

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 Text09：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 07：框架实现：... 09：框架实现：... 下一节 ▶

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

当组件处理完成之后，最后我么来看下 Text、Comment、Fragment 这三个场景下的 VNode。

Text

Text 标记为 **文本**。即：纯文本的 VNode

创建 packages/vue/examples/imooc/runtime/h-other.html 测试实例，查看 Text 的打印：

<> 代码块

```
1  const { h, render, Text, Comment, Fragment } = Vue
2  const vnodeText = h(Text, '这是一个 Text')
3  console.log(vnodeText);
4  // 可以通过 render 进行渲染
5  render(vnodeText, document.querySelector('#app'))
```

查看打印：

<> 代码块

```
1  {
2    __v_isVNode: true
3    children: "这是一个 Text",
4    type: Symbol(Text),
5    shapeFlag: 8 // TEXT_CHILDREN
6  }
```

那么由以上代码可知，对于 Text，它存在的唯一——一个比较特殊的地方就是：****Text 类型是一个 Symbol(Text) ****，这个类型是在 Vue 中被导出的。

Comment

Comment 标记为 **注释**。即：注释节点的 VNode。

在 packages/vue/examples/imooc/runtime/h-other.html 测试实例中，查看 vnodeComment 的打印：

<> 代码块

```
1  const { h, render, Text, Comment, Fragment } = Vue
2  const vnodeComment = h(Comment, '这是一个 Comment')
3  console.log(vnodeComment);
4  render(vnodeComment, document.querySelector('#app'))
```

查看打印：

<> 代码块

```
1  {
2    __v_isVNode: true
3    children: "这是一个 Comment",
4    type: Symbol(Comment),
5    shapeFlag: 8 // TEXT_CHILDREN
6  }
```

那么由以上代码可知，对于 Comment，它存在的唯一——一个比较特殊的地方就是：****Comment 类型是一个**

索引目录

- 08：源码阅读：h
- Text
- Comment
- Fragment
- 总结

📖

?

📱

💬

Fragment

`Fragment` 标记为 **片段**。它相对比较特殊，是 `Vue3` 中新提出的一个概念，主要应对与 **包含多个根节点的模板**。

即：包含多个根节点的模板被表示为一个片段 (`fragment`)。

在 `packages/vue/examples/imooc/runtime/h-other.html` 测试实例中，查看 `vnodeComment` 的打印：

<> 代码块

```
1   const { h, render, Text, Comment, Fragment } = Vue
2   const vnodeFragment = h(Fragment, '这是一个 Fragment')
3   console.log(vnodeFragment);
4   render(vnodeFragment, document.querySelector('#app'))
```

查看打印（可以看一下 `Element` 比较 **特殊**）：

<> 代码块

```
1   {
2     __v_isVNode: true
3     children: "这是一个 Fragment",
4     type: Symbol(Fragment),
5     shapeFlag: 8 // TEXT_CHILDREN
6   }
```

那么由以上代码可知，对于 `Fragment`，它存在的唯一一个比较特殊的地方就是：**`Fragment` 类型是一个 `Symbol(Fragment)`**，这个类型是在 `Vue` 中被导出的。

总结

由以上代码可知，`Text`、`Comment`、`Fragment` 三块的处理还是比较简单的，比较重要的是搞明白他们三者的意思即可。

07: 框架实现：处理组件的 VNode ◀ 上一节 下一节 ▶ 09: 框架实现：实现剩余场景 Text09: 框架...

✎ 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