

使用v-model实现双向绑定

使用v-model实现双向绑定

- Text 数据双向绑定
- Checkbox 数据双向绑定
- Radio 数据双向绑定
- Select 数据双向绑定

使用v-model实现双向绑定

Text 数据双向绑定

- <INPUT>
 - 类型为 text、number、range、date、time、email、url.....
 - checkbox、radio 类型除外
 - file 类型谨慎使用
- <TEXTAREA>

<input v-model="user.name">

<input type="range" v-model="user.age">

<textarea v-model="user.profile"></textarea>

使用v-model实现双向绑定

Checkbox 数据双向绑定

```
<input type="checkbox" v-model="checked">  
<input type="checkbox" value="Jack" v-model="checkedNames">  
<input type="checkbox" value="John" v-model="checkedNames">  
<input type="checkbox" value="Mike" v-model="checkedNames">
```

```
new Vue({  
  el: 'body',  
  data: {  
    checked: false,  
    checkedNames: []  
  }  
})
```

使用v-model实现双向绑定

Radio 数据双向绑定

```
<label><input type="radio" value="One" v-model="picked">One</label>  
<label><input type="radio" value="Two" v-model="picked">Two</label>  
<span>Picked: {{ picked }}</span>
```

```
new Vue({  
  el: 'body',  
  data: {  
    picked: ''  
  }  
})
```

使用v-model实现双向绑定

Select 数据双向绑定

单选

```
<select v-model="selected">
  <option selected>A</option>
  <option>B</option>
  <option>C</option>
</select>
<span>
  Selected: {{ selected }}
</span>
```

多选

```
<select v-model="arrSelected"
multiple>
  <option selected>A</option>
  <option>B</option>
  <option>C</option>
</select>
<br>
<span>
  Selected: {{ arrSelected | json }}
</span>
```

绑定value

绑定value

- Checkbox 绑定 Value
- Radio 绑定 Value
- Select Options 绑定 Value

Checkbox 绑定 Value

单个 Checkbox

```
<input  
  type="checkbox"  
  v-model="toggle"  
  v-bind:true-value="a"  
  v-bind:false-value="b">
```

```
// 当选中时  
vm.toggle === vm.a  
// 当没有选中时  
vm.toggle === vm.b
```

多个 Checkbox

```
<template v-for="insurancetm in insurancetems">  
  <input  
    type="checkbox"  
    v-model="insurances"  
    v-bind:value="insurancetm"> {{insurancetm}}  
</template>
```

绑定value

Radio 绑定 Value

```
<input type="radio" v-model="pick" v-bind:value="a">
```

```
// 当选中时  
vm.pick === vm.a
```

Select Options 绑定 Value

```
<select v-model="selected">  
  <!-- 对象字面量 -->  
  <option v-bind:value="{ number: 123 }">123</option>  
</select>
```

// 当选中时

```
typeof vm.selected // -> 'object'  
vm.selected.number // -> 123
```

参数特性

参数特性

- lazy 特性
- number 特性
- debounce 特性

参数特性

lazy 特性

<!-- 在 "change" 而不是 "input" 事件中更新 -->
<input v-model="msg" lazy>

参数特性

number 特性

```
<input v-model="age" number>
```

参数特性

debounce 特性

```
<input v-model="msg" debounce="500">
```