Cheat Sheet: Introduction to React and Class Components

JSX syntax and Class Component	Description	Code Example
Vite tool	This tool is used to create React projects with the latest version	```npm create vite@latest```
JSX syntax with fragments	Jsx allows to include multiple tags within single parent tag only. For this you can include empty tags also known as fragments <></>	```<>\n This is a heading tag \n</>```
JSX syntax with parent tag	Jsx allows to include multiple tags within single parent tag only. For this you can include tags like div tag	``` \n This is a heading tag \n ```
Class component	A class component extends the `Component` class with a render method	```import React, { Component } from 'react'\nexport default class Extra extends Component {\n render() {\n return (\n <>\n This is class component \n </>\n)\n }\n}```
State management in class component	State can be managed using `this.state` within constructor of class component	```import React, { Component } from 'react'\nexport default class Extra extends Component {\n constructor(){\n super();\n this.state={\n count:0\n }\n }\n render() {\n return (\n <>\n This is paragraph \n </>\n)\n }\n}```

| **Access data of state management within class component** | Data can be accessed by including variable name declared within {} and include variable name within these curly braces | ```jsx
import React, { Component } from 'react'
export default class Extra extends Component {
    constructor(){
    super();
        this.state={
            count:0
        }
    }
  render() {
   return (
    <>
<p>the count is {this.state.count}</p>
    </>
   )
  }
}
``` |
| **Props in class component** | Props can be sent from parent component as attribute along with child component | ```jsx
import React, { Component } from 'react'
import ChildComponent from './ChildComponent'
export default class App extends Component {
  constructor(){
    super()
this.state={
  title:"Manager"
}
  }
  render() {
   return (
     <>
     <Extra title={this.state.title}/>
     </>
   )
  }
}
``` |
| **Access props within child class component** | Props can be accessed easily within the child class component using this.props.variable_name | ```jsx
import React, { Component } from 'react'
export default class Extra extends Component {
    constructor(props){
        super();
        this.state={
            count:0
        }
    }
  render() {
   return (
    <>
<p>The count value is {this.props.title}</p>
    </>
   )
  }
}
``` |
| **Event handling in class component** | Events such as click events are handled by defining a function before the render method | ```jsx
import React, { Component } from 'react';
class MyComponent extends Component {
  handleClick() {
    alert('Button clicked!');
  }
  render() {
    return (
``` |

```
            <div>
                <h1>Event Handling in Class Component</h1>
                <button onClick={this.handleClick}>Click Me</button>
            </div>
        );
    }
}
export default MyComponent;
```