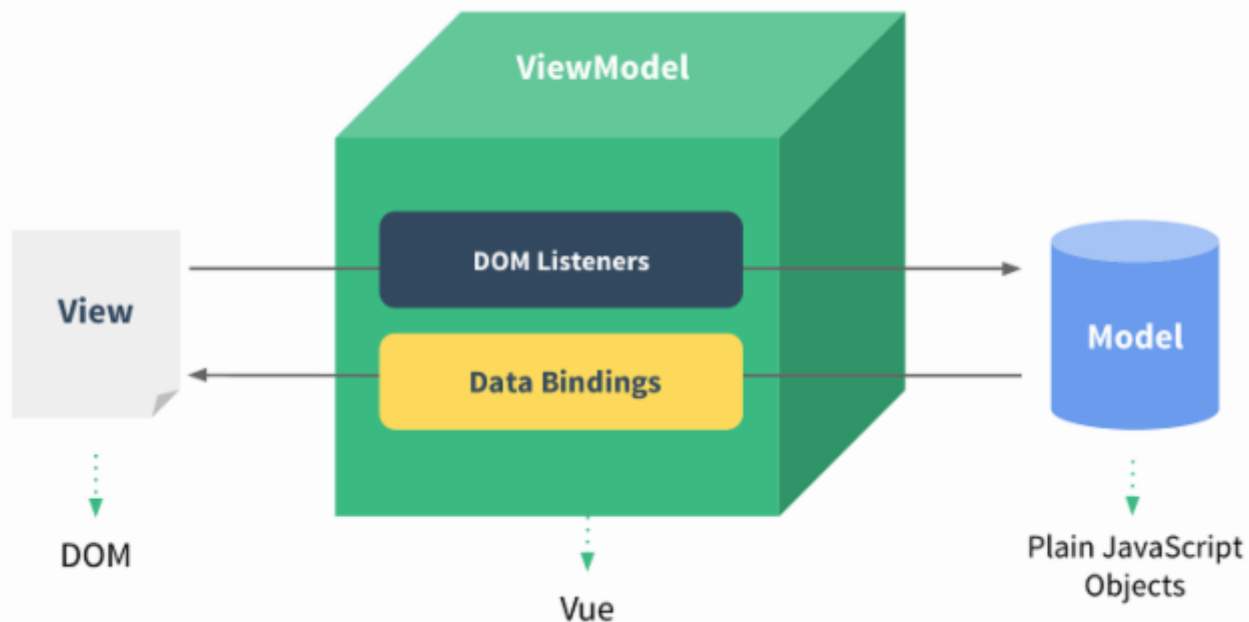


Vue.js 表单

这节我们为大家介绍 Vue.js 表单上的应用。

你可以用 v-model 指令在表单控件元素上创建双向数据绑定。



v-model 会根据控件类型自动选取正确的方法来更新元素。

输入框

实例中演示了 input 和 textarea 元素中使用 v-model 实现双向数据绑定：

```
<div id="app">
  <p>input 元素: </p>
  <input v-model="message" placeholder="编辑我.....">
  <p>消息是: {{ message }}</p>
  <p>textarea 元素: </p>
  <p style="white-space: pre">{{ message2 }}</p>
  <textarea v-model="message2" placeholder="多行文本输入....."></textarea>
</div>
<script>
new Vue({
  el: '#app',
  data: {
    message: 'Runoob',
    message2: '菜鸟教程\r\nhttp://www.runoob.com'
  }
})
</script>
```

[尝试一下 »](#)

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：

复选框

以下实例中演示了复选框的双向数据绑定：

```
<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>
<script>
new Vue({
  el: '#app',
  data: {
    checked : false,
    checkedNames: []
  }
})
</script>
```

尝试一下 »

实例中勾选复选框效果如下所示：

单个复选框：

☒ true

多个复选框：

☒ Runoob ☒ Google ☐ taobao
选择的值为: ["Runoob", "Google"]

单选按钮

以下实例中演示了单选按钮的双向数据绑定：

单选按钮

```
<div id="app">
<input type="radio" id="runoob" value="Runoob" v-model="picked">
<label for="runoob">Runoob</label>
```

```
<br>
<input type="radio" id="google" value="Google" v-model="picked">
<label for="google">Google</label>
<br>
<span>选中值为: {{ picked }}</span>
</div>
<script>
new Vue({
  el: '#app',
  data: {
    picked : 'Runoob'
  }
})
</script>
```

[尝试一下 »](#)

选中后，效果如下图所示：

☒ Runoob
☐ Google
选中值为: Runoob

select 列表

以下实例中演示了下拉列表的双向数据绑定：

select

```
<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>
<script>
new Vue({
  el: '#app',
  data: {
    selected: ''
  }
})
</script>
```

[尝试一下 »](#)

选取 Runoob，输出效果如下所示：

选择的网站是: www.runoob.com

修饰符

.lazy

在默认情况下，v-model 在 input 事件中同步输入框的值与数据，但你可以添加一个修饰符 lazy，从而转变为在 change 事件中同步：

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

.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">
```

这通常很有用，因为在 type="number" 时 HTML 中输入的值也总是会返回字符串类型。

.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">
```

[← Vue.js 事件处理器](#)[Vue.js 组件 →](#)**4 篇笔记** **写笔记**