

Vue 模板

从入门到工作 - Vue 全解

版权声明

本内容版权属杭州饥人谷教育科技有限公司（简称饥人谷）所有。

任何媒体、网站或个人未经本网协议授权不得转载、链接、转贴，
或以其他方式复制、发布和发表。

已获得饥人谷授权的媒体、网站或个人在使用时须注明「资料来源：饥人谷」。

对于违反者，饥人谷将依法追究责任。

联系方式

如果你想要购买本课程
请微信联系 **xiedaimala02** 或 **xiedaimala03**

如果你发现有人盗用本课程
请微信联系 **xiedaimala02** 或 **xiedaimala03**

在讲其他进阶属性之前
先学习一下 Vue 模板语法

模板 template 三种写法

- 一、Vue 完整版，写在HTML里

```
<div id=xxx>  
  {{n}}  
  <button @click="add">+1</button>  
</div>
```

```
new Vue({  
  el: '#xxx',  
  data:{n:0}, // data 可以改成函数  
  methods:{add(){}}  
})
```

模板 template 三种写法

- 二、Vue 完整版，写在选项里

```
<div id=app>  
</div>
```

```
new Vue({  
  template: ` `,  
  <div>  
    {{n}}  
    <button @click="add">+1</button>  
  </div>` ,  
  data:{n:0}, // data 可以改成函数  
  methods:{add(){ this.n += 1 }}  
}).$mount('#app')  
// 注意一个细节：div#app 会被替代
```

模板 template 三种写法

- 三、Vue 非完整版，配合 xxx.vue 文件

```
<template>
  <div>
    {{n}}
    <button @click="add">
      +1
    </button>
  </div>
</template>

<script>
export default {
  data(){ return {n:0} },
  // data 必须为函数
  methods:{add(){ this.n += 1 }}
}
</script>
<style>这里写 css </style>
```

- 然后在另外一个地方写如下代码

```
import Xxx from './xxx.vue'
// Xxx 是一个 options 对象
new Vue({
  render: h => h(Xxx)
}).$mount('#app')
```

模板里有哪些语法

我们把 HTML 模板叫做 template

展示内容

- 表达式

- ✓ `{{ object.a }}` 表达式
- ✓ `{{ n + 1 }}` 可以写任何运算
- ✓ `{{ fn(n) }}` 可以调用函数
- ✓ 如果值为 `undefined` 或 `null` 就不显示
- ✓ 另一中写法为 `<div v-text="表达式"></div>`

- HTML 内容

- ✓ 假设 `data.x` 值为 `hi`
- ✓ `<div v-html="x"></div>` 即可显示粗体的 `hi`

- 我就想展示 `{{ n }}`

- ✓ `<div v-pre>{{ n }}</div>`
- ✓ `v-pre` 不会对模板进行编译

绑定属性

- 绑定 src

```

```

- v-bind: 简写为 :

```

```

- 绑定对象

```
<div  
  :style="{border: '1px solid red', height:  
100}"> 注意这里可以把 '100px' 写成 100  
</div>
```

绑定事件

- **v-on:事件名**

```
<button v-on:click="add">+1</button>
```

- ✓ 点击之后，Vue 会运行 add()

```
<button v-on:click="xxx(1)">xxx</button>
```

- ✓ 点击之后，Vue 会运行 xxx(1)

```
<button v-on:click="n+=1">xxx</button>
```

- ✓ 点击之后，Vue 会运行 n+=1

- ✓ 即发现**函数**就加括号调用之，否则就直接运行代码

- ✓ 这导致一个问题，如果 xxx(1) 返回一个函数咋办

- ✓ 答：**用 Vue 的人怎么可能想出这么复杂的用法**

- **缩写**

```
<button @click="add">+1</button>
```

- ✓ 正常人都用缩写

条件判断

- if...else

```
<div v-if="x > 0">  
  x 大于 0  
</div>  
<div v-else-if="x === 0">  
  x 为 0  
</div>  
<div v-else>  
  x 小于 0  
</div>
```

