

React Js CheatSheet Books

React JS cheatsheet contains useful code syntax with examples which is handy while coding.

▼ Table of Contents

- Create React App
- Import
- Components
 - Class component
 - Functional component
- Props
- Render
- Hooks
- Conditional Rendering
 - Ternary Operator
 - Usage of &&
- Forms
 - Note

Create React App

```
npx create-react-app my-app
```

Import

```
import React from 'react'  
import React, {Component} from 'react'; //multiple imports
```

Components

Class component

```
class Greetings extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

Functional component

```
function Greetings(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

Props

```
function Greetings(props) {  
  return <h1>Hello {props.name}</h1>;  
}  
function App() {  
  return (  
    <div>  
      <Greetings name="foo" />  
      <Greetings name="bar" />  
    </div>  
  );  
}
```

Note: Props are read-only

Render

```
render() {  
  return <div />;  
}
```

Hooks

Below is a sample code, which increases the count value when you click `+` and decreases the count value when you click `-`.

```
import React, { useState } from "react";
function App() {
  const [count, setCount] = useState(0);
  return (
    <div className="container">
      <h1>{count}</h1>
      <button onClick={()=>setCount(count-1)}>-</button>
      <button onClick={()=>setCount(count+1)}>+</button>
    </div>
  );
}
export default App;
```

`useEffect()`- The next most useful Hook you will encounter is the `useEffect` Hook. It performs a function whenever a specified state has changed.

```
function House() {
  const [color, setColor] = useState("white");
  const [door, setDoor] = useState(0); //initialize door as 0
  //add 1 to the current value of door on every button click
  const addDoor = () => {
    setDoor(door + 1);
  };
  //finally, trigger the function to print the door value whenever door is updated
  useEffect(() => {
    console.log(`Door count: ${door}`)
  }, [door]);
  return (
    <div>
      <h2>This is a {color} house</h2>
      <button onClick={addDoor}>Add a Door</button>
    </div>
  );
}
```

Conditional Rendering

Ternary Operator

```
<button type="submit">{isLoggedIn ? "Login" : "Register"}</button>
```

Usage of &&

`&&` also used to execute a block of code only if condition is true.

```
{!isLoggedIn && (  
  <button type="submit">{isLoggedIn ? "Login" : "Register"}</button>  
)}
```

Forms

Below is example of a simple form which displays the given name along with Hello. For example, if you give `Foo` in the input field, then when you click submit button it should display `Hello Foo` .

```
import React, { useState } from "react";  
function App() {  
  const [name, setName] = useState("");  
  const [displayName, setDisplayName] = useState("");  
  function handleClick(event) {  
    setDisplayName(name);  
    event.preventDefault();  
  }  
  return (  
    <div className="container">  
      <h1>Hello {displayName}</h1>  
      <form>  
        <input  
          placeholder="Enter your name?"  
          value={name}  
          onChange={(event)=>setName(event.target.value)}  
        />  
        <button type="submit" onClick={handleClick}>Submit</button>  
      </form>  
    </div>  
  );  
}
```

```
export default App;
```

Note

When you click a submit button, usually page gets refreshed. You can prevent this by preventing the default settings.

```
event.preventDefault();
```