

Frontend & Backend разработка

Лекция 2

Формы ReactJS

Р.В. Шамин

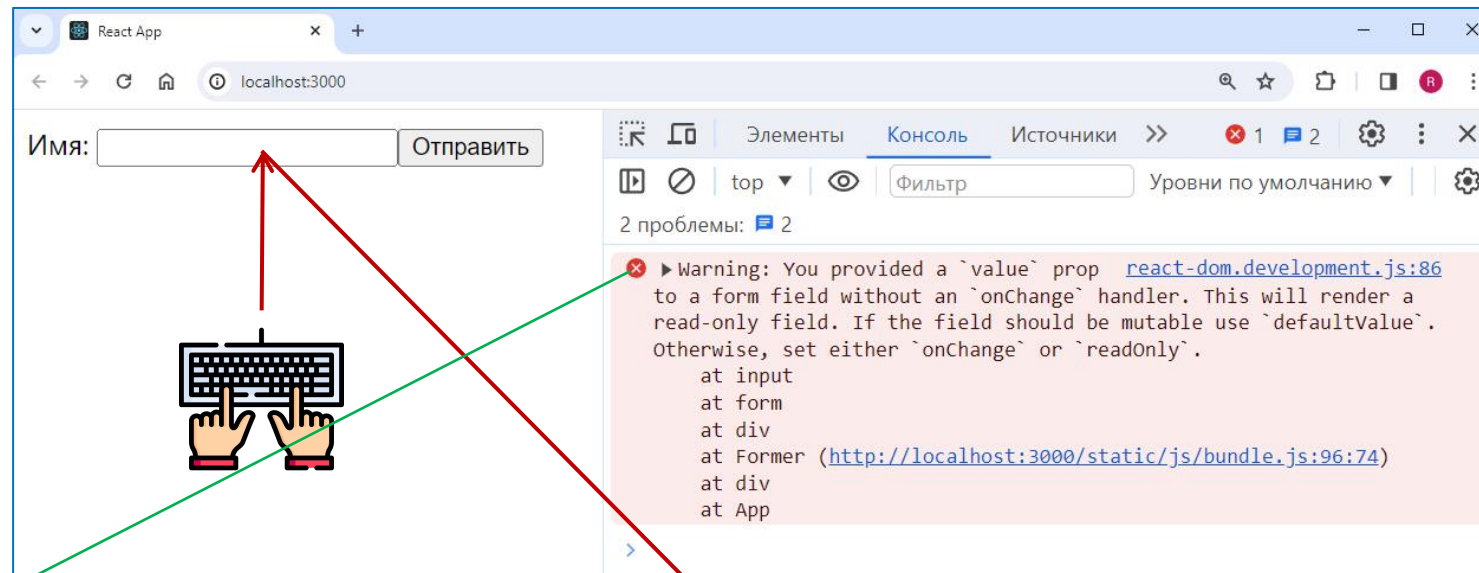
профессор кафедры индустриального программирования

Формы в ReactJS

В ReactJS формы реализуются по принципу Ajax без перезагрузки страницы.

Для этого значение каждого поля должно быть состоянием.

```
1 import { useState } from "react"
2
3 export default function Former() {
4   const [name, setName] = useState("")
5
6   return (
7     <div>
8       <form>
9         <label>Имя: </label>
10        <input type="text" value={name}/>
11        <input type="submit" value="Отправить"/>
12      </form>
13    </div>
14  )
15 }
```



ничего не получается ввести!

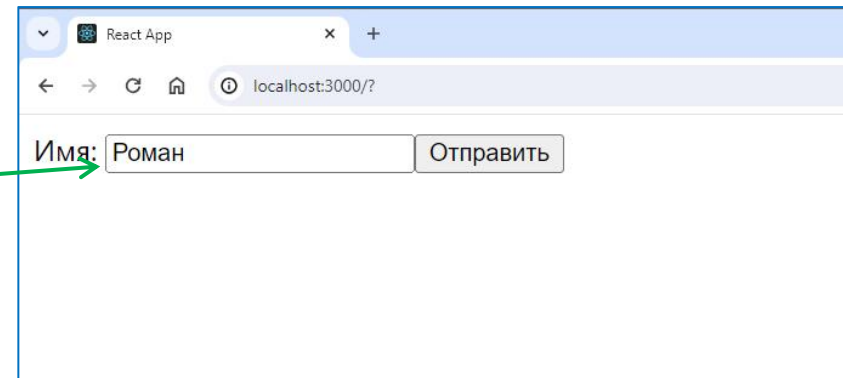
ReactJS сообщает об ошибке, потому что не определен метод onChange

Формы в ReactJS

Добавим метод для обработки события onChange, в котором мы должны обновить состояние.

```
1  import { useState } from "react"
2
3  export default function Former() {
4    const [name, setName] = useState("")
5
6    function handlerChange(e) {
7      setName(e.target.value)
8    }
9
10   return (
11     <div>
12       <form>
13         <label>Имя: </label>
14         <input type="text" value={name} onChange={handlerChange}/>
15         <input type="submit" value="Отправить"/>
16       </form>
17     </div>
18   )
19 }
```

ввод работает!



