



The editor shows the file `index.html` with the following content:

```
02-react-hello > <> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>React Hello World</title>
7    </head>
8    <body>
9      <!-- <div id="app"></div> -->
10     <script src="https://unpkg.com/react@18/umd/react.development.js"></script>
11     <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
12     <script>
13       const helloWorld = React.createElement('h1', null, 'Hello from React');
14       console.log(helloWorld);
15       // const container = document.getElementById('app');
16       // const root = ReactDOM.createRoot(container);
17       // root.render(helloWorld);
18     </script>
19   </body>
20 </html>
21
```

index.html X

04-jsx-hello-world > index.html > html > body > script

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>React Hello World</title>
7   </head>
8   <body>
9     <div id="app"></div>
10    <!-- Добавим Babel который отвечает за трансформацию JSX в JS -->
11    <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
12    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>
13    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
14    <script type="text/babel">
15      // type="" показывает что необходимо выполнить трансформацию кода JSX
16      const helloWorld = <h1>Hello from React</h1>; // код JSX
17      const container = document.getElementById('app');
18      const root = ReactDOM.createRoot(container);
19      root.render(helloWorld);
20    </script>
21  </body>
22 </html>
23
```

index.html X

05-react-JSX-green-button > index.html > html > body > script

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>React JSX Green Button</title>
7     <style>
8       .app {
9         text-align: center;
10      }
11
12      .green-btn {
13        color: white;
14        background-color: green;
15      }
16    </style>
17  </head>
18  <body>
19    <div id="app"></div>
20    <!-- Добавим Babel который отвечает за трансформацию JSX в JS -->
21    <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
22    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>
23    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
24
25    <script
26      src="/05-react-JSX-green-button/script.js"
27      type="text/babel"
28    ></script>
29  </body>
30 </html>
31
32 <!-- npx live-server --open=05-react-JSX-green-button -->
```

```
JS script.js X
05-react-JSX-green-button > JS script.js > ...
1  const App = () => {
2    let [buttonText, setbuttonText] = React.useState('Click me please');
3
4    const onClick = () => {
5      setbuttonText('Hello from React!!!');
6    };
7
8    return (
9      <div className="app">
10        /*
11         - в JSX классы подключаются при помощи className
12         - внутри JSX используются двойные кавычки
13         - при добавлении JS внутри JSX необходимо использовать {}
14        */
15        <button onClick={onClick}>{buttonText}</button>
16      </div>
17    );
18  };
19
20 const container = document.getElementById('app');
21 const root = ReactDOM.createRoot(container);
22 root.render(<App />);
23
```

## Подключение свойств

```
05-react-JSX-green-button > JS script.js > App
1  const App = () => {
2    const [buttonText, setbuttonText] = React.useState('Click me please');
3    const [buttonColor, setbuttonColor] = React.useState('');
4
5    const onClick = () => {
6      setbuttonText('Hello from React!!!');
7      setbuttonColor('green-btn');
8    };
9
10   return (
11     <div className="app">
12       <button className={buttonColor} onClick={onClick}>
13         {buttonText}
14       </button>
15     </div>
16   );
17 };
18
19 const container = document.getElementById('app');
20 const root = ReactDOM.createRoot(container);
21 root.render(<App />);
22
```

## props

```
05-react-JSX-green-button > JS script.js > [App] App
1  const App = ({ initialButtonText }) => {
2    const [buttonText, setbuttonText] = React.useState(initialButtonText);
3    const [buttonColor, setbuttonColor] = React.useState('');
4
5    const onButtonClick = () => {
6      setbuttonText('Hello from React!!!');
7      setbuttonColor('green-btn');
8    };
9
10   return (
11     <div className="app">
12       <button className={buttonColor} onClick={onButtonClick}>
13         {buttonText}
14       </button>
15     </div>
16   );
17 };
18
19 const container = document.getElementById('app');
20 const root = ReactDOM.createRoot(container);
21 root.render(<App initialButtonText="Click me" />);
```

## Задание 1

Передать дополнительным свойством начальные значение для классов кнопки