课程目标

v-model

v-model是用于表单元素<input> <textarea>及<select>上创建双向数据绑定的语法糖，在事件和样式一节课程当中，我们知道父组件传递子组件，通过自定义属性 :props传递到子组件，子组件用props来接收，并通过$emit触发修改通知到父组件。同时在自定义属性添加 .sync修饰符，达到双向绑定。其中.sync修饰符其实是 v-bind:msg和v-on:update:msg的简写

表单处理

那v-model是跟什么样的约定来实现的呢?

其实的v-model作用于不同标签上面会有不同的约定规则。其实在vue源码，是使用了一个策略模式，根据不同的模式来使用不同的策略。下来看下具体的策略。

- <select>元素使用value属性作为props，使用chang事件来监听更新props数据。
- <textarea>和<input>元素使用 value实行作为props，使用input事件来监听更新props数据。如果想通过change事件触发，加上lazy修饰符即可。如:

```
1  <!-- 在“change”时而非“input”时更新 -->
2  <input v-model.lazy="msg" >
```

- <input type='checkbox' />和 <input type='radio' /> 使用checked属性为props，使用change事件来监听更新props数据。

下面来具体解析v-model 语法糖的实现：如下图

<input type='text' /> 的v-model

```
<input type="text" v-model="text" />
```

其实就是 value作为props，input事件作为来监听更新props数据。v-model就是下面实现的语法糖。vue框架做了更深层次的兼容封装，输入法为中文，且还没有输入成功的时候，是不会双向绑定进行的。

```
<input type="text" :value="text" @input="text = $event.target.value" />
```

text-area和input是一样得这里省略了。注意：v-model的双向绑定与vue中子组件向父组件中传递，多一个$emit触发，这个要注意。

注意:v-model除了有 .lazy修饰符，还有trim修饰符(双向绑定去除前后空格) 还有number(字符串转number，无需手动转number)

<input type='radio' /> 和 <input type='checkbox' /> v-model
<select>的v-model就不一一举例了

自定义组件v-model

在了解v-model的语法糖之后，我们可以自定义组件的v-model吗？答案是肯定的。vue2.x之后都加了这个新的特性。下面就是可以自定义一个select组件

子组件

```vue
<template>
  <div>
    <div class="top" @click="showBottom = !showBottom">{{ selected.name }}</div>
    <div class="bottom" v-if="showBottom">
      <div v-for="i in list" :key="i.value" @click="select(i)">{{ i.name }}</div>
    </div>
  </div>
</template>

<script>
export default {
  model: {
    prop: "selected",
    event: "change"
  },
  props: ["list", "selected"],
  data() {
    return {
      showBottom: false
    };
  },
  methods: {
    select(i) {
      this.$emit("change", i);
      this.showBottom = false;
    }
  }
};
</script>
```

子组件新增model， prop event键值

props把model中的prop添加进去

通过$emit触发model中的event事件

父组件

```vue
<template>
  <div>
    <h3>Custom</h3>
    <s-custom-select v-model="selected" :list="list"></s-custom-select>
    你选择了: {{ selected.name }}
  </div>
</template>

<script>
import SCustomSelect from "./SCustomSelect";
```

直接用v-model来接收，无需通过@change 来监听接收