

# Vue.js 组件进阶

Vue.js最精髓的部分是自定义组件，今天就在之前的基础之上进行进一步的探究。

我们今天试着自定义一个名叫 `xun-panel` 的自定义Vue组件。

Vue实例代码：

```
<div id="app">
  <xun-panel></xun-panel>
</div>
```

组件定义：

```
<script>
  Vue.component("xun-panel",{
    template:"<div class='xun-panel'><h3>{{title}}</h3><div v-html='content'></div>
</div>",
    data:function(){
      return {
        title:"标题",
        content:"<p>确定要关闭</p><button>确定</button>"
      }
    }
  })
</script>
```

组件的模版代码可以直接写在“template”属性中，也可以用下面的方法。

## 组件模版的另一种写法和引用方法

可以将组件的HTML模版写在一个 `<template>` 标签当中，并且赋予 `id` 属性。

```
<template id="xun-panel">
  <div class='xun-panel'><h3>{{title}}</h3><div v-html='content'></div></div>
</template>
```

引用的时候可以直接写成 `template:"#xun-panel"`，效果与第一种方式一样，但是对于书写HTML模版代码，显然这种方法更有利些。

浏览器默认 `<template>` 标签的display属性为none，也就是不在页面当中显示。

```

<script>
  Vue.component("xun-panel",{
    template:"#xun-panel",
    data:function(){
      return {
        title:"标题",
        content:"<p>确定要关闭</p><button>确定</button>"
      }
    }
  })
</script>

```

## 为组件添加CSS样式

```

<style scoped>
.xun-panel *{padding: 0px;margin: 0px}
.xun-panel{
  width:200px;
  max-height:200px;
  background: #ccc;
  border-radius: 5px;
}
.xun-panel h3{
  padding: 5px;
  border-radius: 5px 5px 0px 0px;
  background: #aaa
}
.xun-panel>div{
  padding: 5px
}
</style>

```

## css的@import写法

可以将上面的css样式单独存入一个css文件，然后使用 @import 语法引入，写成下面这样：

```

<style scoped>
  @import "xun-panel.css";
</style>

```

运行结果：

**标题**

确定要关闭

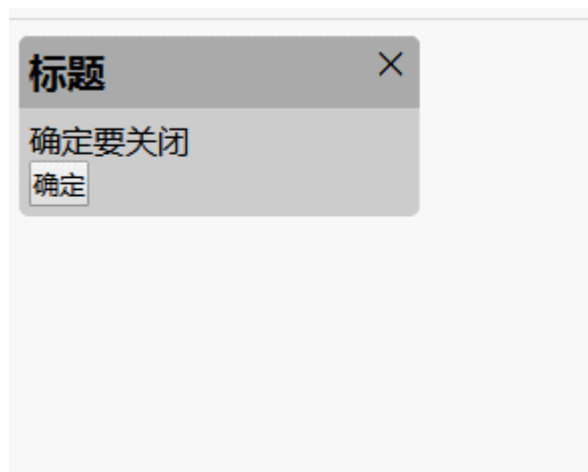
确定

实现关闭：

```
<template id="xun-panel">
<div class='xun-panel' v-if="show"><h3><i class="el-icon-close" style="float:right;"
@click="close()"></i>{{title}}</h3><div v-html='content'></div></div>
</template>
```

首先是在h3标签上添加了 `<i class="el-icon-close" style="float:right;" @click="close()"></i>`,这段是利用Vue的第三方UI组件库elementUI中提供的图标显示“x”号。并且右浮动。对应单击click事件为 `close()` 方法。

其次子啊外层div上添加 `v-if="show"` 来判断组件中data下的show是否为true来确定整个组件是否显示。



```
Vue.component("xun-panel",{
  template:"#xun-panel",
  data:function(){
    return {
      title:"标题",
      content:"<p>确定要关闭</p><button>确定</button>",
      show:true//用于div判断是否显示
    }
  },
  methods:{
    close:function(){//关闭功能
      this.show = false;//变量show取false , vue根据v-if="show"自动更新div的显示状态
    }
  }
})
```

## 完整代码

### index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <script src="https://cdn.bootcss.com/vue/2.6.10/vue.min.js"></script>
```

```

<!-- 引入样式 -->
<link rel="stylesheet" href="https://unpkg.com/element-ui/lib/theme-chalk/index.css">
<!-- 引入组件库 -->
<script src="https://unpkg.com/element-ui/lib/index.js"></script>
</head>
<body>
  <div id="app">
    <xun-panel></xun-panel>
    <xun-panel></xun-panel>
    <xun-panel></xun-panel>
    <xun-panel></xun-panel>
  </div>
  <template id="xun-panel">
    <div class='xun-panel' v-if="show"><h3><i class="el-icon-close"
style="float:right;" @click="close()"></i>{{title}}</h3><div v-html='content'></div></div>
  </template>
  <style scoped>
    @import "xun-panel.css";
  </style>
  <script>
    Vue.component("xun-panel",{
      template:"#xun-panel",
      data:function(){
        return {
          title:"标题",
          content:"<p>确定要关闭</p><button>确定</button>",
          show:true
        }
      },
      methods:{
        close:function(){
          this.show = false;
        }
      }
    })
    var v=new Vue({
      el:"#app",
      data:{

      },
      methods:{

      }
    });
  </script>
</body>
</html>

```

## xun-panel.css

```

.xun-panel *{padding: 0px;margin: 0px}
.xun-panel{
  width:200px;
  max-height:200px;

```

```
    background: #ccc;
    border-radius: 5px;
    margin: 10px 0px;
    border: 1px solid #ccc;
}
.xun-panel:hover{
    background: #fff;
    border: 1px solid #444;
}
.xun-panel:hover h3{
    background: #444;
    color: #fff
}
.xun-panel h3{
    padding: 5px;
    border-radius: 5px 5px 0px 0px;
    background: #aaa
}

.xun-panel>div{
    padding: 5px
}
```