慕课网首页 免费课 体系课 慕课教程 专栏 手记 企业服务 实战课

我的课程

Q

从所有教程的词条中查询…

首页 > 慕课教程 > Vue3源码分析与构建方案 > 02: JS 的程序性



第四章:响应系统-响应系 统的核心设计原则

01: 前言

12: 总结

02: JS 的程序性

03: 如何让你的程序变得更加 "聪明"?

04: vue 2 的响应性核心 API: Object.defineProperty

05: Object.defineProperty 在设计层的缺陷

06: vue3的响应性核心 API: proxy

```
Sunday • 更新于 2022-10-19
                                               ◆ 上一节 01: 前言 03: 如何让你的... 下一节 →
```

02: JS 的程序性

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
    <script>
 1
      // 定义一个商品对象,包含价格和数量
 3
      let product = {
        price: 10,
 5
        quantity: 2
 6
      // 总价格
     let total = product.price * product.quantity;
 8
     // 第一次打印
    console.log(`总价格: ${total}`);
10
11
    // 修改了商品的数量
12
     product.quantity = 5;
      // 第二次打印
1.3
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 就必须要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

索引目录

02: JS 的程序性

?

··

