

# W05-P1: Hooks Tutorial T14\_xx

The screenshot shows a Mac desktop environment with the following components:

- Code Editor:** A dark-themed code editor window titled "T4-26-useState-object.js" showing React code. A red box highlights the following code block:

```
const T4_26_UseStateObject = () => {
  const [person, setPerson] = useState({
    name: 'weiyi',
    age: '36',
    message: 'random message'
  });

  const changeMessage = () => {
    setPerson({...person, message: 'weiyi 210410626'});
  }

  //Fragment
  return =>
    <h3>{person.name}</h3>
    <h3>{person.age}</h3>
    <h3>{person.message}</h3>
    <button className='btn' onClick={changeMessage}>Change Message</button>
  </>;
};

export default T4_26_UseStateObject;
```

- Browser:** A web browser window titled "localhost:3000" displaying a simple UI with a title "Tutorial 4 - 210410626" and three cards: "Weiyi" (age 36, message random message), a button "Change Message", and a placeholder card.
- Terminal:** An integrated terminal window showing build logs. It includes sections for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The OUTPUT section shows the application is running locally at port 3000. The DEBUG CONSOLE section shows a successful webpack compilation. The TERMINAL section shows the command "npm run build".
- System Dock:** The Mac OS X dock at the bottom of the screen with various application icons.

# W05-P2: useEffect 3 cases demo

The screenshot shows a Mac desktop environment with a browser window and a code editor.

**Code Editor (left):**

```
src > tutorial > 2-useEffect > JS T21-26-useEffect-basics.js > T21_26_UseEffectBasics
1 import React, { useState, useEffect } from 'react';
2 // cleanup function
3 // second parameter
4 const T21_26_UseEffectBasics = () => {
5   const [value, setValue] = useState(0);
6   const [value2, setValue2] = useState(0);
7
8
9 // by default runs after every re-render
10 useEffect(() => {
11   console.log('value', value);
12 });
13
14 useEffect(() => {
15   console.log('loading...');
16 }, []);
17
18 useEffect(() => {
19   console.log('value2', value2);
20   if(value2 > 0) {
21     document.title = `New Message(${value2})`;
22   }
23 }, [value2]);
24
25 return <>
26 <h1>{value}</h1>
27 <button className='btn' onClick={() => setValue(value + 1)}>value + 1</button>
28 <h1>{value2}</h1>
29 <button className='btn' onClick={() => setValue2(value2 + 1)}>value2 + 1</button>
30 </>
31 };
32 export default T21_26_UseEffectBasics;
```

**Browser Window (right):**

The browser window displays the title "Tutorial T21-210410626". Below the title, there are three large numbers: "1", "3", and "Value2 + 1". To the right of these elements is a list of log entries from the browser's developer tools:

- value 0 T21-26-useEffect-basics.js:11
- loading... T21-26-useEffect-basics.js:15
- value2 0 T21-26-useEffect-basics.js:19
- value 1 T21-26-useEffect-basics.js:11
- value2 1 T21-26-useEffect-basics.js:19
- value 1 T21-26-useEffect-basics.js:11
- value2 2 T21-26-useEffect-basics.js:19
- value 1 T21-26-useEffect-basics.js:11
- value2 3 T21-26-useEffect-basics.js:19

The browser status bar shows "localhost:3000" and "Dimensi... 270 x 625". The bottom of the screen shows the Mac OS Dock with various application icons.

