

Sunday • 更新于 2022-10-19

◀ 上一节 01: 前言 03: 如何让你的... 下一节 ▶

想要了解响应性，那么首先我们先了解什么叫做：**JS 的程序性**

我们来看下面这段代码：

### <> 代码块

[预览](#) [复制](#)

```
1  <script>
2    // 定义一个商品对象，包含价格和数量
3    let product = {
4      price: 10,
5      quantity: 2
6    }
7    // 总价格
8    let total = product.price * product.quantity;
9    // 第一次打印
10   console.log(`总价格: ${total}`);
11   // 修改了商品的数量
12   product.quantity = 5;
13   // 第二次打印
14   console.log(`总价格: ${total}`);
15 </script>
```

大家可以想一下，在这段代码中，第一次打印的值是什么？第二次打印的值是什么？

这是一个非常简单的 JS 逻辑，两次打印的值应该都是一样的： 总价格：20

但是大家有没有想过一个问题？

那就是当我们要去进行第二次打印的时候，你真的希望它还是 20 吗？

我们知道我们最终希望打印的是 **总价格**，那么当 `quantity` 由 2 变为 5 的时候，总价格不应该是 50 了吗？

我们打印出来的总价格，难道不应该是 50 吗？

那么此时你有没有冒出来一个想法：商品数量发生变化了，如果总价格能够自己跟随变化，那就太好了！

但是 js 本身具备 **程序性**，所谓程序性指的就是：**一套固定的，不会发生变化的执行流程**，在这样的一个程序性之下，我们是 **不可能** 拿到想要的 `50` 的。

那么如果我们想要拿到这个 50 就必须让你的程序变得更加的“聪明”，也就是使其具备 响应性！

01: 前言 ◀ 上一节      下一节 ▶ 03: 如何让你的程序变得更加“聪明”?

 我要提出意见反馈

