

vue基础 - day04

组件进阶 + 自定义指令 + 插槽



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



学习目标

Learning Objectives

1. 知道 **v-model语法糖**，实现组件支持双向绑定
2. 知道 **\$refs**，掌握获取dom和组件实例
3. 知道 **\$nextTick**，掌握数据更新后获取dom
4. 知道 **自定义指令**，掌握自定义指令封装
5. 掌握 **三种插槽使用**，完成组件的结构复用



目录

Contents

- ◆ 组件进阶
- ◆ 自定义指令
- ◆ 插槽
- ◆ 商品列表案例

案例效果：

编号	图片	名称	标签
101		梨皮朱泥三绝清代小品壶经典款紫砂壶	茶具
102		全防水HABU旋钮牛皮户外徒步鞋山宁泰抗菌	男鞋
103		毛茸茸小熊出没，儿童羊羔绒背心73-90cm	儿童服饰
104		基础百搭，儿童套头针织毛衣1-9岁	儿童服饰



目录

Contents

- ◆ 组件进阶
- ◆ 自定义指令
- ◆ 插槽
- ◆ 商品列表案例



思考

v-model 可以实现双向数据绑定？原理是什么呢？

v-model 语法糖

语法糖：v-model本质上是 value属性和input事件的一层包装

v-model的作用：提供数据的双向绑定

- ① 数据发生了改变，页面会自动变 v-bind:value
- ② 页面输入改变，数据会自动变化 v-on:input

```
<template>
  <div>
    <input type="text" v-model="msg">
    <input type="text" :value="msg" @input="msg = $event.target.value">

    <input type="text" :value="car" @input="car = $event.target.value">
    <input type="text" v-model="car">
  </div>
</template>
```



小结

v-model是一个语法糖怎么理解?

本质上是 value属性和input事件的一层包装



思考

v-model 都是给表单元素使用的，可以给组件使用么？

v-model给组件使用

我们经常遇到一种场景：

1. 父组件提供一个数据给子组件使用（父传子）
2. 子组件又需要修改父组件传过来的这个数据，所以需要子传父把值传给父组件。（子传父）

这种场景可以使用v-model进行简写。

```
<demo :value="money" @input="money = $event"></demo>  
<!-- 简写 -->  
<demo v-model="money"></demo>
```

```
export default {  
  props: {  
    value: Number  
  },  
  methods: {  
    add () {  
      this.$emit('input', this.value + 1)  
    }  
  }  
}
```



小结

v-model可以应用于组件么?

可以!

父传子传递的属性名，子传父触发的事件分别是什么?

value 和 input事件



思考

在 Vue中 如何获取真实DOM元素呢？

在 Vue中 能否获取组件对象呢？

ref 和 \$refs

作用：利用 ref 和 \$refs 可以用于 获取 dom 元素, 或者组件实例

- 在mounted生命周期 - 获取原生DOM标签

1. 目标标签 - 添加 ref 属性

```
<h1 ref="myH1" id="h">1. ref/id获取原生dom</h1>
```

2. 恰当时机, 通过 this.\$refs.xxx, 获取目标标签

```
mounted(){  
  console.log(document.getElementById("h"));  
  console.log(this.$refs.myH1);  
}
```



思考

问题：利用 ref 如何获取组件对象呢？

ref 和 \$refs

通过ref属性获取组件对象

1. 给Demo组件目标组件, 添加ref属性-名字随意

```
<Demo ref="de"></Demo>
```

2. 恰当时机, 通过 this.\$refs.xxx 获取组件对象, 可调用组件对象里方法等

```
mounted(){  
  this.$refs.de.fn()  
}
```



小结

ref 和 \$refs 配合使用能做什么？

帮我们获取 dom 或者 组件对象

ref 和 \$refs 怎么用呢？

1. 目标元素添加ref属性
2. this.\$refs.名字，获取内容

如果拿到了组件对象能做什么？

可以调用组件里的属性/方法



思考

问题：data数据改变，想获取到更新后DOM内容怎么办？

\$nextTick

需求：点击改data的数据, 获取原生DOM内容

1. 创建标签显示数据

```
<p ref="a">数字: {{ count }}</p>
<button @click="btn">点击+1, 观察打印</button>
```

