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

04: 框架实现: 场

?

··

索引目录

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 > 04:框架实现:场景一:自前向后的 diff 对比

## 全部开发者教程 Ξ

16: 总结

## 第十二章: runtime 运行时 - diff 算法核心实现

01: 前言

02: 前置知识: VNode 虚拟 节点 key 属性的作用

03: 源码阅读: 场景一: 自前 向后的 diff 对比

## 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

## 

Sunday • 更新于 2022-10-19

根据我们上一小节所描述的内容,下面我们可以直接实现对应逻辑。

04: 框架实现: 场景一: 自前向后的 diff 对比

- 1. 首先我们先让我们的代码支持 ARRAY\_CHILDREN 的渲染。
  - 1. 在 packages/runtime-core/src/renderer.ts 中 mountElement 中:

```
<> 代码块
     else if (shapeFlag & ShapeFlags.ARRAY_CHILDREN) {
2
         // 设置 Array 子节点
3
         mountChildren(vnode.children, el, anchor)
4
```

2. 接下来我们来处理 diff

▶ 意见反馈

♡ 收藏教程

口 标记书签

3. 在 packages/runtime-core/src/renderer.ts 中, 创建 patchKeyedChildren 方法:

```
<> 代码块
1
2
     * diff
3
      */
4
     const patchKeyedChildren = (oldChildren, newChildren, container, parentAnchor) => {
5
        /**
         * 索引
6
         */
8
         let i = 0
         * 新的子节点的长度
10
11
12
         const newChildrenLength = newChildren.length
13
         * 旧的子节点最大(最后一个)下标
14
15
         let oldChildrenEnd = oldChildren.length - 1
17
18
         * 新的子节点最大(最后一个)下标
19
20
         let newChildrenEnd = newChildrenLength - 1
21
         // 1. 自前向后的 diff 对比。经过该循环之后,从前开始的相同 vnode 将被处理
22
23
         while (i <= oldChildrenEnd && i <= newChildrenEnd) {</pre>
24
            const oldVNode = oldChildren[i]
            const newVNode = normalizeVNode(newChildren[i])
2.5
            // 如果 oldVNode 和 newVNode 被认为是同一个 vnode,则直接 patch 即可
26
27
            if (isSameVNodeType(oldVNode, newVNode)) {
28
                patch(oldVNode, newVNode, container, null)
29
30
            // 如果不被认为是同一个 vnode, 则直接跳出循环
31
            else {
32
                break
33
            }
34
            // 下标自增
35
            i++
```

```
37 }
```

4. 在 patchChildren 方法中, 触发 patchKeyedChildren 方法:

```
c>代码块

if (shapeFlag & ShapeFlags.ARRAY_CHILDREN) {

// 这里要进行 diff 运算

patchKeyedChildren(c1, c2, container, anchor)
}
```

创建对应测试实例 packages/vue/examples/runtime/render-element-diff.html:

```
<> 代码块
1 <script>
2
    const { h, render } = Vue
   const vnode = h('ul', [
      h('li', {
       key: 1
 7
      }, 'a'),
      h('li', {
 8
       key: 2
9
10
      }, 'b'),
11
      h('li', {
12
        key: 3
13
      }, 'c'),
    ])
14
    // 挂载
15
   render(vnode, document.querySelector('#app'))
16
17
18 // 延迟两秒, 生成新的 vnode, 进行更新操作
19
   setTimeout(() => {
20
     const vnode2 = h('ul', [
       h('li', {
21
22
         key: 1
       }, 'a'),
23
       h('li', {
24
25
         key: 2
      }, 'b'),
26
       h('li', {
27
         key: 3
28
       }, 'd')
     },
])
29
30
     render(vnode2, document.querySelector('#app'))
31
32 }, 2000);
33 </script>
```

✔ 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 6 4

?

0

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





