

React - Props

Topics : [React JS](#)

Written on [January 02, 2024](#)

In React, props (short for properties) are a way to pass data from a parent component to a child component. Props are immutable, meaning that the child component cannot modify the values received as props. They are used to make components more flexible and reusable.

Here's how you can use props in React:

Passing Props:

1. **Parent Component:** In the parent component, when you render the child component, you can pass data to it by providing attributes with values.

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

const ParentComponent = () => {
  const message = "Hello from parent!";

  return (
    <div>
      <ChildComponent message={message} />
    </div>
  );
};

export default ParentComponent;
```

2. **Child Component:** In the child component, you can access the passed data through the props object.

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

const ChildComponent = (props) => {
  return (
    <div>
      <p>{props.message}</p>
    </div>
  );
};
```

```
);  
};  
  
export default ChildComponent;
```

Default Props:

You can also provide default values for props in case they are not passed from the parent component.

```
// ChildComponent.jsx  
import React from 'react';  
  
const ChildComponent = (props) => {  
  return (  
    <div>  
      <p>{props.message}</p>  
    </div>  
  );  
};  
  
ChildComponent.defaultProps = {  
  message: "Default Message"  
};  
  
export default ChildComponent;
```

Destructuring Props:

You can use destructuring to make the component code cleaner, especially when dealing with multiple props.

```
// ChildComponent.jsx  
import React from 'react';  
  
const ChildComponent = ({ message }) => {  
  return (  
    <div>  
      <p>{message}</p>  
    </div>  
  );  
};  
  
export default ChildComponent;
```

Children Prop:

The props object also has a special property called children, which represents the content between the opening and closing tags of the component.

```
// ParentComponent.jsx  
import React from 'react';
```

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

```
const ParentComponent = () => {  
  return (  
    <div>  
      <ChildComponent>  
        This is passed as the 'children' prop  
      </ChildComponent>  
    </div>  
  );  
};
```

```
export default ParentComponent;
```

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

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

```
export default ChildComponent;
```

Prop Types:

While not mandatory, it is a good practice to define the expected types of props using PropTypes. This helps catch potential errors and provides documentation for the component.

```
import PropTypes from 'prop-types';
```

```
const ChildComponent = ({ message }) => {  
  return (  
    <div>  
      <p>{message}</p>  
    </div>  
  );  
};
```

```
ChildComponent.propTypes = {  
  message: PropTypes.string.isRequired,  
};
```

```
export default ChildComponent;
```