

lesson9

1. 单元产出物

- [搜索列表组件封装](#)
 - v-model指令原理讲解 **【提薪】**

- how:

v-model就是vue的双向绑定的指令，能将页面上控件输入的值同步更新到相关绑定的data属性，也会在更新data绑定属性时候，更新页面上输入控件的值。

- what:

v-model作为双向绑定指令也是vue两大核心功能之一，使用非常方便，提高前端开发效率。在view层，model层相互需要数据交互，即可使用v-model。

- why:

v-model主要提供了两个功能，view层输入值影响data的属性值，data属性值发生改变会更新view层的数值变化。

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
</head>
<body>
<div id="app">
  <p>{{name}}</p>
  <input type="text" v-model="name" />
</div>
</body>
</html>
```

```
node.addEventListener("input",function(e){
  vm.name = e.target.value;
})
Object.defineProperty(data,"name",{
  get(){
    return data["name"];
  },
  set(newVal){
    let val = data["name"];
    if (val === newVal){
      return;
    }
    data["name"] = newVal;
```

```
// 监听到了属性值的变化,假如node是其对应的input节点
node.value = newVal;
    }
  })
```

- v-model作用在组件上

```
<input :value="value" @input="this.value = $event.target.value" />

<input v-model="value" />
```

```
// 父组件
<template>
  <my-children v-model="title" />
</template>
<script>
  export default {
    data(){
      return {
        title:'title'
      }
    }
  }
</script>

// 子组件
<template>
  <div>
    <input :value="inputValue" @input="$emit('update',
    $event.target.value)"/>
  </div>
</template>
<script>
  export default {
    props:{
      value: String
    },
    model: {
      prop: "value", //绑定的值, 通过父组件传递
      event: "update" //自定义时间名
    }
  }
</script>
```

2. 内置组件

- component组件 动态组件

```
<component :is="组件名/组件选项"></component>
```

- 过渡效果 transition

```
<transition>  
  <div v-if="ok">toggled content</div>  
</transition>
```

- 递归组件

组件是可以在它们自己的模板中调用自身的。不过它们只能通过name 选项来做这件事