

函数形式

2020年03月18日, 星期三 14:53

1. 纯函数

一个函数的输出不受外部环境影响，同时也不影响外部环境时，只关注逻辑运算和数学运算，同一个输入总得到同一个输出。

- 相同的输入得到相同的输出
- 不会产生副作用

```
function getLast(arr) {  
  return arr[arr.length - 1];  
}
```



```
> var arr = [1, 2, 3, 4, 5, 6, 7];  
   function getLast(arr) {  
     return arr[arr.length - 1];  
   }  
   getLast(arr);  
  
< 7  
  
> getLast(arr);  
  
< 7
```

//两次调用结果相同（是纯函数）

```
function getLast(arr) {  
  return arr.pop();  
}
```



```
> var arr = [1, 2, 3, 4, 5, 6, 7];  
   function getLast(arr) {  
     return arr.pop();  
   }  
   getLast(arr);  
  
< 7  
  
> getLast(arr);  
  
< 6
```

//两次调用结果不同（不是纯函数）

2. 高阶函数

高阶函数：对其他函数进行操作的函数

- 函数的参数为函数
- 函数的返回值为函数

```
setTimeout(function() {  
  console.log("1s后输出");  
}, 1000);
```

//setTimeout是高阶函数

高阶函数：

- 在封装一个函数时，对于一个**不确定的变量**，我们使用传递**参数**的方式来指定。
- 在封装一个函数时，对于一个**不确定的过程**，我们向函数中传入另一个**函数**来指定。

高阶函数

基础函数

```
var fullAge = persons.filter(function(person) {  
  return person.age >= 18;  
});
```

4. 回调函数 (Callback)

- 一个函数被作为参数传递给另一个函数（在这里我们把另一个函数叫做“otherFunction”），回调函数在 otherFunction 中被调用。
- 回调函数并不会马上被执行。它会在包含它的函数内的某个特定时间点被“回调”（比如定时器）

回调函数

回调函数

```
setTimeout(function(){  
    console.log("1s后输出");  
}, 1000);
```