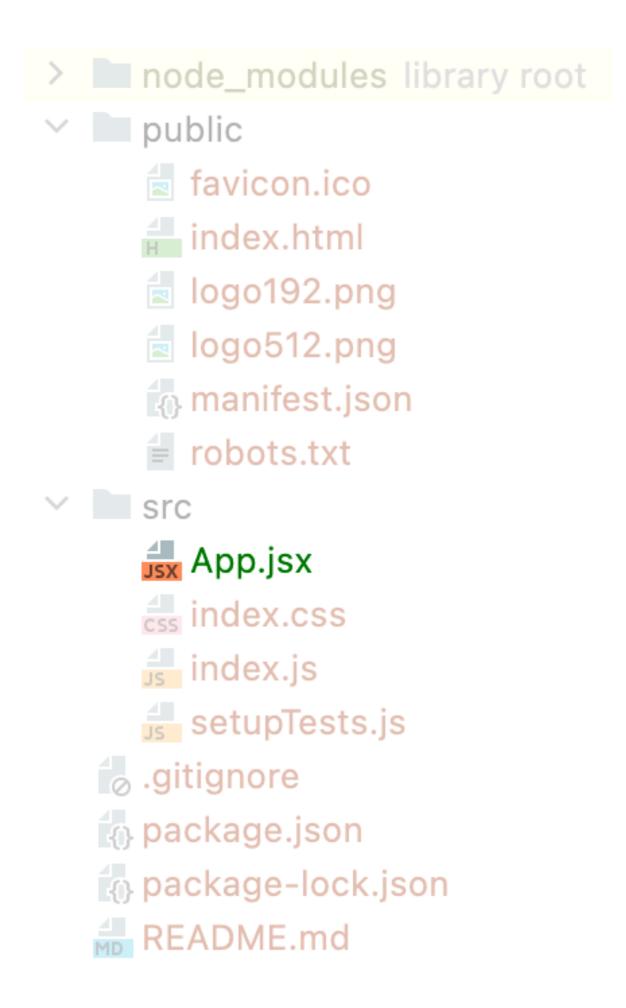




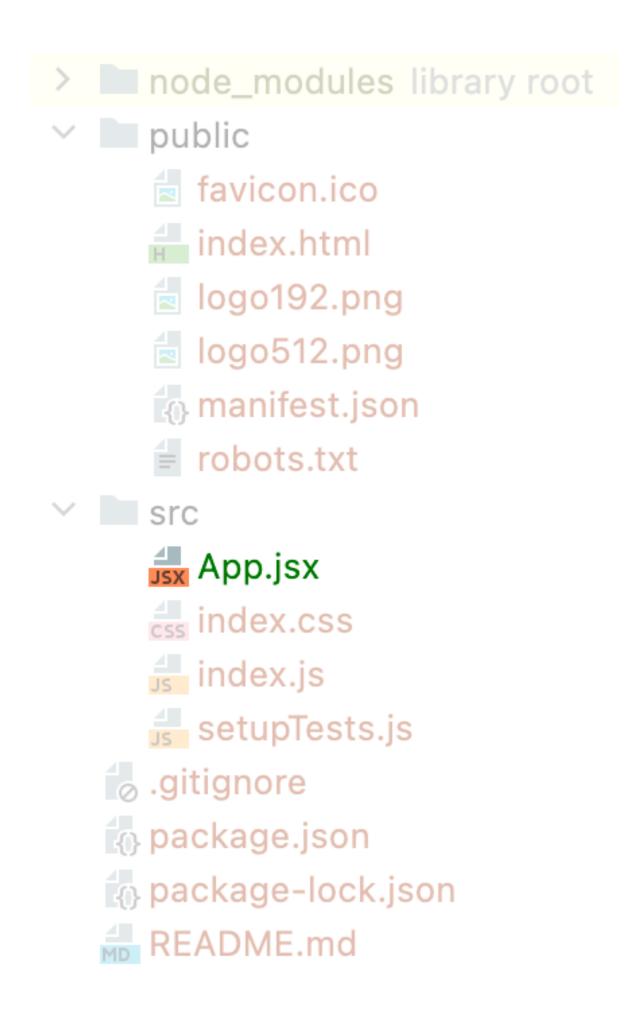
Что такое jsx???

> mode_modules library root public a favicon.ico index.html logo192.png logo512.png manifest.json robots.txt ✓ src App.jsx index.css index.js setupTests.js agitignore package.json package-lock.json README.md

Немного терпения



Откроем App.js



Откроем App.js

```
export const App = () => {
}
```

```
export const App = () => {
}
```

```
export const App = () => {
    return React.createElement('div')
}
```

Добавим его в index.js

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App/>);
```

```
export const App = () => {
    return React.createElement('div')
}
```

Выглядит очень сложно....

```
JSX     JS

<div/> => React.createElement('div')
```

JSX

А что будем делать?

Header Film Details Reviews Recommendations Footer

А что с данными?

```
const filmDetails = {
   id: 'asdjnwjenf2342njna',
    title: 'The Simpsons',
    seasonsCount: 33,
   genre: 'Comedy',
   similar: [...],
   reviews: [...]
}
```

А что с данными?

```
const filmDetails = {
   id: 'asdjnwjenf2342njna',
    title: 'The Simpsons',
    seasonsCount: 33,
    genre: 'Comedy',
    similar: [
            id: 'sa87f68as6f8as68',
            title: 'South Park',
    reviews: [...]
```

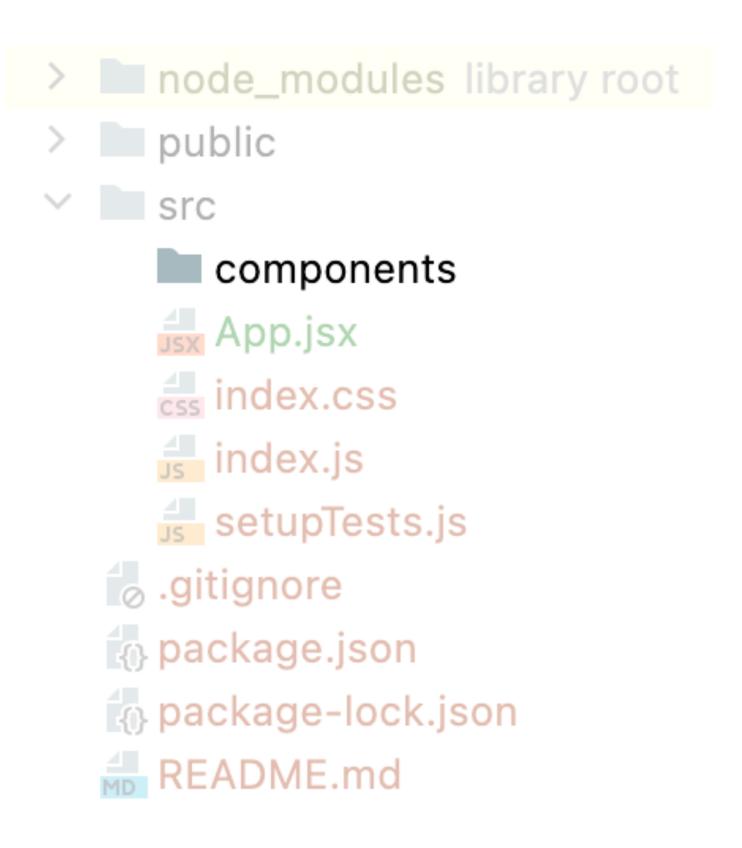
А что с данными?

```
const filmDetails = {
    id: 'asdjnwjenf2342njna',
    title: 'The Simpsons',
    seasonsCount: 33,
    genre: 'Comedy',
    similar: [...],
    reviews: [
            id: 'xc754vcx5',
            author: 'SP',
            text: 'Самый лучший мультфильм',
            rating: '10',
        },
```

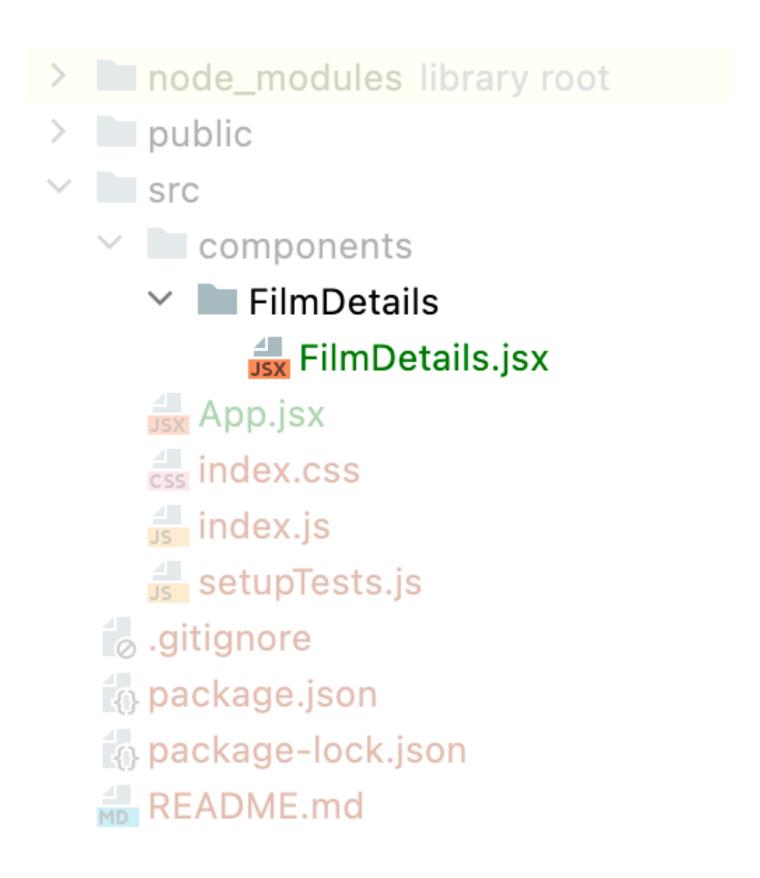
Создадим FilmDetails

Создадим FilmDetails

Начнем с общей для компонентов директории



Создадим компонент FilmDetails



Создадим компонент FilmDetails

