

全部开发者教程

三

- 01: 前言
- 02: 源码阅读：ref 复杂数据类型的响应性
- 03: 框架实现：ref 函数 - 构建复杂数据类型的响应性
- 04: 总结：ref 复杂数据类型的响应性
- 05: 源码阅读：ref 简单数据类型的响应性
- 06: 框架实现：ref 函数 - 构建简单数据类型的响应性
- 07: 总结：ref 简单数据类型响应性
- 08: 总结
- 第七章：响应系统 - computed && watch
- 01: 开篇
- 02: 源码阅读：computed 的响应
- 03: 框架实现：构建



Sunday • 更新于 2022-10-19

◀ 上一节 04: 总结：ref ... 06: 框架实现：... 下一节 ▶

05：源码阅读：ref 简单数据类型的响应性

这一小节，我们将来看一下：ref 针对简单数据类型的响应性。

和之前一样我们需要先创建对应的测试案例：

1. 我们创建如下测试案例 创建测试实例 packages/vue/examples/imooc/ref-shallow.html：

<> 代码块

```
1    const { ref, effect } = Vue
2
3    const obj = ref('张三')
4
5    // 调用 effect 方法
6    effect(() => {
7      document.querySelector('#app').innerText = obj.value
8    })
9
10   setTimeout(() => {
11     obj.value = '李四'
12   }, 2000);
```

该测试案例，利用 ref 创建了一个简单数据类型的响应性。那么我们接下来就借助该案例，来看下 vue3 对简单数据类型的响应性的处理。

ref 函数

整个 ref 初始化的流程与 02 小节完全相同，但是有一个不同的地方，需要 特别注意：因为当前不是复杂数据类型，所以在 toReactive 函数中，不会通过 reactive 函数处理 value。所以 this._value 不是一个 proxy。即：无法监听 setter 和 getter。

effect 函数

整个 effect 函数的流程与 02 小节完全相同。

get value()

整个 effect 函数中引起的 get value() 的流程与 02 小节完全相同。

大不同：set value()

延迟两秒钟，我们将要执行 obj.value = '李四' 的逻辑。我们知道在复杂数据类型下，这样的操作（obj.value.name = '李四'），其实是触发了 get value 行为。

但是，此时，在简单数据类型之下，obj.value = '李四' 触发的将是 set value 形式，这里也是 ref 可以监听到简单数据类型响应性的关键。

1. 跟踪代码，进入到 set value(newVal)：

1. 通过 hasChanged 方法，对比数据是否发生变化
2. 发生变化，则触发 triggerRefValue，进入：

- 1 执行 triggerEffects 触发依赖。完成响应性

索引目录

- 05: 源码阅读：ref
- ref 函数
- effect 函数
- get value()
- 大不同：set va
- 总结



由以上代码可知：

1. 简单数据类型的响应性，不是基于 `proxy` 或 `Object.defineProperty` 进行实现的，而是通过：`set` 语法，将对象属性绑定到查询该属性时将被调用的函数上，使其触发 `xxx.value = '李四'` 属性时，其实是调用了 `xxx.value('李四')` 函数。
2. 在 `value` 函数中，触发依赖

所以，我们可以说：对于 `ref` 标记的简单数据类型而言，它其实“**并不具备响应性**”，所谓的响应性只不过是因为我们 **主动触发了 `value` 方法** 而已。

总结

目前我们已经明确了 `ref` 针对于简单数据类型的“响应性”的问题，那么下面我们就可以针对我们的理解，来对这一块进行对应的实现了。

04：总结：ref 复杂数据类型的响应性 ◀ 上一节 下一节 ▶ 06：框架实现：ref 函数 - 构建简单数据类型...

 我要提出意见反馈

企业服务 网站地图 网站首页 关于我们 联系我们 讲师招募 帮助中心 意见反馈 代码托管

Copyright © 2022 imooc.com All Rights Reserved | 京ICP备 12003892号-11 京公网安备11010802030151号



 意见反馈

 收藏教程

 标记书签