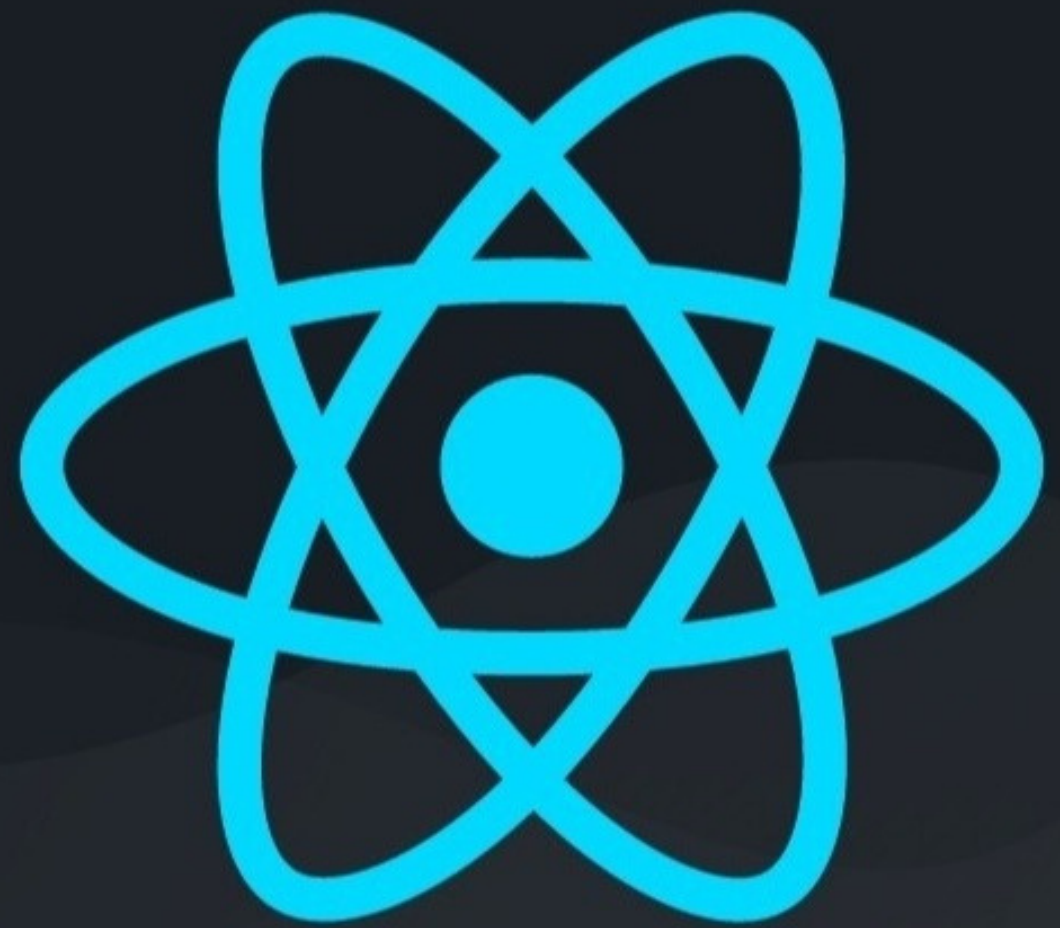# React JS
Context API

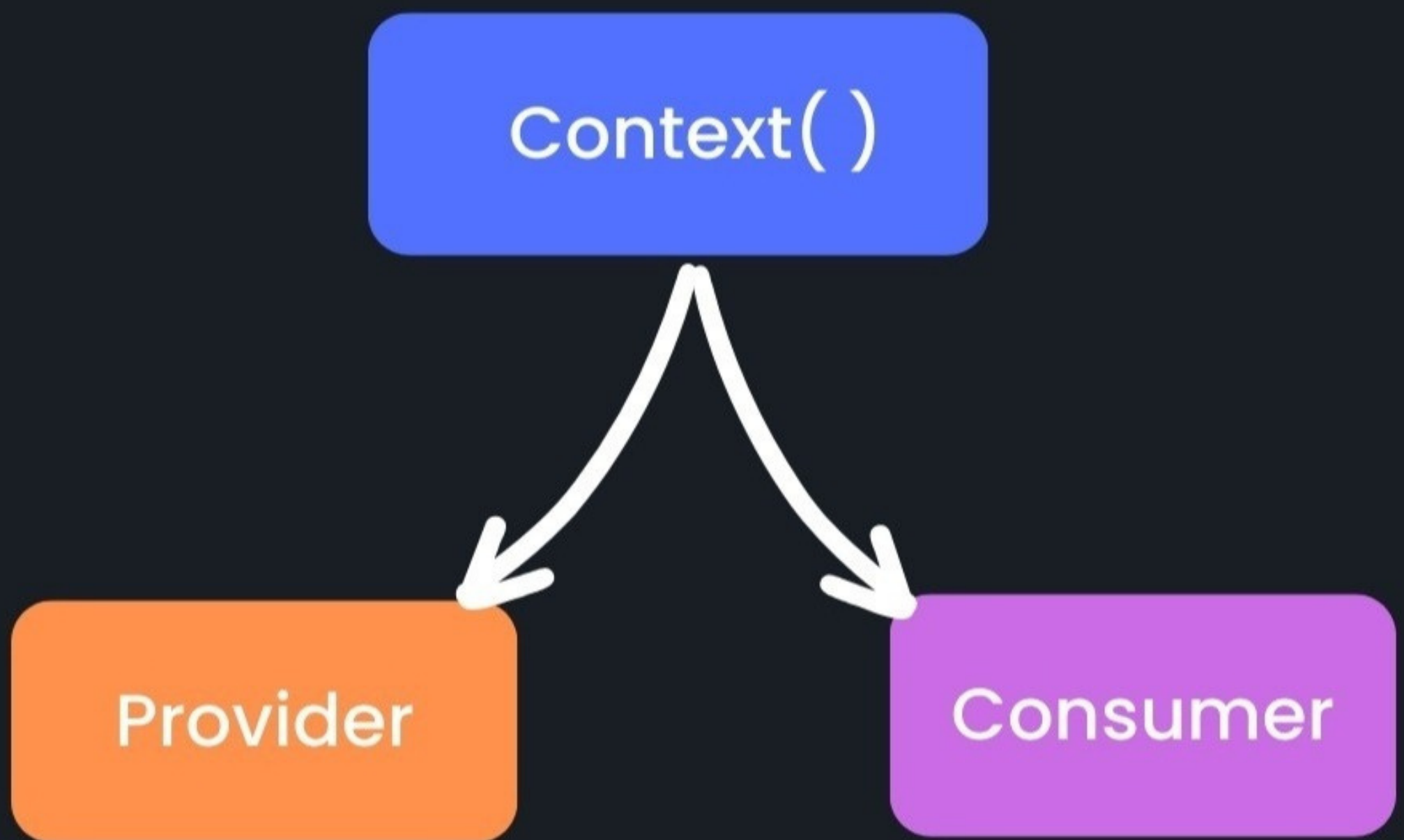Without useContext( )

Mallikarjun | @CodeBustler

Context API | React

In React, the Context API is a built-in feature that provides a way to **manage global state** or **share data** across components in a component tree **without the need to pass props** explicitly through every intermediate component.

It's designed to **solve the problem** of **"prop drilling,"** where data needs to be passed down multiple levels in the component hierarchy

Context API | React

Context API consists of two main components

Context( )

Provider

Consumer

✅ Context Provider

This component creates a context and **holds the state** that you want to share with other components. It wraps around the components that need access to the shared state.

✅ Context Consumer

This component consumes the data stored in the context and allows components to **access and use** that data.

Create Context API

You can create context API using simple 3 steps

1. Create
2. Provide
3. Consume

---

1. Create Context

Import React and create a new context using **React.createContext()**

```javascript
import React from "react";
const MyContext = React.createContext();
```

2. Provide

Create a **Provider component** that wraps around the components needing access to the context's data. **Provide** the **shared data using the value prop**.

```jsx
const MyContextProvider = ({ children }) => {
  // Shared data
  const sharedData = "Hello from Context!";

  return (
    <MyContext.Provider value={sharedData}>
      {children} {/* Child Component */}
    </MyContext.Provider>
  );
};
```

3. Consume

Consume the context data in the child components using the **Consumer component.** (or the **useContext** hook for functional components)

```jsx
const ChildComponent = () => {
  return (
    <MyContext.Consumer>
      {(data) => (
        <div>
          <h3>Child Component</h3>
          <p>Received data: {data}</p>
        </div>
      )}
    </MyContext.Consumer>
  );
};
```