```
export const FilmDetails = () => {
}
```

```
export const FilmDetails = (props) => {
}
```

```
export const FilmDetails = ({}) => {
}
```

```
export const FilmDetails = ({}) => {
}
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
}
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
}
```

```
<FilmDetails
    title={filmDetails.title}
    seasonsCount={filmDetails.seasonsCount}
    genre={filmDetails.genre}
/>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
    return
}
```

Добавим состояние

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let count = 0;
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre}\}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button>-</button>
               {count}
               <button>+</button>
           </div>
       </div>
```

Добавим состояние

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let count = 0;
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Нету'}
           <div>
               <button onClick={() => {}}>-</button>
               {count}
               <button onClick={() => {}}>+</button>
           </div>
       </div>
```

Добавим состояние

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let count = 0;
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre}\}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

Так работать не будет(

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let count = 0;
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre}\}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

Нам нужны хуки!

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let count = 0;
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

Что такое хуки???

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let count = 0;
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

Что такое хуки???

функции

Что такое хуки???

функции с особыми возможностями

Правила хуков

1. Имена хуков начинаются с use... (useState, useEffect, ...);

Правила хуков

- 1. Имена хуков начинаются с use... (useState, useEffect, ...);
- 2. Хуки можно вызывать только в функциональных компонентах или других хуках;

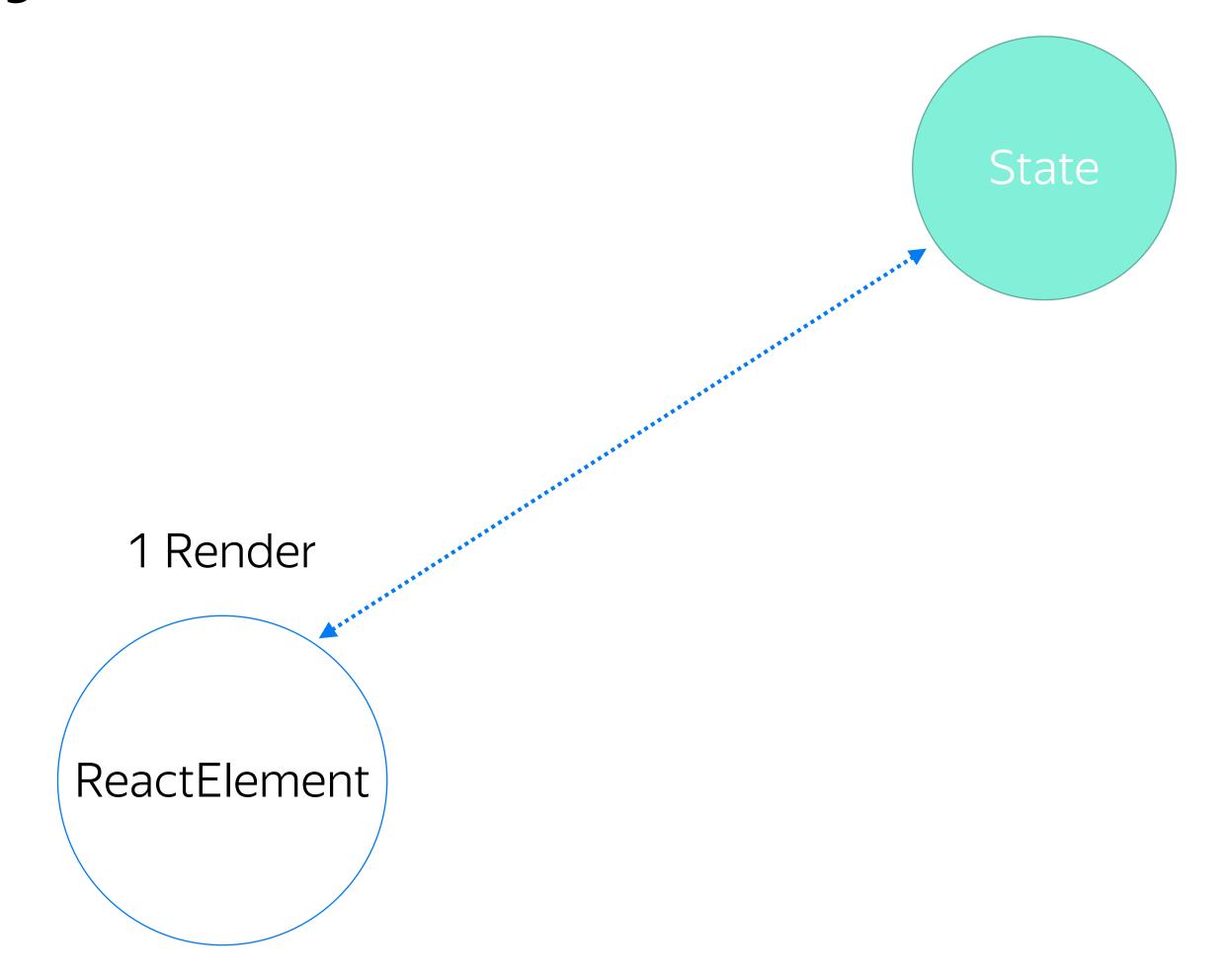
Правила хуков

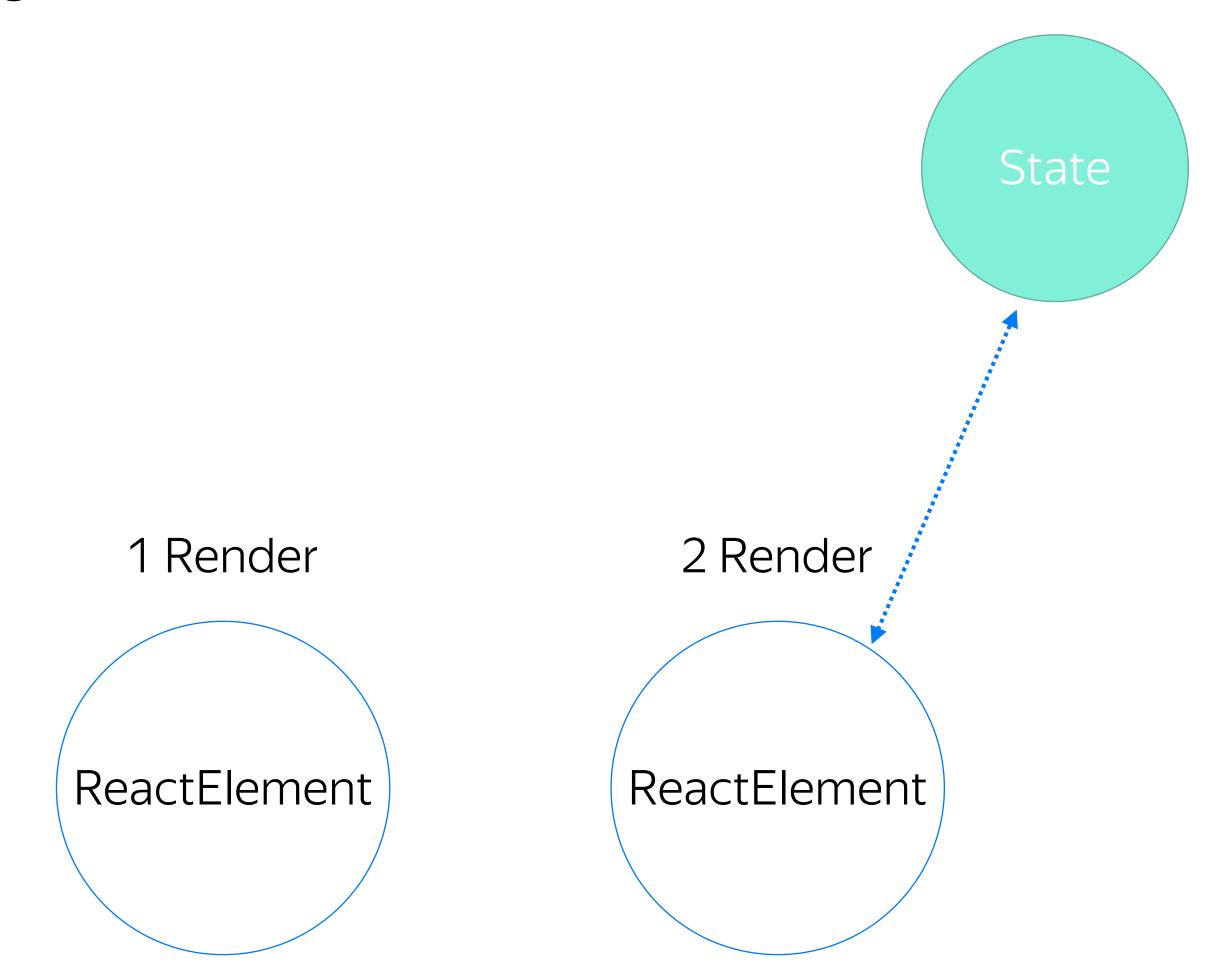
- 1. Имена хуков начинаются с use... (useState, useEffect, ...);
- 2. Хуки можно вызывать только в функциональных компонентах или других хуках;
- 3. Хуки нельзя вызывать условно, в циклах, в вложенных функциях.

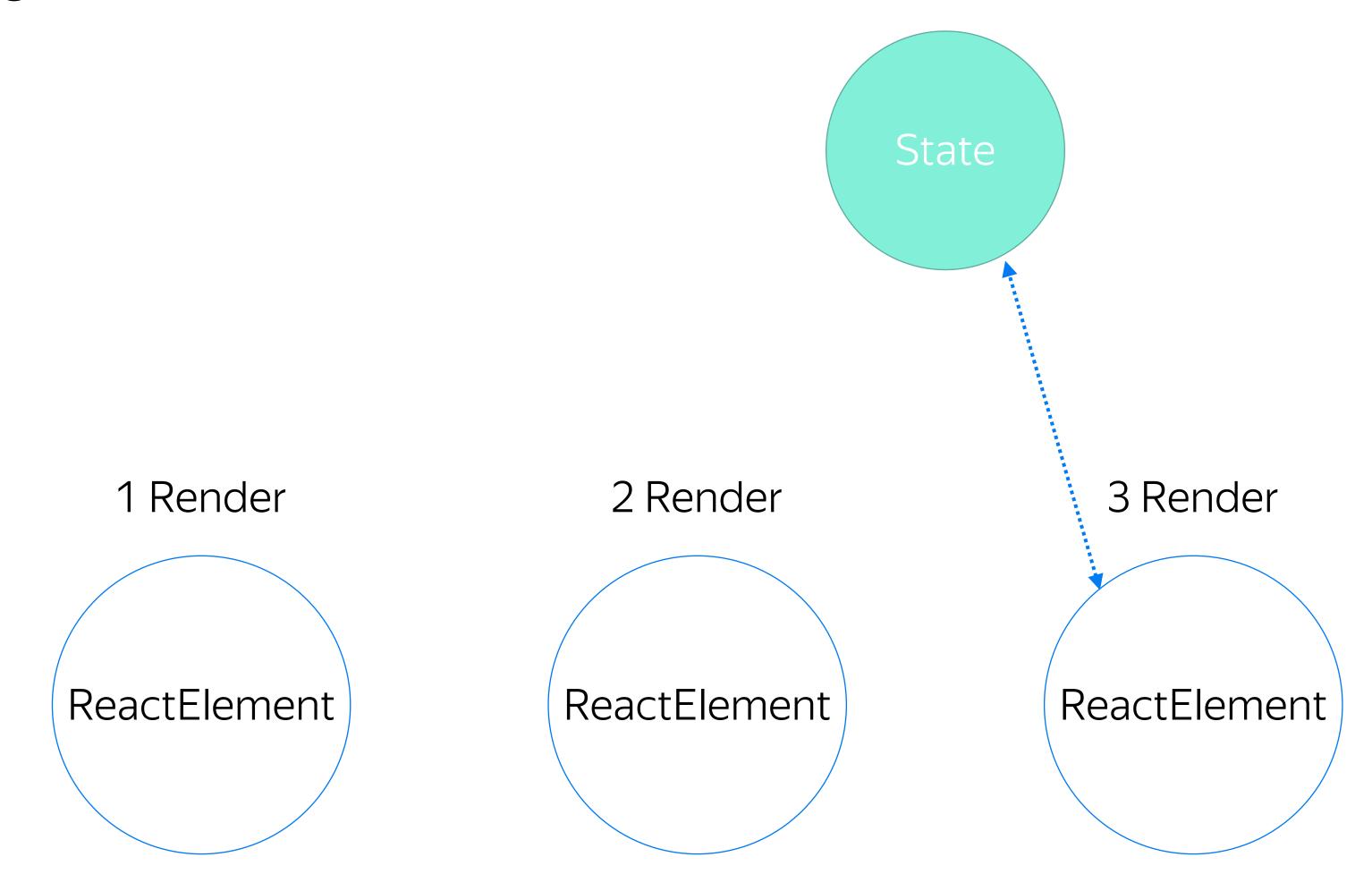
```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let state = useState();
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

