

全部开发者教程

06: 响应性数据的编辑器处理: generate 生成 render 函数

07: 响应性数据的编辑器处理: render 函数的执行处理

08: 多层级模板的编辑器处理: 多层级的处理逻辑

09: 基于编辑器的指令(v-xx)处理: 指令解析的整体逻辑

10: 基于编辑器的指令(v-xx)处理: AST 解析逻辑 (困难)

11: 基于编辑器的指令(v-xx)处理: JavaScript AST , 构建 vif 转化模块 (困难)

12: 基于编辑器的指令(v-xx)处理: JavaScript AST , transform 的转化逻辑

13: 基于编辑器的指令(v-xx)处理: 生成 render 函数

14: 总结

Sunday • 更新于 2022-10-19

◀ 上一节 03: 基于 templ... 01: 总结 下一节 ▶

04: 总结

到这里我们已经成功的完成了 **运行时 + 编译器** 的合并逻辑，那么我们可以运行如下测试实例，来查看一个相对完整的测试实例：

<> 代码块

```
1  <script>
2    const { createApp } = Vue
3    const APP = {
4      template: `<div>hello world, <h1 v-if="isShow">{{ msg }}</h1></div>`,
5      data() {
6        return {
7          msg: '你好，世界',
8          isShow: false
9        }
10     },
11     created() {
12       setTimeout(() => {
13         this.isShow = true
14       }, 2000);
15     }
16   },
17
18   const app = createApp(APP)
19   app.mount('#app')
20 </script>
```

03: 基于 template 渲染的 createApp 的构... ◀ 上一节 下一节 ▶ 01: 总结

✎ 我要提出意见反馈

索引目录

04: 总结