

21. 作用域插槽

学习要点：

1. 作用域插槽

本节课我们来开始学习 Vue 的组件使用作用域插槽的方法。

一. 作用域插槽

1. 对于具名插槽，除了使用指令 `v-slot` 来进行调用，还支持#号的简写方式；

```
<html-a v-bind:text="message" #header>导航</html-a>
```

2. 由于父子组件之间是存在作用域的，它们的模版内容只能在自己作用域编译；
3. 作用域的问题导致我们使用插槽时，无法通过父组件区域直接访问子组件内容；

```
<div id="app">  
  <html-a v-bind:text="message"></html-a>  
</div>
```

```
<template id="html-a">  
  <div>  
    <span>{{text}}</span>  
    <slot>{{obj.name}}</slot>  
  </div>  
</template>
```

```
'html-a' : {  
  data() {  
    return {  
      obj : {  
        id : 1001,  
        name : 'Mr.Lee'  
      }  
    }  
  },  
  template : '#html-a',  
  props : ['text']  
}
```

4. 上面的例子中，如果在`<html-a>`中使用`{{obj.id}}`来替换掉插槽会如何？

```
<html-a v-bind:text="message">{{obj.id}}</html-a>
```

PS：作用域的问题，非子组件模版是无法获取到 `obj` 这个对象数据的；

5. 这个时候，系统提供了作用域插槽来解决这个问题，我们来看下面代码：

```
<div id="app">
  <html-a v-bind:text="message" v-slot:default="slotProps">
    {{slotProps.obj.id}}
  </html-a>
</div>

<template id="html-a">
  <div>
    <span>{{text}}</span>
    <slot v-bind:obj="obj">{{obj.name}}</slot>
  </div>
</template>
```

PS：我们通过给插槽 `<slot>` 增加一个 `v-bind` 来创建一个 `props` (插槽 `props`)；

PS：此时，父级模板中，就可以使用 `v-slot` 指令去获取子组件的数据；

6. 对于作用域插槽，还有简写方案：

```
<html-a v-bind:text="message" v-slot="slotProps">
  {{slotProps.obj.id}}
</html-a>
```

7. 如果支持 ES6(es2015)，我们还可以对其进行解构操作；

```
<html-a v-bind:text="message" v-slot="{obj:user}">
  {{user.id}}
</html-a>
```