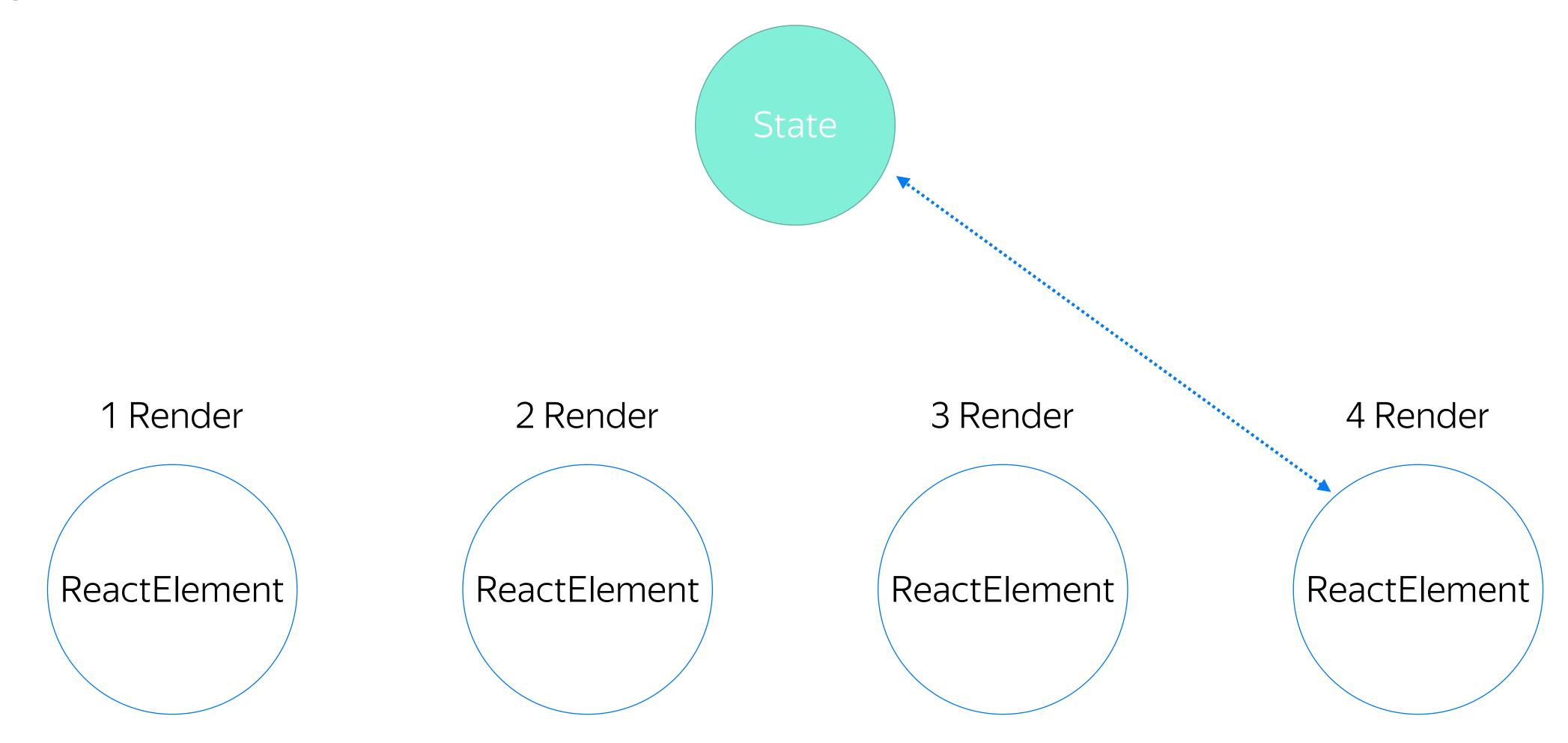
```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let state = useState();
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

1 Render
ReactElement









```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let state = useState();
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [element0, element1] = useState();
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [stateValue, element1] = useState();
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Нету'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [stateValue, element1] = useState();
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Нету'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [count, element1] = useState();
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [count, setStateValue] = useState();
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Нету'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [count, setCount] = useState();
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

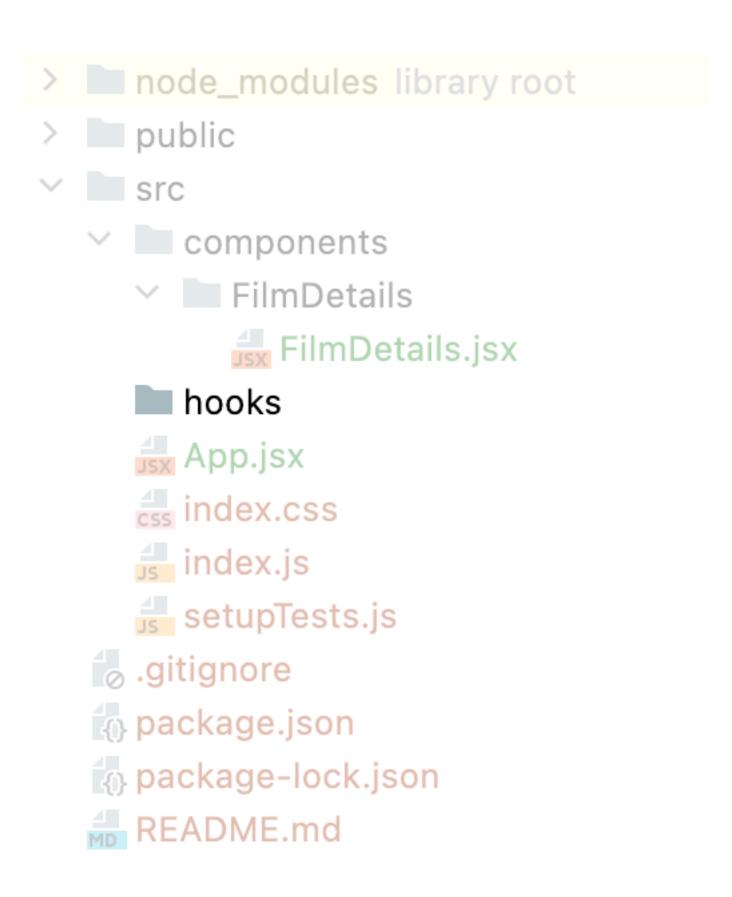
```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [count, setCount] = useState(0);
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Нету'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

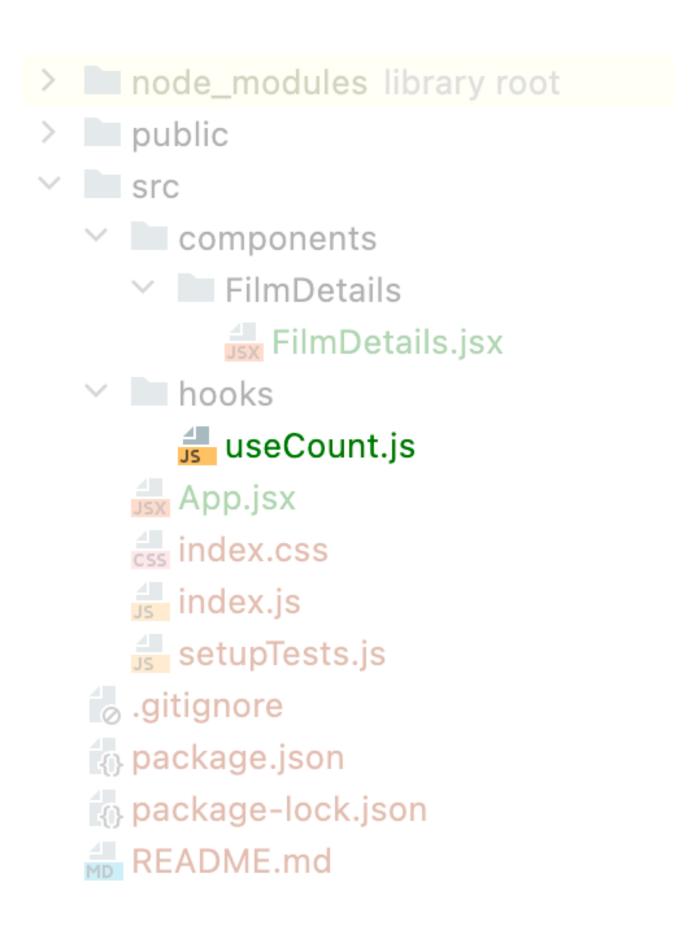
```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [count, setCount] = useState(() => {});
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [count, setCount] = useState(0);
   return (
       <div>
           {title | 'Неизвестный'}
           {Boolean(genre) && {genre}}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Нету'}
           <div>
               <button onClick={() => {count = count - 1}}>-</button>
               {count}
               <button onClick={() => {count = count + 1}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [count, setCount] = useState(0);
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {setCount(count - 1)}}>-</button>
               {count}
               <button onClick={() => {setCount(count - 1)}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let [count, setCount] = useState(0);
   return (
       <div>
           {title | 'Неизвестный'}
           {Boolean(genre) && {genre}}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}` : 'Heтy'}
           <div>
               <button onClick={() => {setCount(count - 1)}}>-</button>
               {count}
               <button onClick={() => {setCount(count - 1)}}>+</button>
           </div>
       </div>
```