循环

- `for (value, key) in 对象或数组`

```
<ul>
  <li v-for="(u,index) in users" :key="index">
    索引: {{index}} 值: {{u.name}}
  </li>
</ul>
```

```
<ul>
  <li v-for="(value, name) in obj" :key="name">
    属性名: {{name}}, 属性值: {{name}}
  </li>
</ul>
```

坑预警: `:key="index"` 有 bug, 后面讲

显示、隐藏

- **v-show**

- ✓ <div v-show="n%2==0"> n 是偶数 </div>

- **近似等于**

```
<div :style="{display:n%2==0?'block':'none'}>  
" n 是偶数 </div>
```

- ✓ 但你要注意，看得见的元素 display 不只有 block
- ✓ 如 table 的 display 为 table
- ✓ 如 li 的 display 为 list-item

总结

- **Vue模板主要特点有**
 - ✓ 使用 XML 语法 (不是 HTML)
 - ✓ 使用 {{ }} 插入表达式
 - ✓ 使用 v-html v-on v-bind 等指令操作 DOM
 - ✓ 使用 v-if v-for 等指令实现条件判断和循环
- **还没讲的指令**
 - ✓ v-model 在 Vue 表单里讲
 - ✓ v-slot 在 Vue 插槽里讲
 - ✓ v-cloak 自己看看[文档](#)先，在代码里讲
 - ✓ v-once 看[文档](#)，太简单不讲

指令 Directive

同义词：命令、指示

指令

- **什么是指令**

```
<div v-text="x"></div>
```

```
<div v-html="x"></div>
```

- ✓ 以 **v-** 开头的东西就是指令（名字起得不太好）

- **语法**

- ✓ **v-指令名:参数=值**, 如 **v-on:click=add**
- ✓ 如果**值**里没有特殊字符, 则可以不加引号
- ✓ 有些指令没有参数和值, 如 **v-pre**
- ✓ 有些指令没有值, 如 **v-on:click.prevent**

修饰符

- **有些指令支持修饰符**
 - ✓ @click.stop="add" 表示阻止事件传播/冒泡
 - ✓ @click.prevent="add" 表示阻止默认动作
 - ✓ @click.stop.prevent="add" 同时表示两种意思
- **一共有多少修饰符呢？**
 - ✓ v-on 支持的有 .{keycode | keyAlias}
.stop .prevent .capture .self .once .passive .native
快捷键相关 .ctrl .alt .shift .meta .exact
 - 鼠标相关 .left .right .middle
 - ✓ v-bind 支持的有 .prop .camel .sync
 - ✓ v-model 支持的有 .lazy .number .trim

修饰符

- **有些指令支持修饰符**
 - ✓ @click.stop="add" 表示阻止事件传播/冒泡
 - ✓ @click.prevent="add" 表示阻止默认动作
 - ✓ @click.stop.prevent="add" 同时表示两种意思
- **一共有多少修饰符呢？**
 - ✓ v-on 支持的有 .{keycode | keyAlias}
.stop .prevent .capture .self .once .passive .native
快捷键相关 .ctrl .alt .shift .meta .exact
 - ✓ 鼠标相关 .left .right .middle
 - ✓ v-bind 支持的有 .prop .camel .sync
 - ✓ v-model 支持的有 .lazy .number .trim

.sync 修饰符

- 场景描述

- ✓ 爸爸给儿子钱，儿子要花钱怎么办，[示例](#)
- ✓ 答：儿子打电话（触发事件）向爸爸要钱
- ✓ Vue 规则：组件不能修改 props 外部数据
- ✓ Vue 规则：this.\$emit 可以触发事件，并传参
- ✓ Vue 规则：\$event 可以获取 \$emit 的参数
- ✓ 不要问我为什么有上面三条规则，这是尤雨溪定的

- 由于这种场景很常见

- ✓ 所以尤雨溪发明了 .sync，[示例](#)
- ✓ :money.sync="total" 等价于
:money="total" v-on:update:money="total = \$event"
- ✓ 我们再讲事件的时候还会重新讲这个知识点

总结

- 搞清楚这4个修饰符就行了

- ✓ `@click.stop="xxx"`
- ✓ `@click.prevent="xxx"`
- ✓ `@keypress.enter="xxx"`
- ✓ `:money.sync="total"`

下节课回到进阶选项

下节课见