2. 点击+1, 马上获取原生DOM内容 （拿到的是更新前的dom? ）

原因: Vue更新DOM是异步的

```
methods: {
  btn() {
    this.count++;
    console.log(this.$refs.a.innerHTML);
  }
}
```

\$nextTick

\$nextTick: 等DOM更新后, 才会触发执行此方法里的函数体

- 语法: this.\$nextTick(函数体)

```
methods: {  
  btn() {  
    this.count++;  
    this.$nextTick(() => {  
      console.log("DOM更新后触发$nextTick函数");  
      console.log(this.$refs.a.innerHTML);  
    })  
  }  
}
```



小结

data改变，更新DOM是同步还是异步的？

异步的

我们可以如何获取到更新后的DOM呢？

`this.$nextTick`里的函数体

\$nextTick练习

需求：点击按钮自身隐藏，出现输入框并马上处于激活状态

效果演示：



注意：一定要用v-if来写（保证DOM异步更新前获取不到输入框，可以体现\$nextTick的价值）

```
<input
  type="text"
  placeholder="输入要搜索的关键字"
  v-if="isShow"
  ref="inp"
/>
<button v-else @click="searchFn">点击搜索</button>
```



思考

动态组件是什么呢？

动态组件能解决什么问题？

动态组件应该怎么用呢？

dynamic 动态组件 - 是什么

问题：动态组件是什么？

是 可以改变 的 组件

问题：动态组件能解决什么需求呢？

解决多组件同一位置, 切换显示的需求

基本语法：

1. component 组件(位置) + is 属性 (哪个组件)

```
<component :is="要显示的组件" />
```

2. 修改 is 属性绑定的值 => 切换组件

思考：需要2个组件, 互斥的显示隐藏切换, 应该怎么做？

动态组件dynamic的学习

v-if => 也可以

动态组件 => 更简单

账号密码填写

个人信息填写

人生格言:

个人简介:

兴趣爱好:

dynamic 动态组件- 怎么用(案例)

1. 准备被切换的 2 个组件，并引入注册
2. 准备变量来承载要显示的"组件名"
3. 设置挂载点<component>，is属性设置要显示的组件 (**component + is**)
4. 点击按钮 - 修改comName变量里的"组件名" (**修改 is 的值**)



```
import UserAccount from './components/UserAccount'
import UserInfo from './components/UserInfo'
```

```
export default ({
  components: {
    UserAccount,
    UserInfo
  },
  data() {
    return {
      comName: 'UserAccount'
    }
  }
})
```

```
<div id="app">
  <h1>动态组件dynamic的学习</h1>
  <button class="my-btn-primary" @click="comName = 'UserAccount'">
    账号密码填写
  </button>
  <button class="my-btn-success" @click="comName = 'UserInfo'">
    个人信息填写
  </button>
  <component :is="comName" />
</div>
```




总结

1. 动态组件是什么？

是 可以改变 的 组件

2. 可以解决什么问题？

多组件同一位置, 切换显示隐藏的需求

3. 如何使用动态组件？

component组件(位置) + is 属性(显示哪个组件)



目录

Contents

- ◆ 组件进阶
- ◆ 自定义指令
- ◆ 插槽
- ◆ 商品列表案例



思考

问题1：我们学过的内置指令有哪些？

问题2：能否自己定义一些指令来使用？

11-自定义指令

自定义指令：自己定义指令，封装dom操作，扩展额外功能

- 全局注册 - 语法

```
Vue.directive("指令名", {  
  "inserted" (el) {  
    // 可以对el标签扩展额外功能  
  }  
})
```

需求：当页面加载时，让元素将获得焦点，
(autofocus 在 safari 浏览器有兼容性)

- 局部注册 - 语法

```
directives: {  
  "指令名字": {  
    inserted(el){  
      // 对el进行操作  
    }  
  }  
}
```



