

Handling State

Accessing arrays

Lets say we have an array

```
const animals = ["giraffe", "zebra", "bear"];
console.log(animals[0]);
```

React Hooks

```
const what = useState();
```

Lift state up in React, we want to be high up in the tree as possible

Here we are lifting setStatus and status to function Main

```
<Main dishes={dishObjects} openStatus={status} onStatus = {setStatus}/>
```

```
useReducer()
```

```
const [status, dispatch] = useReducer((status), true)
```

```
const [status, dispatch] = useReducer((status) => !status, true);
```

whenever we call userReducer(), it is going to return an array of two items

instead of useState(), useReducer() is doing the toggler functionality

```
const [status, toggle] = useReducer(
  (status) => !status,
  true
);
```

we just pass in the function without worrying about the logic built into the function

useEffect() → manages side effects that are not related to a component's render

Things like console messages, loading data, working with animations, can benefit from useEffect() hook

ctrl + alt + j for the console window

by default, useEffect() runs whenever the state value changes

```
useEffect(() => {
  console.log(`The restaurant is ${status ? "open" : "closed"}`);
}, []);
```

Empty dependency array [] means, call this function only when the App first renders not after that whenever the state changes. [status] this will make the function to execute whenever 'status' value changes. useEffect() takes in a callback function and dependency array.

The restaurant is currently Closed

[Open Restaurant](#)

▶ (3) [{ - }, { - }, { - }]
The restaurant is open
The restaurant is closed
The restaurant is open
The restaurant is closed
➤ [ctrl + i](#) to turn on code suggestions

Alex's kitchen

Copyright 2026

We serve the most delicious food around

[I want to be open](#)

Welcome to this beautiful restaurant! It is closed.



Macaroni and cheese
Salmon with potatoes
Tofu with rice and vegetables