


```
4
5     return (_openBlock(), _createElementBlock("div", null, "hello world"))
6   }
7 }
```

对于以上代码，存在一个 `with` 语法，这个语法是一个 **不被推荐** 的语法，我们无需太过于关注它，只需要知道它的作用即可：

摘自：《JavaScript 高级程序设计》

`with` 语句的作用是：将代码的作用域设置到一个特定的对象中...

由于大量使用 `with` 语句会导致性能下降，同时也会给调试代码造成困难，因此在开发大型应用程序时，不建议使用 `with` 语句。

我们可以把该代码（`render`）略作改造，直接应用到 `render` 的渲染中：

```
<> 代码块

1  <script>
2    const { compile, h, render } = Vue
3    // 创建组件
4    const component = {
5      render: function (_ctx, _cache) {
6        with (_ctx) {
7          const { openBlock: _openBlock, createElementBlock: _createElementBlock } = Vue /
8
9          return (_openBlock(), _createElementBlock("div", null, "hello world"))
10         }
11       }
12     }
13
14     // 通过 h 函数，生成 vnode
15     const vnode = h(component)
16
17     // 通过 render 函数渲染组件
18     render(vnode, document.querySelector('#app'))
19   </script>
```

发现可以得到与：

```
<> 代码块

1  render() {
2    return h('div', 'hello world')
3  }
```

同样的结果。

那么观察两个 `render` 可以发现：

1. `compiler` 最终生成的 `render` 函数，与我们自己的写的 `render` 会略有区别。
 1. 它会直接通过 `createElementBlock` 来渲染 **块级元素** 的方法，比 `h` 函数更加“精确”
 2. 同时这也意味着，生成的 `render` 函数会触发更精确的方法，比如：
 1. `createTextVNode`
 2. `createCommentVNode`
 3. `createElementBlock`
 4. ...
2. 虽然，生成的 `render` 更加精确，但是本质的逻辑并没有改变，已然是一个：`return vnode` 进行 `render` 的过程。

 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