```
export function useCount() {
}
```

```
export function useCount() {
   let [count, setCount] = useState(0);
}
```

```
export function useCount(initialValue) {
   let [count, setCount] = useState(initialValue);
}
```

```
export function useCount(initialValue) {
   let [count, setCount] = useState(initialValue);
   return {}
}
```

```
export function useCount(initialValue) {
   let [count, setCount] = useState(initialValue);
   return { count }
}
```

```
export function useCount(initialValue) {
  let [count, setCount] = useState(initialValue);

  const decrement = () => {};
  const increment = () => {};

  return { count }
}
```

```
export function useCount(initialValue) {
   let [count, setCount] = useState(initialValue);

   const decrement = () => {setCount(count - 1)};
   const increment = () => {setCount(count + 1)};

   return { count }
}
```

```
export function useCount(initialValue) {
  let [count, setCount] = useState(initialValue);

  const decrement = () => {setCount(count - 1)};
  const increment = () => {setCount(count + 1)};

  return { count, decrement, increment }
}
```

```
export function useCount(initialValue) {
  let [count, setCount] = useState(initialValue);

  const decrement = () => {setCount(count - 1)};
  const increment = () => {setCount(count + 1)};

  return { count, decrement, increment }
}
```

```
export function useCount(initialValue) {
   let [count, setCount] = useState(initialValue);

   const decrement = useCallback();
   const increment = () => {setCount(count + 1)};

   return { count, decrement, increment }
}
```

```
export function useCount(initialValue) {
  let [count, setCount] = useState(initialValue);

  const decrement = useCallback(() => {});
  const increment = () => {setCount(count + 1)};

  return { count, decrement, increment }
}
```

```
export function useCount(initialValue) {
  let [count, setCount] = useState(initialValue);

  const decrement = useCallback(() => {setCount(count - 1)});
  const increment = () => {setCount(count + 1)};

  return { count, decrement, increment }
}
```

```
export function useCount(initialValue) {
   let [count, setCount] = useState(initialValue);

   const decrement = useCallback(
        () => {setCount(count - 1)},
        []
   );
   const increment = () => {setCount(count + 1)};

   return { count, decrement, increment }
}
```

```
export function useCount(initialValue) {
   let [count, setCount] = useState(initialValue);

   const decrement = useCallback(
        () => {setCount(count - 1)},
        [count]
   );
   const increment = () => {setCount(count + 1)};

   return { count, decrement, increment }
}
```

```
export function useCount(initialValue) {
    let [count, setCount] = useState(initialValue);
    const decrement = useCallback(
        () => {setCount(count - 1)},
        [count]
    const increment = useCallback(
        () => {setCount(count + 1)},
        [count]
    return { count, decrement, increment }
```

```
export function useCount(initialValue) {
    let [count, setCount] = useState(initialValue);
    const decrement = useCallback(
        () => {setCount(() => {})},
        [count]
    const increment = useCallback(
        () => {setCount(count + 1)},
        [count]
    return { count, decrement, increment }
```

```
export function useCount(initialValue) {
    let [count, setCount] = useState(initialValue);
    const decrement = useCallback(
        () => {setCount(() => {})},
        [count]
    const increment = useCallback(
        () => {setCount(count + 1)},
        [count]
    return { count, decrement, increment }
```

```
export function useCount(initialValue) {
    let [count, setCount] = useState(initialValue);
    const decrement = useCallback(
        () => {setCount((currentCount) => {})},
        [count]
    const increment = useCallback(
        () => {setCount(count + 1)},
        [count]
    return { count, decrement, increment }
```

```
export function useCount(initialValue) {
    let [count, setCount] = useState(initialValue);
    const decrement = useCallback(
        () => {setCount((currentCount) => currentCount - 1)},
        [count]
    const increment = useCallback(
        () => {setCount(count + 1)},
        [count]
    return { count, decrement, increment }
```

```
export function useCount(initialValue) {
    let [count, setCount] = useState(initialValue);
    const decrement = useCallback(
        () => {setCount((currentCount) => currentCount - 1)},
    const increment = useCallback(
        () => {setCount(count + 1)},
        [count]
    return { count, decrement, increment }
```

```
export function useCount(initialValue) {
    let [count, setCount] = useState(initialValue);
    const decrement = useCallback(
        () => {setCount((currentCount) => currentCount - 1)},
    const increment = useCallback(
        () => {setCount((currentCount) => currentCount + 1)},
    return { count, decrement, increment }
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
    let [count, setCount] = useState(0);
    return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {seasonsCount > 0 ? `Koл-вo ${seasonsCount}` : 'Hety'}
           <div>
               <button onClick={() => {setCount(count - 1)}}>-</button>
               {count}
               <button onClick={() => {setCount(count - 1)}}>+</button>
           </div>
       </div>
```

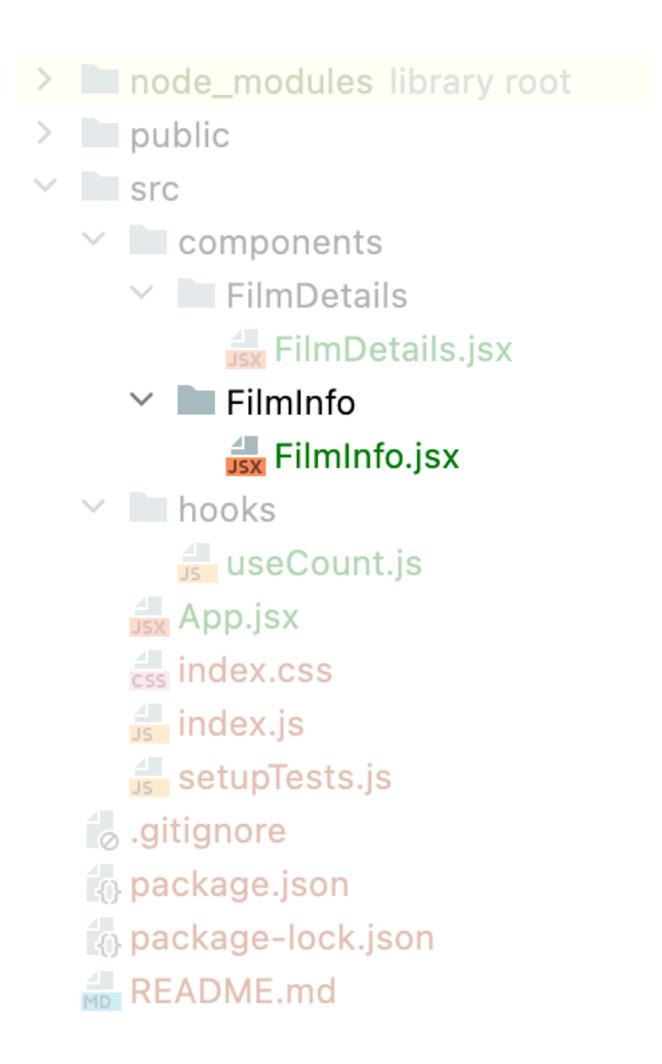
```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let {count, increment, decrement} = useCount(0);
    return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {seasonsCount > 0 ? `Koл-вo ${seasonsCount}` : 'Hety'}
           <div>
               <button onClick={() => {setCount(count - 1)}}>-</button>
               {count}
               <button onClick={() => {setCount(count - 1)}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let {count, increment, decrement} = useCount(0);
    return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre} \}
           {seasonsCount > 0 ? `Koл-вo ${seasonsCount}` : 'Hety'}
           <div>
               <button onClick={() => {setCount(count - 1)}}>-</button>
               {count}
               <button onClick={() => {setCount(count - 1)}}>+</button>
           </div>
       </div>
```

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
    let {count, increment, decrement} = useCount(0);
    return (
       <div>
           {title | 'Неизвестный'}
            \{Boolean(genre) \&\& \{genre} \}
           {p>{seasonsCount > 0 ? `Кол-во ${seasonsCount}`: 'Heтy'}
           <div>
               <button onClick={decrement}>-</button>
               {count}
               <button onClick={increment}>+</button>
           </div>
       </div>
```

Хотим переиспользовать!

