

第17章：ElementUI

第1节：概述

Element，一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库。
是基于Vue的一个UI框架，该框架基于Vue开发了很多相关组件，方便我们快速开发页面。
饿了么前端团队 基于Vue进行开发并开源，提供了封装好的组件。
官网：<https://element.eleme.cn/#/zh-CN>

第2节：安装

方式1：CDN

目前可以通过 unpkg.com/element-ui 获取到最新版本的资源，在页面上引入 js 和 css 文件即可开始使用。

```
<!-- 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>
```

建议使用 CDN 引入 Element 的用户在链接地址上锁定版本，以免将来 Element 升级时受到非兼容性更新的影响。参考：unpkg.com

```
<!-- 引入 elementUI 样式 -->
<link rel="stylesheet" href="https://unpkg.com/element-ui@2.13.0/lib/theme-chalk/index.css">
<!-- 引入elementUI 组件库 -->
<script src="https://unpkg.com/element-ui@2.13.0/lib/index.js"></script>
```

方式2：npm

① 通过脚手架创建项目

```
vue init webpack 项目名
```

② 下载ElementUI的依赖

```
npm i element-ui -S
```

③ 在main.js中写入以下内容

```
// 引入ElementUI 依赖
import ElementUI from 'element-ui';
// 引入ElementUI依赖样式
import 'element-ui/lib/theme-chalk/index.css';
// 在vue中使用ElementUI
Vue.use(ElementUI);
```

方式3：本地资源 下载地址：<https://unpkg.com/browse/element-ui@2.13.0/>

```
<!-- 引入 elementUI 样式 -->
```

```
<link rel="stylesheet" href="./element-ui-2.13.0/lib/theme-chalk/index.css"> <!--
引入 vue 的js文件:      elementUI基于Vue开发, 必须在elementUI的js文件之前引入 --> <script
src="./vue-v2.6.10.js"></script>
<!-- 引入elementUI 组件库 -->
<script src="./element-ui-2.13.0/lib/index.js"></script>
```

第3节：使用

3.1 Layout 布局

通过基础的24分栏，迅速过渡地创造布局。

3.1.1 单一分栏基础的布局

通过row和col组件，并通过col组件的span属性我们就可以自由地组合布局。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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>
<style>
  .el-row {
    margin-bottom: 20px;

  }
  .el-col {
    border-radius: 4px;
  }
  .bg-purple-dark {
    background: #99a9bf;
  }
  .bg-purple {
    background: #d3dce6;
  }
  .bg-purple-light {
    background: #e5e9f2;
  }
  .grid-content {
```

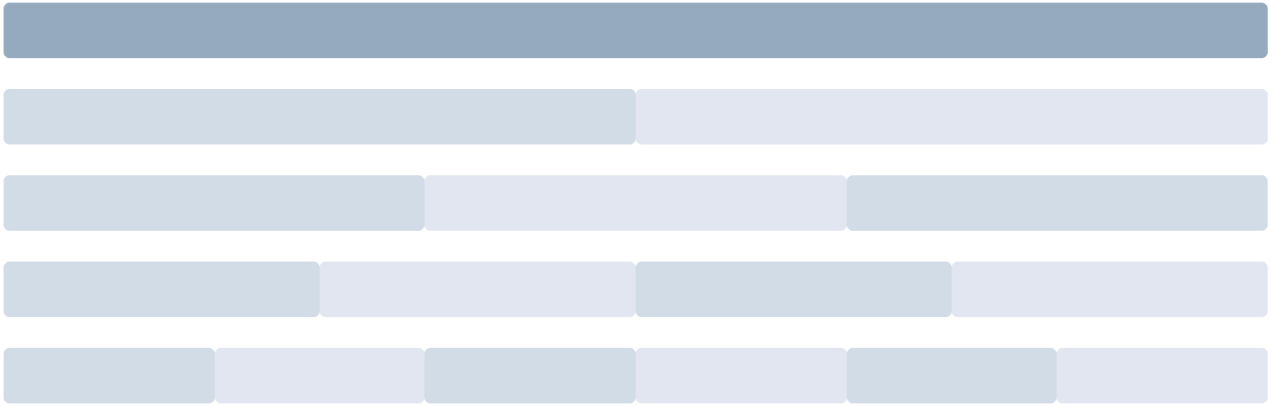
```

