Mint UI Unit02

1.表单组件

• `Field`组件

`Field`组件用于实现表单控件，其语法结构是：

```
<mt-field type="输入框的类型" label="标签"
          placeholder="占位内容"
          state="检测状态"
          v-model="变量名称"
          :attr="原生属性"
          readonly
          disabled
          disableClear>
</mt-field>
```

> 输入框的类型有：
>
> - `text`，单行文本框
> - `password`，密码框
> - `textarea`，多行文本框
> - `number`，数字
> - `tel`，电话
> - `url`，URL地址
> - `date`，日期
> - `datetime-local`，日期和时间
>
> `state`属性表示输入框的状态，其值有：
>
> - `error`，表示错误
> - `success`，表示成功
> - `warning`，表示警告
>
> `:attr`属性用于控制表单控件的原生属性，对象类型，如：
>
> ```
> <mt-field type="password"
>           label="密码"
>           :attr="{autocomplete:'off',maxlength:'10'}">
> </mt-field>
> ```
>
> `disabled`属性用于控制表单控件是否为禁用状态，布尔类型，默认为`false`
>
> `readonly`属性用于控制表单控件是否为只读状态，布尔类型，默认为`false`
>
> `disableClear`属性用于控制表单控件是否禁用`clear`按钮，布尔类型，默认为`false`
>
> 如果为表单控件添加失去焦点、获取焦点的事件，其语法结构是：

```
<mt-field @blur.native.capture="事件名称"></mt-field>

<mt-field @focus.native.capture="事件名称"></mt-field>
```

Vue.js的事件修饰符https://cn.vuejs.org/v2/guide/events.html

• `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" align="对齐方式(left|right)"
              :option="选项列表"
              v-model="变量名称">
</mt-checklist>
```

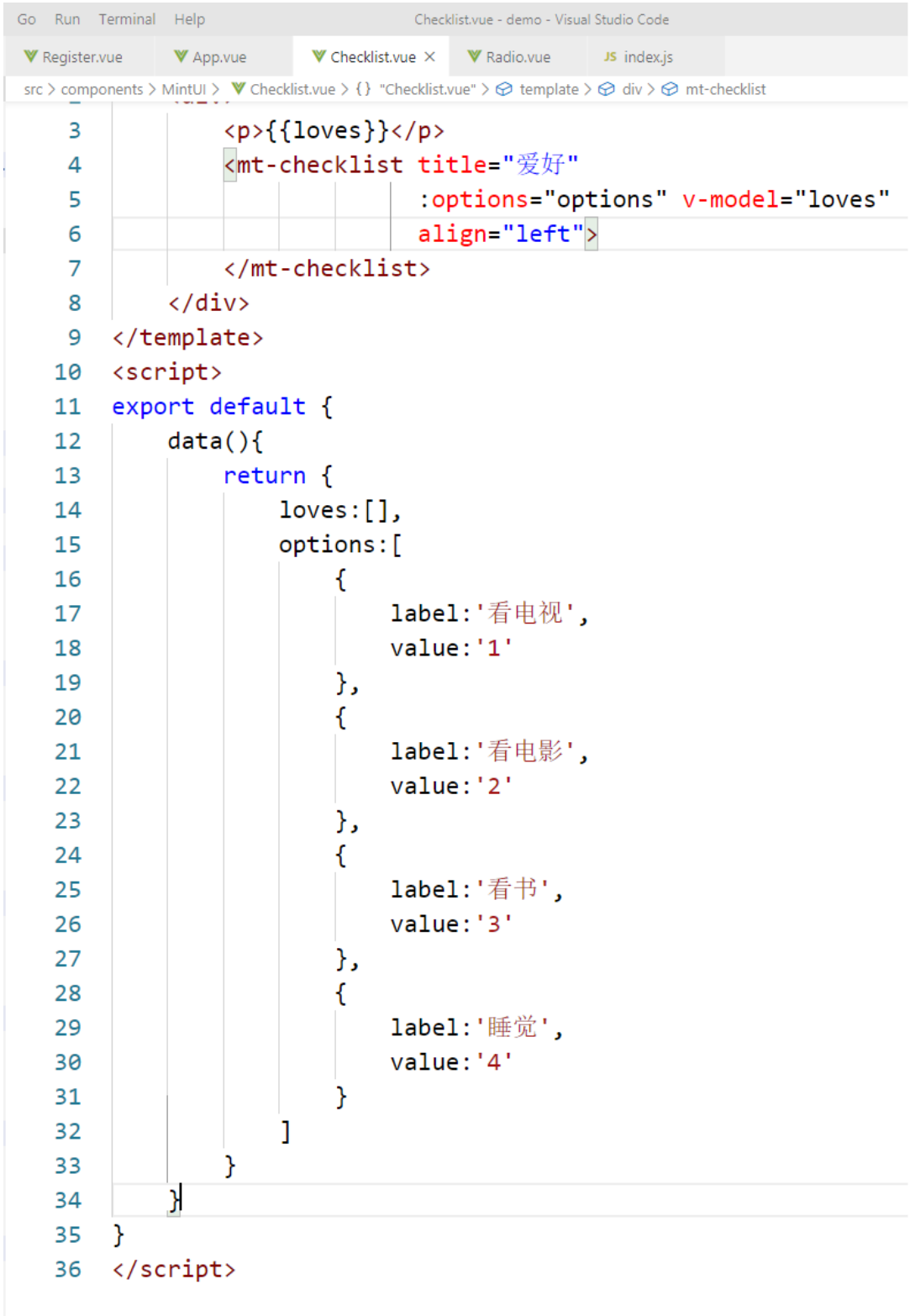
`:options`属性用于控制复选框的列表项，数据类型为数组，数组中既可以是字符串，也可以是对象，如：

