

React:

useContext Hook



NEXT →

Hey Everyone



In a typical React application, data is passed **top-down** via props, but this can be **difficult for certain types** of props.

In this Post, we **will learn** how we **use Context** to pass the data between components.

Do **Like, save** and **Share** This Post If You Found This Helpful.

NEXT →

React useContext()

- The **React context** provides data to components no matter how **deep they are** in the components **tree**.
- The **context** is used to manage **global data**, e.g. **global state**, **theme**, **services**, **user settings**, and **more**.

NEXT →

use the context

- Using the context in React **requires 3 simple steps:**
 - a. **creating** the context
 - b. **providing** the context
 - c. **consuming** the context.

NEXT →

A. Creating the context

- To **create context**, you must Import `createContext` and **initialize** it.
- It's worth to note that context should always be **initialized in a separate file** , because we will **reuse it** across the entire **application**.

```
import React, { createContext } from 'react';

const UserContext = createContext('Default Value');

export { UserContext };
```

B. Providing the context

- Wrap child components in the **Context Provider** and supply the **state value**.
- To set the value of **context** use the **value prop** available on the `<Context.Provider value={value} />`.

```
import React from 'react';
import { UserContext } from './currency-context';
...

const App = () => {
  return (
    <UserContext.Provider value="Imtiyaz">
      <User />
    </UserContext.Provider>
  );
};
```

C. Consuming the context

- Consuming the context can be performed in 2 ways.
- The first way, the one I recommend, is to use the `useContext(Context)` React hook.

```
import React, { useContext } from 'react';
import { UserContext } from './currency-context';

const User = () => {
  const user = useContext(UserContext);
  return (
    <h2>`Hello ${user}, welcome to CODECLASH`</h2>
  )
}

export default User;
```

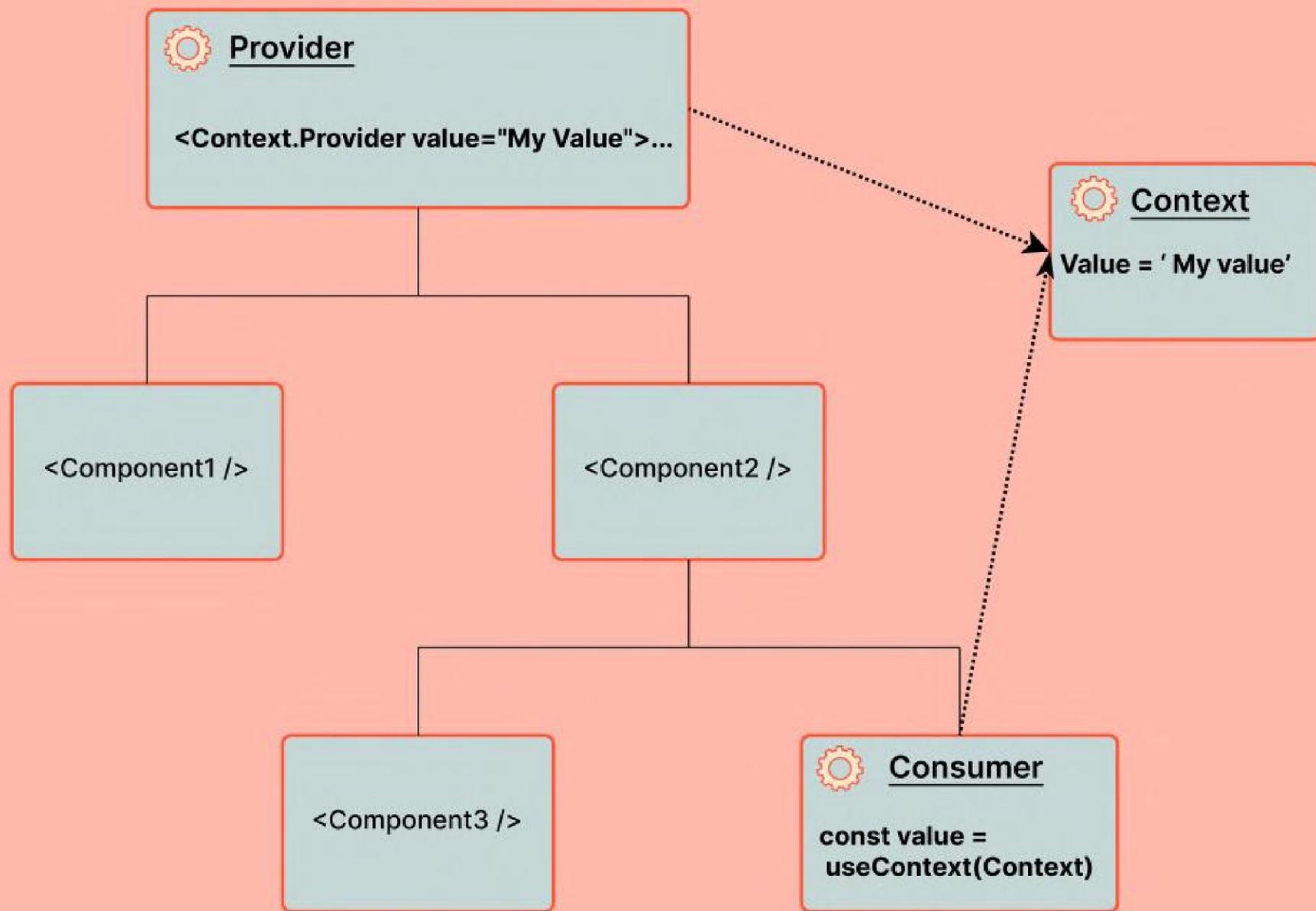
- The second way is by using a render function supplied as a child to `Context.Consumer` special component available on the context instance.

```
import React, { useContext } from 'react';
import { UserContext } from './currency-context';

const User = () => {
  return (
    <UserContext.Consumer>
      { value => <h2>`Hello ${user}, welcome to CODECLASH`</h2> }
    </UserContext.Consumer>
  )
}

export default User;
```

React Context



NEXT →