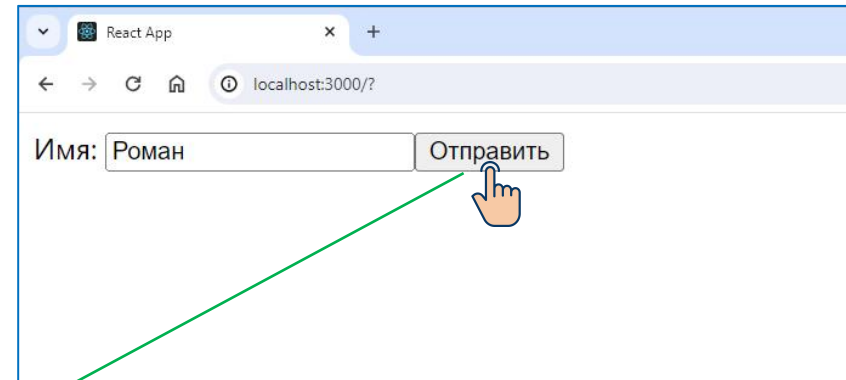
Однако при нажатии кнопки «Отправить» происходит перезагрузка страницы, что не в ходит в наши планы.

Для исключения перезагрузки страницы необходимо прослушивать событие onSubmit у формы.

Формы в ReactJS

```
1 import { useState } from "react"
2
3 export default function Former() {
4   const [name, setName] = useState("")
5
6   function handlerChange(e) {
7     setName(e.target.value)
8   }
9
10  function handlerSubmit(e) {
11    e.preventDefault();
12  }
13
14  return (
15    <div>
16      <form onSubmit={handlerSubmit}>
17        <label>Имя: </label>
18        <input type="text" value={name} onChange={handlerChange}/>
19        <input type="submit" value="Отправить"/>
20      </form>
21    </div>
22  )
23 }
```

этот метод отменяет обработку
события по умолчанию

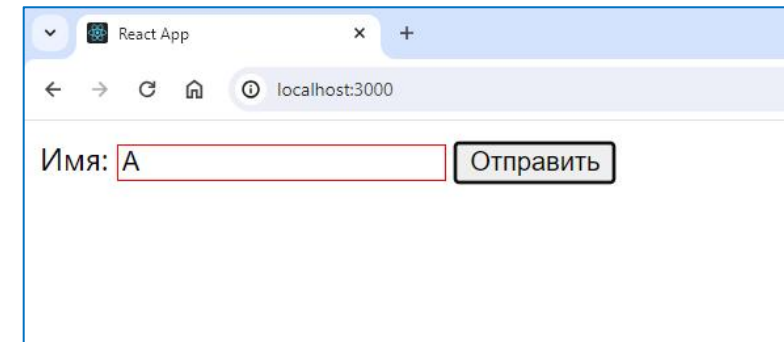
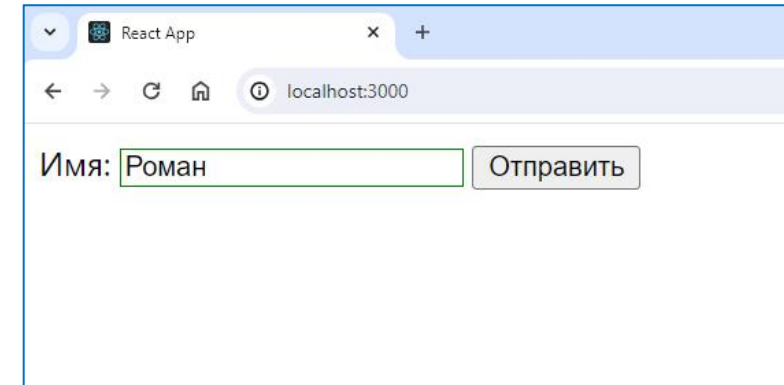


