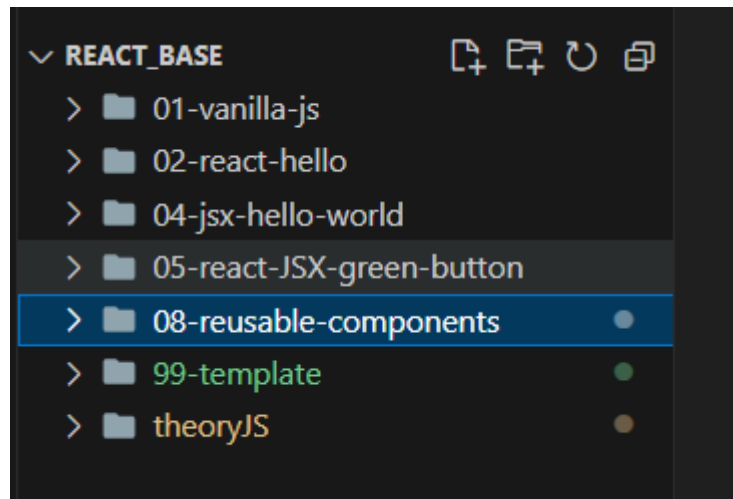


Однократное и повторное использование компонентов



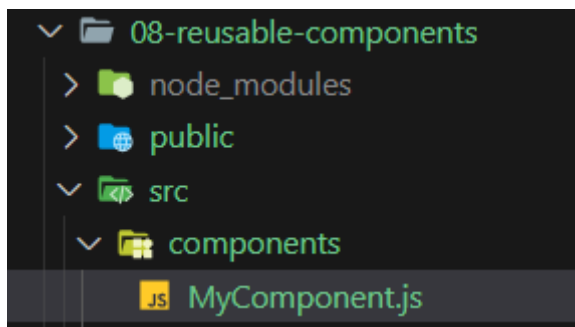
08-reusable-components

Однократное использование компонента MyComponent

```
JS App.js U X
08-reusable-components > src > JS App.js > ...
1  import './App.css';
2
3  function MyComponent() {
4    return (
5      <div>
6        <h1>Hello from the reusable component</h1>
7        <button>Like!</button>
8      </div>
9    );
10 }
11
12 function App() {
13   return (
14     <div className="App">
15       <MyComponent />
16     </div>
17   );
18 }
19
20 export default App;
```

Самостоятельно: выполнить повторное (многократное) использование компонента MyComponent

Вынесем компонент в отдельный файл



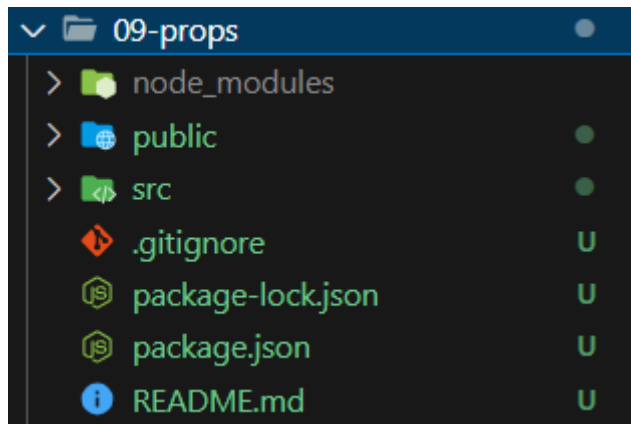
```
JS MyComponent.js U X
08-reusable-components > src > components > JS MyComponent.js > ...
1  function MyComponent() {
2      return (
3          <div>
4              <h1>Hello from the reusable component</h1>
5              <button>Like!</button>
6          </div>
7      );
8  }
9
10 export default MyComponent;
11
```

Самостоятельно:

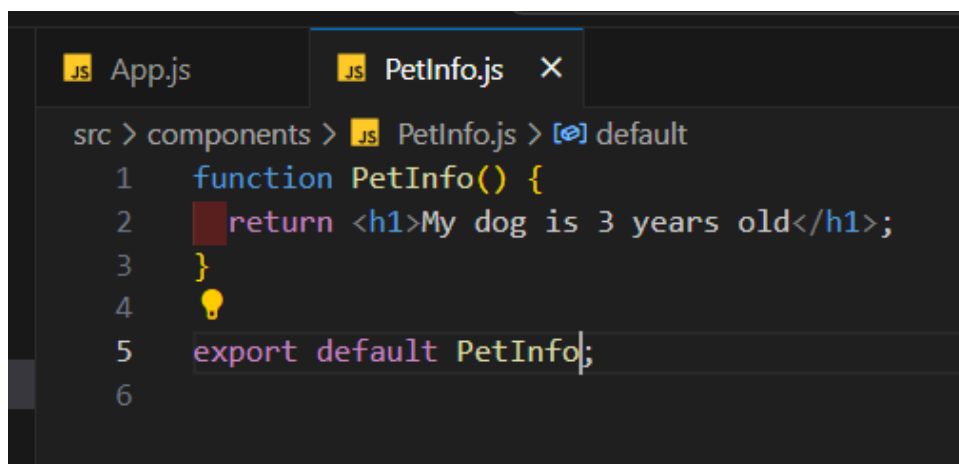
1. выполнить подключение компонента MyComponent (import)
2. создать еще один компонент (например, OtherComponent.js) и выполнить его подключение

Передача свойств компоненту

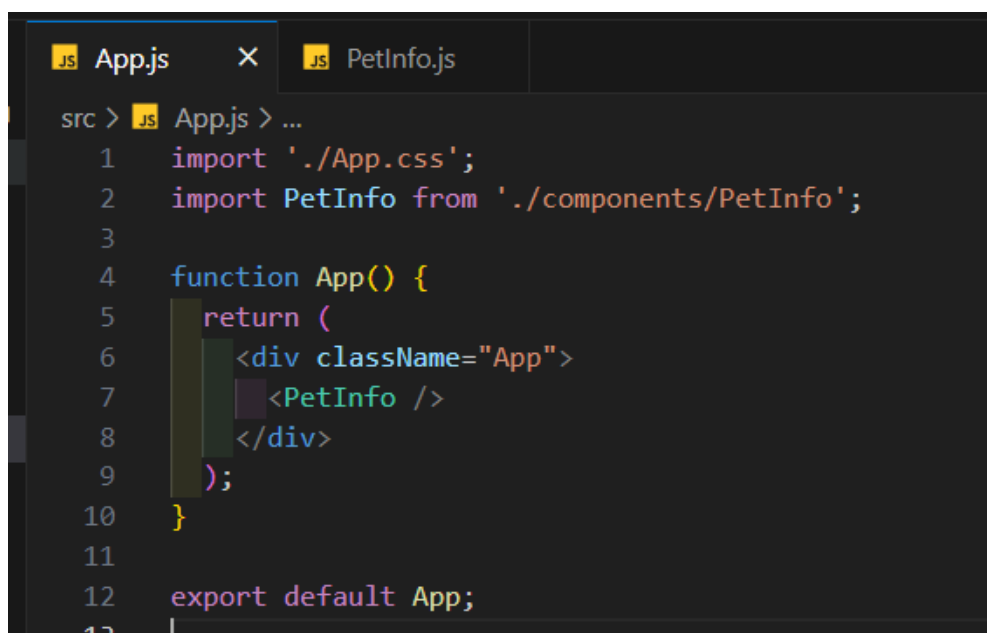
09-props



Создадим компонент PetInfo



Выполним его подключение



Теперь через props передадим в компонент свойство animal

```
<div className="App">  
  <PetInfo animal="cat" />  
</div>
```

А в компоненте PetInfo примем это свойство

```
JS App.js  JS PetInfo.js X  
src > components > JS PetInfo.js > ...  
1  function PetInfo(props) {  
2    return <h1>My {props.animal} is 3 years old</h1>;  
3  }  
4  
5  export default PetInfo;  
6
```

Самостоятельно:

1. повторно использовать этот же компонент для dog возрастом 5 лет
2. выполнить деструктуризацию свойств