## **Attributes 继承**

就是属性继承：当一个父组件给子组件绑定属性时（props属性、class属性、自定义事件、style属性等等）子组件的根元素（即最外层的元素）会自动继承除去 props、emits 之外的属性

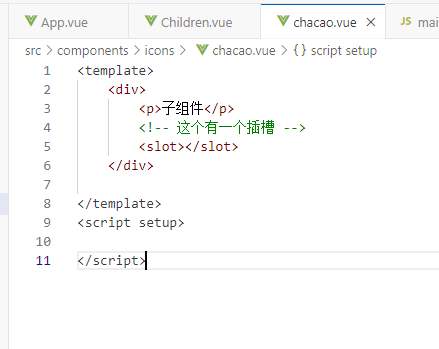
插槽：

理解成有个槽可以插入一些信息。

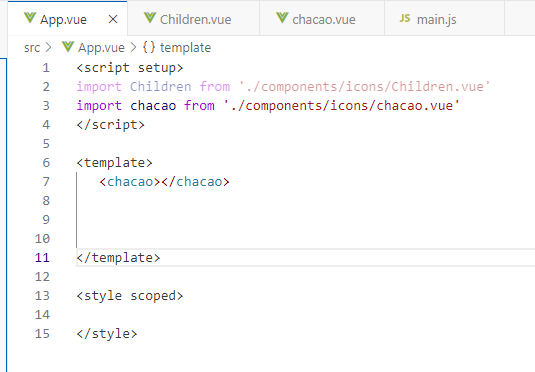
使用场景：通常是父子组件之间

举例：

子组件：



父组件：

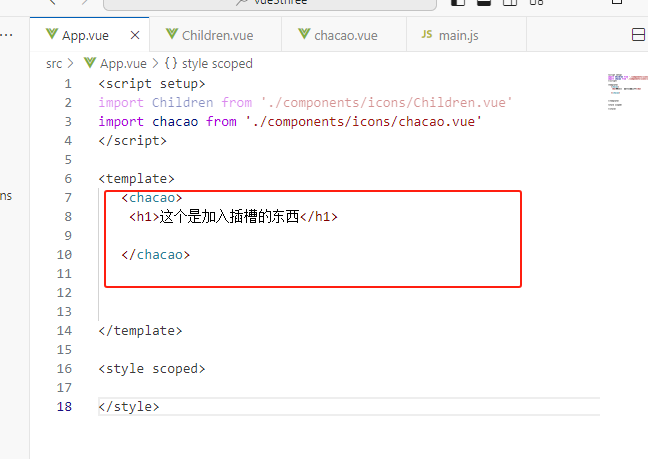


页面显示：



接下来往插槽里加入东西：

在父组件中：



页面显示：



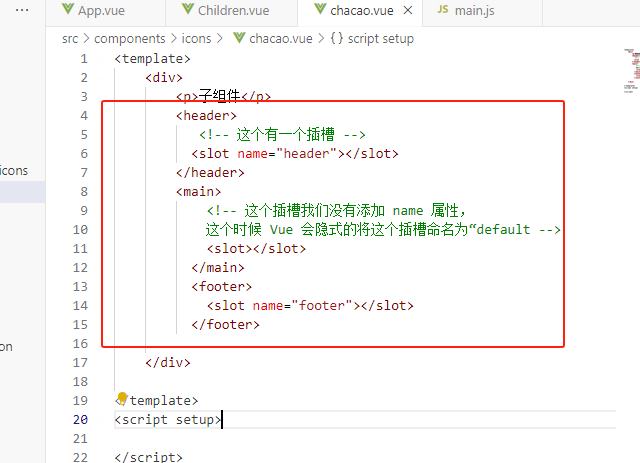
在页面可以看到<h1>这个是加入插槽的东西</h1>加到了chacao.vue的<slot></slot>的位置，

