Sunday • 更新于 2022-10-19

上一节 02：JS 的程序性 04：vue 2 的响… 下一节

03：如何让你的程序变得更加“聪明”？

> 你为了让你的程序变得更加 “聪明”，所以你开始想：”如果数据变化了，重新执行运算就好了“。

那么怎么去做呢？你进行了一个这样的初步设想：

1. 创建一个函数 `effect`，在其内部封装 **计算总价格的表达式**
2. 在第一次打印总价格之前，执行 `effect` 方法
3. 在第二次打印总价格之前，执行 `effect` 方法

那么这样我们是不是就可以在第二次打印时，得到我们想要的 `50` 了呢？

所以据此，你得到了如下的代码：

代码块

```
<script>
  // 定义一个商品对象，包含价格和数量
  let product = {
    price: 10,
    quantity: 2
  }
  // 总价格
  let total = 0;
  // 计算总价格的匿名函数
+ let effect = () => {
+   total = product.price * product.quantity;
+ };
  // 第一次打印
+ effect();
  console.log(`总价格：${total}`); // 总价格：20
  // 修改了商品的数量
  product.quantity = 5;
  // 第二次打印
+ effect();
  console.log(`总价格：${total}`); // 总价格：50
</script>
```

在这样的代码中，我们成功的让第二次打印得到了我们期望的结果：**数据变化了，运算也重新执行了。**

但是大家也可以发现，在我们当前的代码中存在一个明显的问题，那就是：**必须主动在数量发生变化之后，重新主动执行 `effect` **才可以得到我们想要的结果。那么这样未免太麻烦了。有什么好的办法吗？

肯定是有的，我们继续来往下看~~~~

02：JS 的程序性 上一节 下一节 04：vue 2 的响应性核心 API：Object.defin…