

全部开发者教程

三

行为

22: 源码阅读：renderer 渲染器下，Comment 节点的挂载行为

23: 框架实现：renderer 渲染器下，Comment 节点的挂载行为

24: 源码阅读：renderer渲染器下，Fragment 节点的挂载、更新行为

25: 框架实现：renderer渲染器下，Fragment 节点的挂载、更新行为

26: 总结

第十一章：runtime 运行时 - 组件的设计原理与渲染方案

01: 前言

02: 源码阅读：无状态基础组件挂载逻辑

03: 框架实现：完成无状态基



Sunday • 更新于 2022-10-19

上一节 24: 源码阅读：... 26: 总结 下一节

## 25: 框架实现：renderer渲染器下，Fragment 节点的挂载、更新行为

明确好了 Fragment 的渲染逻辑之后，那么下面我们对此进行下实现。

1. 在 packages/runtime-core/src/renderer.ts 中，为 patch 新增 Fragment 的方法触发：

<> 代码块

```
1 case Fragment:
2   // Fragment
3   processFragment(oldVNode, newNode, container, anchor)
4   break
```

2. 创建 processFragment 方法：

<> 代码块

```
1 /**
2  * Fragment 的打补丁操作
3  */
4 const processFragment = (oldVNode, newNode, container, anchor) => {
5   if (oldVNode == null) {
6     mountChildren(newVNode.children, container, anchor)
7   } else {
8     patchChildren(oldVNode, newNode, container, anchor)
9   }
10 }
```

3. 构建 mountChildren 渲染逻辑：

<> 代码块

```
1 /**
2  * 挂载子节点
3  */
4 const mountChildren = (children, container, anchor) => {
5   // 处理 Cannot assign to read only property '0' of string 'xxx'
6   if (isString(children)) {
7     children = children.split('')
8   }
9   for (let i = 0; i < children.length; i++) {
10     const child = (children[i] = normalizeVNode(children[i]))
11     patch(null, child, container, anchor)
12   }
13 }
```

4. 在 packages/runtime-core/src/componentRenderUtils.ts 中，为 normalizeVNode 增加处理 Text 节点逻辑：

<> 代码块

```
1 /**
2  * 标准化 VNode
3  */
4 export function normalizeVNode(child) {
5   if (typeof child === 'object') {
6     // ...
7   }
8 }
```

索引目录

25: 框架实现：re



意见反馈

收藏教程

标记书签

```
8         return createVNode(Text, null, String(child))
9     }
10 }
11
12 /**
13  * clone VNode
14  */
15 export function cloneIfMounted(child) {
16     return child
17 }
```

5. 更新逻辑之前已经完成过，不需要额外处理

至此，Fragment 的渲染处理完成。

我们可以创建对应测试实例 `packages/vue/examples/runtime/render-fragment.html`：

<> 代码块

```
1 <script>
2   const { h, render, Fragment } = Vue
3
4   const vnode = h(Fragment, 'hello world')
5   // 挂载
6   render(vnode, document.querySelector('#app'))
7
8   setTimeout(() => {
9     const vnode2 = h(Fragment, '你好，世界')
10    // 挂载
11    render(vnode2, document.querySelector('#app'))
12  }, 2000);
13 </script>
```

挂载、更新成功

24: 源码阅读: renderer渲染器下, Fragn... < 上一节 下一节 > 26: 总结

✎ 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