小结

自定义指令的作用？

封装dom操作，扩展额外功能

自定义指令 - 指令的值

需求：定义color指令-传入一个颜色, 给标签设置文字颜色

- 语法：在绑定指令时，可以通过“等号”的形式为指令绑定具体的参数值

```
<div v-color="color">我是内容</div>
```

- 通过 `binding.value` 可以拿到指令值，指令值修改会触发 `update` 函数。

```
directives: {  
  // 自定义一个局部指令  
  color: {  
    // 指令所在的元素渲染的时候  
    inserted (el, {value}) {  
      el.style.color = value  
    },  
    // update指令的值改变时触发，binding.value指令的值修改触发  
    update (el, binding) {  
      el.style.color = binding.value  
    }  
  }  
}
```



小结

指令如何传值？

v-指令名="值"

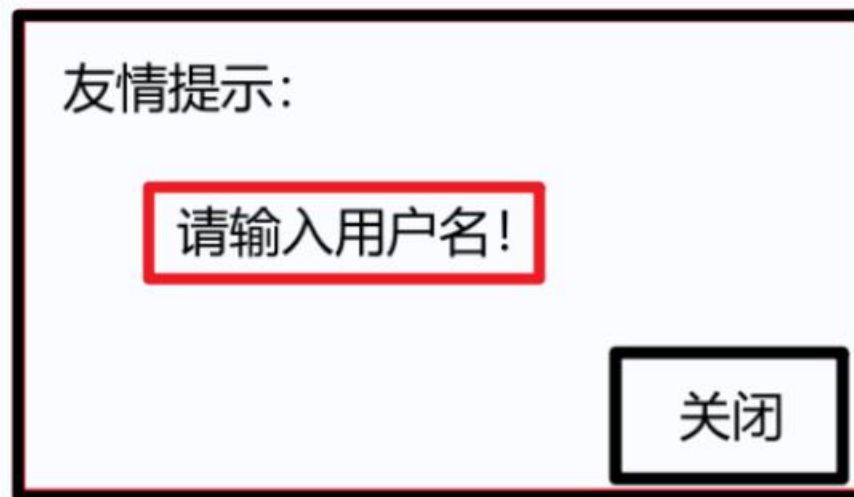
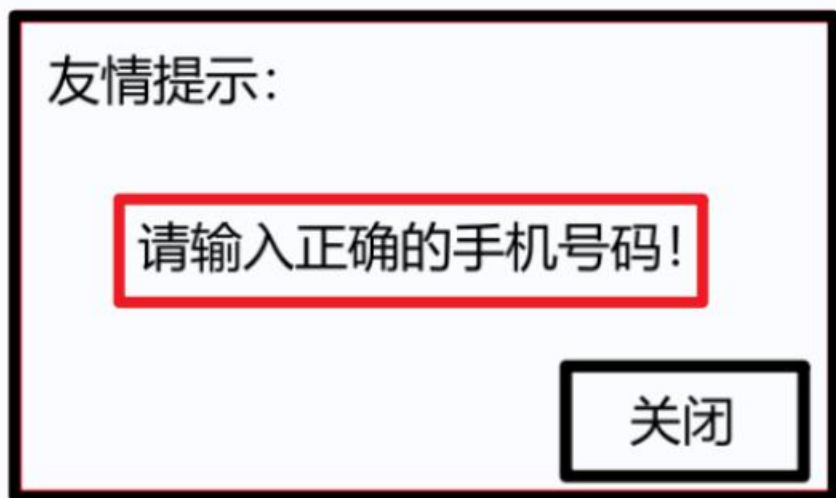
指令值变化会触发什么方法？

自定义指令的update方法

插槽 - 默认插槽

需求: 要在页面中显示一个对话框, 封装成一个组件

问题: 组件的内容部分, **不希望写死**, 希望能使用的时候**自定义**。怎么办?



- **父传子**: 父传子固然可以完成一定层面的组件文本的定制, 但是自定义性较差, 且**无法自定义结构**。
- 如果希望能够自定义组件内部的一些结构 => **就需要用到插槽**

插槽 - 默认插槽

插槽基本语法：

1. 组件内用<slot></slot>占位
2. 使用组件时<MyDialog></MyDialog>夹着的地方，传入标签替换slot

```
<template>
  <div class="my-dialog">
    <div class="header">
      <h3>友情提示</h3>
    </div>
    <div class="content">
      <slot></slot>
    </div>
    <div class="footer">
      <button>关闭</button>
    </div>
  </div>
</template>
```

```
<div id="app">
  <MyDialog>
    <p>请输入正确的手机号码</p>
  </MyDialog>
</div>
```



小结

当组件内某一部分标签不确定，想要自定义怎么办？

用插槽 slot 技术

插槽使用步骤？

1. 先在组件内用 slot 占位
2. 使用组件时，传入具体标签插入



思考

问题1: 使用我封装组件的人, 不给slot传标签, 怎么办?