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
   let {count, increment, decrement} = useCount(0);
   return (
       <div>
           {title | 'Неизвестный'}
           \{Boolean(genre) \&\& \{genre}\}
           {seasonsCount > 0 ? `Koл-вo ${seasonsCount}` : 'Hety'}
           <div>
               <button onClick={decrement}>-</button>
               {count}
               <button onClick={increment}>+</button>
           </div>
       </div>
```



```
export const FilmInfo = () => {
   return
}
```

```
export const FilmInfo = ({ title, genre, seasonsCount }) => {
    return
}
```

Так работать не будет(

Так работать не будет(

Компонент возвращает только 1 элемент

Исправляемся

Фрагмент

Фрагмент

Воспользуемся нашим новым компонентом

```
export const FilmDetails = ({ title, genre, seasonsCount }) => {
    let {count, increment, decrement} = useCount(0);
    return (
        <div>
            <FilmInfo
                title={title}
                genre={genre}
                seasonsCount={seasonsCount}
            />
            <div>
                <button onClick={decrement}>-</button>
                {count}
                <button onClick={increment}>+</button>
            </div>
        </div>
```

Нужно отобразить рейтинг

Рейтинг фильма - средний рейтинг всех отзывов

Нужно отобразить рейтинг

Рейтинг фильма = суммарный рейтинг всех отзывов / количество отзывов

Нужно отобразить рейтинг

Рейтинг фильма = суммарный рейтинг всех отзывов / количество отзывов

```
const filmRating = filmDetails.reviews.reduce((sum, review) => {
    return sum + review.rating
}, 0)
```

```
const filmRating = filmDetails.reviews.reduce((sum, review) => {
    return sum + review.rating
}, 0) / filmDetails.reviews.length;
```

```
const filmRating = filmDetails.reviews.reduce((sum, review) => {
    return sum + review.rating
}, 0) / filmDetails.reviews.length;
```

```
export const App = () => {
    const filmRating = Math.floor(
        filmDetails.reviews.reduce((sum, review) => {
            return sum + review.rating
        }, 0) / filmDetails.reviews.length
    );
    return (...)
}
```

Пересчет при каждом перерендере

```
export const App = () => {
    const filmRating = Math.floor(
        filmDetails.reviews.reduce((sum, review) => {
            return sum + review.rating
        }, 0) / filmDetails.reviews.length
    );
    return (...)
}
```

```
export const App = () => {
    const filmRating = Math.floor(
        filmDetails.reviews.reduce((sum, review) => {
            return sum + review.rating
        }, 0) / filmDetails.reviews.length
    );
    return (...)
}
```

```
export const App = () => {
   const filmRating = useMemo();
   return (...)
}
```

```
export const App = () => {
   const filmRating = useMemo(() => {});
   return (...)
}
```

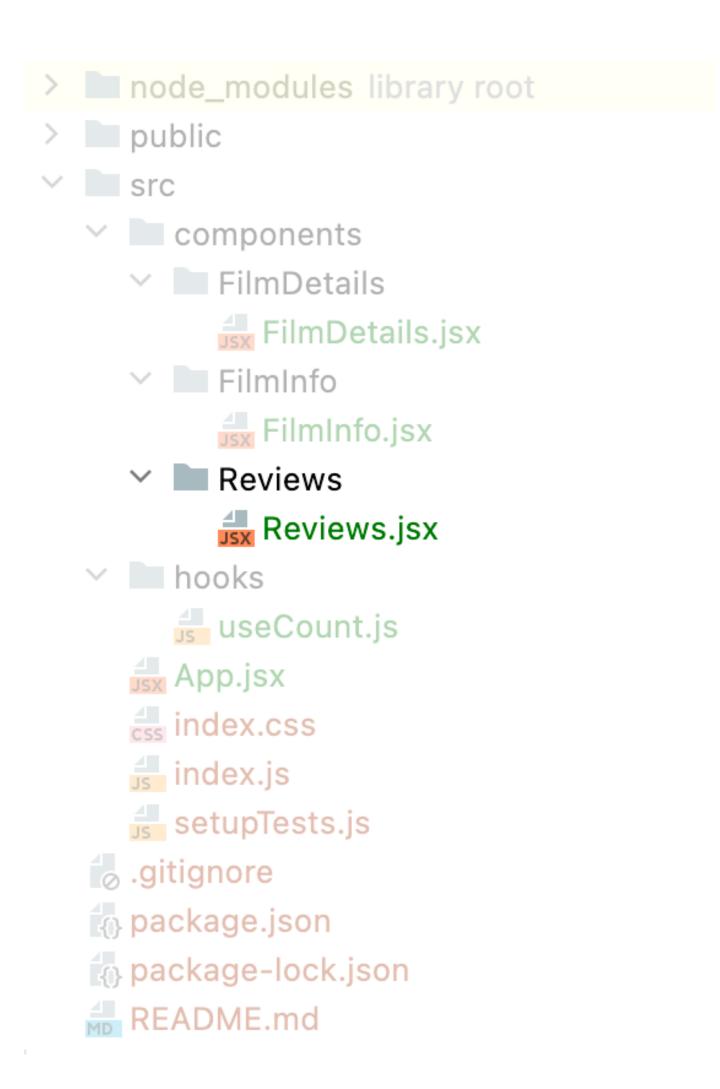
```
export const App = () => {
    const filmRating = useMemo(() => {
        return Math.floor(
            filmDetails.reviews.reduce((sum, review) => {
                return sum + review.rating
            }, 0) / filmDetails.reviews.length
        )
    }, [filmDetails.reviews]);
    return (...)
}
```

```
export const App = () => {
    const filmRating = useMemo(() => {
        return Math.floor(
            filmDetails.reviews.reduce((sum, review) => {
                return sum + review.rating
            }, 0) / filmDetails.reviews.length
        )
    }, []);
    return (...)
}
```

```
export const App = () => {
    const filmRating = useMemo(() => {
        return Math.floor(
            filmDetails.reviews.reduce((sum, review) => {
                return sum + review.rating
            }, 0) / filmDetails.reviews.length
        )
    }, [filmDetails.reviews]);
    return (...)
}
```

```
export const App = () => {
    const filmRating = useMemo(() => {
        return Math.floor(
            filmDetails.reviews.reduce((sum, review) => {
                return sum + review.rating
            }, 0) / filmDetails.reviews.length
        )
    }, [filmDetails.reviews]);
    return (...)
}
```

Отзывы



Отзывы

```
export const Reviews = () => {
}
```

Отзывы

```
export const Reviews = ({reviews}) => {
}
```

А как отрисовать массив?

Мб так?

```
export const Reviews = ({reviews}) => {
    return (
        <div>
            <div>
                <span>{reviews[0].author}</span>
                <span>{reviews[0].text}</span>
                <span>{reviews[0].rating}</span>
            </div>
            <div>
                <span>{reviews[1].author}</span>
                <span>{reviews[1].text}</span>
                <span>{reviews[1].rating}</span>
            </div>
        </div>
```

Это не гибко

```
export const Reviews = ({reviews}) => {
    return (
        <div>
            <div>
                <span>{reviews[0].author}</span>
                <span>{reviews[0].text}</span>
                <span>{reviews[0].rating}</span>
            </div>
            <div>
                <span>{reviews[1].author}</span>
                <span>{reviews[1].text}</span>
                <span>{reviews[1].rating}</span>
            </div>
        </div>
```

Вспоминаем что jsx превратится в js

```
export const Reviews = ({reviews}) => {
    return (
        <div>
            <div>
                <span>{reviews[0].author}</span>
                <span>{reviews[0].text}</span>
                <span>{reviews[0].rating}</span>
            </div>
            <div>
                <span>{reviews[1].author}</span>
                <span>{reviews[1].text}</span>
                <span>{reviews[1].rating}</span>
            </div>
        </div>
```

Необходимо для каждого объекта создать элемент

Необходимо для каждого объекта создать элемент

```
export const Reviews = ({reviews}) => {
    return
        <div>
                reviews.map((review) => (
                    <div>
                        <span>{review.author}</span>
                        <span>{review.text}</span>
                        <span>{review.rating}</span>
                    </div>
        </div>
```

Добавим проверку на количество элементов

```
export const Reviews = ({reviews}) => {
    return (
        <div>
                reviews?.length && reviews.map((review) => (
                    <div>
                        <span>{review.author}</span>
                        <span>{review.text}</span>
                        <span>{review.rating}</span>
                    </div>
        </div>
```

Добавим проверку на количество элементов

Добавим проверку на количество элементов

```
export const Reviews = ({reviews}) => {
    return (
        <div>
                reviews?.length && reviews.map((review) => (
                    <div>
                        <span>{review.author}</span>
                        <span>{review.text}</span>
                        <span>{review.rating}</span>
                    </div>
        </div>
```

Исправим

```
export const Reviews = ({reviews}) => {
    return (
        <div>
                Boolean(reviews.length) && reviews.map((review) => (
                    <div>
                        <span>{review.author}</span>
                        <span>{review.text}</span>
                        <span>{review.rating}</span>
                    </div>
        </div>
```

Исправим

```
export const Reviews = ({reviews}) => {
    return (
        <div>
                !!reviews.length && reviews.map((review) => (
                    <div>
                        <span>{review.author}</span>
                        <span>{review.text}</span>
                        <span>{review.rating}</span>
                    </div>
        </div>
```

Исправим

```
export const Reviews = ({reviews}) => {
    return (
        <div>
                reviews.length > 0 && reviews.map((review) => (
                    <div>
                        <span>{review.author}</span>
                        <span>{review.text}</span>
                        <span>{review.rating}</span>
                    </div>
        </div>
```

Передадим в наш компонент данные (App.jsx)

А это что? А почему?

