note.md

# Vue组件库 - Unit02

# 1.关于Mint UI表单控件获取/失去焦点

<mt-field type="text"
  
 @blur.native.capture="函数名称"
  
 @focus.native.capture="函数名称">
  
</mt-field>

native、capture、stop、prevent属于Vue的事件修饰符

native - 监听组件根元素的原生事件

capture - 添加事件侦听器时使用 capture 模式

stop - 调用 event.stopPropagation()

prevent - 调用 event.preventDefault()

参照https://cn.vuejs.org/v2/api/#v-on

src/views/Register.vue的示例代码如下：

<template>
  
 <div>
  
 <!-- 顶部导航开始 -->
  
 <mt-header title="用户注册">
  
 <router-link to="/" slot="left">
  
 <mt-button icon="back"></mt-button>
  
 </router-link>
  
 </mt-header>
  
 <!-- 顶部导航结束 -->
  
 <!-- 表单区域开始 -->
  
 <div>
  
 <mt-field
  
 type="text"
  
 label="用户名"
  
 placeholder="请输入用户名"
  
 v-model="username"
  
 @blur.native.capture="checkUsername"
  
 :state="usernameState">
  
 </mt-field>
  
 <mt-field
  
 type="password"
  
 label="密码"
  
 placeholder="请输入密码"
  
 v-model="password"
  
 @blur.native.capture="checkPassword"
  
 :state="passwordState">
  
 </mt-field>
  
 <mt-field
  
 type="password"
  
 label="确认密码"
  
 placeholder="请再次输入密码"
  
 v-model="conpassword"
  
 @blur.native.capture="checkConpassword"
  
 :state="conpasswordState">
  
 </mt-field>
  
 <mt-button type="primary" size="large" @click="register">快速注册</mt-button>
  
 </div>
  
 <!-- 表单区域结束 -->
  
 </div>
  
</template>
  
<script>
  
export default {
  
 data() {
  
 return {
  
 //用户名的初始值
  
 username: "",
  
 //密码初始值
  
 password: "",
  
 //确认密码初始值
  
 conpassword: "",
  
 //用户名状态的初始值
  
 usernameState: "",
  
 //密码状态的初始值
  
 passwordState: "",
  
 //确认密码状态的初始值
  
 conpasswordState: ""
  
 };
  
 },
  
 methods: {
  
 //检测用户名
  
 checkUsername() {
  
 let usernameExp = /^[a-zA-Z0-9\_]{6,12}$/;
  
 if (usernameExp.test(this.username)) {
  
 this.usernameState = "success";
  
 return true;
  
 } else {
  
 this.usernameState = "error";
  
 return false;
  
 }
  
 },
  
 //检测密码
  
 checkPassword() {
  
 let passwordExp = /^[a-zA-Z0-9\_]{8,20}$/;
  
 if (passwordExp.test(this.password)) {
  
 this.passwordState = "success";
  
 return true;
  
 } else {
  
 this.passwordState = "error";
  
 return false;
  
 }
  
 },
  
 //检测确认密码
  
 checkConpassword() {
  
 if (this.password != this.conpassword) {
  
 this.conpasswordState = "error";
  
 return false;
  
 } else {
  
 this.conpasswordState = "success";
  
 return true;
  
 }
  
 //可简化成
  
 //this.conpasswordState = this.password != this.conpassword ? 'error' : 'success';
  
 //return this.password != this.conpassword;
  
 },
  
 //用户单击注册按钮时,进行验证
  
 register() {
  
 if (this.checkUsername() && this.checkPassword() && this.checkConpassword()) {
  
 console.log("要发送AJAX请求到服务器了");
  
 } else {
  
 console.log("你肯定有不对的地方,现在我什么也不干");
  
 }
  
 }
  
 }
  
};
  
</script>

课外作业：

1.完成用户登录界面(login.vue)及业务功能的实现

2.尝试对检测用户名、检测密码、检测确认密码的函数进行简化的操作

# 2.表单组件

## • Checklist组件

Checklist组件用于实现复选框列表，其语法结构是：

<mt-checklist title="标题" :options="选项列表" v-model="变量名称">
  
   
</mt-checklist>

:options和v-model属性必须存在

:options属性的数据类型为array，数组中既可以存储字符串，也可以存储对象，如：

<!-- 字符串类型 -->
  
<mt-checklist
  
 title="爱好"
  
 :options="['学习','跑步','打球','打游戏','刷抖音']"
  
 v-model="loves">
  
</mt-checklist>
  
   
<!--对象类型-->
  
<mt-checklist
  
 title="技术列表"
  