```
//数组形态
['文本','文本',...]

//对象形态
[
 {
    label:'标签文本',
    value:'值',
    disabled:是否禁用(true|false)
 }
]
```

`v-model`指令绑定的变量的数据类型必须为数组类型（因为其有可能要选择多个，所以为数组类型）

必须为`mt-checklist`组件赋予`v-model`、`:options`属性



• Radio组件

Radio 组件用于实现单选框列表，其语法结构是：

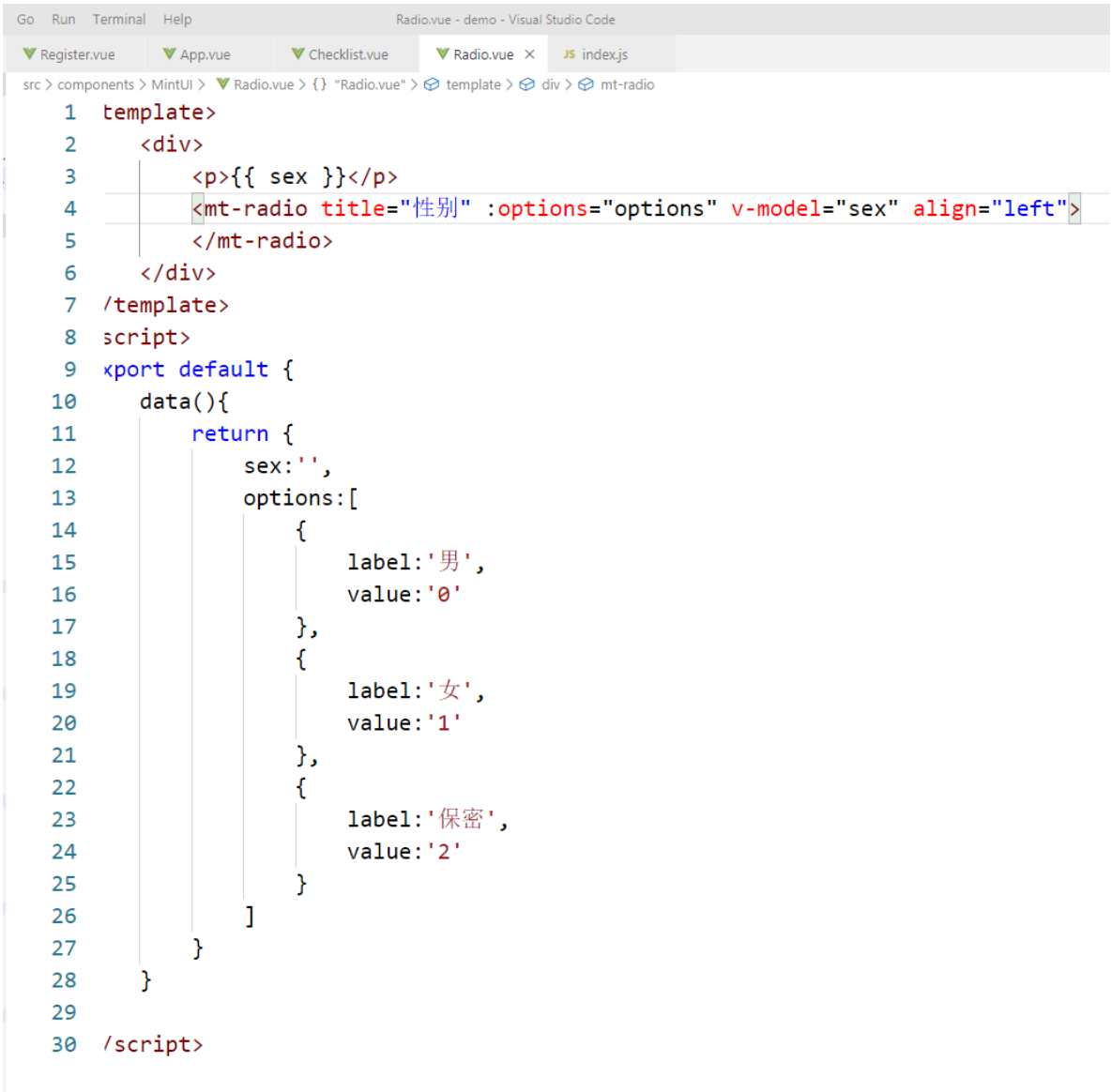
```
<mt-radio title="标题" align="对齐方式(left|right)"
          :options="选项列表" v-model="变量名称">
</mt-radio>
```

> v-model 指令绑定的变量的数据类型为字符类型

`:options`属性用于控制单选框的列表项，数据类型为数组，数组中既可以是字符串，也可以是对象，如：

