

23. 具名插槽和作用域

学习要点：

1. 具名插槽
2. 作用域问题

本节课我们来开始学习 Vue3.x 的组件插槽的名称设置和作用域问题。

一. 具名插槽

1. 具名插槽也就是给插槽起一个名字，然后调用对应的内容，而其它则被忽略：

```
<div id="app">
  <html-a>****</html-a>
  <html-a v-slot:header>header</html-a>
</div>
```

```
// 模板
template : `
  <div>{{message}}</div>
  <div><slot name="header"></slot></div>
`
```

PS：具名插槽的好处不言而喻了，可以有效控制各种复杂的布局和内容展示；

PS：v-slot 也支持缩写方案，用#号代替即可；

```
<html-a #header>header</html-a>
```

二. 作用域问题

1. 用于作用域的问题，插槽内的值是无法获取到子组件 data 属性的内容的；

```
<html-a>{{message}}</html-a>
```

PS：这里的{{message}}到底是子组件的 message 还是父组件的 message？

2. 为了解决这个问题，插槽提供了 v-slot:default 方案来处理，具体如下：

```
<html-a v-slot:default="slotProps">
  {{slotProps.message}}
</html-a>
```

```
template : `
  <div>{{message}}</div>
  <div><slot v-bind:message="message"></slot></div>
`
```

PS: 这里命名会有误导性，这里的 `slotProps` 是可以自定义的，下面左边的 `message` 自定义；

PS: 如果改为 `v-bind:abc`，那么调用就是：`slotProps.abc` 即可；

3. 下面是作用域插槽的一些简写，具体如下：

```
<html-a v-slot="slotProps">
  {{slotProps.message}}
</html-a>
```

4. 使用 ES6+ 的解构语法，更加方便，具体如下：

```
<html-a v-slot="{message}">
  {{message}}
</html-a>
```

//或

```
<html-a v-slot="{message : info}">
  {{info}}
</html-a>
```