

全部开发者教程

04：源码阅读：h 函数，跟踪 ELEMENT + ARRAY_CHILDREN 场景下的源码实现

05：框架实现：构建 h 函数，处理 ELEMENT + ARRAY_CHILDREN 场景

06：源码阅读：h 函数，组件的本质与对应的 VNode

07：框架实现：处理组件的 VNode

08：源码阅读：h 函数，跟踪 Text、Comment、Fragment 场景

09：框架实现：实现剩余场景 Text、Comment、Fragment

10：源码阅读：对 class 和 style 的增强处理



Sunday • 更新于 2022-10-19

◀ 上一节 06：源码阅读：... 08：源码阅读：... 下一节 ▶

07：框架实现：处理组件的 VNode

根据上一小节的了解，我们知道 组件的 VNode 其实只存在两个不同的地方：

1. type
2. shapeFlag

对于 type 而言，它是 h 函数的第一个参数，我们其实不需要单独进行处理，所以我们只需要处理 shapeFlag 即可。

在我们的代码中，处理 shapeFlag 的地方有两个：

1. createVNode：第一次处理，表示 node 类型（比如：Element）
2. createBaseVNode：第二次处理，表示 子节点类型（比如：Text Children）

因为我们这里不涉及到子节点，所以我们只需要在 createVNode 中处理即可：

<> 代码块

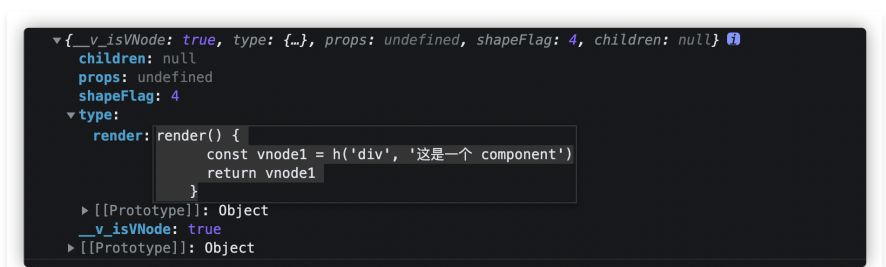
```
1 // 通过 bit 位处理 shapeFlag 类型
2 const shapeFlag = isString(type)
3   ? ShapeFlags.ELEMENT
4   : isObject(type)
5   ? ShapeFlags.STATEFUL_COMPONENT
6   : 0
```

此时创建测试实例 packages/vue/examples/runtime/h-component.html：

<> 代码块

```
1 <script>
2   const { h, render } = Vue
3
4   const component = {
5     render() {
6       const vnode1 = h('div', '这是一个 component')
7       return vnode1
8     }
9   }
10  const vnode2 = h(component)
11  console.log(vnode2);
12 </script>
```

可以得到相同的打印结果



索引目录

07：框架实现：处理组件的 VNode



 我要提出意见反馈