перезагрузки формы не происходит

Валидация форм в ReactJS

При работе с формами необходимо контролировать корректность данных которые вводит пользователь. Такую проверку нужно осуществлять «на лету».

```
1  import { useState } from "react"
2
3  export default function Form() {
4    const [name, setName] = useState("")
5    const [isValid, setIsValid] = useState(false) ← флаг корректности ввода
6    const [nameColor, setNameColor] = useState("") ← цвет рамки поля ввода
7
8    function handlerChange(e) {
9      setName(e.target.value)
10     if (e.target.value.length > 2) {
11       setIsValid(true)
12     } else {
13       setIsValid(false)
14     }
15   }
16
17   function handlerSubmit(e) {
18     e.preventDefault();
19     if (!isValid) {
20       setNameColor("1px solid red")
21     } else {
22       setNameColor("1px solid green")
23     }
24   }
25
26   return (
27     <div>
28       <form onSubmit={handlerSubmit}>
29         <label>Имя: </label>
30         <input type="text" value={name} onChange={handlerChange} style={{border:nameColor}}/>
31         &nbsp;
32         <input type="submit" value="Отправить"/>
33       </form>
34     </div>
35   )
36 }
```



Элементы управления и пользовательского интерфейса

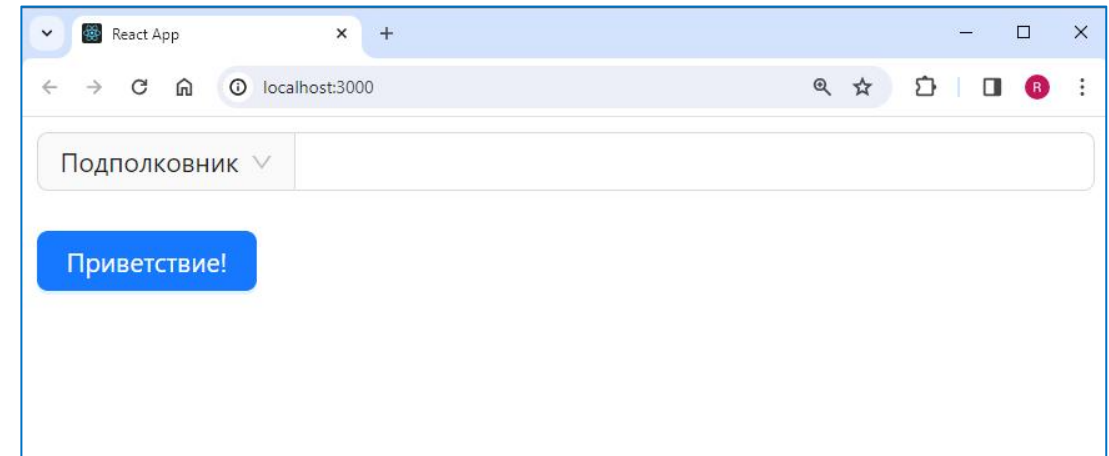
Для ReactJS разработано большое количество сторонних библиотек и фреймворков, которые реализуют компоненты пользовательского интерфейса.

Мы рассмотрим библиотеку Ant Design. Установим Ant Design:



