

🚩 Current Skill Advanced

Rerendering on prop changes.

React components will render if the props change. React runs a comparison between the new and previous Props and will automatically render the new ones.

We have this simple React component that displays the title on screen :

```
import React from "react";

const MyComponent = ({ title }) => <h1>{title}</h1>;

export default MyComponent;
```

We are simply going to pass a title props to MyComponent component :

```
<MyComponent title="I'm learning React" />;

//output on screen: I'm Learning React
```



Destructuring with props

In React, it's very common to pass multiple props to the component. So it's no wonder that many of the React component functions interact with a few or more props.

Let's see an example:



```
import React from "react";

const Row = props => (
  <div>
    <div>
      <span>First Name: {props.firstName}</span>
    </div>
    <div>
      <span>Last Name: {props.lastName}</span>
    </div>
    <div>
      <span>Email: {props.email}</span>
    </div>
    <button onClick={props.doSomethingAmazing}>Click me</button>
  </div>
);
```

Advantages of Destructuring

The advantages of ES6 destructuring assignment looks even cleaner in the function component.

Applying the props destructuring technique allows us to write code like this:

```
import React from "react";

const Row = ({ firstName, lastName, email, doSomethingAmazing }) => (
  <div>
    <div>
      <span>First Name: {firstName}</span>
    </div>
    <div>
      <span>Last Name: {lastName}</span>
    </div>
    <div>
      <span>Email: {email}</span>
    </div>
    <button onClick={doSomethingAmazing}>Click me</button>
  </div>
);
```

< Previous

next >