```
Warning: Each child in a list should have a unique "key" prop.
Check the render method of `Reviews`. See <a href="https://reactjs.org/link/warning-keys">https://reactjs.org/link/warning-keys</a> for more information.
at div
at Reviews (<a href="http://localhost:3000/static/js/bundle.js:360:5">http://localhost:3000/static/js/bundle.js:360:5</a>)
at div
at App (<a href="http://localhost:3000/static/js/bundle.js:33:68">https://localhost:3000/static/js/bundle.js:33:68</a>)
```

А это что? А почему? А что за кеу???

```
Warning: Each child in a list should have a unique "key" prop.
Check the render method of `Reviews`. See <a href="https://reactjs.org/link/warning-keys">https://reactjs.org/link/warning-keys</a> for more information.
at div
at Reviews (<a href="http://localhost:3000/static/js/bundle.js:360:5">http://localhost:3000/static/js/bundle.js:360:5</a>)
at div
at App (<a href="http://localhost:3000/static/js/bundle.js:33:68">https://localhost:3000/static/js/bundle.js:33:68</a>)
```

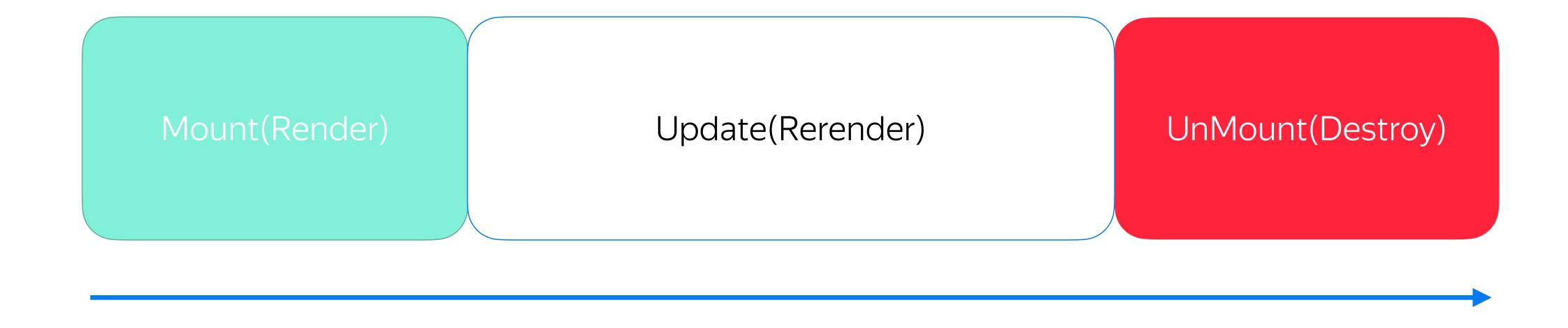
Рождение компонента

Mount(Render)

Он стал совсем взрослым...

Mount(Render) Update(Rerender)

Ничто не вечно...



Render array elements

Review1

Review2

Render array elements

Review1

Review2

Review3

Render array elements

Без изменений Review1 Review1 Без изменений Review2 Review2 Вмонтируется Review3

Review1

Review2

Review1

Review3

Review2

Review1 Review1 Review2 Review3 Review2

Без изменений Review1 Review1 ВМонтируется Review3 Без изменений Review2 Review2

Без изменений Review1 Review1 Обновится Review2 Review3 ВМонтируется Review2

Key Key Без изменений Review1 Review1 Key ВМонтируется Review3 Key Key Без изменений Review2 Review2

dwd3 dwd3 Без изменений Review1 Review1 czcd5 ВМонтируется Review3 F344 F344 Без изменений Review2 Review2

Добавим ключи

```
export const Reviews = ({reviews}) => {
    return (
        <div>
                reviews.length > 0 && reviews.map((review) => (
                    <div>
                        <span>{review.author}</span>
                        <span>{review.text}</span>
                        <span>{review.rating}</span>
                    </div>
        </div>
```

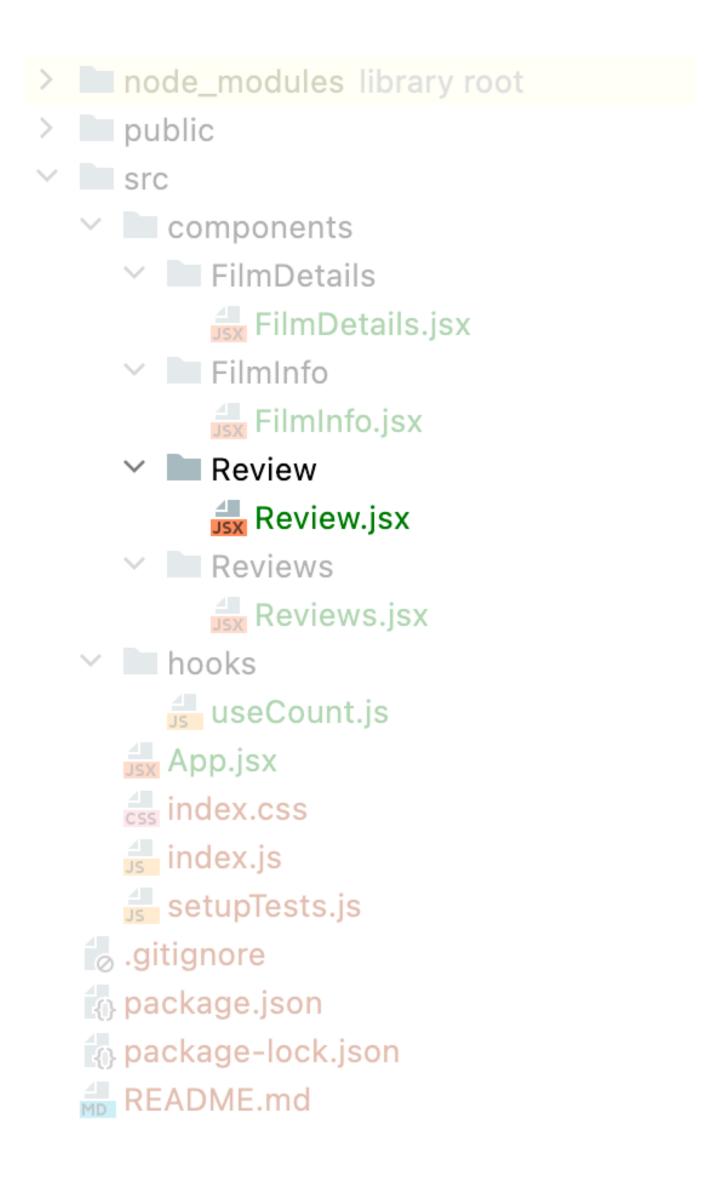
Добавим ключи

```
export const Reviews = ({reviews}) => {
    return (
        <div>
                reviews.length > 0 && reviews.map((review) => (
                    <div key={review.id}>
                        <span>{review.author}</span>
                        <span>{review.text}</span>
                        <span>{review.rating}</span>
                    </div>
        </div>
```

Вынесем отображение отзыва в отдельный компонент

```
export const Reviews = ({reviews}) => {
    return (
        <div>
                reviews.length > 0 && reviews.map((review) => (
                    <div key={review.id}>
                        <span>{review.author}</span>
                        <span>{review.text}</span>
                        <span>{review.rating}</span>
                    </div>
        </div>
```

Вынесем отображение отзыва в отдельный компонент



Вынесем отображение отзыва в отдельный компонент

Воспользуемся им

```
export const Reviews = ({reviews}) => {
    return (
        <div>
                reviews.length > 0 && reviews.map(
                     (review) => (
                        <Review
                             rating={review.rating}
                             text={review.text}
                             author={review.author}
```

Агде ключ???

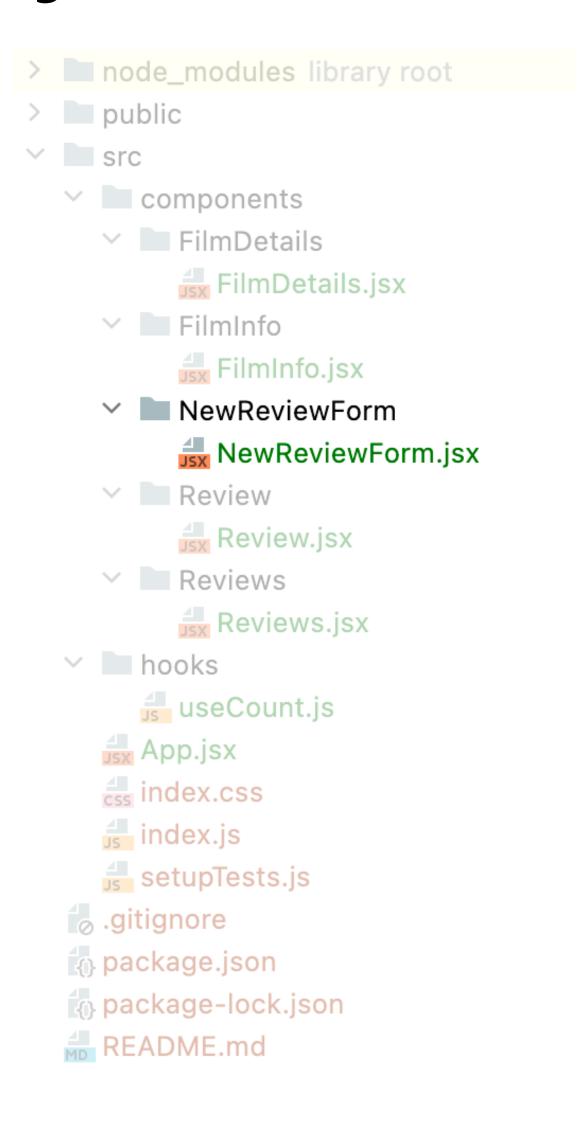
```
export const Reviews = ({reviews}) => {
    return (
        <div>
                reviews.length > 0 && reviews.map(
                     (review) => (
                        <Review
                             rating={review.rating}
                             text={review.text}
                             author={review.author}
        </div>
```



Исправили

```
export const Reviews = ({reviews}) => {
    return (
        <div>
                reviews.length > 0 && reviews.map(
                     (review) => (
                        <Review
                             key={review.id}
                             rating={review.rating}
                             text={review.text}
                             author={review.author}
```

Добавим формочку для нового отзыва



Добавим формочку для нового отзыва

```
export const NewReviewForm = () => {
  return ()
}
```

Добавим формочку для нового отзыва

```
export const NewReviewForm = () => {
  return (
      <div>
          <label>
              Name:
              <input type="text" />
          </label>
          <label>
              Text:
              <input type="text" />
          </label>
          <label>
              Rating:
              <input type="number" />
          </label>
      </div>
```

И как нам быть со всем этим?