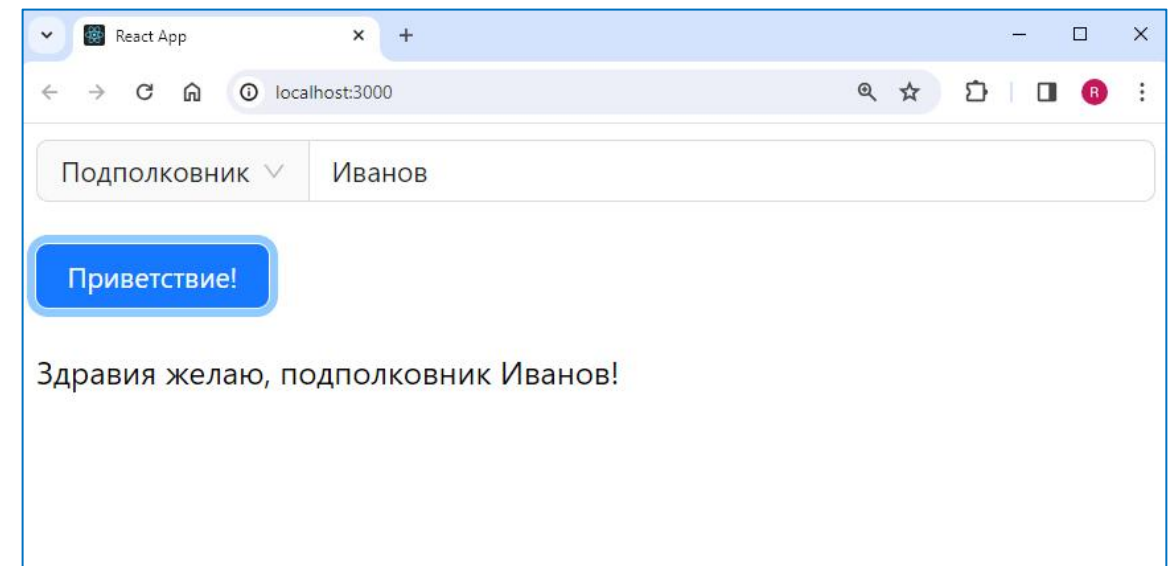
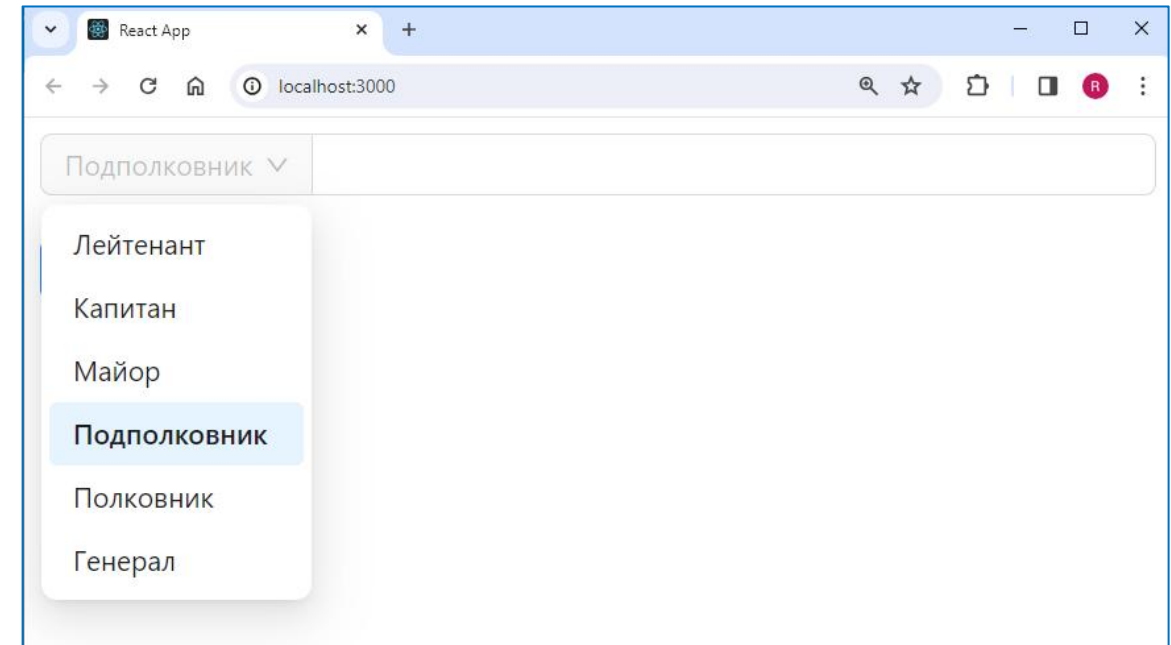
```
PS C:\Users\User> npm install antd
```

```
1  import { Input, Select } from "antd";
2  const { Option } = Select;
3
4  export default function AntHello() {
5    const selectBefore = (
6      <Select defaultValue="Подполковник">
7        <Option value="Лейтенант">Лейтенант</Option>
8        <Option value="Капитан">Капитан</Option>
9        <Option value="Майор">Майор</Option>
10       <Option value="Подполковник">Подполковник</Option>
11       <Option value="Полковник">Полковник</Option>
12       <Option value="Генерал">Генерал</Option>
13     </Select>
14   );
15
16   return (
17     <div>
18       <Input addonBefore={selectBefore}/>
19     </div>
20   )
21 }
```