    border-radius: 4px;
    min-height: 36px;
  }
  .row-bg {
    padding: 10px 0;
    background-color: #f9fafc;
  }
</style>
</head>
<body>
  <div id="app">
    <el-row>
      <el-col :span="24"><div class="grid-content bg-purple-dark"></div></el-
col>
    </el-row>
    <el-row>
      <el-col :span="12"><div class="grid-content bg-purple"></div></el-col>
      <el-col :span="12"><div class="grid-content bg-purple-light"></div></el-
col>
    </el-row>
    <el-row>
      <el-col :span="8"><div class="grid-content bg-purple"></div></el-col>
      <el-col :span="8"><div class="grid-content bg-purple-light"></div></el-
col>
      <el-col :span="8"><div class="grid-content bg-purple"></div></el-col>
    </el-row>
    <el-row>
      <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
      <el-col :span="6"><div class="grid-content bg-purple-light"></div></el-
col>
      <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
      <el-col :span="6"><div class="grid-content bg-purple-light"></div></el-
col>
    </el-row>
    <el-row>
      <el-col :span="4"><div class="grid-content bg-purple"></div></el-col>
      <el-col :span="4"><div class="grid-content bg-purple-light"></div></el-
col>
      <el-col :span="4"><div class="grid-content bg-purple"></div></el-col>
      <el-col :span="4"><div class="grid-content bg-purple-light"></div></el-
col>
      <el-col :span="4"><div class="grid-content bg-purple"></div></el-col>
      <el-col :span="4"><div class="grid-content bg-purple-light"></div></el-
col>
    </el-row>
  </div>
<script>
  var app = new Vue({
    el: "#app"
  })

```

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

效果：



3.1.2 分栏间隔

分栏之间存在间隔。Row组件提供gutter属性来指定每个栏之间的间隔，交替间隔为0。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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>

  <style>
    .el-row {
      margin-bottom: 20px;
    }
    .el-col {
      border-radius: 4px;
    }
    .bg-purple-dark {
      background: #99a9bf;
    }
  </style>
```

```

    .bg-purple {
      background: #d3dce6;
    }
    .bg-purple-light {
      background: #e5e9f2;
    }
    .grid-content {
      border-radius: 4px;
      min-height: 36px;
    }
    .row-bg {
      padding: 10px 0;
      background-color: #f9fafc;
    }
  </style>
</head>
<body>
  <div id="app">
    <el-row :gutter="20">
      <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
      <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
      <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
      <el-col :span="6"><div class="grid-content bg-purple"></div></el-col>
    </el-row>
  </div>
  <script>
    var v = new Vue({
      el: "#app"
    })
  </script>
</body>
</html>

