[Vue 插槽详解](https://www.cnblogs.com/chinabin1993/p/9115396.html)

Vue插槽，是学习vue中必不可少的一节，当初刚接触vue的时候，对这些掌握的一知半解，特别是作用域插槽一直没明白。

后面越来越发现插槽的好用。

分享一下插槽的一些知识吧。

**分一下几点：**

1、插槽内可以放置什么内容？

2、默认插槽

3、具名插槽

4、作用域插槽

**一、插槽内容**

　　一句话：插槽内可以是任意内容。

　　　先看一下下面的代码：声明一个child-component组件，

　　　如果现在我想在<child-component></child-component>内放置一些内容，结果会是怎样？

[复制代码](javascript:void(0);)

<div id="app">

<child-component></child-component>

</div>

<script>

Vue.component('child-component',{

template:`

<div>Hello,World!</div>

`

})

let vm = new Vue({

el:'#app',

data:{

}

})

</script>

[复制代码](javascript:void(0);)

<child-component>你好</child-component>

 输出内容还是在组件中的内容，在 <child-component>内写的内容没起作用。



这就是插槽出现的作用。

我们现在给组件增加一个<slot></slot>插槽

我们在<child-component></child-component>内写的"你好"起作用了！！！

[复制代码](javascript:void(0);)

Vue.component('child-component',{

template:`

<div>

Hello,World!

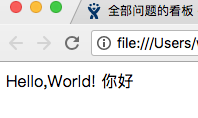
<slot></slot>

</div>

`

})

[复制代码](javascript:void(0);)



到现在，我们知道了什么是插槽：

插槽就是Vue实现的一套内容分发的API，将<slot></slot>元素作为承载分发内容的出口。

这句话的意思就是，没有插槽的情况下在组件标签内些一些内容是不起任何作用的，当我在组件中声明了slot元素后，在组件元素内写的内容

就会跑到它这里了！

**二、具名插槽**

具名插槽，就是给这个插槽起个名字

在组件中，我给插槽起个名字，一个名字叫"girl"，一个名字叫"boy"，还有一个不起名字。

然后再<child-component></child-component>内，slot属性对应的内容都会和组件中name一一对应。

而没有名字的，就是默认插槽！！

[复制代码](javascript:void(0);)

<div id="app">

<child-component>

<template slot="girl">

漂亮、美丽、购物、逛街

</template>

<template slot="boy">

帅气、才实

</template>

<div>

我是一类人，

我是默认的插槽

</div>

</child-component>

</div>

<script>

Vue.component('child-component',{

template:`

<div>

<h4>这个世界不仅有男人和女人</h4>

<slot name="girl"></slot>

<div style="height:1px;background-color:red;"></div>

<slot name="boy"></slot>

<div style="height:1px;background-color:red;"></div>

<slot></slot>

</div>

`

})

let vm = new Vue({

el:'#app',

data:{

}

})

</script>

[复制代码](javascript:void(0);)

**三、默认插槽**

上面已经介绍过了，这里不做描述

**四、作用域插槽**

之前一直没搞懂作用域插槽到底是什么！！！

**说白了就是我在组件上的属性，可以在组件元素内使用！**

先看一个最简单的例子！！

我们给<slot></slot>元素上定义一个属性say（随便定义的！），接下来在使用组件child，然后在template元素上添加属性slot-scope！！随便起个名字a

我们把a打印一下发现是 {"say" : "你好"}，也就是slot上面的属性和值组成的键值对！！！

这就是作用域插槽！

我可以把组件上的属性/值，在组件元素上使用！！

[复制代码](javascript:void(0);)

<div id="app">

<child>

<template slot-scope="a">  
　　　　　　<!-- {"say":"你好"} -->

{{a}}

</template>

</child>

</div>

<script>

Vue.component('child',{

template:`

<div>

<slot say="你好"></slot>

</div>

`

})

let vm = new Vue({

el:'#app',

data:{

}

})

</script>

[复制代码](javascript:void(0);)

再看一下下面的例子：

[复制代码](javascript:void(0);)

<div id="app">

<child :lists="nameList">

<template slot-scope="a">

{{a}}

</template>

</child>

</div>

<script>

Vue.component('child',{

props:['lists'],

template:`

<div>

<ul>

<li v-for="list in lists">

<slot :bbbbb="list"></slot>

</li>

</ul>

</div>

`

})

let vm = new Vue({

el:'#app',

data:{

nameList:[

{id:1,name:'孙悟空'},

{id:2,name:'猪八戒'},

{id:3,name:'沙和尚'},

{id:4,name:'唐僧'},

{id:5,name:'小白龙'},

]

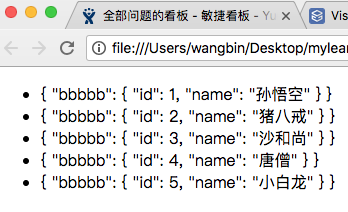
}

})

</script>

[复制代码](javascript:void(0);)

看一下输出结果



这太有用了兄弟们！！！

这样我就可以在这元素上随便玩了啊！！

当id等于1的时候，我前面加个你好。

我可以随便根据这个对象的属性值进行操作！

[复制代码](javascript:void(0);)

<child :lists="nameList">

<template slot-scope="a">

<div v-if='a.bbbbb.id==1'>你好：<span>{{a.bbbbb.name}}</span></div>

<div v-else>{{a.bbbbb.name}}</div>

</template>

</child>

[复制代码](javascript:void(0);)

最后！如果用过elementui的同学，一定知道表格就是这样来的！！

表格的本质就是这样！

如果你的才华还实现不了你的野心，那就静下心来，埋头苦干。有志者事竟成破釜成舟百二秦关终属楚，苦心人天不负卧薪尝胆三千越甲可吞吴！

参考：https://www.cnblogs.com/chinabin1993/p/9115396.html