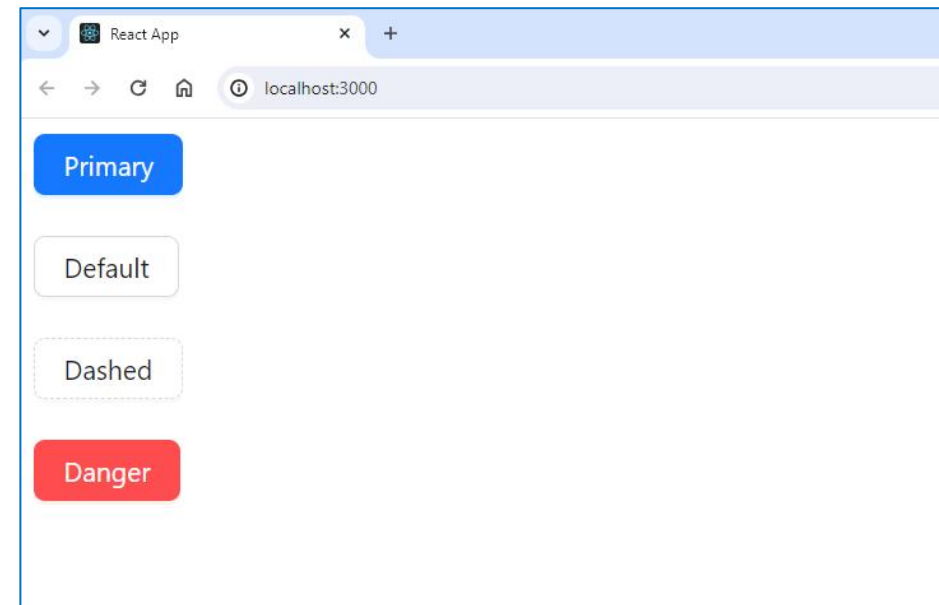


Элементы управления и пользовательского интерфейса

```
1 import { Input, Select, Button } from "antd";
2 import { useState } from "react";
3 const { Option } = Select;
4
5 export default function AntHello() {
6   const [name, setName] = useState()
7   const [title, setTitle] = useState()
8   const [hello, setHello] = useState("")
9   function handlerChangeTitle (e) {
10     setTitle(e)
11   }
12   function handlerChange(e) {
13     setName(e.target.value)
14   }
15   function handlerClick(e) {
16     setHello("Здравия желаю, " + title + " " + name + "!")
17   }
18   const selectBefore = (
19     <Select defaultValue="Подполковник" onChange={handlerChangeTitle}>
20       <Option value="лейтенант">Лейтенант</Option>
21       <Option value="капитан">Капитан</Option>
22       <Option value="майор">Майор</Option>
23       <Option value="подполковник">Подполковник</Option>
24       <Option value="полковник">Полковник</Option>
25       <Option value="генерал">Генерал</Option>
26     </Select>
27   );
28   return (
29     <div>
30       <Input addonBefore={selectBefore} onChange={handlerChange}/>
31       <br/><br/>
32       <Button type="primary" onClick={handlerClick}>Приветствие!</Button>
33       <br/><br/>
34       <span>{hello}</span>
35     </div>
36   )
37 }
```

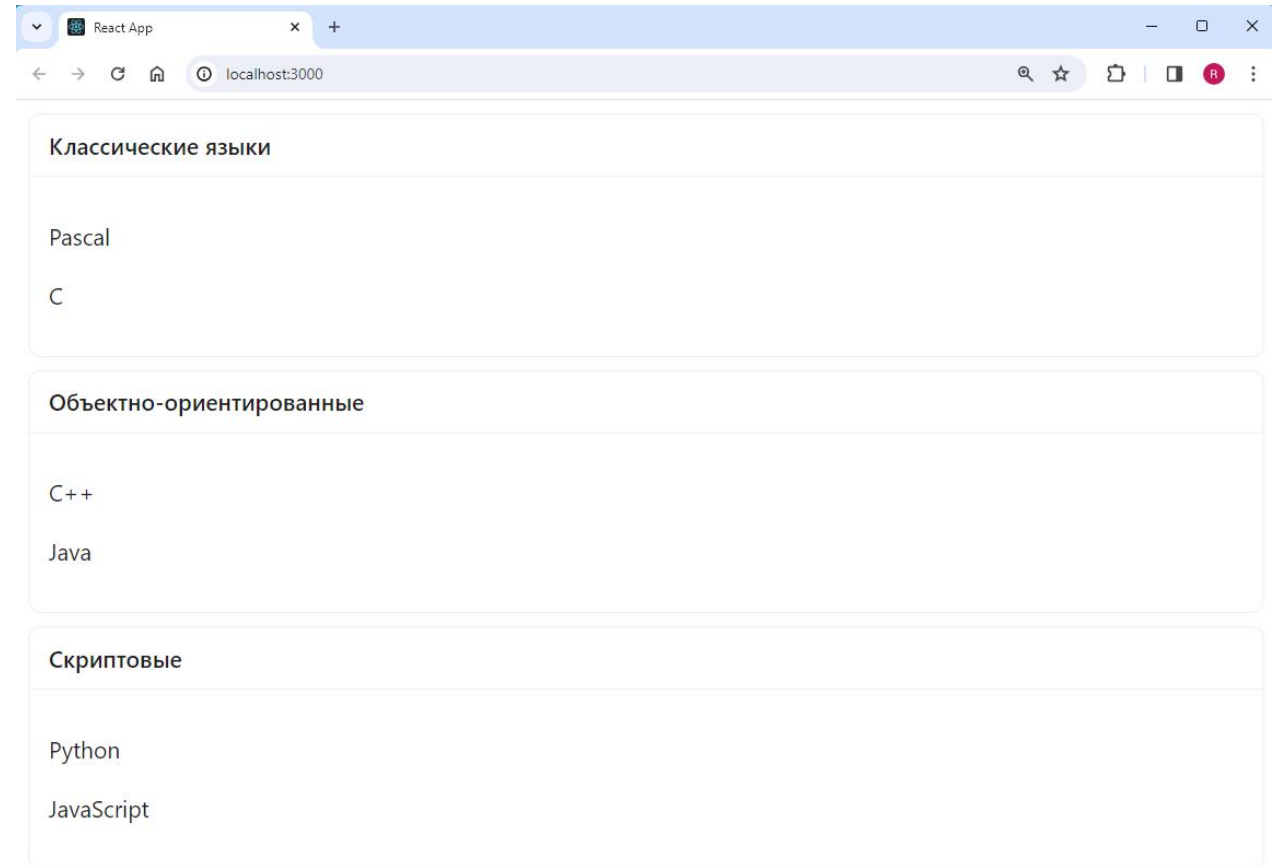


```
1 import React from 'react';
2 import { Button } from 'antd';
3
4 export default function AntButtun () {
5   return (
6     <div>
7       <Button type="primary">Primary</Button>
8       <br/><br/>
9       <Button>Default</Button>
10      <br/><br/>
11      <Button type="dashed">Dashed</Button>
12      <br/><br/>
13      <Button type="primary" danger>Danger</Button>
14    </div>
15  )
16 }
```