```
export const NewReviewForm = () => {
  return (
      <div>
          <label>
              Name:
              <input type="text" />
          </label>
          <label>
              Text:
              <input type="text" />
          </label>
          <label>
              Rating:
              <input type="number" />
          </label>
      </div>
```

```
export const NewReviewForm = () => {
    const [] = useState();
    return (
      <div>
          <label>
              Name:
              <input type="text" />
          </label>
      </div>
```

```
export const NewReviewForm = () => {
   const [name, setName] = useState('Tema');
    return (
      <div>
          <label>
              Name:
              <input type="text" />
          </label>
      </div>
```

```
export const NewReviewForm = () => {
    const [name, setName] = useState('Tema');
    return (
      <div>
          <label>
              Name:
              <input
                  value={name}
                  onChange={event => setName(event.target.value)}
                  type="text" />
          </label>
      </div>
```

```
export const NewReviewForm = () => {
   const [name, setName] = useState('Tema');
   const [text, setText] = useState('Empty Text');
   const [rating, setRating] = useState(10);
   return (...)
}
```

```
export const NewReviewForm = () => {
    const [name, setName] = useState('Tema');
    const [text, setText] = useState('Empty Text');
   const [rating, setRating] = useState(10);
   const onNameChange = (event) => {
      setName(event.target.value);
      setText('');
      setRating(5);
    const onTextChange = (event) => setText(event.target.value);
    const onRatingChange = (event) => setRating(event.target.value);
   return (...)
```

Как-то сложно(

```
export const NewReviewForm = () => {
    const [name, setName] = useState('Tema');
   const [text, setText] = useState('Empty Text');
   const [rating, setRating] = useState(10);
   const onNameChange = (event) => {
      setName(event.target.value);
      setText('');
      setRating(5);
   const onTextChange = (event) => setText(event.target.value);
   const onRatingChange = (event) => setRating(event.target.value);
   return (...)
```

Мбесть еще хуки???

```
export const NewReviewForm = () => {
    const [name, setName] = useState('Tema');
    const [text, setText] = useState('Empty Text');
   const [rating, setRating] = useState(10);
   const onNameChange = (event) => {
      setName(event.target.value);
      setText('');
      setRating(5);
   const onTextChange = (event) => setText(event.target.value);
    const onRatingChange = (event) => setRating(event.target.value);
   return (...)
```

useReducer

```
export const NewReviewForm = () => {
  const state = useReducer();

return (...)
}
```

useReducer

```
export const NewReviewForm = () => {
   const [state, dispatch] = useReducer();
   return (...)
}
```

useReducer

```
export const NewReviewForm = () => {
   const [state, dispatch] = useReducer(reducer);
   return (...)
}
```

Что такое reducer???

```
export const NewReviewForm = () => {
   const [state, dispatch] = useReducer(reducer);
   return (...)
}
```

```
const reducer = () => {
}
```

```
const reducer = (state, action) => {
}
```

```
const reducer = (state, action) => {

const action = {
   type: 'DO_SOMETHING',
   payload: {}
}
```

```
const reducer = (state, action) => {
    switch (action.type) {
        case 'DO_SOMETHING':
            // ...
            return {...}
        default:
            return state;
const action = {
    type: 'DO_SOMETHING',
   payload: {}
```

```
const reducer = (state, action) => {
    switch (action.type) {
        case 'DO_SOMETHING':
           // ...
            return {...}
        default:
            return state;
const action = {
    type: 'DO_SOMETHING',
   payload: {}
```

```
const reducer = (state, action) => {
    switch (action.type) {
        case 'DO_SOMETHING':
           // ...
            return {...}
        default:
            return state;
const action = {
    type: 'DO_SOMETHING',
   payload: {}
```

```
const reducer = (state, action) => {
    switch (action?.type) {
        case "setName":
            return { name: action.payload.name, text: "", rating: 0 };
        case "setText":
            return { ...state, text: action.payload.text };
        case "setRating":
            return { ...state, rating: action.payload.rating };
        default:
            return state;
    }
};
```

Actions

```
const FORM_ACTIONS = {
    setName: "setName",
    setText: "setText",
    setRating: "setRating",
};
```

reducer

```
const reducer = (state, action) => {
    switch (action?.type) {
        case FORM_ACTIONS.setName:
            return { name: action.payload.name, text: "", rating: 0 };
        case FORM_ACTIONS.setText:
            return { ...state, text: action.payload.text };
        case FORM_ACTIONS.setRating:
            return { ...state, rating: action.payload.rating };
        default:
            return state;
    }
};
```

Вернемся к форме

```
export const NewReviewForm = () => {
   const [state, dispatch] = useReducer(reducer);
   return (...)
}
```

InitialState

```
const initialState = {
   name: "Tema",
   text: "text",
   rating: 10,
}

export const NewReviewForm = () => {
   const [state, dispatch] = useReducer(reducer, initialState);
   return (...)
}
```

```
export const NewReviewForm = () => {
    const [state, dispatch] = useReducer(reducer, initialState);
    return (
        <div>
            <label>
                Name:
                <input type="text" />
            </label>
        </div>
```

```
export const NewReviewForm = () => {
    const [state, dispatch] = useReducer(reducer, initialState);
    const onNameChange = (event) => dispatch(
        {type: FORM_ACTIONS.setName, payload: {name: event.target.value}}
    return (
        <div>
            <label>
                Name:
                <input
                    value={state.name}
                    onChange={}
                    type="text"
                />
            </label>
        </div>
```

```
export const NewReviewForm = () => {
    const [state, dispatch] = useReducer(reducer, initialState);
    const onNameChange = (event) => dispatch(
        {type: FORM_ACTIONS.setName, payload: {name: event.target.value}}
    return (
        <div>
            <label>
                Name:
                <input
                    value={state.name}
                    onChange={onNameChange}
                    type="text"
                />
            </label>
        </div>
```

```
export const NewReviewForm = () => {
    const [state, dispatch] = useReducer(reducer, initialState);
    const onNameChange = (event) => dispatch(
        {type: FORM_ACTIONS.setName, payload: {name: event.target.value}}
    return (
        <div>
            <label>
                Name:
                <input
                    value={state.name}
                    onChange={onNameChange}
                    type="text"
                />
            </label>
        </div>
```

Красота...

```
export const NewReviewForm = () => {
    const [state, dispatch] = useReducer(reducer, initialState);
    const onNameChange = (event) => dispatch(
        {type: FORM ACTIONS.setName, payload: {name: event.target.value}}
    return (
        <div>
            <label>
                Name:
                <input
                    value={state.name}
                    onChange={onNameChange}
                    type="text"
                />
            </label>
        </div>
```

Попробуем установить фокус на инпут

```
export const NewReviewForm = () => {
    const [state, dispatch] = useReducer(reducer, initialState);
    const onNameChange = (event) => dispatch(
        {type: FORM ACTIONS.setName, payload: {name: event.target.value}}
    return (
        <div>
            <label>
                Name:
                <input
                    value={state.name}
                    onChange={onNameChange}
                    type="text"
            </label>
        </div>
```

А как взаимодействовать с DOM?

```
export const NewReviewForm = () => {
    const [state, dispatch] = useReducer(reducer, initialState);
    const onNameChange = (event) => dispatch(
        {type: FORM ACTIONS.setName, payload: {name: event.target.value}}
    return (
        <div>
            <label>
                Name:
                <input
                    value={state.name}
                    onChange={onNameChange}
                    type="text"
            </label>
        </div>
```

useRef

```
export const NewReviewForm = () => {
    const ref = useRef();
    return (
        <div>
            <label>
                Name:
                <input
                    type="text"
                />
            </label>
        </div>
```

useRef

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    return (
        <div>
            <label>
                Name:
                <input
                   type="text"
                />
            </label>
        </div>
```

useRef можно использовать как хранилище!

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    return (
        <div>
            <label>
                Name:
                <input
                    type="text"
                />
            </label>
        </div>
```

useRef

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    return (
        <div>
            <label>
                Name:
                <input
                    ref={ref}
                    type="text"
                />
            </label>
        </div>
```

useRef

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    ref.current?.focus();
    return (
        <div>
            <label>
                Name:
                <input
                    ref={ref}
                    type="text"
                />
            </label>
        </div>
```

Это так не работает(

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    ref.current?.focus();
    return (
        <div>
            <label>
                Name:
                <input
                    ref={ref}
                    type="text"
                />
            </label>
        </div>
```

Нам нужен новый гер... ХУК!