问题2: 可否设置默认内容?

插槽 - 后备内容（默认值）

插槽后备内容：封装组件时，可以为预留的 `` 插槽提供**后备内容**（默认内容）。

- 语法：在<slot>标签内放置内容, 作为默认显示内容
- 效果：
 - 外部使用组件时，不传东西，则slot会显示后备内容（slot标签内的结构）
 - 外部使用组件时，传东西了，则slot整体会被换掉

```
<template>
  <div class="my-dialog">
    <div class="header">
      <h3>友情提示</h3>
    </div>
    <div class="content">
      <slot>这是后备内容</slot>
    </div>
    <div class="footer">
      <button>关闭</button>
    </div>
  </div>
</template>
```



小结

如何给插槽设置默认显示内容？

在slot标签内，写好后备内容

什么时候插槽后备内容会显示？

当使用组件并未给我们传入具体标签或内容时



思考

组件内有 2 处以上 的内容，不确定怎么办？

插槽 - 具名插槽

需求：一个组件内有多处，需要外部传入标签，进行定制

语法：

1. 多个slot使用name属性区分名字 `<slot name="content"></slot>`
2. template配合v-slot:名字来分发对应标签

v-slot:可以简化成#

```
<template v-slot:content>
  <p>你确认要退出么</p>
</template>
```

```
<template #content>
  <p>你确认要退出么</p>
</template>
```

友情提示

请输入正确的手机号码

关闭

警告

你确认要退出么

确定

取消



小结

组件内 有多处不确定的结构 怎么办?

1. slot占位, 给name属性起名字来区分
2. template配合 **v-slot:插槽名** 分发内容

v-slot:插槽名 可以简化成什么?

#插槽名



思考

问题：使用插槽时，想使用组件内的变量？

插槽 - 作用域插槽

作用域插槽: 定义 slot 插槽的同时, **是可以传值的**。给插槽上可以**绑定数据**, 将来使用组件时可以用。

基本使用步骤:

1. 给 slot 标签, 以 添加属性的方式传值

```
<slot name="bottom" :yes="yes" :no="no" money="100"></slot>
```

2. 所有添加的属性, 都会被收集到一个对象中

```
{ yes: '确认', no: '取消', money: '100' }
```

3. 在template中, 通过 `v-slot:插槽名="obj"` 接收

```
<template #bottom="obj">
  <!-- {{ obj }} -->
  <button>{{ obj.yes }}</button>
  <button>{{ obj.no }}</button>
  <button>{{ obj.money }}</button>
</template>
```

作用域插槽的作用是什么？

可以给插槽上绑定数据，供将来使用组件时使用

作用域插槽使用步骤？

- (1) 给 slot 标签, 以 添加属性的方式传值
- (2) 所有属性都会被收集到一个对象中
- (3) template中, 通过 `v-slot:插槽名= "obj"` 接收



小结



目录

Contents

- ◆ 组件进阶
- ◆ 自定义指令
- ◆ 插槽
- ◆ 商品列表案例

案例

商品列表（表格组件封装）

编号	图片	名称	标签
101		梨皮朱泥三绝清代小品壶经典款紫砂壶	茶具
102		全防水HABU旋钮牛皮户外徒步鞋山宁泰抗菌	男鞋
103		毛茸茸小熊出没，儿童羊羔绒背心73-90cm	儿童服饰
104		基础百搭，儿童套头针织毛衣1-9岁	儿童服饰



思考

1. ref 和 \$refs 的作用？
2. vue的dom更新是同步的还是异步的？\$nextTick 的作用？
3. 什么是动态组件？
4. 插槽的作用？插槽的分类？
5. 作用域插槽的作用？
6. 自定义指令作用？



今日作业

把课上例子, 练习, 案例做一遍

预习第二天内容



传智教育旗下高端IT教育品牌