Элементы управления и пользовательского интерфейса

```
1 import React from 'react';
2 import { Card, Space } from 'antd';
3
4 export default function Card1() {
5   return (
6     <Space
7       direction="vertical"
8       size="small"
9       style={{display: 'flex', }}
10     >
11       <Card title="Классические языки" size="small">
12         <p>Pascal</p>
13         <p>C</p>
14       </Card>
15       <Card title="Объектно-ориентированные" size="small">
16         <p>C++</p>
17         <p>Java</p>
18       </Card>
19       <Card title="Скриптовые" size="small">
20         <p>Python</p>
21         <p>JavaScript</p>
22       </Card>
23     </Space>
24   )
25 }
```



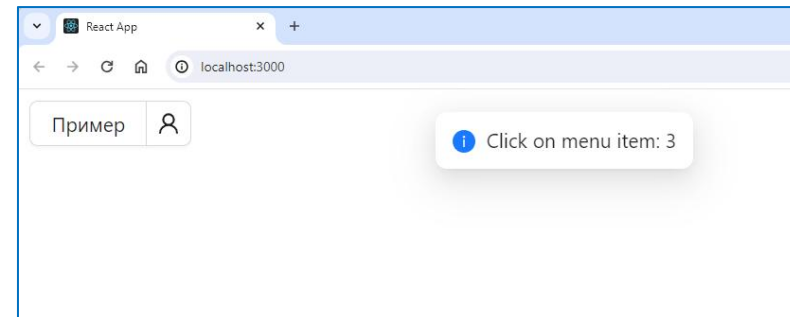
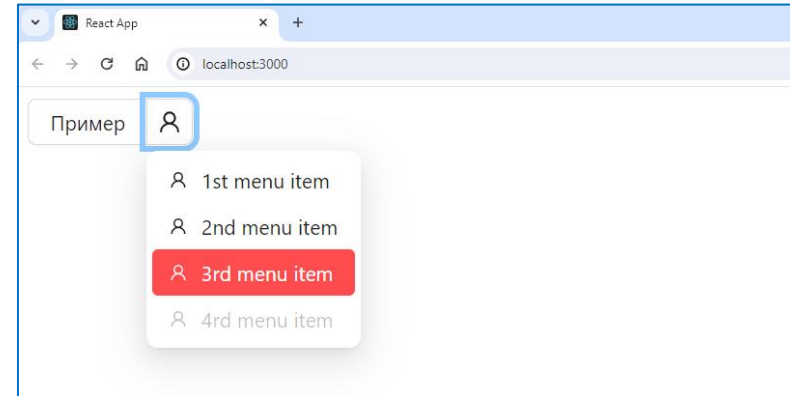
Элементы управления и пользовательского интерфейса

```
1 import { DownOutlined, UserOutlined } from '@ant-design/icons';
2 import { Button, Dropdown, message, Space, Tooltip } from 'antd';
3 const handleClick = (e) => {
4   message.info('Click on button.');
```

5 };

```
6 const handleMenuClick = (e) => {
7   message.info('Click on menu item: ' + e.key);
8 };
9 const items = [
10   {
11     label: '1st menu item',
12     key: '1',
13     icon: <UserOutlined />,
14   },
15   {
16     label: '2nd menu item',
17     key: '2',
18     icon: <UserOutlined />,
19   },
20   {
21     label: '3rd menu item',
22     key: '3',
23     icon: <UserOutlined />,
24     danger: true,
25   },
26   {
27     label: '4rd menu item',
28     key: '4',
29     icon: <UserOutlined />,
30     danger: true,
31     disabled: true,
32   },
33 ];
34 const menuProps = {
35   items,
36   onClick: handleMenuClick,
37 };
```

```
38
39 export default function Dropdown1 () {
40   return (
41     <div>
42       <Dropdown.Button menu={menuProps} icon={<UserOutlined />} onClick={handleButtonClick}>
43         Пример
44       </Dropdown.Button>
45     </div>
46   )
47 }
```