```

效果：



3.1.3 响应式布局

参照了 Bootstrap 的 响应式设计，预设了五个响应尺寸：xs、sm、md、lg 和 xl。

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```
<title>Document</title>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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>
<style>
  .el-col {
    border-radius: 4px;
  }
  .bg-purple-dark {
    background: #99a9bf;
  }
  .bg-purple {
    background: #d3dce6;
  }
  .bg-purple-light {
    background: #e5e9f2;
  }
  .grid-content {
    border-radius: 4px;
    min-height: 36px;
  }
</style>
</head>
<body>
  <div id="app">
    <el-row :gutter="10">
      <el-col :xs="8" :sm="6" :md="4" :lg="3" :xl="1">
        <div class="grid-content bg-purple"></div>
      </el-col>
      <el-col :xs="4" :sm="6" :md="8" :lg="9" :xl="11">
        <div class="grid-content bg-purple-light"></div>
      </el-col>
      <el-col :xs="4" :sm="6" :md="8" :lg="9" :xl="11">
        <div class="grid-content bg-purple"></div>
      </el-col>
      <el-col :xs="8" :sm="6" :md="4" :lg="3" :xl="1">
        <div class="grid-content bg-purple-light"></div>
      </el-col>
    </el-row>
  </div>
  <script>
    var app = new Vue({
      el: "#app"
    })
  </script>
</body>
```

</html>

属性如下：

Row Attributes

| 参数 | 说明 | 类型 | 可选值 | 默认值 |
|---------|-----------------------|--------|---|-------|
| gutter | 栅格间隔 | number | — | 0 |
| type | 布局模式，可选 flex，现代浏览器下有效 | string | — | — |
| justify | flex 布局下的水平排列方式 | string | start/end/center/space-around/space-between | start |
| align | flex 布局下的垂直排列方式 | string | top/middle/bottom | top |
| tag | 自定义元素标签 | string | * | div |

Col Attributes

| 参数 | 说明 | 类型 | 可选值 | 默认值 |
|--------|------------------------|--|-----|-----|
| span | 栅格占据的列数 | number | — | 24 |
| offset | 栅格左侧的间隔格数 | number | — | 0 |
| push | 栅格向右移动格数 | number | — | 0 |
| pull | 栅格向左移动格数 | number | — | 0 |
| xs | <768px 响应式栅格数或者栅格属性对象 | number/object (例如： {span: 4, offset: 4}) | — | — |
| sm | ≥768px 响应式栅格数或者栅格属性对象 | number/object (例如： {span: 4, offset: 4}) | — | — |
| md | ≥992px 响应式栅格数或者栅格属性对象 | number/object (例如： {span: 4, offset: 4}) | — | — |
| lg | ≥1200px 响应式栅格数或者栅格属性对象 | number/object (例如： {span: 4, offset: 4}) | — | — |
| xl | ≥1920px 响应式栅格数或者栅格属性对象 | number/object (例如： {span: 4, offset: 4}) | — | — |
| tag | 自定义元素标签 | string | * | div |

3.2 Container 布局容器

用于布局的容器组件，方便快速搭建页面的基本结构。

常见标签：

`<el-container>`：外层容器。当子元素中包含 `<el-header>` 或 `<el-footer>` 时，全部子元素会垂直上下排列，否则会水平左右排列。

`<el-header>`：顶栏容器。

`<el-aside>`：侧边栏容器。

`<el-main>`：主要区域容器。

`<el-footer>`：底栏容器。

注意：

以上组件采用了 flex 布局，使用前请确定目标浏览器是否兼容。此外，`<el-container>` 的子元素只能是后四者，后四者的父元素也只能是 `<el-container>`。

3.2.1 常见页面布局

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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>
  <style>
    .el-header, .el-footer {
      background-color: #B3C0D1;
      color: #333;
      text-align: center;
      line-height: 60px;
    }

    .el-aside {
      background-color: #D3DCE6;
      color: #333;
      text-align: center;
      line-height: 200px;
    }
  </style>
</head>
<body>
  <div>
    <el-container>
      <el-header>
        顶部
      </el-header>
      <el-aside>
        侧边
      </el-aside>
      <el-main>
        主体
      </el-main>
      <el-footer>
        底部
      </el-footer>
    </el-container>
  </div>
</body>
</html>
```



```

.el-main {
  background-color: #E9EEF3;
  color: #333;
  text-align: center;
  line-height: 160px;
}

body > .el-container {
  margin-bottom: 40px;
}

.el-container:nth-child(5) .el-aside,
.el-container:nth-child(6) .el-aside {
  line-height: 260px;
}

.el-container:nth-child(7) .el-aside {
  line-height: 320px;
}
</style>
</head>
<body>
  <div id="app">
    <el-container>
      <el-header>Header</el-header>
      <el-container>
        <el-aside width="200px">Aside</el-aside>
        <el-main>Main</el-main>
      </el-container>
    </el-container>
  </div>
  <script>
    var app = new Vue({
      el: "#app"
    })
  </script>
</body>
</html>

```



Container Attributes

| 参数 | 说明 | 类型 | 可选值 | 默认值 |
|-----------|----------|--------|-----------------------|--|
| direction | 子元素的排列方向 | string | horizontal / vertical | 子元素中有 <code>el-header</code> 或 <code>el-footer</code> 时为 vertical，否则为 horizontal |

Header Attributes

| 参数 | 说明 | 类型 | 可选值 | 默认值 |
|--------|------|--------|-----|------|
| height | 顶栏高度 | string | — | 60px |

Aside Attributes

| 参数 | 说明 | 类型 | 可选值 | 默认值 |
|-------|-------|--------|-----|-------|
| width | 侧边栏宽度 | string | — | 300px |

Footer Attributes

| 参数 | 说明 | 类型 | 可选值 | 默认值 |
|--------|------|--------|-----|------|
| height | 底栏高度 | string | — | 60px |

3.2.2 实例

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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>
  <style>
    .el-header {
      background-color: #B3C0D1;
      color: #333;
      line-height: 60px;
```

