Sunday • 更新于 2022-10-19

上一节 03：源码阅读：… 05：源码阅读：… 下一节

04：框架实现：场景一：自前向后的 diff 对比

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。

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) {
        // 设置 Array 子节点
        mountChildren(vnode.children, el, anchor)
    }
```

2. 接下来我们来处理 `diff`

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

代码块

```
/**
 * diff
 */
const patchKeyedChildren = (oldChildren, newChildren, container, parentAnchor) => {
    /**
     * 索引
     */
    let i = 0
    /**
     * 新的子节点的长度
     */
    const newChildrenLength = newChildren.length
    /**
     * 旧的子节点最大（最后一个）下标
     */
    let oldChildrenEnd = oldChildren.length - 1
    /**
     * 新的子节点最大（最后一个）下标
     */
    let newChildrenEnd = newChildrenLength - 1

    // 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
    while (i <= oldChildrenEnd && i <= newChildrenEnd) {
        const oldVNode = oldChildren[i]
        const newVNode = normalizeVNode(newChildren[i])
        // 如果 oldVNode 和 newVNode 被认为是同一个 vnode，则直接 patch 即可
        if (isSameVNodeType(oldVNode, newVNode)) {
            patch(oldVNode, newVNode, container, null)
        }
        // 如果不被认为是同一个 vnode，则直接跳出循环
        else {
            break
        }
        // 下标自增
        i++
```

```
37        }
      }
```

4. 在 `patchChildren` 方法中，触发 `patchKeyedChildren` 方法：

代码块

```
1     if (shapeFlag & ShapeFlags.ARRAY_CHILDREN) {
2         // 这里要进行 diff 运算
3         patchKeyedChildren(c1, c2, container, anchor)
4     }
```

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

03：源码阅读：场景一：自前向后的 diff 对比 ◂ 上一节　　下一节 ▸ 05：源码阅读：场景二：自后向前的 diff 对比

我要提出意见反馈