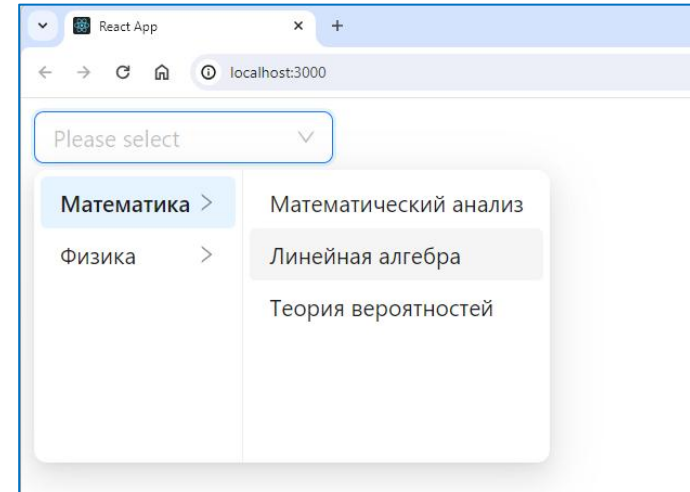
Элементы управления и пользовательского интерфейса

```
1 import { LoadingOutlined, SmileOutlined, SolutionOutlined, UserOutlined } from '@ant-design/icons';
2 import { Steps } from 'antd';
3
4 export default function Steps1 () {
5   return (
6     <Steps
7       items={[
8         {
9           title: 'Login',
10          status: 'finish',
11          icon: <UserOutlined />,
12        },
13        {
14          title: 'Verification',
15          status: 'finish',
16          icon: <SolutionOutlined />,
17        },
18        {
19          title: 'Pay',
20          status: 'process',
21          icon: <LoadingOutlined />,
22        },
23        {
24          title: 'Done',
25          status: 'wait',
26          icon: <SmileOutlined />,
27        },
28      ]}
29    />
30  )
31 }
```



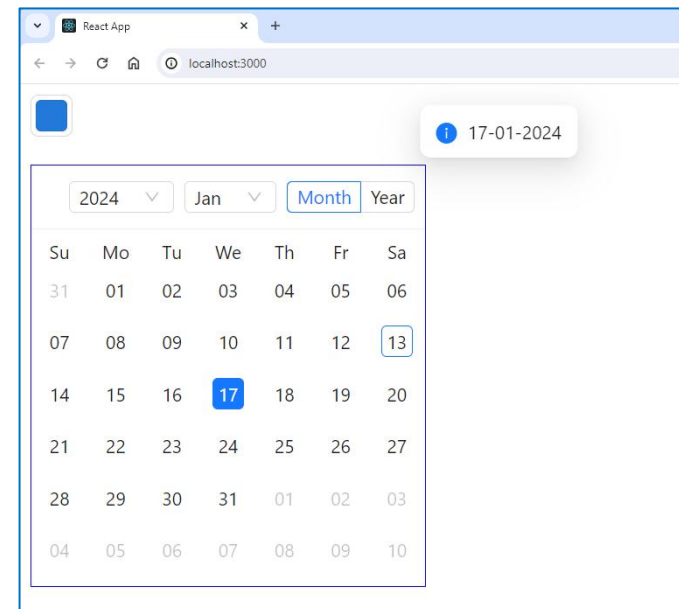
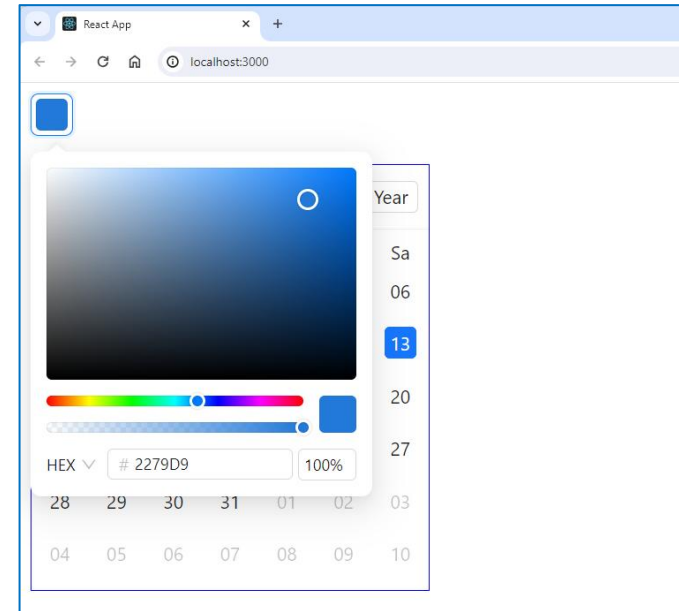
Элементы управления и пользовательского интерфейса

```
1 import { Cascader } from 'antd';
2
3 const options = [
4   {
5     value: 'Математика',
6     label: 'Математика',
7     children: [
8       {
9         value: 'Математический анализ',
10        label: 'Математический анализ',
11      },
12      {
13        value: 'Линейная алгебра',
14        label: 'Линейная алгебра',
15      },
16      {
17        value: 'Теория вероятностей',
18        label: 'Теория вероятностей',
19      },
20    ],
21  },
22  {
23    value: 'Физика',
24    label: 'Физика',
25    children: [
26      {
27        value: 'Механика',
28        label: 'Механика',
29      },
30      {
31        value: 'Термодинамика',
32        label: 'Термодинамика',
33      },
34      {
35        value: 'Оптика',
36        label: 'Оптика',
37      },
38    ],
39  },
40 ];
41 const onChange = (value) => {
42   console.log(value);
43 };
44
45 export default function Cascader1 () {
46   return (
47     <Cascader options={options} onChange={onChange} placeholder="Please select" />
48   )
49 }
```



