

陣列的方法 filter()

filter() 介紹

filter() 是 JavaScript 的一個陣列處理方法，用於篩選出指定元素，並回傳一個陣列。filter() 不會修改原來的陣列，而是會產生一個新的陣列，並且可以指派給其他變數。範例語法如下：

```
1 let arr = [];  
2 //回傳一個新的陣列，並指派給變數  
3 let newArr = arr.filter();
```

以下為 filter() 的用法：

1. filter() 內需放入一個函式，並回傳一個值。
2. filter() 會一一將陣列中的每個元素帶入函式測試，如果函式的回傳值為 true，則會將當前的元素保留至新陣列中，若為 false 則不保留。

callback 函式常見會帶有三種參數，範例如下：

```
1 let food = [  
2   {  
3     foodName: '鍋燒意麵',  
4     price: 70  
5   },  
6   {  
7     foodName: '炒麵',  
8     price: 65  
9   },  
10  {  
11    foodName: '蘿蔔泥',  
12    price: 20  
13  },  
14  {  
15    foodName: '荷包蛋',  
16    price: 10  
17  }  
18 ];  
19  
20 // item: 目前正在處理的元素。  
21 // index: 目前正在處理的元素之索引  
22 // array: 呼叫 filter 的原始陣列，在此為 food。此參數較為少用  
23  
24 let target = food.filter(function(item, index, array){  
25   // 回傳值，如果為 true，則將正在處理的這個 item 加入新陣列  
26   return item.price >= 20;  
27 })  
28 console.log(target);  
29 console.log(food);
```

Chrome 的輸出結果:

▼ (3) [{...}, {...}, {...}] ⓘ

- ▶ 0: {foodName: "鍋燒意麵", price: 70}
- ▶ 1: {foodName: "炒麵", price: 65}
- ▶ 2: {foodName: "蘿蔔泥", price: 20}

length: 3

▶ __proto__: Array(0)

▼ (4) [{...}, {...}, {...}, {...}] ⓘ

- ▶ 0: {foodName: "鍋燒意麵", price: 70}
- ▶ 1: {foodName: "炒麵", price: 65}
- ▶ 2: {foodName: "蘿蔔泥", price: 20}
- ▶ 3: {foodName: "荷包蛋", price: 10}

length: 4

▶ __proto__: Array(0)