```

    }

    .el-aside {
        color: #333;
    }
</style>
</head>

<body>
    <div id="app">
        <el-container style="height: 500px; border: 1px solid #eee">
            <el-aside width="200px" style="background-color: rgb(238, 241, 246)">
                <el-menu :default-openeds=['1', '3']>
                    <el-submenu index="1">
                        <template slot="title"><i class="el-icon-message"></i>导航一</template>

                        <el-menu-item-group>
                            <template slot="title">分组一</template>
                            <el-menu-item index="1-1">选项1</el-menu-item>
                            <el-menu-item index="1-2">选项2</el-menu-item>
                        </el-menu-item-group>
                        <el-menu-item-group title="分组2">
                            <el-menu-item index="1-3">选项3</el-menu-item>
                        </el-menu-item-group>
                        <el-submenu index="1-4">
                            <template slot="title">选项4</template>
                            <el-menu-item index="1-4-1">选项4-1</el-menu-item>
                        </el-submenu>
                    </el-submenu>
                    <el-submenu index="2">
                        <template slot="title"><i class="el-icon-menu"></i>导航二

                        <el-menu-item-group>
                            <template slot="title">分组一</template>
                            <el-menu-item index="2-1">选项1</el-menu-item>
                            <el-menu-item index="2-2">选项2</el-menu-item>
                        </el-menu-item-group>
                        <el-menu-item-group title="分组2">
                            <el-menu-item index="2-3">选项3</el-menu-item>
                        </el-menu-item-group>
                        <el-submenu index="2-4">
                            <template slot="title">选项4</template>
                            <el-menu-item index="2-4-1">选项4-1</el-menu-item>
                        </el-submenu>
                    </el-submenu>
                    <el-submenu index="3">
                        <template slot="title"><i class="el-icon-setting"></i>导航三</template>

                        <el-menu-item-group>

```

```

        <template slot="title">分组一</template>
        <el-menu-item index="3-1">选项1</el-menu-item>
        <el-menu-item index="3-2">选项2</el-menu-item>
    </el-menu-item-group>
    <el-menu-item-group title="分组2">
        <el-menu-item index="3-3">选项3</el-menu-item>
    </el-menu-item-group>
    <el-submenu index="3-4">
        <template slot="title">选项4</template>
        <el-menu-item index="3-4-1">选项4-1</el-menu-item>
    </el-submenu>
</el-submenu>
</el-menu>
</el-aside>

<el-container>
    <el-header style="text-align: right; font-size: 12px">
        <el-dropdown>
            <i class="el-icon-setting" style="margin-right: 15px"></i>
            <el-dropdown-menu slot="dropdown">
                <el-dropdown-item>查看</el-dropdown-item>
                <el-dropdown-item>新增</el-dropdown-item>
                <el-dropdown-item>删除</el-dropdown-item>
            </el-dropdown-menu>
        </el-dropdown>
        <span>王小虎</span>
    </el-header>

    <el-main>
        <el-table :data="tableData">
            <el-table-column prop="date" label="日期" width="140">
            </el-table-column>
            <el-table-column prop="name" label="姓名" width="120">
            </el-table-column>
            <el-table-column prop="address" label="地址">
            </el-table-column>
        </el-table>
    </el-main>
</el-container>
</el-container>
</div>
<script>
    var app = new Vue({
        el: "#app",
        data() {
            const item = {
                date: '2016-05-02',
                name: '王小虎',
                address: '上海市普陀区金沙江路 1518 弄'
            }
        }
    })

```

```

    };
    return {
      tableData: Array(20).fill(item)
    }
  }
}
}))

</script>
</body>

</html>

```

效果：



3.3 导航菜单

参考案例：

```

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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>
  <style>
    a {
      /*设置超链接的颜色和 删除下划线*/
    }
  </style>

```

```

        text-decoration: none;
        display: block;
    }

    .el-menu-item a {
        color: #fff
    }

    .el-menu-item:hover a,
    .el-menu-item.is-active a {
        color: #ffd04b;
    }
</style>
</head>