Элементы управления и пользовательского интерфейса

```
1 import { ColorPicker, Calendar, message } from 'antd';
2
3 const onPanelChange = (value) => {
4   message.info(value.format('DD-MM-YYYY'));
5 };
6
7 export default function Picker () {
8   const wrapperStyle = {
9     width: 300,
10    border: `1px solid blue`,
11  };
12
13  return (
14    <div>
15      <ColorPicker defaultValue="#22D924" />
16      <br/><br/>
17      <div style={wrapperStyle}>
18        <Calendar fullscreen={false} onChange={onPanelChange} />
19      </div>
20    </div>
21  )
22 }
```



Хуки useEffect

При работе с формами (и не только) часто возникает такая задача: вызвать определенный код при изменении состояния.

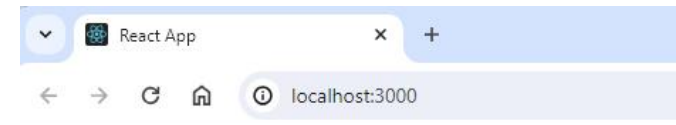
Хуки состояния `useState` вызывают рендеринг компонента, когда меняется значение состояния, но при этом мы не можем вызвать пользовательский код. Для этого необходимо использовать хук `useEffect`.

Пользовательская функция всегда вызывается при создании компонента



Пример хука useEffect

```
1 import { useEffect, useState } from "react";
2
3 export default function Effect() {
4   const [count, setCount] = useState(0)
5   const [effect, setEffect] = useState('')
6   const [clear, setClear] = useState('')
7
8   useEffect(() => {
9     setEffect('Use Effect! ' + count)
10
11     return () => {
12       setClear('Clear = ' + count)
13     }
14   }, [count])
15
16   return (
17     <div>
18       <div>{count}</div>
19       <button onClick={() => {setCount(count + 1)}}>Нажми на меня</button>
20       <div>{effect}</div>
21       <div>{clear}</div>
22     </div>
23   );
24
25 }
```



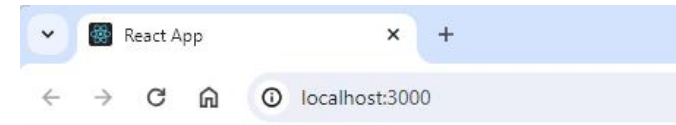
Привет!

0

Нажми на меня

Use Effect! 0

До нажатия



Привет!

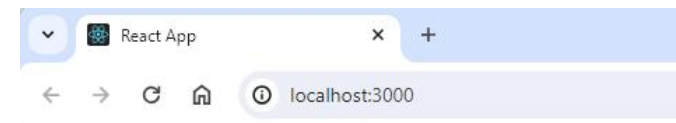
1

Нажми на меня

Use Effect! 1

Clear = 0

После первого нажатия



Привет!

2

Нажми на меня

Use Effect! 2

Clear = 1

После второго нажатия

Хук useRef

Хук useRef позволяет хранить значение какого-либо объекта в течение всего жизненного цикла компонента.

В отличие от глобальных переменных значение каждого хука будет локальным в каждом компоненте.

```
1  import { useRef } from "react";
2
3  export default function Effect() {
4    const inputRef = useRef(null)
5
6    function handleClick() {
7      inputRef.current.focus()
8    }
9
10   return (
11     <div>
12       <input type="text"/><br/>
13       <input type="text" ref={inputRef}/>
14       <button onClick={handleClick}>Фокус сюда!</button><br/>
15       <input type="text"/><br/>
16       <input type="text"/>
17     </div>
18   );
19 }
```

