

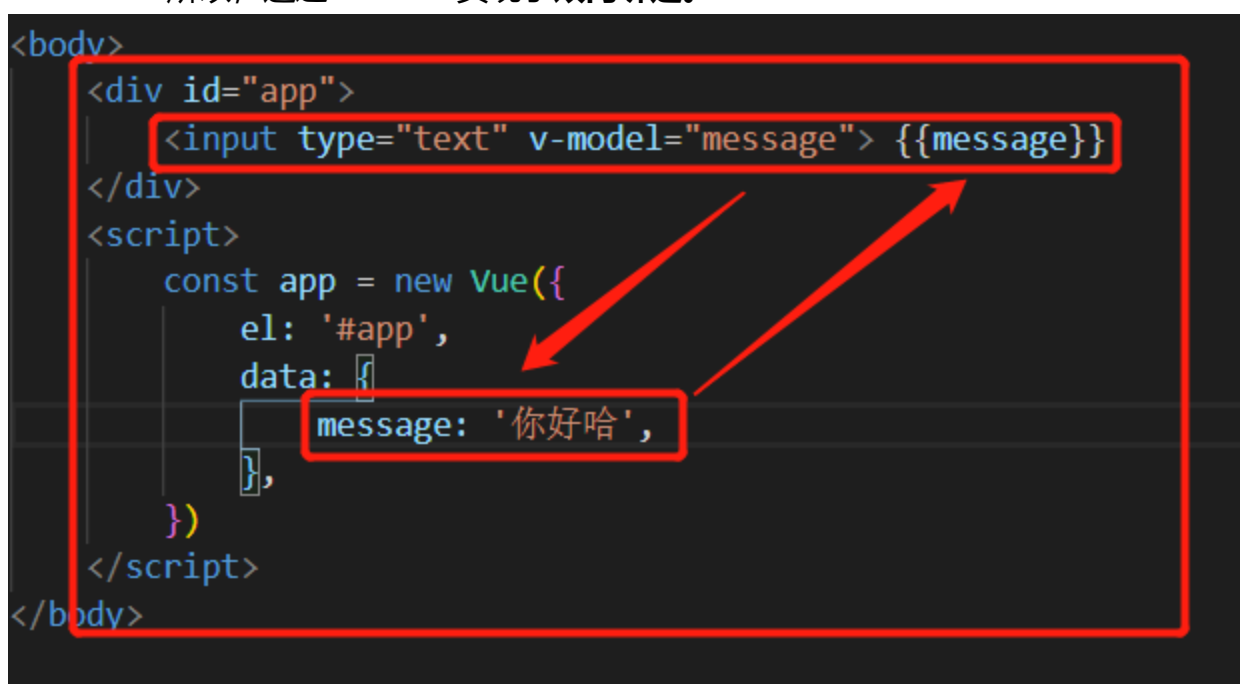
一、v-model

1. 表单控件在实际开发是非常常见的。特别是对于用户信息的提交，需要大量的表单。

2. Vue 中使用v-model指令来实现表单元素和数据的双向绑定。

3. 案例的解析：

- 当我们在输入框输入内容时
- 因为input中的v-model绑定了message，所以会实时将输入的内容传递给message，message发生改变。
- 当message发生改变时，因为上面我们使用Mustache语法，将message的值插入到DOM中，所以DOM会发生响应的改变。
- 所以，通过v-model实现了**双向绑定**。

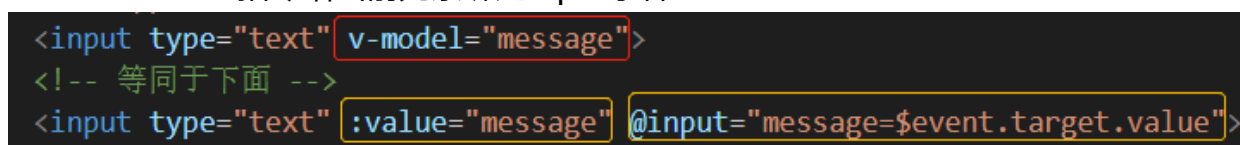


4. 当然，我们也可以将v-model用于textarea元素

二、v-model原理

v-model其实是一个语法糖，它的背后本质上是包含两个操作：

- v-bind 绑定一个value属性
- v-on指令给当前元素绑定input事件



三、v-model:radio



```
<div id="app">
  <label for="male">
    <input type="radio" id="male" value="男" v-model='sex'>男
  </label>
  <label for="female">
    <input type="radio" id="female" value="女" v-model='sex'>女
  </label>
  <h2>您选择的性别是:{{sex}}</h2>
</div>
<script>
  const app = new Vue({
    el: '#app',
    data: {
      sex: '男' //默认值
    }
  })
</script>
```

四、v-model: checkbox

复选框分为两种情况：单个勾选框和多个勾选框

1. 单个勾选框：

- v-model即为Boolean值
- 此时input的value并不影响v-model的值

2. 多个复选框：

- 当是多个复选框时，因为可以选中多个，所以对应的data中属性是一个数组。
- 当选中某一个时，就会将input的value添加到数组中。



```
<div id="app">
  <!-- label的好处是点文字也会选中 -->
  <!-- 1.checkbox单选框 -->
  <label for="agree">
    <input type="checkbox" id="agree" v-model="isAgree">同意协议
  </label>
  <h2>是否选中: {{isAgree}}</h2>
  <button :disabled="!isAgree">下一步</button>
  <hr>
  <!-- 2.checkbox多选框 -->
  <label for="hobbies">
    <input type="checkbox" value="篮球" v-model="hobbies">篮球
    <input type="checkbox" value="足球" v-model="hobbies">足球
    <input type="checkbox" value="乒乓球" v-model="hobbies">乒乓球
    <input type="checkbox" value="羽毛球" v-model="hobbies">羽毛球
  </label>
  <h2>您的选择的兴趣爱好是:{{hobbies}}</h2>
</div>
<script>
  const app = new Vue({
    el: '#app',
    data: {
      message: '你好哈',
      isAgree: false, //单选框
      hobbies: [] //多选框
    }
  })
</script>
```

五、v-model:select

和checkbox一样，select也分单选和多选两种情况

1. 单选：只能选中一个值。

- v-model绑定的是一个值
- 当我们选中option中的一个时，会将对应的value赋值到mySelect中

2. 多选：可以选中多个值

- v-model绑定的是一个数组

- 当选中多个值时，就会将选中的option对应的value添加到数组mySelects中

```
<div id="app">
  <!-- 1. 选择一个 -->
  <select name="myFruit" v-model="fruit">
    <option value="苹果">苹果</option>
    <option value="香蕉">香蕉</option>
    <option value="榴莲">榴莲</option>
    <option value="葡萄">葡萄</option>
  </select>
</div>
<h2>您选择的水果是:{{fruit}}</h2>
<hr>
<!-- 2. 选择多个 -->
<select name="myFruit" v-model="fruits" multiple>
  <option value="苹果">苹果</option>
  <option value="香蕉">香蕉</option>
  <option value="榴莲">榴莲</option>
  <option value="葡萄">葡萄</option>
</select>
</div>
<h2>您选择的水果是:{{fruits}}</h2>
</div>
<script>
const app = new Vue({
  el: '#app',
  data: {
    fruit: "香蕉",
    fruits: []
  },
})
</script>
```

六、值绑定

初看Vue官方值绑定的时候，我很疑惑：what the hell is that?

但是仔细阅读之后，发现很简单，就是动态的给value赋值而已：

- 我们前面的value中的值，可以回头去看一下，都是在定义input的时候直接给定的。
- 但是真实开发中，这些input的值可能是从网络获取或定义在data中的。
- 所以我们可以通过v-bind:value动态的给value绑定值。
- 这不就是v-bind吗？

这不就是v-bind在input中的应用吗？会用**v-bind**，就会值绑定的应用了

```

<label v-for="item in originHobbies" :for="item">
  <input type="checkbox" :id="item" :value="item" v-model="hobbies" {{item}}
</label>

</div>
<script>
  const app = new Vue({
    el: '#app',
    data: {
      message: '你好哈',
      isAgree: false, // 单选框
      hobbies: [], // 多选框
      originHobbies: ['篮球', '足球', '乒乓球', '羽毛球', '台球', '高尔夫球']
    },
  })
</script>

```

七、修饰符

1.lazy修饰符：

- 默认情况下，v-model默认是在input事件中同步输入框的数据的，
- 也就是说，一旦有数据发生改变对应的data中的数据就会自动发生改变
- lazy修饰符可以让数据在失去焦点或者回车时才会更新：

2.number修饰符：

- 默认情况下，在输入框中无论我们输入的是字母还是数字，都会被当做字符串类型进行处理。
- 但是如果我们希望处理的是数字类型，那么最好直接将内容当做数字处理。
- number修饰符可以让在输入框输入的内容自动转成数字类型：

3.trim修饰符：

- 如果输入的内容首尾有很多空格，通常我们希望将其去除
- trim修饰符可以过滤内容左右两边的空格