

# Лабораторная: создание модального окна

---

## Упражнение 1: Создание модального окна

1.0. Создайте CSS-файл **src\components\css\Modal.css** с стилями оформления модального окна

```
.Modal{
  background: rgba(118, 118, 118, 0.5);
  height: 100vh;
  width: 100vw;
  position: fixed;
  top: 0;
  left: 0;
  overflow: hidden;
  display: flex;
  align-items: center;
  justify-content: center;
  opacity: 0;
  animation: modal .1s 0s ease-in forwards;
}

@keyframes modal {
  from { opacity: 0 }
  to { opacity: 1}
}

.Modal-window{
  border-radius: 3px;
  box-shadow: 0 0 15px rgb(107, 107, 107);
  width: 50%;
  min-height: 150px;
  background: #fff;
}

.Modal-header,
.Modal-footer,
.Modal-content{
  padding: 15px 20px;
}

.Modal-input-text{
  border: 1px solid rgb(187, 187, 187);
  padding: 3px 4px;
  outline: 0;
  border-radius: 3px;
  display: block;
  width: 100%;
  margin: 4px 0;
}

.Modal-input-text:focus{
```

```
    box-shadow: 0 0 0 3px rgb(70, 172, 255);
  }
```

1.1. В файле **src\components\Modal.js** создайте описание модального окна

```
import React, { useState } from "react";
import "../css/Modal.css";

const Modal = function (props) {
  return (
    <div className="Modal">
      <div className="Modal-window">
        <h3 className="Modal-header">{props.title}</h3>

        <div className="Modal-content">
          {props.children}
        </div>

        <div className="Modal-footer">
          {props.buttons}
        </div>
      </div>
    </div>
  );
};

export default Modal;
```

1.2. В файле **src\App.js** создайте состояния: флаг для отображения окна и состояние хранящее данные ввода в `<input />`

```
const [isShow, setIsShow] = useState(false);
const [name, setName] = useState('Гость');
```

1.3. В теле компонента **App**, но до метода **render** опишите константу **modal** и присвойте ей значение компонента `<Modal />`

```
const modal = <Modal
  title="Модальное окно"
  children={
    <div>
      <p>Это диалоговое окно</p>
      <input
        value={name}
        onChange={ev => setName(ev.target.value)}
        className="Modal-input-text"
      />
    </div>
  }
/>
```

```
        </div>
      }
      buttons={<Button
        value="Заккрыть"
        onClickHandler={() => setIsShow(false)}
      />}
    />
  />
```

1.4. Измените содержимое, которое возвращает `return` внутри `App`

```
return (
  <div className="App">
    <header className="App-header">

      <Button
        value="Показать модальное окно"
        onClickHandler={() => setIsShow(true)}

      />
      <h1>Корзина: {name}</h1>

      {basketPlace}
    </header>

    { isShow ? modal : null }
  </div>
);
```

1.5. Проверьте работу модального окна. Результат должен выглядеть приблизительно как на изображении

