## 第8章 Vue组件进阶

### 8.1 Vue的插槽slot学习

#### 8.1.1 插槽slot的基本使用

为什么要使用插槽，软件是来源于生活，如我们见到的USB插槽，电源插槽，宾馆的房间，主要目的是为了扩展和管理。

**组件使用插槽的目的**：

（1）让封装的组件更有扩展性。

（2）让使用者决定在组件内部插入具体内容。比如使用者可以向USB插入手机或键盘。

**组件使用插槽的基本原则：**

在组件开发中，保留共性的东西，个性的东西写成插槽。这样可以确保组件的复用。例：

<template id**="comp"**>  
 <div>  
 <h2>**我是标题，永远不变**</h2>  
 <!--<p>内容，随时改写,就用slot吧，让使用者去改写吧-->  
 <slot></slot>  
 </div>  
</template>

在父组件里，调用comp组件，分别放入不同的内容，在子组件的slot就呈现父组件放入的内容。

<comp><button>**按钮**</button></comp>  
<comp><input type**="text"**> </comp>  
<comp><p>**我是插入的P标签**</p></comp>

父组件调用子组件分别放入button，input，p三个标签，渲染效果如图8.1-1：



图8.1-1

子组件通过slot，以不变应万变，1个组件变成3个模样。

#### 8.1.2 插槽slot的具名使用

8.1.1节讲的插槽是没有命名，叫匿名插槽。如果给插槽起个名字，就叫具名插槽。具名插槽比较实用，就像宾馆的房间号，没有房间号的宾馆，大家只能滚大铺了：），看一下用法：

<body>  
<!-- 父组件模板 -->  
<div id**="app"**>  
 <comp><span slot**="room001"**>**老总得到插槽是room001**</span></comp>  
 <comp><span slot**="room002"**>**主管得到插槽是room002**</span></comp>  
 <comp><span slot**="room003"**>**秘书得到插槽是room003**</span></comp>  
  
 <comp><p>**员工张三没有房间号**</p></comp>  
 <comp><p>**司机李四没有房间号**</p></comp>

</div>  
<!-- 子组件模板 -->  
<template id**="comp"**>  
 <div>  
 <h2>**我是固定部分**</h2>  
 <!--房间号-->  
 <slot name**="room001"**></slot>  
 <slot name**="room002"**></slot>  
 <slot name**="room003"**></slot>  
 <!--没起名字，没分到房间号的全部到这里睡觉-->  
 <slot ></slot>  
 </div>  
</template>  
<script src**="../static/vue.js"**></script>  
<script>  
 const *app* = new Vue({  
 el:'#app',  
 data:{  
 message:"我是父Vue实例"  
 },  
 components:{  
 comp:{  
 template:"#comp"  
 }  
 }  
 })  
</script>  
</body>

看上面的代码注释，用法一目了然，再通过图8.1-2加深一下理解：

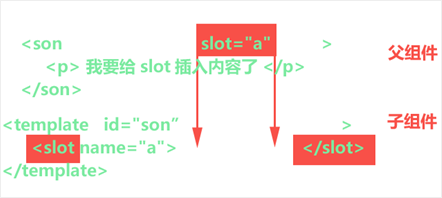


图8.1-2

形象地描述一下：父组件携带了一个名为a的slot家伙，插入到子组件<slot></slot>之间。

#### 8.1.3 作用域插槽slot的使用案例

父组件有父的作用域，子组件有子的作用域，相互之间不能直接访问。看一下父作用域的例子：

<!-- 父组件模板 -->  
<div id**="app"**>  
 <comp v-show**="isShow"**></comp>  
 <comp v-for**="item in names"**>**{{item}}** </comp>  
</div>

父组件调用子组件comp，v-show和v-for所使用的数据和变量，都是在父的作用域，即：在父的data选项里定义的数据和变量。如果父组件想使用子组件的数据，该怎么使用呢？

这就需要子组件把私有的数据暴露出来，看一下怎么将子组件数据暴露出来呢。就是通过:data属性，将私有数据books暴露出来：

<!-- 子组件模板 -->  
<template id**="comp"**>  
 <div>  
 <slot :data**="books"**>  
 <ul>  
 <li v-for**="book in books"**>**{{book}}**</li>  
 </ul>  
 </slot>  
 </div>  
</template>

如果父组件要访问子组件的data，在默认的情况下，依然访问不了，因为上面讲了，那些指令访问的依然是父作用域，这个时候，需要显示地指定插槽作用域，才可以。看一下代码：

<!-- 父组件模板 -->  
<div id**="app"**>  
 <comp>  
 <template slot-scope**="slot"**>  
 <span v-for**="item in slot.data"**>**{{item}}——**</span>  
 </template>  
 </comp>  
</div>

在父组件中，显示指定目前的作用域是slot插槽，即：<template slot-scope="slot">，这样就可以访问子组件的数据：slot.data。整个代码示例如下：

<body>  
<!-- 父组件模板 -->  
<div id**="app"**>  
 <comp>  
 <template slot-scope**="slot"**>  
 <span v-for**="item in slot.data"**>**{{item}}——**</span>  
 </template>  
 </comp>  
</div>  
<!-- 子组件模板 -->  
<template id**="comp"**>  
 <div>  
 <slot :data**="books"**>  
 <ul>  
 <li v-for**="book in books"**>**{{book}}**</li>  
 </ul>  
 </slot>  
 </div>  
</template>  
<script src**="../static/vue.js"**></script>  
<script>  
 const *app* = new Vue({  
 el:'#app',  
 data:{  
 isShow : true  
 },  
 components:{  
 comp:{  
 template:"#comp",  
 data(){  
 return {  
 books : ['语文','数学','外语','物理']  
 }  
 }  
 }  
 }  
 })  
</script>  
</body>

其中，子组件<slot :data**="books"**>这句话的意思是把数据books以data这个名称，暴露在slot位置上，父组件这句<template slot-scope**="slot"**>的意思是要引用插槽里的数据，即：slot.data。就这样，父组件成功访问子组件的数据了。