<body>
    <div id="app">
        <el-menu default-active="2" class="el-menu-vertical-demo" background-
color="#545c64" text-color="#fff"
        active-text-color="#ffd04b">
            <el-submenu index="1">
                <template slot="title">
                    <i class="el-icon-location"></i>
                    <span>组件介绍</span> </template>
                <el-menu-item-group>
                    <template slot="title">官网</template>
                    <el-menu-item index="1-1">
                        <a href="http://www.ujiuye.com/" target="right">xxx</a>
                    </el-menu-item>
                    <el-menu-item index="1-2"><a href="http://www.ujiuye.com/"
target="right">mmm</a></el-menu-item>
                </el-menu-item-group>
                <el-menu-item-group title="其他">
                    <el-menu-item index="1-3">选项3</el-menu-item>
                </el-menu-item-group>
            </el-submenu>
            <el-menu-item index="2">
                <i class="el-icon-menu"></i>
                <span slot="title">导航二</span>
            </el-menu-item>
        </el-menu>
    </el-aside>
    <el-main>
        <iframe name="right" width="100%" height="100%" frameborder="0"></iframe>
    </el-main>
</div>
<script>
    var app = new Vue({
        el: "#app"
    })

```

```
    })
  </script>
</body>

</html>
```

效果：

默认颜色



自定义颜色



3.4 按钮

常见的几种按钮

```
<el-button>默认按钮</el-button>
<el-button type="primary">主要按钮</el-button>
<el-button type="success">成功按钮</el-button>
<el-button type="info">信息按钮</el-button>
<el-button type="warning">警告按钮</el-button>
<el-button type="danger">危险按钮</el-button>
```

默认按钮

主要按钮

成功按钮

信息按钮

警告按钮

危险按钮

朴素按钮

主要按钮

成功按钮

信息按钮

警告按钮

危险按钮

圆角按钮

主要按钮

成功按钮

信息按钮

警告按钮

危险按钮

🔍

✍️

✅

✉️

☆

🗑️

3.5 消息提示

几种常见的消息提示：

```
<el-button :plain="true" @click="open2">成功</el-button>
<el-button :plain="true" @click="open3">警告</el-button>
<el-button :plain="true" @click="open1">消息</el-button>
<el-button :plain="true" @click="open4">错误</el-button>
```

参考代码：

```
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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">
    <el-button :plain="true" @click="open2">成功</el-button>
    <el-button :plain="true" @click="open3">警告</el-button>
    <el-button :plain="true" @click="open1">消息</el-button>
    <el-button :plain="true" @click="open4">错误</el-button>
  </div>
  <script>
    var app = new Vue({
      el: "#app",
      methods: {
        open1() {
          this.$message.info('这是一条消息提示');
        },
        open2() {
          this.$message.success('恭喜你，这是一条成功消息');
        },
        open3() {
          this.$message.warning('警告哦，这是一条警告消息');
        },
        open4() {
          this.$message.error('错了哦，这是一条错误消息');
        }
      }
    });
```



```

    }
  })
</script>
</body>

</html>

```

效果：



3.6 对话框

参考代码：

```

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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">
  <!-- 点击按钮，显示对话框 -->
  <!-- 引用vue中的变量 dialogVisible -->
  <el-button type="text" @click="dialogVisible = true">点击打开 Dialog</el-button>
  <!--
    dialog 在页面中存在，只是处于隐藏状态
    title 标题
    :visible.sync: visible 是否显示 Dialog
  -->

```

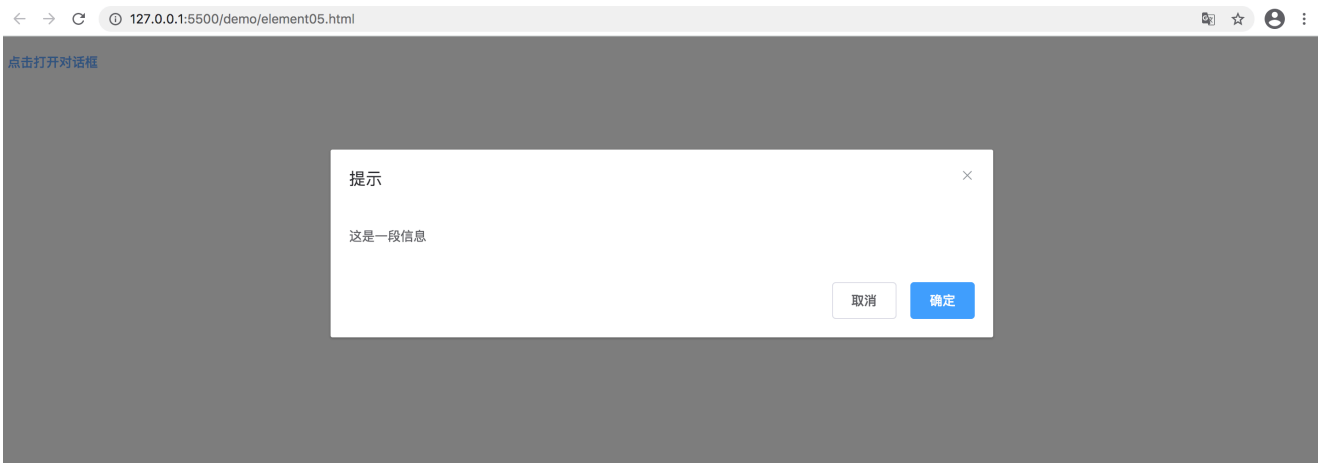
```

