

React: Props

НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

Повторим;)

■ Что такое React компонент?

■ Для чего используется JSX?

■ Какие виды экспорта и импорта вы знаете?

ЦЕЛЬ

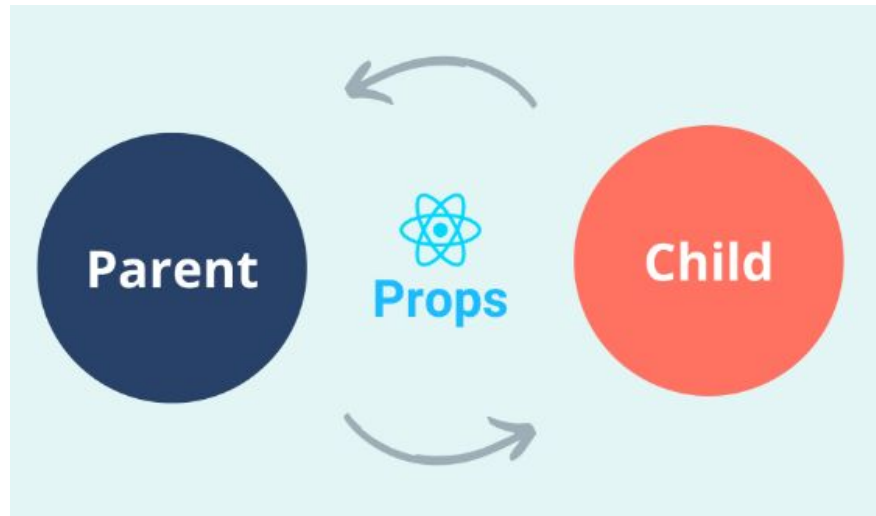
Изучить работу с props компонента

ПЛАН ЗАНЯТИЯ



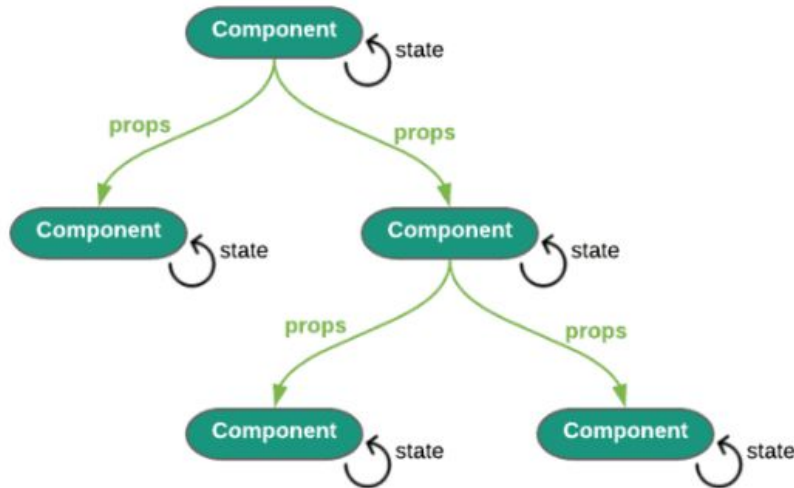
1. Props
2. Props Children

Props



Когда React видит элемент, представляющий пользовательский компонент, он передаёт JSX-атрибуты этому компоненту в виде единственного объекта. Мы называем этот объект «props».

Props используются для передачи данных от родительских компонентов дочерним компонентам. Это один из основных механизмов передачи данных в React.



Пример использования props

Дочерний компонент

Значения для переменных в JSX мы берем из объекта props

```
import React from 'react';

const ChildComponent = (props) => {
  return (
    <div>
      <p>Name: {props.name}</p>
      <p>Age: {props.age}</p>
    </div>
  );
}

export default ChildComponent;
```


Пример использования props

Родительский компонент

ParentComponent передает name и age в ChildComponent через props.

```
import React from 'react';
import ChildComponent from './ChildComponent';

const ParentComponent = () => {
  return (
    <ChildComponent name="John" age={30} />
  );
}

export default ParentComponent;
```

Несколько важных моментов о props :

1. **Props только для чтения (read-only):** Компонент не должен изменять свои собственные props.



Несколько важных моментов о props :

2. Деструктуризация Props: В компонентах часто используется деструктуризация для более удобного доступа к props.

```
const ChildComponent = ({ name, age }) => {  
  return (  
    <div>  
      <p>Name: {name}</p>  
      <p>Age: {age}</p>  
    </div>  
  );  
};
```

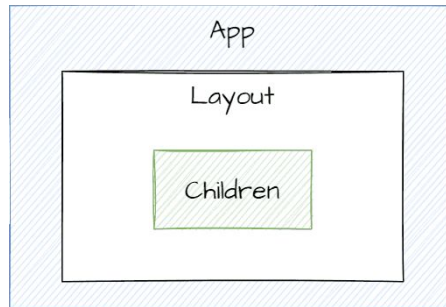


Несколько важных моментов о props :

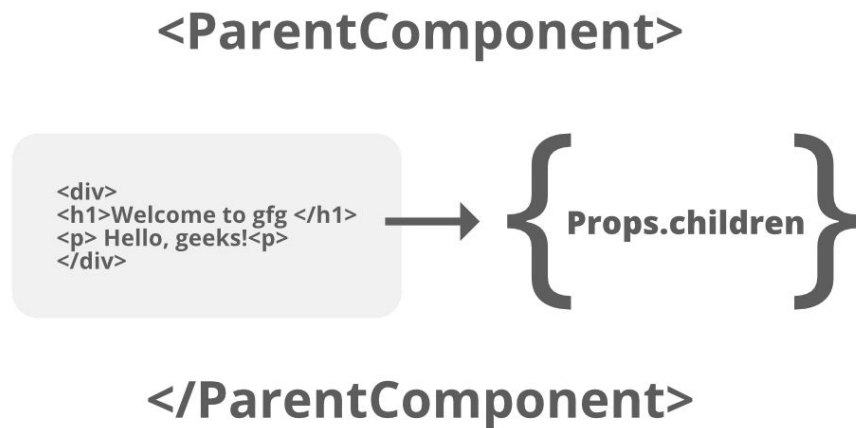
3. Default Props: Компонент может иметь значения по умолчанию для props, которые будут использоваться, если соответствующие значения не были переданы.

```
const ChildComponent = ({ name = 'Default Name', age = 25 }) => {  
  return (  
    <div>  
      <p>Name: {name}</p>  
      <p>Age: {age}</p>  
    </div>  
  );  
};
```

Children



В React children представляет собой специальный проп, который позволяет передавать компоненту содержимое между открывающим и закрывающим тегами компонента. Это особенно полезно, когда вам нужно внедрить дочерние элементы внутрь компонента без явного передачи их как отдельных props.



Пример использования

Дочерний компонент

Добавляем специальный проп children после заголовка в компонент ChildComponent

```
const ChildComponent = (props) => {  
  return (  
    <div>  
      <h1>Дочерний компонент</h1>  
      {props.children}  
    </div>  
  );  
}
```


Пример использования props

Родительский компонент

В родительском компоненте App, компонент `<ChildComponent>` записан с открывающимся и закрывающимся тегом между которыми добавлены элементы `<p>` и `<button>`

```
import ChildComponent from './ChildComponent';

const App = () => {
  return (
    <ChildComponent>
      <p>Этот текст будет дочерним элементом.</p>
      <button>Кнопка</button>
    </ChildComponent>
  );
}
```

Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH