用 v-model 指令在表单控件元素上创建双向数据绑定。

<template>

  <div id="app">

    <input v-model="message" placeholder="编辑我……" />

    <textarea v-model="message2" placeholder="多行文本输入……"></textarea>

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      message: "Runoob",

      message2: "菜鸟教程\r\nhttp://www.runoob.com"

    };

  }

};

</script>

**复选框**

复选框如果是一个为逻辑值，如果是多个则绑定到同一个数组：

<template>

  <div id="app">

    <p>单个复选框：</p>

    <input type="checkbox" id="checkbox" v-model="checked" />

    <label for="checkbox">{{ checked }}</label>

    <p>多个复选框：</p>

    <input type="checkbox" id="runoob" value="Runoob" v-model="checkedNames" />

    <label for="runoob">Runoob</label>

    <input type="checkbox" id="google" value="Google" v-model="checkedNames" />

    <label for="google">Google</label>

    <input type="checkbox" id="taobao" value="Taobao" v-model="checkedNames" />

    <label for="taobao">taobao</label>

    <br />

    <span>选择的值为: {{ checkedNames }}</span>

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      checked: false,

      checkedNames: []

    };

  }

};

</script>

**单选按钮**

以下实例中演示了单选按钮的双向数据绑定：

<template>

  <div id="app">

    <input type="radio" id="runoob" value="Runoob" v-model="picked" />

    <input type="radio" id="google" value="Google" v-model="picked" />

    <span>选中值为: {{ picked }}</span>

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      picked: "Runoob"

    };

  }

};

</script>

**select 列表**

以下实例中演示了下拉列表的双向数据绑定：

<template>

  <div id="app">

    <select v-model="selected" name="fruit">

      <option value>选择一个网站</option>

      <option value="www.runoob.com">Runoob</option>

      <option value="www.google.com">Google</option>

    </select>

    <div id="output">选择的网站是: {{selected}}</div>

  </div>

</template>

<script>

export default {

  name: "app",

  data() {

    return {

      selected: ""

    };

  }

};

</script>

**修饰符**

.number

如果想自动将用户的输入值转为 Number 类型（如果原值的转换结果为 NaN 则返回原值），可以添加一个修饰符 number 给 v-model 来处理输入值：

<input v-model.number="age" type="number">

.trim

如果要自动过滤用户输入的首尾空格，可以添加 trim 修饰符到 v-model 上过滤输入：

<input v-model.trim="msg">