插槽就是可以填充一些内容的，父组件中调用子组件时，子组件标签之间的内容元素就是要放置的内容，它会把 slot 标签替换掉

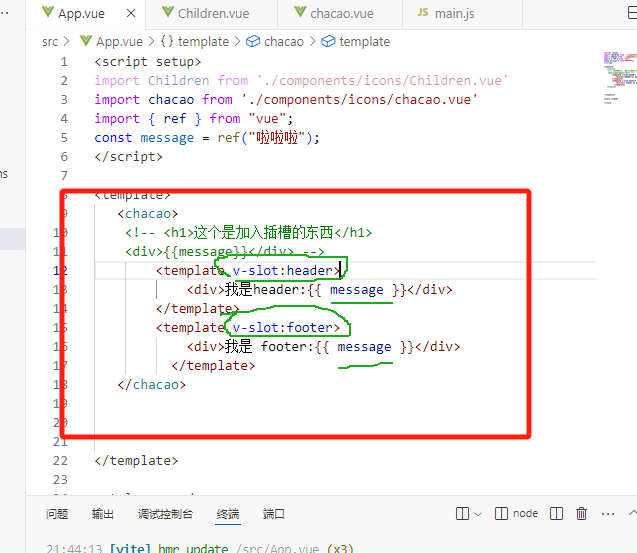
具名插槽

给子组件写入：

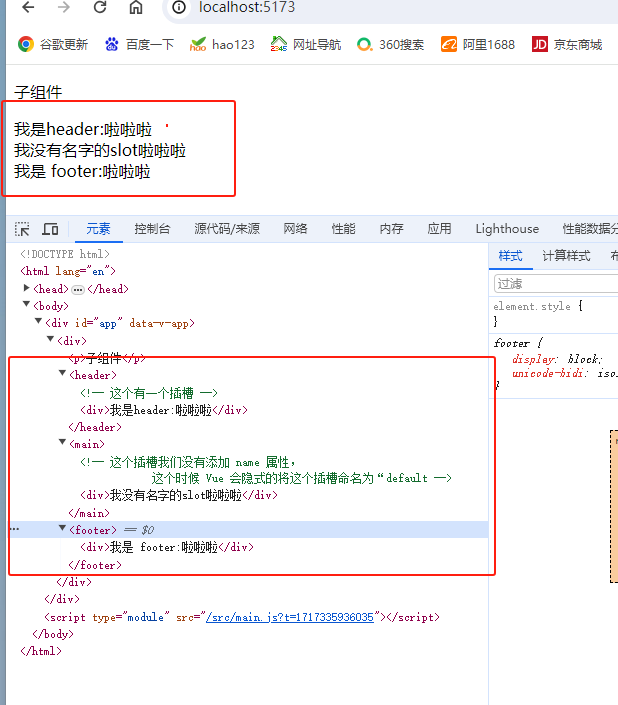
slot 和内容都加上一个名name



父组件：



页面显示：



通过<slot name="footer"> 给slot赋予名字，父组件中传入内容时就会和名字关系对应起来

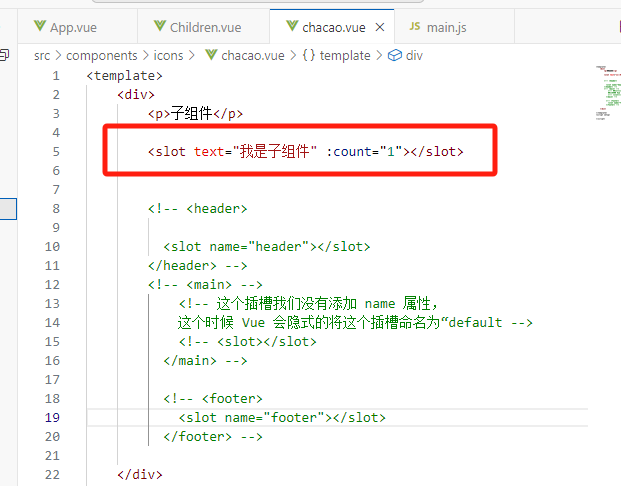
在父组件可以通过v-slot:header 指令的形式找到对应的插槽，需要注意的是该指令需要作用在 template 元素上，页面的最终结果就是传入的内容都渲染到了对应的插槽内，没有命名的插槽渲染了我们传入的未添加指令的内容。

