

1 / 1 Vue

指令：

v-model：在表单控件中使用。（将数据和视图绑定 -> 双向数据绑定）

（我感觉这个相当于给他绑定一个东西。绑定的东西随意，根据需求而定。）

在表单控件中只用 v-model：

```
<input>
<select>
<textarea>
```

当在表单元素中，绑定了v-model，那么只要用户改变视图中的内容，那么就等同于直接改变数据。当然改变数据一样会改变视图。

在使用v-model的时候，如果放个数组，那么会自动查找表单元素中的value。如果没有val，那么就是null，如果要统计多个表单元素的信息，可以把同一个数据绑定到v-model上。

v-model：中要赋值数据变量。当它单独用时，变量可以是字符串、数组、对象。

如果和 v-for 组合使用时，v-model赋值的变量，只能为 数组（数组不行的话就在里面每一项套个对象，套的对象的val值不能用引号包。），对象没进行测试。

```
<body>
  <div id="app">
    <input
      type="text"
      @keyup.13 = 'up'
      v-model = 'val'
    />{{val}} // 此处的val值是数据里面的val值。
    <ul>
      <li
        v-for="(val, key) in arr"
      >{{val}} // 此处的val 是数组arr中的val。
      </li>
    </ul>
  </div>

  <script src="./vue.js"></script>
```

```

<script>
  new Vue({
    el:"#app",
    data:{
      arr:[1, 2, 3],
      val:''
    },
    methods:{
      up(){
        console.log(1)
        this.arr.push(this.val)
        this.val = ''
      }
    }
  })
</script>
</body>

```

<!--小例子 001 拿到选中的人名-->

```

<div id="app">
  小明:<input type="checkbox" value="小明" v-model="arr">
  <hr>
  小刚:<input type="checkbox" value="小刚" v-model="arr">
  <hr>
  小红:<input type="checkbox" value="小红" v-model="arr">
  <hr>
  {{arr}}
</div>

```

```

// js 部分
new Vue({
  el:"#app",
  data:{
    arr:[]
  }
})

```

<!--如果要统计多个表单元素的信息，可以把同一个变量（数组或对象），通过v-model绑定在一块，然后通过这个数组统计多个表单信息-->

<!--小例子 002-->

```

<div id="app" 绑定数据或对象，操作视图>
  <button @click="fn001">全选</button>
  <button @click="fn002">反选</button>
  <button @click="fn003">全不选</button>

```

```

    <input
      type="checkbox"
      v-for = "(val,key) in arr"
      v-model = "val.c"
    >
    {{arr}}
  </div>

// js 部分
new Vue({
  el:"#app",
  data:{
    arr:[false,false,false,false,false]
    arr:[{"c":false},{"c":false},{"c":false},{"c":false},
{"c":false}]
  },
  methods:{
    fn001(){
      console.log()
      this.arr.map(e=>e.c =true)
    },
    fn002(){
      this.arr.map(e=> e.c= !e.c)
    },
    fn003(){
      this.arr.map(e=> e.c=false)
    }
  }
})

```

<!--如果数组中是对象，那么在改变对象中的属性时候，不用直接数据数组（原数组）如果数据组中不是对象，那改变数组的某个值的时候，要改变原数组。-->

```

<!-- 小例子003 单选 -->
<div id="app">
  小明:<input type="radio" name="a" value= '小明' v-model= 'b'>
  <hr>
  小刚:<input type="radio" name="a" value= '小刚' v-model= 'b'>
  <hr>
  小红:<input type="radio" name="a" value= '小红' v-model= 'b'>
  <hr>
  {{b}}
</div>

// js 部分
new Vue({
  el:' #app',
  data:{

```

```

        b: ''
    }
  })

<!-- 选地名 -->

<div id="app">
  <select name="" id="" v-model="b">
    <option disabled value="">请选择</option> // disabled 有这个属性该项就不可被选定了。
    <option value="北京">北京</option>
    <option value="上海">上海</option>
    <option value="杭州">杭州</option>
  </select>
  {{b}}
</div>

```

// 小例子：选地名：js部分

```

new Vue({
  el: '#app',
  data: {
    b: ''
  }
})

```

<!--在select中，直接绑上v-model，如果option中有value，优先走value，如果没有value（没有不是为'',是直接不写" val= '' "），则走里面的内容。-->