冒号:v-bind缩写
-->
<el-dialog title="提示" :visible.sync="dialogVisible" width="50%">
  <span>这是一段信息</span>
  <span slot="footer" class="dialog-footer">
    <!--点击取消时, 关闭窗口 -->
    <el-button @click="dialogVisible = false">取 消</el-button>
    <!--点击确认时 关闭窗口, 真实环境下, 应该指向添加或者修改的js方法-->
    <el-button type="primary" @click="dialogVisible = false">确 定</el-
button>
  </span>
</el-dialog>
</div>
<script>
  var app = new Vue({
    el: "#app",
    data: {
      dialogVisible: false //默认隐藏对话框
    }
  })
</script>
</body>

</html>

```

效果：



3.7 标签页

参考代码：

```

<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<title>Document</title>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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">
    <el-button type="text" @click="dialogVisible = true">点击打开 Dialog</el-button>
    <el-dialog title="提示" :visible.sync="dialogVisible" width="50%">
      <!--标签页-->
      <!--v-model : 双向绑定, 绑定vue中的数据 -->
      <!--actionName 指定默认选中的选项卡, 在js中actionName="first",所以默认选中第一个选项卡 -->
      <el-tabs v-model="activeName">
        <el-tab-pane label="用户管理" name="first">
          用户管理<br /> 用户管理<br /> 用户管理<br />
        </el-tab-pane>
        <el-tab-pane label="配置管理" name="second">
          配置管理<br /> 配置管理<br /> 配置管理<br />
        </el-tab-pane>
        <el-tab-pane label="角色管理" name="third">
          角色管理<br /> 角色管理<br /> 角色管理<br />
        </el-tab-pane>
        <el-tab-pane label="定时任务补偿" name="fourth">
          定时任务补偿<br /> 定时任务补偿<br /> 定时任务补偿<br />
        </el-tab-pane>
      </el-tabs>
      <span slot="footer" class="dialog-footer">
        <!--点击取消时, 关闭窗口 -->
        <el-button @click="dialogVisible = false">取 消</el-button>
        <!--点击确认时 关闭窗口, 真实环境下, 应该指向添加或者修改的js方法-->
        <el-button type="primary" @click="dialogVisible = false">确 定</el-button>
      </span>
    </el-dialog>
  </div>
<script>
  var app = new Vue({
    el: "#app",
    data: {
      dialogVisible: false, //默认隐藏对话框
      activeName: 'first' //选中选项卡的 name
    }
  })

```

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

效果：



3.8 表格

参考代码：

```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
<meta charset="UTF-8">  
<meta name="viewport" content="width=device-width, initial-scale=1.0">  
<title>Document</title>  
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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">  
  <!--  
    :data: 绑定表格对应的数据  
  -->  
<el-table :data="tableData" style="width: 100%">  
  <!--  
    type="index" :类型为行的索引，注意：索引不是集合中的id  
    label: 表头
```

```

-->
<el-table-column label="序号" width="180" type="index"></el-table-column>
<el-table-column label="姓名" width="180" prop="name"></el-table-column>
<el-table-column label="地址" width="240" prop="address"></el-table-
column>
<el-table-column label="操作">
  <!--
    slot-scope:作用域插槽，可以获取表格数据 scope:代表表格数据(tableData)
    可以通过scope.row来获取表格当前行数据，scope不是固定写法 @click:绑定vue中
的方法
  -->
  <template slot-scope="scope">
    <el-button size="mini" type="primary"
@click="handleEdit(scope.row)">编辑
  </el-button>
    <el-button size="mini" type="danger"
@click="handleDelete(scope.row)">删除
  </el-button>
  </template>
</el-table-column>
</el-table>
</div>
<script>
var app = new Vue({
  el: "#app",
  data: {
    tableData: [{
      id: 7,
      name: '王小虎111',
      address: '上海市普陀区金沙江路 1111 弄'
    }, {
      id: 8,
      name: '王小虎222',
      address: '上海市普陀区金沙江路 2222 弄'
    }, {
      id: 9,
      name: '王小虎333',
      address: '上海市普陀区金沙江路 3333 弄'
    }, {
      id: 10,
      name: '王小虎444',
      address: '上海市普陀区金沙江路 4444 弄'
    }
  ]
},
  methods: {
    andleEdit(row) {
      console.log(row);
    },
    handleDelete(row) {

```