v-slot:name 可以简写成：#name 比如：v-slot:footer简写为#footer

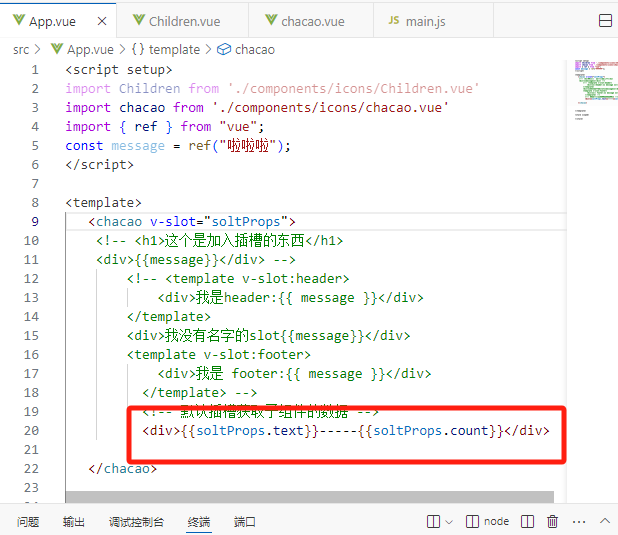
作用域插槽

默认插槽获取子组件的数据：

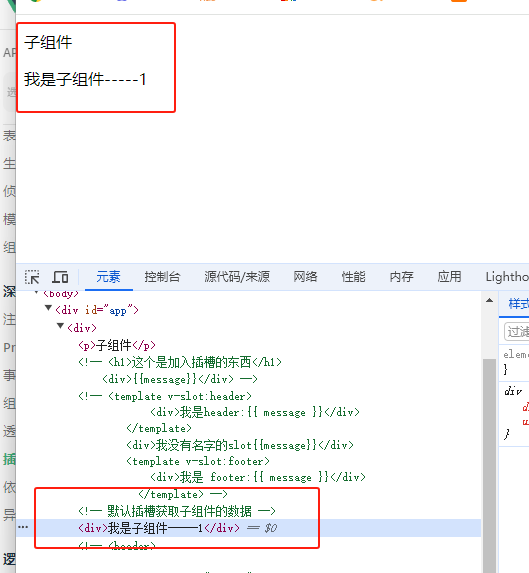
子组件：



父组件：



结果显示：



父组件 App.vue 中通过 v-slot="slotProps"等形式接收子组件传毒过来的数据，slotProps 的名字是可以任意取的，它是一个对象，包含了所有传递过来的数据

具名插槽作用域传值：

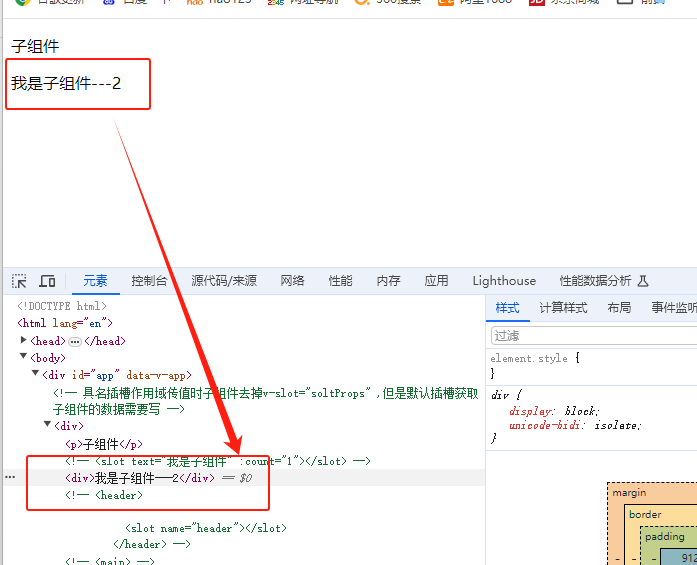
子组件：



父组件：



显示：



给 slot 添加了一个 name，在父组件中接收数据的时候不在采用 v-slot=""形式了，而是直接再插槽内容上采用#header=""形式，