```
//数组形态
['文本','文本',...]

//对象形态
[
 {
    label:'标签文本',
    value:'值',
    disabled:是否禁用(true|false)
 }
]
```



• `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" disabled>
    ...
</mt-switch>
```

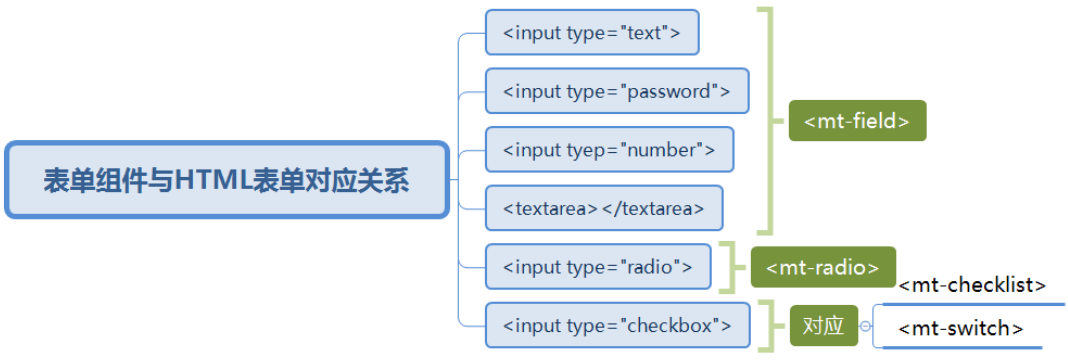
> v-model指令绑定的变量的数据类型为布尔类型

Go Run Terminal Help Switch.vue - demo - Visual Studio Code

Register.vue App.vue Checklist.vue Radio.vue Switch.vue

src > components > MintUI > Switch.vue > { } "Switch.vue" > script

```
1  <template>
2      <div>
3          <p>{{allow}}</p>
4          <mt-switch v-model="allow">
5              是否接收消息提示
6          </mt-switch>
7      </div>
8  </template>
9  <script>
10 export default {
11     data(){
12         return {
13             allow:false
14         }
15     }
16 }
17 </script>
```