**W05-P3: T23\_xx: fetch data from Github and display them**

# W05-P4: T31\_xx: conditional rendering -- 3 cases

The screenshot shows a Mac desktop with two browser windows open. The left window is a code editor showing the file `T31-26-multiple-returns.js`. A red box highlights the fetch call and its handling. The right window is a browser showing the output of the code execution. The user 'weiyi' is logged in.

```
src > tutorial > 3-conditional-rendering > JS T31-26-multiple-returns.js - 1111-wpt1-demo-26
1 import React, { useState, useEffect } from 'react';
2 const url = 'https://api.github.com/users/weiyi';
3
4 const T31_26_MultipleReturns = () => {
5   const [isLoading, setLoading] = useState(true);
6   const [isError, setIsError] = useState(false);
7   const [user, setUser] = useState('default user');
8
9   //run one
10  useEffect(() => {
11    fetch(url)
12      .then(responses) => {
13        if(responses.status >= 200 && responses.status <= 299) {
14          return responses.json();
15        }
16      })
17      .then(user) => {
18        console.log('user', user.login);
19        setUser(user.login);
20        setLoading(false);
21      })
22    .catch(error) => {
23      console.error(error);
24    }
25  }, []);
26  , []);
27}
28 > if(isLoading) {
29   return (
30     <div>
31       <h1>Loading...</h1>
32     </div>
33   )
34 }
35 > if(isError) {
36   return (
37     <div>
38       <h2>Error</h2>
39     </div>
40   )
41 }
42 > return <h2>{user}</h2>;
43
44 };
45
```

Tutorial T31-  
210410626  
Weiyi

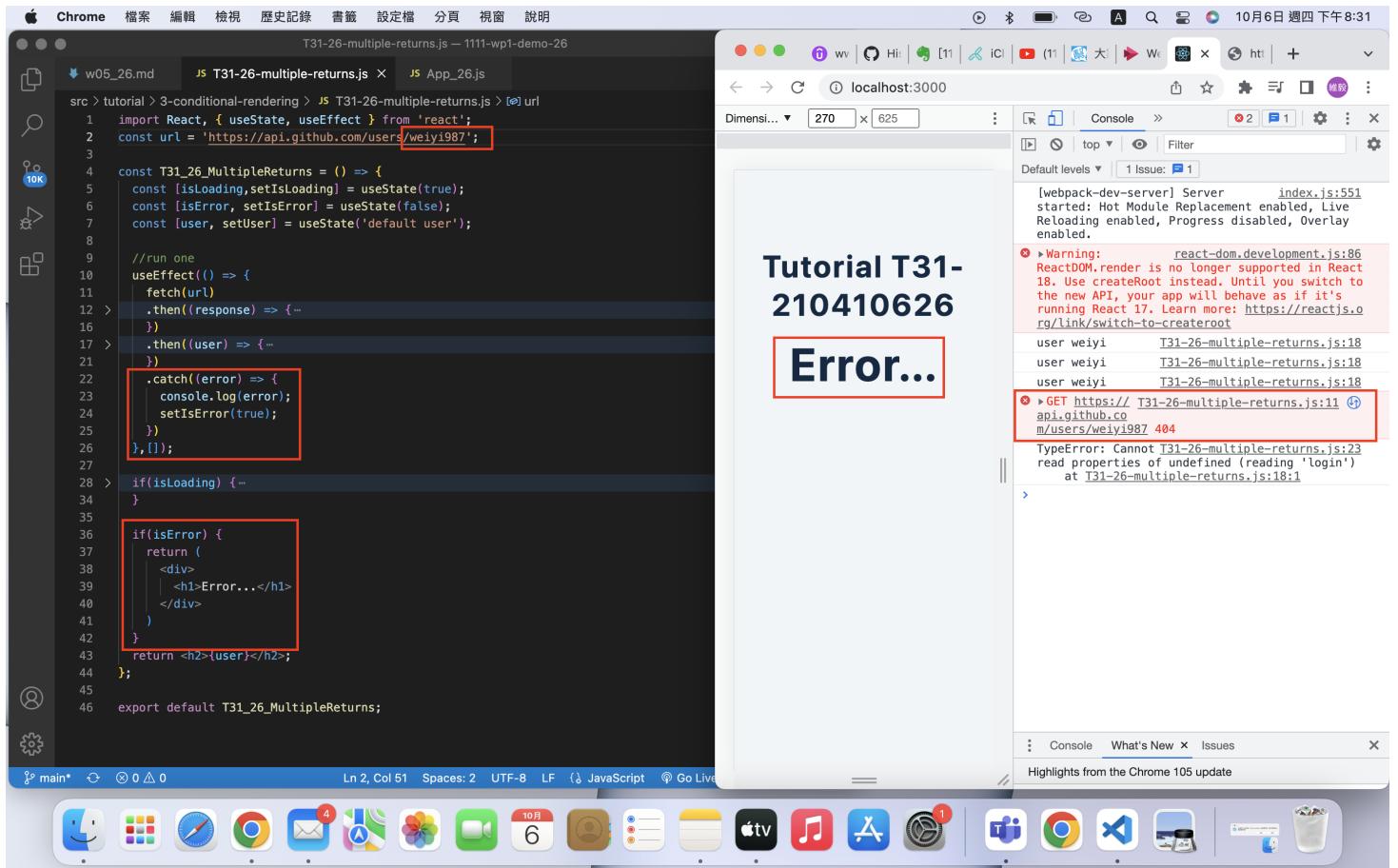
[webpack-dev-server] Server index.js:551 started: Hot Module Replacement enabled, Live Reloading enabled, Progress disabled, Overlay enabled.  
Warning: react-dom.development.js:86 ReactDOM.render is no longer supported in React 18. Use createRoot instead. Until you switch to the new API, your app will behave as if it's running React 17. Learn more: https://reactjs.org/link/switch-to-createRoot  
user weiyi T31-26-multiple-returns.js:18

