

38. Компоненты в React

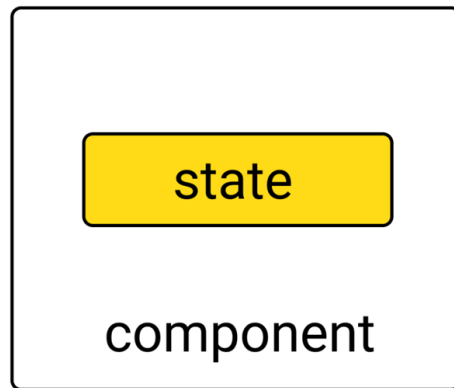
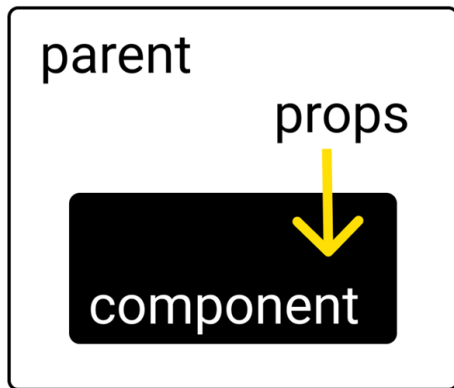
Цель:

Познакомиться со следующими понятиями и возможностями react:

- state в компонентах
- stateless и stateful компоненты
- рендеринг списков
- элементы форм
- условный рендеринг
- подъем состояния

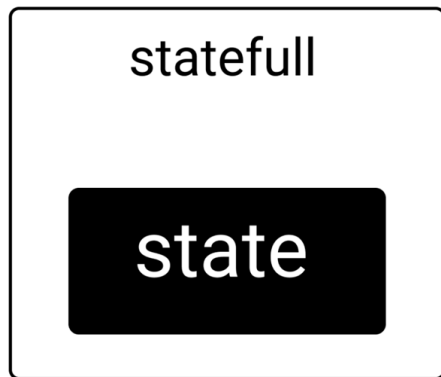
State в компонентах react:

State это JavaScript объект,
который хранит динамические внутренние данные компонента.



Stateful и stateless компоненты:

Компоненты в React содержащие внутреннее состояние — state, называются **Stateful**. Без него — **Stateless**.



Списки в react:

- можно создать коллекцию элементов и встроить её в JSX с помощью фигурных скобок {}.
- встроить `map()` в JSX

Keys:

- Ключи нужно определять непосредственно внутри массивов
- Ключи должны быть уникальными среди соседних элементов
- Ключи служат подсказками для React, но они никогда не передаются в ваши компоненты

```
<ul>
  {users.map(({ name, id }) => (
    <li key={id}>{name}</li>
  ))}
</ul>
```

Работа с формами. Input:

Input:

- value
- checked
- type
- onChange

```
<label>  
  Работает:  
  <input  
    name="job"  
    type="checkbox"  
    checked={user.hasJob}  
    onChange={handleInputChange}  
  />  
</label>
```

Работа с формами. Select:

```
<select value={myPet} onChange={handleChange}>  
  <option value="Cat">Cat</option>  
  <option value="Dog">Dog</option>  
  <option value="Parrot">Parrot</option>  
</select>
```


Условный рендеринг. Переменные-элементы:

Переменные-элементы: объявить переменную и сохранить туда элемент и условно рендерить компонент if-выражением

```
const Avatar = (props) => {  
  const hasAvatar = props.avatar;  
  if (hasAvatar) {  
    return <UserAvatar avatar={props.avatar} />;  
  }  
  return <DefaultAvatar />;  
};
```

Условный рендеринг. Логический оператор &&:

Встроенные условия if с логическим оператором &&: можно использовать логический оператор && заключив его в фигурные скобки

```
const PersonalInfo = (props) => {  
  const { name, job, pets } = props;  
  
  return (  
    <>  
      <p>My name: {name}</p>  
      <p>Job: {job}</p>  
      {pets?.length && <p>I have {pets.length} pets.</p>}  
    </>  
  );  
};
```

Условный рендеринг. if-else:

Встроенные условия if-else с тернарным оператором: можно воспользоваться тернарным оператором `condition ? true : false`, также заключив его в фигурные скобки

```
<>
  { isAuthorized ? <UserPage/> : <LoginPage/> }
</>
```

Предотвращение рендеринга компонента:

```
const Notification = (props) => {  
  if (!props.message) {  
    return null;  
  }  
  
  return (  
    <p>  
      {message}  
    </p>  
  );  
}
```

Предотвращение рендеринга компонента:

```
const Notification = (props) => {  
  if (!props.message) {  
    return null;  
  }  
  
  return (  
    <p>  
      {message}  
    </p>  
  );  
}
```

Подъем состояния:

