

# Cheat Sheet: Introduction to React and Class Components

JSX syntax and Class Component	Description	Code Example
Vite tool	This is used to create React project with latest version	1. 1
		1. 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 <></>	Copied!
		1. 1 2. 2 3. 3
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	1. <> 2. <h1>This is a heading tag</h1> 3. </>
		Copied!
Class component	Class component extends property from Component class with render method	1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10
		1. import React, { Component } from 'react' 2. export default class Extra extends Component { 3.   render() { 4.     return ( 5.       <> 6.       <p>This is class component</p> 7.     </> 8.   ) 9. } 10. }
State management in class component	State can be managed using this.state within constructor of class component	Copied!
		1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10 11. 11 12. 12 13. 13 14. 14 15. 15
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	1. import React, { Component } from 'react' 2. export default class Extra extends Component { 3.   constructor(){ 4.     this.state={ 5.       count:0 6.     } 7.   } 8.   render() { 9.     return ( 10.      <> 11.      <p>This is paragraph</p> 12.    </> 13.  ) 14.  } 15. }
		Copied!
		1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9

10. 10  
11. 11  
12. 12  
13. 13  
14. 14  
15. 15  
16. 16

```
1. import React, { Component } from 'react'  
2. export default class Extra extends Component {  
3.   constructor(){  
4.     super();  
5.     this.state={  
6.       count:0  
7.     }  
8.   }  
9.   render() {  
10.    return (  
11.      <>  
12. <p>the count is {this.state.count}</p>  
13. </>  
14.    )  
15.  }  
16. }
```

Copied!

1. 1  
2. 2  
3. 3  
4. 4  
5. 5  
6. 6  
7. 7  
8. 8  
9. 9  
10. 10  
11. 11  
12. 12  
13. 13  
14. 14  
15. 15  
16. 16  
17. 17  
18. 18  
19. 19

**Props in class component**

Props can be sent from parent component as attribute along with child component

```
1. import React, { Component } from 'react'  
2. import ChildComponent from './ChildComponent'  
3.  
4. export default class App extends Component {  
5.   constructor(){  
6.     super()  
7.     this.state={  
8.       title:"Manager"  
9.     }  
10.  }  
11.  
12.   render() {  
13.     return (  
14.       <>  
15.         <Extra title={this.state.title}/>  
16.       </>  
17.     )  
18.   }  
19. }
```

Copied!

1. 1  
2. 2  
3. 3  
4. 4  
5. 5  
6. 6  
7. 7  
8. 8  
9. 9  
10. 10  
11. 11  
12. 12  
13. 13  
14. 14  
15. 15  
16. 16  
17. 17

**Access props within child class component**

Props can be accessed easily within the child class component using this.props.variable\_name

```
1. import React, { Component } from 'react'  
2.  
3. export default class Extra extends Component {  
4.   constructor(props){  
5.     super();  
6.     this.state={  
7.       count:0  
8.     }  
9.   }  
10.   render() {  
11.     return (  
12.       <>
```

```
13. <p>The count value is {this.props.title}</p>
14.     </>
15.   )
16. }
17. }
```

Copied!

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. 14
15. 15
```

```
1. import React, { Component } from 'react';
2. class MyComponent extends Component {
3.   handleClick() {
4.     alert('Button clicked!');
5.   }
6.   render() {
7.     return (
8.       <div>
9.         <h1>Event Handling in Class Component</h1>
10.        <button onClick={this.handleClick}>Click Me</button>
11.      </div>
12.    );
13.  }
14. }
15. export default MyComponent;
```

Copied!

**Event handling in class component** Events such as a click event can be performed by calling function which is declared before the render method



# Skills Network