

Understanding state in class based components | Complete React Course in Hindi #25

In the last tutorial, we have seen how we can use props in class components. In today's video, we will talk about state, props, and constructors.

Passing ImageURL as a Prop

First of all, you have to use the Image Url with the props object. After that, We would pass the image URL in the 'img src' tag. Now, let's pass content (URL of Image) in this 'imageUrl' from News.js as shown below:

```
components > News.js > News > render
import React, { Component } from 'react'
import NewsItem from './NewsItem'

export class News extends Component {
  render() {
    return (
      <div className="container my-3">
        <h2>NewsMonkey - Top Headlines</h2>
        <div className="row">
          <div className="col-md-4">
            <NewsItem title="myTitle" description="mydesc" imageUrl="https://cdn.24.co.za/files/Cms/General/d/1074397d776dc91734e98906c0e1b7f3blafa.jpg"/>
          </div>
          <div className="col-md-4">
            <NewsItem title="myTitle" description="mydesc"/>
          </div>
          <div className="col-md-4">
            <NewsItem title="myTitle" description="mydesc"/>
          </div>
        </div>
      </div>
    )
  }
}
```

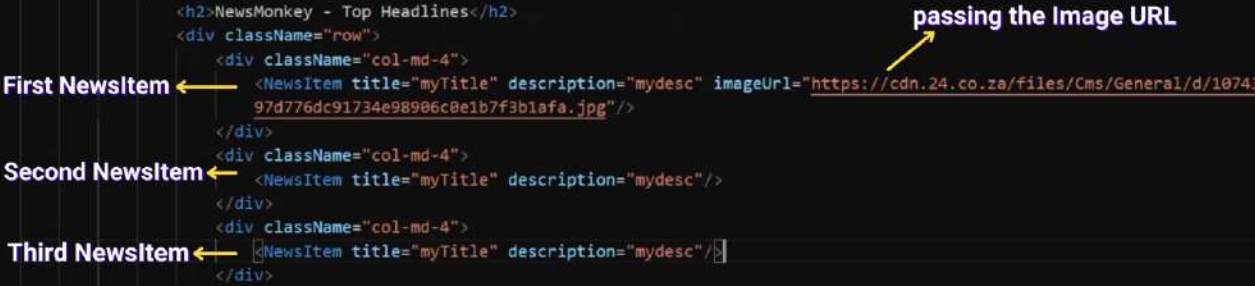


Figure 1.1: Passing Image as a Prop
Hence, we have **successfully** passed the image as a prop in the first News Item.

Using States in Class Components

The first step is to create a state object and initialize it, and this step is usually done inside the class constructor:

Constructor

If you don't know, the constructor is a method used to initialize an object's state in a class. Remember, within the constructor, we will call the super method. This is required because we extend react component class, and a call has to be made to the base class constructor.

Syntax:

```
constructor(){  
  super()  
}
```

Creating the State object

After doing this, we create our state object. For now, let 'this.state' equals an empty object.

```
constructor(){  
  super()  
  this.state = {  
  }  
}
```

Remember, we are using the class-based components, so don't forget to use the 'this' keyword. By the way, State is the reserved keyword in react. So, when you declare this.state, react understands your intention.

Hence, we have **successfully** created a state object. Now let's go ahead and initialize a property.

Create a variable

Here, we will be creating an article variable. We would be using all the articles of 'sampleoutput.json' in this article variable as an array.

Accessing this array

We can easily access this array from the constructor by using 'this.articles' in the state. We can even add some more properties in this.state. For example, we are adding a loading property to our application.

```
constructor(){  
  super();  
  console.log("Hello I am a constructor from News component");  
  this.state = {  
    articles: this.articles,   
    loading: false  
  }  
}
```

Annotations in the code:

- An arrow points from the text "Articles Property" to the `articles` key in the state object.
- An arrow points from the text "Using Articles Variable as a State" to the `this.articles` value.
- An arrow points from the text "Loading Property is set as false" to the `false` value.

Figure 1.2: Using States in Class Components

Revising Props and State

At this point, If you are confused with the concept of Props and State in React. Here is a quick comparison between the two to clear your mind.

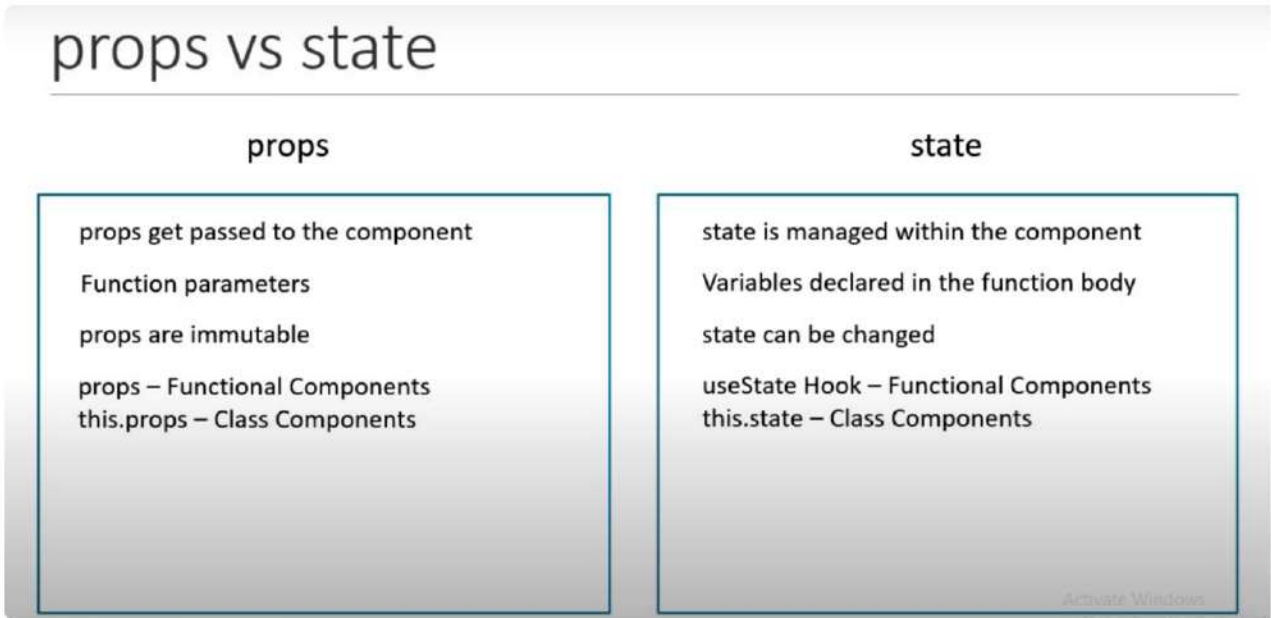



Figure 1.3: Quick Review between State and prop


NewsMonkey - Top Headlines



myTitle

mydesc

Read More



myTitle

mydesc

Read More



myTitle

mydesc

Read More