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    ref.current?.focus();
    return (
        <div>
            <label>
                Name:
                <input
                    ref={ref}
                    type="text"
                />
            </label>
        </div>
```

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    useEffect()
    return (
        <div>
            <label>
                Name:
                <input
                    ref={ref}
                    type="text"
                />
            </label>
        </div>
```

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    useEffect(() => \{\})
    return (
        <div>
            <label>
                Name:
                <input
                     ref={ref}
                    type="text"
                 />
            </label>
        </div>
```

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    useEffect(() => {}, [])
    return (
        <div>
            <label>
                Name:
                <input
                    ref={ref}
                    type="text"
                />
            </label>
        </div>
```

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    useEffect(() => {}, [...])
    return (
        <div>
            <label>
                Name:
                <input
                    ref={ref}
                    type="text"
                />
            </label>
        </div>
```

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}

    useEffect(() => {
        if (ref.current) {
            ref.current.focus();
        }
    }, [])

    return (...)
}
```

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}

    useEffect(() => {
        if (ref.current) {
            ref.current.focus();
        }
    }, [])

    return (...)
}
```

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    useEffect(() => {
        if (ref.current) {
            ref.current.focus();
        return () => {}
    }, [])
    return (...)
```

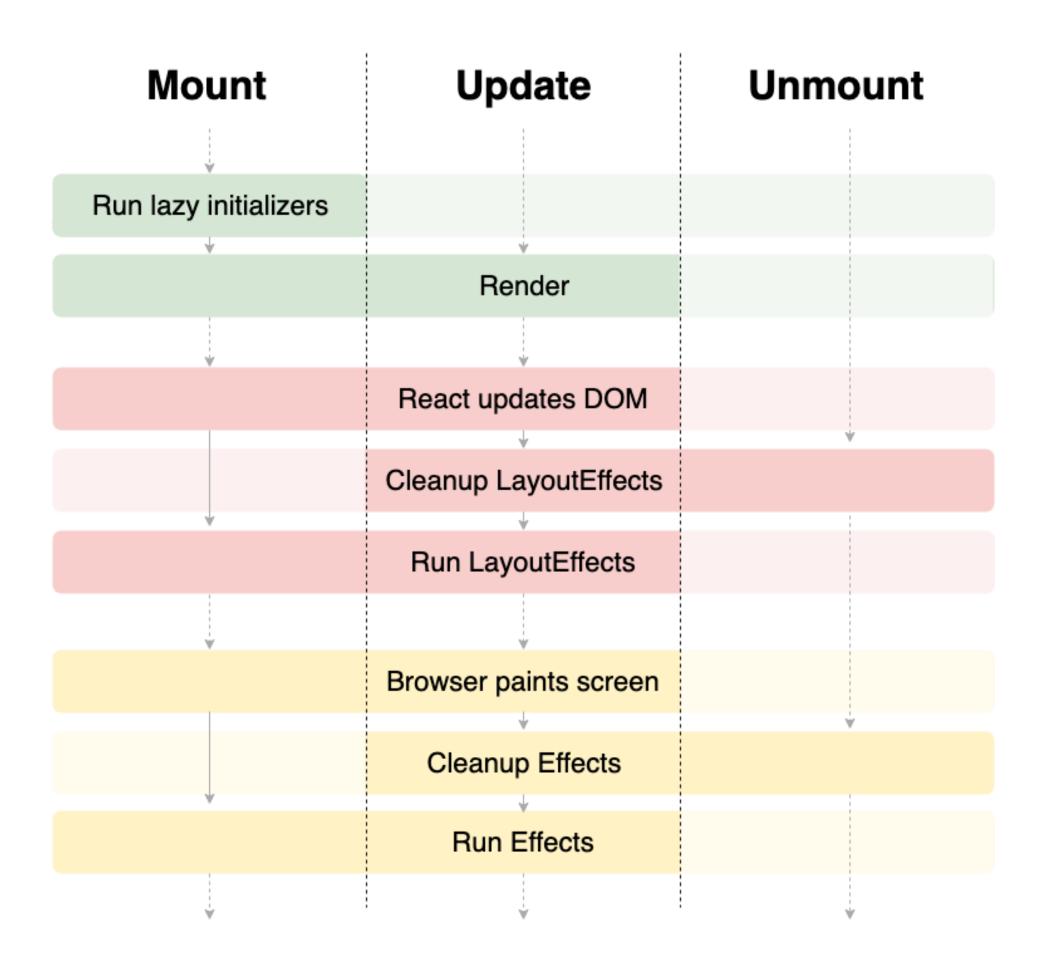
useLayoutEffect

```
export const NewReviewForm = () => {
    const ref = useRef(); // {current: ...}
    useLayoutEffect(() => {
        if (ref.current) {
            ref.current.focus();
        return () => {}
    }, [])
    return (...)
```

useLayoutEffect u useEffect

React Hook Flow Diagram

v1.3.1 github.com/donavon/hook-flow



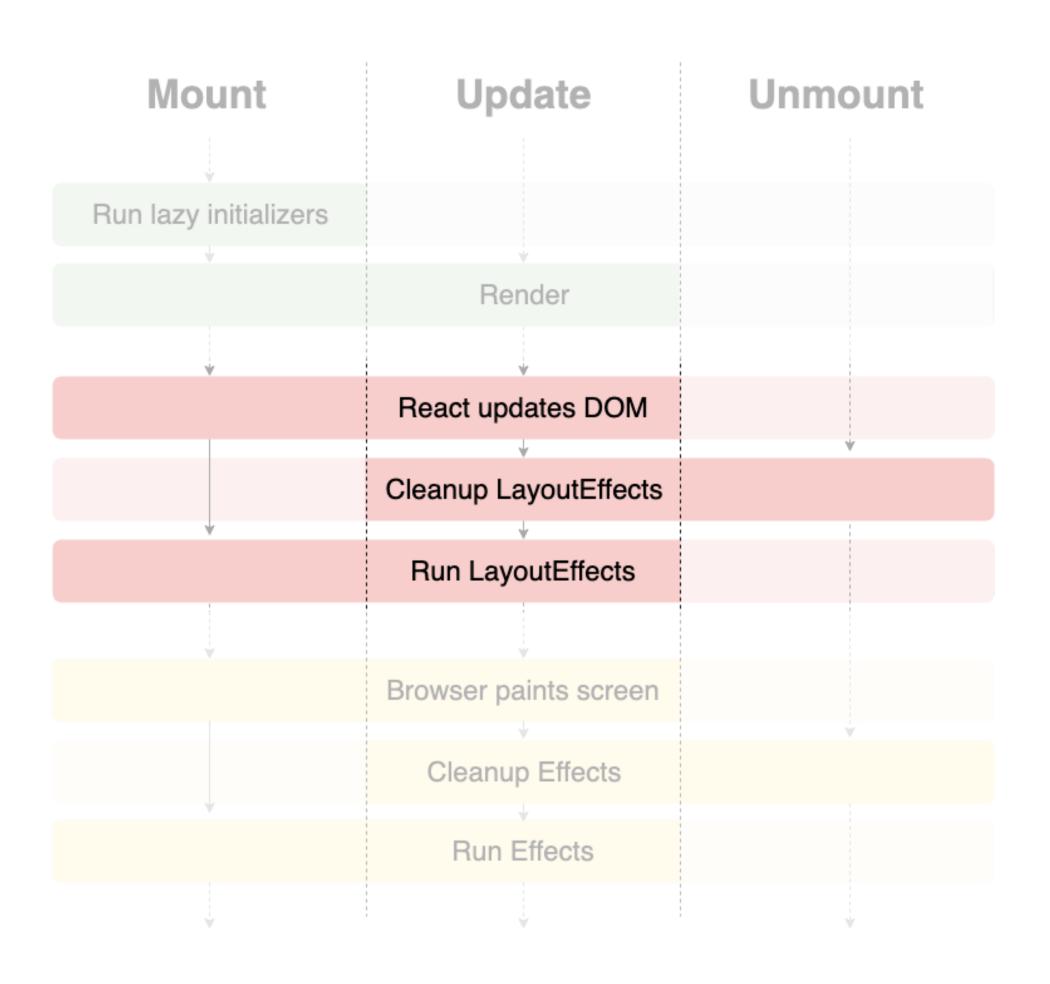
Notes:

- 1. Updates are caused by a parent re-render, state change, or context change.
- 2. Lazy initializers are functions passed to useState and useReducer.

useLayoutEffect u useEffect

React Hook Flow Diagram

v1.3.1 github.com/donavon/hook-flow



Notes:

- 1. Updates are caused by a parent re-render, state change, or context change.
- 2. Lazy initializers are functions passed to useState and useReducer.

useLayoutEffect u useEffect

React Hook Flow Diagram

v1.3.1 github.com/donavon/hook-flow



Notes:

- 1. Updates are caused by a parent re-render, state change, or context change.
- 2. Lazy initializers are functions passed to useState and useReducer.

Без useEffect и useRef

```
export const NewReviewForm = () => {
    return
        <div>
            <label>
                Name:
                <input
                    type="text"
                />
            </label>
        </div>
```

Без useEffect и useRef

```
export const NewReviewForm = () => {
    const setFocus = useCallback(
        (element) => {element.focus()},
    return (
        <div>
            <label>
                Name:
                <input
                    ref={setFocus}
                    type="text"
                />
            </label>
        </div>
```

А что со стилями?

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import {App} from "./App";

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App/>);
```

А что со стилями?

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import {App} from "./App";

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App/>);
```

Index.css

```
body {
 margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
```

> node_modules library root > public ✓ src components FilmDetails → FilmInfo FilmInfo.jsx styles.css NewReviewForm > Review > Reviews > hooks App.jsx index.css index.js setupTests.js gitignore. package.json package-lock.json README.md

> node_modules library root > public ✓ src components FilmDetails FilmInfo FilmInfo.jsx styles.module.css NewReviewForm > Review Reviews > hooks App.jsx index.css index.js setupTests.js agitignore package.json package-lock.json README.md

```
.title {
    color: red;
}
```

```
{
title: 'styles_title__5hfTN'
}
```

```
title: 'styles_title__5hfTN'
}
```

```
title: 'styles_title__5hfTN'
```

```
<NewReviewForm className={styles.reviewForm}/>
export const NewReviewForm = ({ className }) => {
   return <div className={className}>...</div>}
```