

Solution: Create your own custom hook, usePrevious

Your task was to complete the custom hook named `usePrevious` so that the `h1` heading shows both the current day and the previous current day before the update.

Here is the completed solution code for the `App.js` file:

```
1  import { useState, useEffect, useRef } from "react";
2  export default function App() {
3    const [day, setDay] = useState("Monday");
4    const prevDay = usePrevious(day);
5    const getNextDay = () => {
6      if (day === "Monday") {
7        setDay("Tuesday")
8      } else if (day === "Tuesday") {
9        setDay("Wednesday")
10     } else if (day === "Wednesday") {
11       setDay("Thursday")
12     } else if (day === "Thursday") {
13       setDay("Friday")
14     } else if (day === "Friday") {
15       setDay("Monday")
16     }
17   }
18   return (
19     <div style={{padding: "40px"}}>
20       <h1>
21         Today is: {day}<br />
22         {
23           prevDay && (
24             <span>Previous work day was: {prevDay}</span>
25           )
26         }
27       </h1>
28       <button onClick={getNextDay}>
29         Get next day
30       </button>
31     </div>
32   );
33 }
34 function usePrevious(val) {
35   const ref = useRef();
36   useEffect(() => {
37     ref.current = val;
38   }, [val]);
39   return ref.current;
40 }
```

Steps

Step 1

You should have located the `usePrevious()` function.

```
1 function usePrevious(val) {  
2  
3 }
```

Inside the `usePrevious()` function's code block, you needed to invoke the `useRef` hook without any arguments, and assign this invocation to a variable named `ref`, declared using the `const` keyword.

```
1 function usePrevious(val) {  
2   const ref = useRef();  
3 }
```

Step 2

Next, inside the `usePrevious()` function declaration, you needed to add a call to the `useEffect()` hook.

```
1 function usePrevious(val) {  
2   const ref = useRef();  
3   useEffect();  
4 }
```

Step 3

Then, you needed to pass two parameters as arguments to the `useEffect()` function call.

The first parameter should have been an arrow function, without any arguments. Inside the arrow function's body, you should have assigned the `val` value to the `current` property on the `ref` object.

The second parameter needed to be the dependencies array. The dependencies array needed to list a single variable - namely, the `val` variable.

```
1 function usePrevious(val) {  
2   const ref = useRef();  
3   useEffect(() => {  
4     ref.current = val;  
5   }, [val]);  
6 }
```

Step 4

You needed to add one more line to the body of the `usePrevious()` function declaration, to specify the `return` value of that function.

The `usePrevious()` function should have returned the `ref.current` value, as follows:

```
1 function usePrevious(val) {  
2   const ref = useRef();  
3   useEffect(() => {  
4     ref.current = val;  
5   }, [val]);  
6   return ref.current;  
}
```

