

# REACT Cheat Sheet

This cheat sheet is for the course [React 18 Course - Learn React JS the fast way](#) by Jannick Leismann.

## ContextAPI

Use for **handling** Global **state management without** prop drilling. Too much prop drilling complicates code and makes it very hard to maintain. This is where context can be an advantage because it can **share context information** or data to **multiple components**. Is designed to share data between components without having to pass props down manually at every level of the **component tree**.

### Create a Context

Use **createContext** to create a context. This will hold the data to be shared.

### Providing a Context Value

The **MyProvider** component uses the **Provider** to provide context value to its children.

*MyContext.js*

```
import React, { createContext, useState } from "react";

// Create a context with a default value
const MyContext = createContext("defaultValue");

//Creating a provider
const MyProvider = ({ children }) => {
  const [value, setValue] = useState("Hello from context!");

  return <MyContext.Provider
value={value}>{children}</MyContext.Provider>;
};

export { MyContext, MyProvider };
```

## Consume the Content

Use the **useContext** hook to consume the context value.

*MyComponent.js*

```
import React, { useContext } from "react";
import { MyContext } from "../MyContext";

const MyComponent = () => {
  const contextValue = useContext(MyContext);

  return <div>{contextValue}</div>;
};

export default MyComponent;
```

## Use the Provider

Wrap the **MyComponent** with **MyProvider** to make the context value available to **MyComponent**

*App.js*

```
import React from "react";
import { MyProvider } from "../components/MyContext";
import MyComponent from "../components/MyComponent";

function App() {
  return (
    <div className="App">
      <div className="container">
        <MyProvider>
          <MyComponent />
        </MyProvider>
      </div>
    </div>
  );
}

export default App;
```