```

        console.log(row);
    }
}
})
</script>
</body>

</html>

```

效果：

| | | | |
|----|--------|-------------------|---------------------------------------|
| 序号 | 姓名 | 地址 | 操作 |
| 1 | 王小虎111 | 上海市普陀区金沙江路 1111 弄 | 编辑 删除 |
| 2 | 王小虎222 | 上海市普陀区金沙江路 2222 弄 | 编辑 删除 |
| 3 | 王小虎333 | 上海市普陀区金沙江路 3333 弄 | 编辑 删除 |
| 4 | 王小虎444 | 上海市普陀区金沙江路 4444 弄 | 编辑 删除 |

3.9 分页

参考代码：

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
  <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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">
    <!--
      分页组件
      @size-change：当改变每页条数时触发的钩子函数
      @current-change：当改变页码时触发的钩子函数
      current-page ：当前页数
      :page-sizes：每页显示个数选择器的选项设置
      :page-size ： 每页显示条目个数
    -->

```

```

        layout: 分页组件的布局
        total(总条数),
        sizes(每页条数),
        prev(上一页),
        pager(所有的页码),
        next(下一 页),
        jumper(跳转页码)
        :total: 总条目数

-->
<el-pagination @size-change="handleSizeChange" @current-
change="handleCurrentChange" :current-page="currentPage"
        :page-sizes="[100, 200, 300, 400]" :page-size="100" layout="total, sizes,
prev, pager, next, jumper"
        :total="400">
</el-pagination>
</div>
<script>
    var app = new Vue({
        el: "#app",
        data: {
            currentPage: 1, //当前页码
        },
        methods: {
            handleSizeChange(val) {
                console.log(`每页 ${val} 条`);
            },
            handleCurrentChange(val) {
                console.log(`当前页: ${val}`);
            }
        }
    })
</script>
</body>

</html>

```

效果：



3.10 表单

参考代码：

```

<!DOCTYPE html>
<html lang="en">

```

```

    <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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">
        <!--
:model: 引入vue中的 ruleForm 数据 :rules: 引入规则 rules
-->
        <el-form :model="ruleForm" :rules="rules" ref="ruleForm" label-width="100px"
class="demo-ruleForm">
            <!--
prop="name" 引用规则中的name规则, 该属性必须配置 v-model="ruleForm.name": 双向绑定
-->
            <el-form-item label="活动名称" prop="name">
                <el-input v-model="ruleForm.name"></el-input>
            </el-form-item>
            <el-form-item label="活动区域" prop="region">
                <el-select v-model="ruleForm.region" placeholder="请选择活动区域">
                    <el-option label="区域一" value="shanghai"></el-option>
                    <el-option label="区域二" value="beijing"></el-option>
                </el-select>
            </el-form-item>
            <el-form-item label="活动时间" required>
                <el-col :span="11">
                    <el-form-item prop="date1">
                        <el-date-picker type="date" placeholder="选择日期" v-
model="ruleForm.date1" style="width: 100%">
                    </el-date-picker>
                </el-form-item>
            </el-col>
            <el-col class="line" :span="2"></el-col>
            <el-col :span="11">
                <el-form-item prop="date2">
                    <el-time-picker placeholder="选择时间" v-model="ruleForm.date2"
style="width: 100%;">
                </el-tim>
            </el-form-item>
        </el-col>
    </el-form-item>

