To create a class component in React.js, you can use the class keyword to define a JavaScript class that extends the React.Component class.

Example  
import React from 'react';

class MyComponent extends React.Component {  
  render() {  
    return (  
<div>  
<h1>Hello, World!</h1>  
</div>  
    );  
  }  
}

export default MyComponent;  
In this example:

We define a class named MyComponent that extends React.Component.  
Inside this class, we define a render() method. This method returns the JSX that represents the UI of the component.  
The JSX returned by render() can contain any valid React elements, including other components.  
You can then use this component in other parts of your application by importing it:

import React from 'react';  
import MyComponent from './MyComponent';

function App() {  
  return (  
<div>  
<h1>My App</h1>  
<MyComponent />  
</div>  
  );  
}

export default App;  
In this example, MyComponent is imported and used within the App component's JSX. When the App component is rendered, MyComponent will also be rendered as a part of its UI.

Class components are useful when you need to manage state or use lifecycle methods. However, with the introduction of hooks in React 16.8, functional components are now the preferred way to create components in most cases. If you're starting a new project or working on an existing project with React 16.8 or later, you might consider using functional components with hooks instead of class components.