 :options="[
  
 {
  
 label:'HTML5',
  
 value:1
  
 },
  
 {
  
 label:'CSS3',
  
 value:2
  
 },
  
 {
  
 label:'JavaScript',
  
 value:3
  
 },
  
 {
  
 label:'jQuery',
  
 value:4
  
 }]"
  
 v-model="lists">
  
</mt-checklist>

如果:options属性成员为字符串的话，则提交值与显示文本相同

如果:options属性成员为对象的话，则提交值为对象的value属性来决定

v-model绑定变量的数据类型必须为array，如果要存在默认选定选项的话，可以直接将文本或value添加到绑定的变量中即可，如：

<script>
  
 export default {
  
 data(){
  
 return {
  
 loves:['跑步','打球'],
  
 lists:[2,3]
  
 }
  
 }
  
 }
  
</script>

## • radio组件

radio组件用于实现单选框列表，其语法结构是：

<mt-radio title="标题" :options="选项列表" v-model="变量名称">
  
</mt-radio>

:options和v-model属性必须存在

:options属性的数据类型为array，数组中既可以存储字符串，也可以存储对象，如：

<mt-radio
  
 title="性别"
  
 :options="['男','女','保密']"
  
 v-model="sex">
  
</mt-radio>
  
   
<mt-radio
  
 title="学历"
  
 :options="[
  
 {
  
 label:'高职',
  
 value:'1'
  
 },
  
 {
  
 label:'专科',
  
 value:'2'
  
 },
  
 {
  
 label:'本科',
  
 value:'3'
  
 },
  
 {
  
 label:'研究生',
  
 value:'4'
  
 }]"
  
 v-model="education">
  
</mt-radio>

v-model绑定变量的数据类型必须为string，如果要存在默认选定选项的话，可以直接将文本或value添加到绑定的变量中即可，如：

<script>
  
export default {
  
 data(){
  
 return {
  
 sex:'保密',
  
 education:'1'
  
 }
  
 }
  
}
  
</script>

## • switch组件

switch组件用于实现开关，其语法结构是：

<mt-switch v-model="变量名称">
  
 ...
  
</mt-switch>

v-model属性绑定的变量数据类型必须为boolean

示例代码如下：

<template>
  
 <div>
  
 <mt-switch v-model="flyMode">飞行模式</mt-switch>
  
 </div>
  
</template>
  
   
<script>
  
export default {
  
 data(){
  
 return {
  
 flyMode:false
  
 }
  
 }
  
}
  
</script>

# 3.CSS组件

## • cell 组件

cell组件用于实现单元格，其语法结构是：

<mt-cell title="标题" label="备注信息" value="内容" icon="图标" isLink to="路由地栏">
  
 ...
  
</mt-cell>

icon属性值有：more(更多)、back(返回)

isLink属性用于控制单元格是否链接单元格，boolean类型

可以在<mt-cell>组件中嵌套图像，并且图像添加slot="icon"属性的话，该图像将作为单元格图标显示

示例代码1如下：

<template>
  
 <div>
  
 <mt-cell title="飞行模式" Label="在此种模式下禁止通话">
  
 <img src="../../assets/images/fly.png" alt="" slot="icon">
  
 <mt-switch v-model="flyMode"></mt-switch>
  
 </mt-cell>
  
 </div>
  
</template>
  
   
<script>
  
export default {
  
 data(){
  
 return {
  
 flyMode:true
  
 }
  
 }
  
}
  
</script>

示例代码2如下：

<mt-cell title="无限局域网" isLink>
  
 <img src="../../assets/images/network.png" alt="" slot="icon">
  
 未连接
  
</mt-cell>
  
   
<mt-cell title="蓝牙" isLink to="/">
  
 <img src="../../assets/images/blue.png" alt="" slot="icon">
  
</mt-cell>

## • Navbar组件

Navbar组件用于实现顶部选项卡，其语法结构是：

<mt-navbar v-model="变量名称" fixed>
  
 <mt-tab-item id="当前选项卡的ID">
  
 ...
  
 </mt-tab-item>
  
 ...
  
</mt-navbar>

fixed属性用于控制是否将顶部选项卡固定，boolean类型

<mt-tab-item>的id属性必须保证在当前顶部选项卡中唯一即可

在<mt-tab-item>中嵌套图像并且设置slot="icon" 属性的话，可以将图像作为选项卡的图标显示

作业：

A.复习MySQL中 INSERT、UPDATE、DELETE及SELECT

B.复习axios、cors、mysql模块