

### **//props**

Bu Componentlarga parameter berish uchun ishlatiladi. Componentga props berish uchun avval rootga boglanga asosiy fayilga boglash kerak. Asosiy fayilga chaqirilgan componentga (props) attribute sifatida beriladi hamda obyekt kurinishida malumotlarni berish kerak (title="jdnhshui") hamda props obyekt malumot qaytaradi.attribute qiymat sifatida tutli xil turdagi malumot berish mumkin obyekt funcshin.

### **//children**

BU bir component boshqa componentlar va elementlar uchun ote element sifatida ishlatish uchun ishlatiladi. Har doim children so'zi bilan yoziladi va doim destruction qilib olinadi.

```
export const Post = ({ children }) => {  
  return (  
    <section>  
      ...  
      <div>{children}</div>  
      ...  
    </section>  
  ) ;  
};
```

### **//HOOK**

Hookni function ichida ishlatib yoki chaqirib bo'lmaydi

Hookni ichida ishlatish mumkin

Hook use so'zi bilan boshlanadi

### **//useState**

bu React Hook bo'lib, u funksional komponentlarda **state (holat)** bilan ishlash imkonini beradi. Ya'ni, agar komponent ichida ma'lumotni saqlash va uni yangilash da foydalaniladi

import React, { useState } from 'react';

function Counter() {

const [count, setCount] = useState(0); // boshlang'ich qiymat 0

const increase = () => {

setCount(count + 1);

};

return (

<div>

<h2>Hisoblagich: {count}</h2>

<button onClick={increase}>+1</button>

</div>

);

}export default Counter;

### Mapping

React-da `map()` funksiyasi juda **ko'p ishlatiladi**, ayniqsa **ro'yxat (array)** ichidagi ma'lumotlarni ekranga chiqarishda. Bu — massivdagi har bir element uchun alohida JSX element hosil qilishga imkon beradi.

`import React from 'react';`

```
function NameList() {  
  const names = ['Ali', 'Vali', 'Gulbahor'];
```

```
  return (  
    <div>  
      <h2>Ismlar ro'yxati:</h2>  
      <ul>  
        {names.map((name, index) => (  
          <li key={index}>{name}</li>  
        ))}  
      </ul>  
    </div>  
  );  
}
```

`export default NameList;`

### Conditional Rendering

Bu `if`, `&&`, va `?:` operatorlari Reactda turli holatlarda komponentlarni ko'rsatish yoki yaratish uchun ishlatiladi.

If SHarti bilan rendering

If operatori biolan render qilishda komponentning ichida yoki undan tashqarida ishlatish mumkin.

`Function MyCom({isLogin}){`

```
  if(isLogin) {  
    return <h1> Welcome, User!</h1>  
  }else{  
    return <h1> Plase , Login!</h1>  
  }  
}
```

Shart true bulsa Welcome, User! Bu chiqadiaks holada plese login

### // inlineStyle

Reatda inline style yozish

```
Function App () {  
  return {  
    <div>  
      <h1 style={{color: "red"}}> hello Style!</h1>  
    </div>
```

2  
2