<!-- todo： 小例子：添加，显示、删除、全选-->

```

<div id="app">
  <input type="text" @keyup.13="fn" v-model.trim="val">{{val}}
  <button @click="rmAll">删除</button>
  <button @click="all">全选</button>
  <ul>
    <li v-for= "(val,key) in arr">
      <input type="checkbox" v-model="val.checked">
        {{val.txt}}
    </li>
  </ul>
</div>

```

// js 部分

```

new Vue({
  el: "#app",
  data: {
    arr: [

```

```

        {
            checked: true,
            txt: '小刚'
        },
        {
            checked: false,
            txt: '小明'
        },
        {
            checked: false,
            txt: '小红'
        },
        {
            checked: false,
            txt: '小美'
        }
    ],
    al: ''
},
methods: {
    fn() {
        // this.arr.push({checked: 'false', txt: this.val});
        this.arr.unshift({
            checked: false,
            txt: this.val
        })
        this.val = ''
    },
    rmAll() {
        this.arr = this.arr.filter(e => !e.checked) // 过滤方法
    },
    all() {
        for (let i = 0; i < this.arr.length; i++) {
            if (!this.arr[i].checked) {
                this.arr[i].checked = !this.arr[i].checked
            }
        }
    }
}
}

/* 数据为数组中包含对象情况的时候：直接赋值一个新的对象，使用filter是可以的。
*/
})

```

修饰符(在事件名称后面，有些可以连着放多个修饰符（好像有些会有冲突，看实际情况）)

常用的：

.stop -> event.stopPropagation(); 阻止冒泡。
.prevent -> event. : 阻止默认行为（拖动效果等。。。）
.trim -> 去掉首尾空格。

computed（属于Vue的计算属性）

定义：（有两项技能，主动技能，和被动技能）

- 是Vue专有的一个计算属性：
- 当数据发生变化的时候，通过变化的数据，做另一件事，只要使用，一开始就会执行一次，之后只要数据发生变化就执行。

函数（ ）{

使用data中的数据，自动帮你监听数据的变化，返回的结果就是通过改变指定数据做另一件事情。

比如：通过数据的变化，来进行累计。

比如：通过单选按钮的变化，来进行判断

}

使用：

在html（模板中）中{函数名}

注意：函数名，不要和data中的数据一样。

优点：指定看那个数据就看那个数据

get(){ 改变数据 }：获取（当绑定的内容发生变化时触发）-> 被动技能

set(newValue){ }：改变之后的结果（当这个元素本身发生变化的时候触发）-> 主动技能

newValue j就是改变之后的数据结果

当使用get、set的时候，那么computed中定义的属性，为一个对象，当不使用get、set 的时候，computed中定义的属性是可以为函数。

```
<body>
  <div id="app">
    全选:<input
      type="checkbox"
      v-model="all"
    >
    <hr>
    <input
      v-for="(val,key) in arr"
      type="checkbox"
      v-model="val.checked"
    >
    {{arr}}
```

```

    </div>
<script src="./vue.js"></script>
<script>
    new Vue({
      el: '#app',
      data: {
        arr: [
          {
            checked: false
          },
          {
            checked: false
          },
          {
            checked: false
          }
        ],
        computed: {
          all: {
            get() {
              return this.arr.every(e=>e.checked); // 被动技能-
            },
            set(newValue) {
              return this.arr.forEach(e=>e.checked = newValue); // 主动技能 --> 当这个函数绑定的本身标签改变时，set函数里面的内容。
            }
          }
        }
      }
    });

```

// 难点: v-model为true时 全选会选上, 为false时, 全选未选上。 因为: 有v-model时, 这个标签的视图和v-model后面的数据会绑定到一块。all在这个里面为一个对象(相当于all为一个变量名), 当对象为true时, 所以全选会选中, 当为false时, 全选会被取消。

获取节点：

```

var chils= s.childNodes; //得到s的全部子节点
var par=s.parentNode; //得到s的父节点
var ns=s.nextSibling; //获得s的下一个兄弟节点
var ps=s.previousSibling; //得到s的上一个兄弟节点
var fc=s.firstChild; //获得s的第一个子节点
var lc=s.lastChild; //获得s的最后一个子节点

```