React Fragments

In React, whenever you want to render something on the screen, you need to use a render method inside the component. This render method can return **single** elements or **multiple** elements. The render method will only render a single root node inside it at a time. However, if you want to return multiple elements, the render method will require a '**div**' tag and put the entire content or elements inside it. This extra node to the DOM sometimes results in the wrong formatting of your HTML output and also not loved by the many developers.

Example

// Rendering with div tag

class App extends React.Component {

render() {

return (

//Extraneous div element

<div>

<h2> Hello World! </h2>

<p> Welcome to the JavaTpoint. </p>

</div>

);

}

}

To solve this problem, React introduced **Fragments** from the **16.2** and above version. Fragments allow you to group a list of children without adding extra nodes to the DOM.

Syntax

<React.Fragment>

      <h2> child1 </h2>

    <p> child2 </p>

      .. ..... .... ...

</React.Fragment>

Example

// Rendering with fragments tag

**class** App **extends** React.Component {

    render() {

**return** (

       <React.Fragment>

            <h2> Hello World! </h2>

        <p> Welcome to the JavaTpoint. </p>

         </React.Fragment>

     );

    }

}

**The main reason to use Fragments tag is:**

1. **It makes the execution of code faster as compared to the div tag.**
2. **It takes less memory.**

**class** App **extends** React.Component {

    render() {

**return** (

       < >

            <h2> Hello World! </h2>

        <p> Welcome to the JavaTpoint. </p>

         </>

     );

    }

}

Fragmentwith keys:

## Keyed Fragments

The shorthand syntax does not accept key attributes. You need a key for mapping a collection to an array of fragments such as to create a description list. If you need to provide keys, you have to declare the fragments with the explicit <**React.Fragment**> syntax.

### Example

1. Function  = (props) {
2. **return** (
3. <Fragment>
4. {props.items.data.map(item => (
5. // Without the 'key', React will give a key warning
6. <React.Fragment key={item.id}>
7. <h2>{item.name}</h2>
8. <p>{item.url}</p>
9. <p>{item.description}</p>
10. </React.Fragment>
11. ))}
12. </Fragment>
13. )
14. }