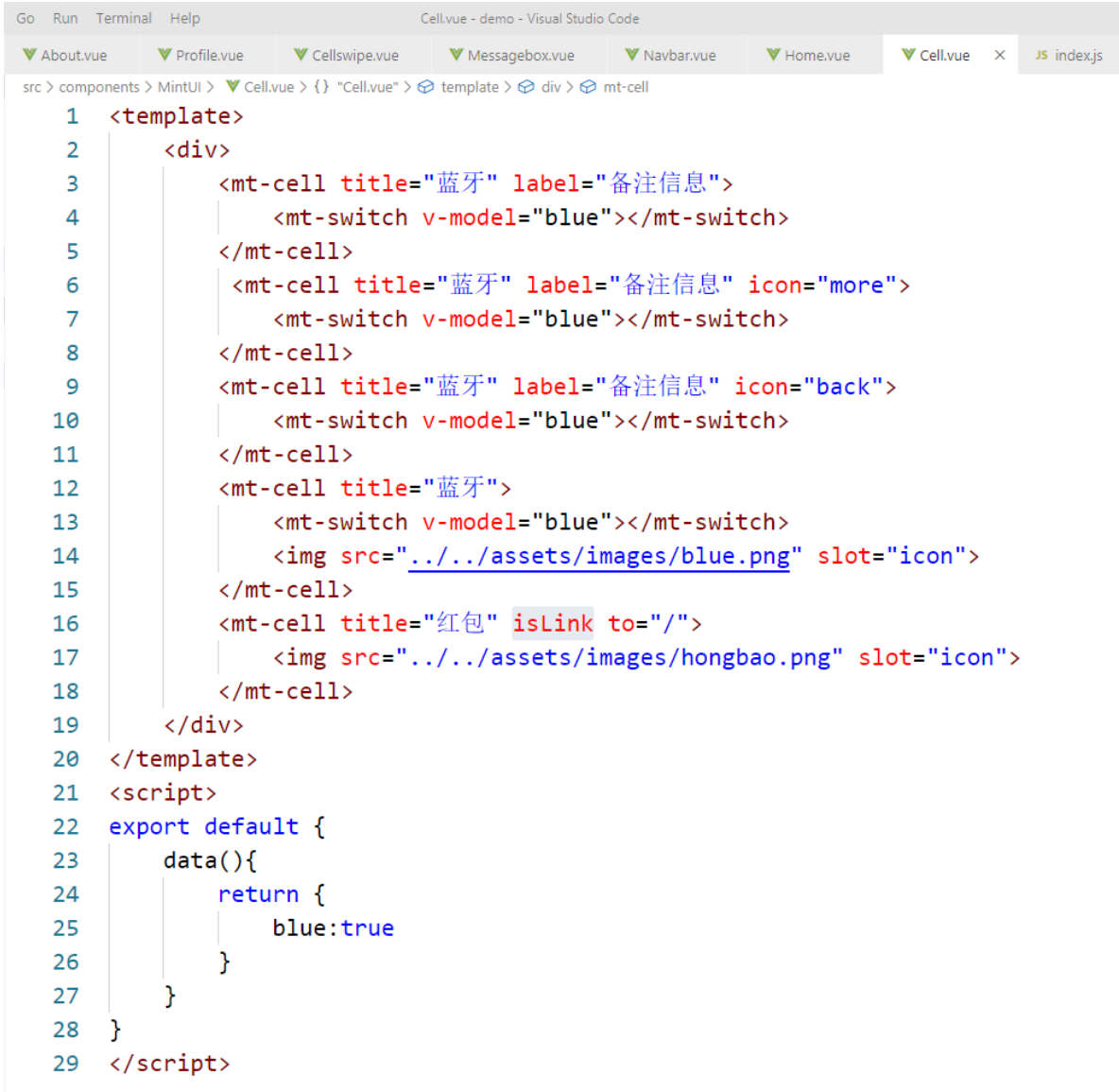
2.CSS组件

• 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="URL地址">
    ...
</mt-cell>
```

> icon属性用于控制单元格图标，其值可为more、back或字体图标类名称
>
> 单元格内的图像可以通过slot="icon"来修改单元格图标
>
> isLink属性用于控制单元格是否为可点击的链接，布尔类型，默认为true
>
> to属性用于控制链接单元格的目标路由，字符类型



• `cell Swipe`组件

`Cell Swipe`组件用于实现可滑动的单元格，其语法结构是：

```
<mt-cell-swipe title="标题" label="备注信息" value="内容" icon="图标"
         isLink to="URL地址"
         :left="左侧按钮组"
         :right="右侧按钮组">
    ...
</mt-cell-swipe>
```

> `:left`和`right`属性分别用于控制滑动单元格的按钮组，数据类型为数组，结构如下：
>
> ```
> :right = "[
>     {
>         content:'按钮显示文本',
>         style:{属性名称:值,属性:值,...},
>         handle:()=>{ ... }
>     }
>     ,...
> ]"
> ```
>
> 针对`[Intervention] Ignored attempt to cancel a touchmove event with cancelable=false, for example because scrolling is in progress and cannot be interrupted.`错误提示的解决方案：
>
> 在对应的组件文件内添加以下样式：
>
> ```
> <style>
>     html,body{
>         touch-action:none;
>     }
> </style>
> ```

• Navbar组件

Navbar组件用于实现顶部选项卡，其语法结构是：

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

Go Run Terminal Help Navbar.vue - demo - Visual Studio Code

About.vue | Profile.vue | Cellswipe.vue | Messagebox.vue | Navbar.vue | Home.vue | Cell.vue

src > components > MintUI > Navbar.vue > {} "Navbar.vue" > template > div > mt-navbar