The screenshot shows a Mac desktop with two browser windows open. The left window is a code editor showing the file `T31-26-multiple-returns.js`. A red box highlights the loading state logic. The right window is a browser showing the output of the code execution. The word 'Loading...' is displayed.

```
src > tutorial > 3-conditional-rendering > JS T31-26-multiple-returns.js - 1111-wpt1-demo-26
1 import React, { useState, useEffect } from 'react';
2 const url = 'https://api.github.com/users/weiyi';
3
4 const T31_26_MultipleReturns = () => {
5   const [isLoading, setLoading] = useState(true);
6   const [isError, setIsError] = useState(false);
7   const [user, setUser] = useState('default user');
8
9   //run one
10  useEffect(() => {
11    fetch(url)
12      .then(responses) => {
13        if(responses.status >= 200 && responses.status <= 299) {
14          return responses.json();
15        }
16      })
17      .then(user) => {
18        console.log('user', user.login);
19        setUser(user.login);
20        //setLoading(false);
21      })
22    .catch(error) => {
23      console.error(error);
24    }
25  }, []);
26  , []);
27}
28 > if(isLoading) {
29   return (
30     <div>
31       <h1>Loading...</h1>
32     </div>
33   )
34 }
35 > if(isError) {
36   return (
37     <div>
38       <h2>Error</h2>
39     </div>
40   )
41 }
42 > return <h2>{user}</h2>;
43
44 };
45
```

Tutorial T31-  
210410626  
Loading...

[webpack-dev-server] Server index.js:551 started: Hot Module Replacement enabled, Live Reloading enabled, Progress disabled, Overlay enabled.  
Warning: react-dom.development.js:86 ReactDOM.render is no longer supported in React 18. Use createRoot instead. Until you switch to the new API, your app will behave as if it's running React 17. Learn more: https://reactjs.org/link/switch-to-createRoot  
user weiyi T31-26-multiple-returns.js:18



## W05-logs: show all four logs done today

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Total 18 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), completed with 8 local objects.
To https://github.com/WeiYiHuangfelix/1111-wp1-demo-26.git
  5fb75ad..101220f main -> main
branch 'main' set up to track 'origin/main'.
● (base) huangweiyi@huangweiyideMacBook-Pro 1111-wp1-demo-26 % git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2022-10-04"
101220f WeiYiHuangfelix Thu Oct 6 20:34:26 2022 +0800 W05-P4: T31_xx: conditional rendering -- 3 cases
5fb75ad WeiYiHuangfelix Thu Oct 6 17:25:57 2022 +0800 W05-P2: useEffect 3 cases demo
4183475 WeiYiHuangfelix Thu Oct 6 16:32:28 2022 +0800 W05-P3: T23_xx: fetch data from Github and display them
041d79b WeiYiHuangfelix Thu Oct 6 13:27:11 2022 +0800 W05-P1: Hooks Tutorial T14_xx
○ (base) huangweiyi@huangweiyideMacBook-Pro 1111-wp1-demo-26 %

```