- Go >>>>

慕课网首页 免费课 实战课 体系课 慕课教程 专栏 手记 企业服务

〕 葉 ▲ 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 > 09: 源码阅读: 场景四: 旧节点多于新节点时...

全部开发者教程 :Ξ

04:框架实现:场景一:自前 向后的 diff 对比

05: 源码阅读: 场景二: 自后 向前的 diff 对比

06: 框架实现: 场景二: 自后 向前的 diff 对比

07:源码阅读:场景三:新节 点多余旧节点时的 diff 比对

08: 框架实现: 场景三: 新节 点多余旧节点时的 diff 比对

09:源码阅读:场景四:旧节 点多于新节点时的 diff 比对

10: 框架实现: 场景四: 旧节 点多于新节点时的 diff 比对

11: 局部总结: 前四种 diff 场 景的总结与乱序场景

12: 前置知识: 场景五: 最长 递增子序列

13:源码逻辑:场景五:求解 最长递增子序列 Sunday • 更新于 2022-10-19

◆ 上一节 08: 框架实现: ... 10: 框架实现: ... 下一节 ▶

索引目录

09: 源码阅读: 场

?

 \odot

09: 源码阅读: 场景四: 旧节点多于新节点时的 diff 比对

接下来我们来看场景四 **旧节点多于新节点时**,根据场景三的经验,其实我们也可以明确,对于旧节点多于新节点时,对应的场景也可以细分为两种:

- 1. 多出的旧节点位于 尾部
- 2. 多出的旧节点位于 头部

那么我们先创建第一种场景下的测试实例 packages/vue/examples/imooc/runtime/render-element-diff-4.html:

```
<> 代码块
1
   <script>
2
     const { h, render } = Vue
 3
 4
      const vnode = h('ul', [
 5
        h('li', {
          key: 1
         }, 'a'),
         h('li', {
 8
 9
          key: 2
10
        }, 'b'),
        h('li', {
11
          key: 3
12
13
        }, 'c'),
14
     ])
      // 挂载
1.5
16
      render(vnode, document.querySelector('#app'))
17
       // 延迟两秒,生成新的 vnode,进行更新操作
18
       setTimeout(() => {
19
20
        const vnode2 = h('ul', [
          h('li', {
21
           key: 1
22
          }, 'a'),
23
          h('li', {
24
           key: 2
26
          }, 'b'),
27
       ])
28
        render(vnode2, document.querySelector('#app'))
      }, 2000);
29
    </script>
30
```

跟踪代码,直接进入场景四即可:

1. 代码进入场景四, 此时的参数为:

✔ 意见反馈

♡ 收藏教程

□ 标记书签

- 2. 满足 i > e2
- 3. 满足 while 循环 i <= e1 的判断
 - 直接执行 unmount(c1[i], parentComponent, parentSuspense, true) 方法, 对旧节点进行 卸载操作
 - 2. 执行 i++

以上代码比较简单,对于多出的旧节点位于头部的场景,同样执行该逻辑。

由以上代码可知:

1. 旧节点多于新节点时,整体的处理比较简单,只需要 卸载旧节点即可

✔ 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

★ % 6 1

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