

20. 插槽的使用

学习要点：

1. 插槽使用

本节课我们来开始学习 Vue 的组件使用插槽的功能。

一. 插槽使用

1. 组件中有一些高级的小功能，比如：插槽<slot>，可以实现内容的动态分发；
2. 有时，我们需要在组件模板中定义大量重复的内容区域，可以用插槽来避免重复；
3. 首先，我们先看下如下代码中的场景：

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

<template id="html-a">
  <div>
    <span>{{text}}</span>
    <span>****</span>
  </div>
</template>
```

PS：其中****我们不希望是静态不变的，而是动态内容；

4. 这时，我们将需要动态内容的模板部分使用<slot>插槽来实现；

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

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

PS：<slot>插槽会获取到组件元素包含的内容进行渲染；没有插槽，组件元素内部无法被识别；

5. 当如果动态内容有较多一样，只有少部分不同的，可以设置插槽默认值；

```
<html-a v-bind:text="message"></html-a>
<slot>****</slot>
```

6. 为了实现更加更多的复杂布局，系统提供了具名插槽来实现调用机制；

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

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