

Callback 、 Promise Async 、 Await

Web 前端框架實戰 | 許季秦 | 113.03.27

將函數當成物件或參數傳遞

JS promise.js > ...

```
1  const handleLogin = (response) => {  
2      if (response.result === 0) {  
3          console.log(' Login succeeded ');  
4      } else {  
5          console.log('login failed');  
6      }  
7  };  
8  
9  login('username', 'password', handleLogin);  
10
```

CallBack

```
12
13 login('username', 'password', (response)=>{
14     if (response.result === 0) {
15         console.log(' Login succeeded ');
16     } else {
17         console.log('login failed');
18     }
19 });
```


```
21
22 login('username', 'password', (response)=>{
23     if (response.result === 0) {
24         empLeaveTotal(response.id, (hour)=>{
25             if(hour.total>0){
26                 console.log('您這個月總共請假36個小時');
27                 getLeaveDetail((official)=>{
28                     if(official!==0)
29                         console.log('公假不扣錢');
30                 })
31             }else{
32                 console.log('您這個月全勤!');
33             }
34         });
35     } else {
36         console.log('login failed');
37     }
38 });
```




Promise


```
41 login('username', 'password')
42   .then((response)=>{
43     if (response.result === 0)
44       return empLeaveTotal(response.id);
45     return console.log('login failed');
46   }).then((hour)=>{
47     if(hour.total>0)
48       return getLeaveDetail(official);
49     return console.log('您這個月全勤!');
50   }).then((officialHour)=>{
51     if(official!==0)
52       console.log('公假不扣錢');
53     else
54       console.log('您這個月全勤!');
55   });
56
```

Promise

 有 `async` 和 `await` 的地方，即有 `promise` 的存在

 「保證執行之後才會做什麼事情」






async

 關鍵字置於函式前面

 用於表示函式是一個非同步函式

 意味著該函式的執行不會阻塞後面程式碼的執行。

await

-  必須在有關鍵字 `async` 的函式中使用
-  暫停 `async` 式的執行，繼續等待函式傳遞至表達式的 `Promise` 的解析，解析完之後會回傳值的解析，並以此 `async` 式的執行。
-  確保一個 `promise` 物件都解決 (`resolve`) 或出錯 (`reject`) 後才會進行下一步
-  `Promise` 物件代表一個即將完成、或失敗的非同步操作，以及它所產生的值
-  `async function` 的內容全都結束後，會傳回一個 `promise`，這表示後方可以使用 `.then` 語法來做連接

~THE END~

```
1
2  async function getInfo() {
3      const info1 = await promiseFunction(1);
4      const info2 = await promiseFnction(2);
5      console.log(info1, info2);
6  }
7  getInfo();
8
```