```vue
<template>
    <div>
        <p>{{ selected }}</p>
        <mt-navbar v-model="selected">
            <mt-tab-item id="a">推荐</mt-tab-item>
            <mt-tab-item id="b">生活</mt-tab-item>
            <mt-tab-item id="c">娱乐</mt-tab-item>
            <mt-tab-item id="d">汽车</mt-tab-item>
        </mt-navbar>
    </div>
</template>
<script>
export default {
    data(){
        return {
            selected:'a'
        }
    }
}
</script>
```

3.JS组件

• MessageBox组件

MessageBox组件用于实现弹出式提示框，其语法结构是：

```
//简捷方式
this.$messagebox("标题","提示信息")
```

> MessageBox还提供了alert()、confirm()及prompt()三个方法，它们都返回Promise，语法结构分别是：
>
> ```
> //alert()方法
> this.$messagebox.alert("提示信息","标题",选项)
>
> //comfirm()方法
> this.$messagebox.confirm("提示信息","标题",选项)
>
> //prompt()方法
> this.$messagebox.prompt("提示信息","标题",选项)
> ```
>
> 选 项参数用于控制提示框的相关配置，对象类型，包括：

- `showConfirmButton`，控制是否显示确认按钮，默认`true`
- `confirmButtonText`，控制确认按钮的文本，默认为`确认`
- `confirmButtonClass`，控制确认按钮的`CSS`类名称
- `confirmButtonHighLight`，控制确认按钮文本是否高亮显示，默认为`false`，需重新定义`CSS`类，如：

```
<style>
    .mint-msgbox-confirm:active {
        color: #0563a9;
        font-weight: bold;
    }

    .mint-msgbox-btn:active{
        background-color:#fff;
    }
</style>
```

- `showCancelButton`，控制是否显示取消按钮，默认`true`
- `cancelButtonText`，控制取消按钮的文本，默认为`取消`
- `cancelButtonClass`，控制取消按钮的`CSS`类名称
- `cancelButtonHighLight`，控制取消按钮文本是否高亮显示，默认为`false`,需重新定义`CSS`类，如：

```
<style>
    .mint-msgbox-cancel:active {
        color: #0563a9;
        font-weight: bold;
    }

    .mint-msgbox-btn:active{
        background-color:#fff;
    }
</style>
```

- `closeOnClickModal`，控制是否在单击遮罩层时关闭提示框，默认为`true`
- `closeOnPressEscape`，控制是否通过键盘`ESC`按钮关闭提示框，默认为`true`
- `lockScroll`,属性是否锁定滚动条，默认为`false`
- `inputType`属性用于控制输入框的类型，默认为`text`
- `inputPlaceholder`属性用于控制输入框的占位符

```vue
<template>
    <div>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <mt-button type="primary" @click="simple">简捷提示框</mt-button>
        <mt-button type="primary" @click="normal">标准提示框</mt-button>
        <mt-button type="primary" @click="confirm">确认提示框</mt-button>
        <mt-button type="primary" @click="prompt">输入提示框</mt-button>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
        <p>1111</p>
    </div>
</template>
<script>
export default {
    methods:{
        prompt(){
            this.$messagebox.prompt('请输入QQ群名称:','新建QQ群',{
                inputType:'text',
                inputPlaceholder:'QQ群名称'
            }).then(action=>{
                console.log(action);
            }).catch(err=>{

            });
        },
        simple(){
            this.$messagebox('标题','提示内容');
        },
        normal(){
            this.$messagebox.alert('提示内容','标题');
        },
        confirm(){
            this.$messagebox.confirm('提示内容','标题',{
                confirmButtonText:'OK',
                confirmButtonHighlight:true
```

```
                confirmButtonHighLight:true,
                cancelButtonText:'Cancel',
                cancelButtonHighLight:true,
                //是否在单击遮罩层时关闭对话框
                closeOnClickModal:false,
                //是否允许通过键盘ESC按钮关闭对话框
                closeOnPressEscape:false,
                //是否锁定滚动条
                lockScroll:true
            }).then(action=>{
                console.log(action);
            }).catch(err=>{

            });
        }
    }
}
</script>
<style>
.mint-msgbox-confirm{
    color: #F00 !important;
}
.mint-msgbox-confirm:active {

    font-weight: bold;
}

.mint-msgbox-cancel:active {
    color: #0563a9;
    font-weight: bold;
}
</style>
```

4.作业

- 修改 `src/views/Register.vue` 组件文件，如果用户名为 `admin` 且密码为 `admin` 时弹出提示框，内容为 `对不起，该用户已存在`，否则询问 `用户注册成功，是否立即登录？`，如果选择 `是` 时，跳转到登录的路由，如果选择 `否`，则跳转到首页跳由
- 复习MySQL的 `CURD` 操作
- 复习 `MySQL连接池` 的相关知识