```



```

<el-form-item label="即时配送" prop="delivery">
  <el-switch v-model="ruleForm.delivery"></el-switch>
</el-form-item>
<el-form-item label="活动性质" prop="type">
  <el-checkbox-group v-model="ruleForm.type">
    <el-checkbox label="美食/餐厅线上活动" name="type"></el-checkbox>
    <el-checkbox label="地推活动" name="type"></el-checkbox>
    <el-checkbox label="线下主题活动" name="type"></el-checkbox>
    <el-checkbox label="单纯品牌曝光" name="type"></el-checkbox>
  </el-checkbox-group>
</el-form-item>
<el-form-item label="特殊资源" prop="resource">
  <el-radio-group v-model="ruleForm.resource">
    <el-radio label="线上品牌商赞助"></el-radio>
    <el-radio label="线下场地免费"></el-radio>
  </el-radio-group>
</el-form-item>
<el-form-item label="活动形式" prop="desc">
  <el-input type="textarea" v-model="ruleForm.desc"></el-input>
</el-form-item>
<el-form-item>
  <!-- 点击按钮，调用 submitForm ruleForm：参数为表单的ref值 -->
  <el-button type="primary" @click="submitForm('ruleForm')">提交</el-
button>

  <!-- 点击按钮，调用 resetForm -->
  <el-button @click="resetForm('ruleForm')">重置</el-button>
</el-form-item>
</el-form>
</div>
<script>
  var app = new Vue({
    el: "#app",
    data: {
      ruleForm: {
        name: '',
        region: '',
        date1: '',
        date2: '',
        delivery: false,
        type: [],
        resource: '',
        desc: ''
      }
    },
    rules: {
      //name规则，上表单中使用prop属性关联
      //required:true，必填项
      //message: 不满足规则的提示消息
      //trigger: 触发条件 blur :失去焦点 change: 改变

```

```

name: [
  { required: true, message: '请输入活动名称', trigger: 'blur' },
  { min: 3, max: 5, message: '长度在 3 到 5 个字符', trigger: 'blur' }
],
region: [
  { required: true, message: '请选择活动区域', trigger: 'change' }
],
//type:数据的类型 date: 日期类型 array : 多选框, 数组类型
date1: [
  {
    type: 'date', required: true, message: '请选择日期', trigger:
      'change'
  }
],
date2: [
  {
    type: 'date', required: true, message: '请选择时间', trigger:
      'change'
  }
],
type: [
  {
    type: 'array', required: true, message: '请至少选择一个活动性质',
    trigger: 'change'
  }
],
resource: [
  { required: true, message: '请选择活动资源', trigger: 'change' }
], desc: [
  { required: true, message: '请填写活动形式', trigger: 'blur' } ]
},
methods: {
  submitForm(formName) {
    // formName :参数值
    //this.$refs[formName]:根据表单的ref值获取表单对象
    //validate: 校验表单的数据
    //valid: 表单校验的结果 , 校验成功:true 校验失败:false
    this.$refs[formName].validate((valid) => {
      if (valid) {
        alert('submit!');
      } else {
        console.log('error submit!!');
        return false;
      }
    });
  },
  resetForm(formName) {
    //清空表单数据和校验数据
    this.$refs[formName].resetFields();
  }
}

```

```

    }
  }
})
</script>
</body>

</html>

```

效果：

Activity Name:

Activity Region:

* Activity Time: -

Immediate Delivery: ☐

Activity Nature: ☐ Food/Restaurant Online Activity ☐ Ground Promotion Activity ☐ Offline Theme Activity ☐ Single Brand Exposure

Special Resources: ☐ Online Brand Sponsorship ☐ Offline Venue Free

Activity Form:

3.11 下拉菜单

参考代码：

```

<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.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>
    <style>
      .el-dropdown-link {
        cursor: pointer;
        color: #409EFF;
      }

      .el-icon-arrow-down {
        font-size: 12px;
      }
    </style>
  </head>

```

```

</style>
</head>

<body>
<div id="app">
  <!--@command="handleCommand" 点击菜单项触发的事件回调 -->
  <el-dropdown @command="handleCommand">
    <span class="el-dropdown-link">
      下拉菜单<i class="el-icon-arrow-down el-icon--right"></i>
    </span>
    <el-dropdown-menu slot="dropdown">
      <!--command : 选项对应的指令-->
      <el-dropdown-item command="a">黄金糕</el-dropdown-item>
      <el-dropdown-item command="b">狮子头</el-dropdown-item>
      <el-dropdown-item command="c">螺蛳粉</el-dropdown-item>
      <el-dropdown-item command="d" disabled>双皮奶</el-dropdown-item>
      <el-dropdown-item command="e" divided>蚵仔煎</el-dropdown-item>
    </el-dropdown-menu>
  </el-dropdown>
</div>
<script>
  var app = new Vue({
    el: "#app",
    methods: {
      handleCommand(command) {

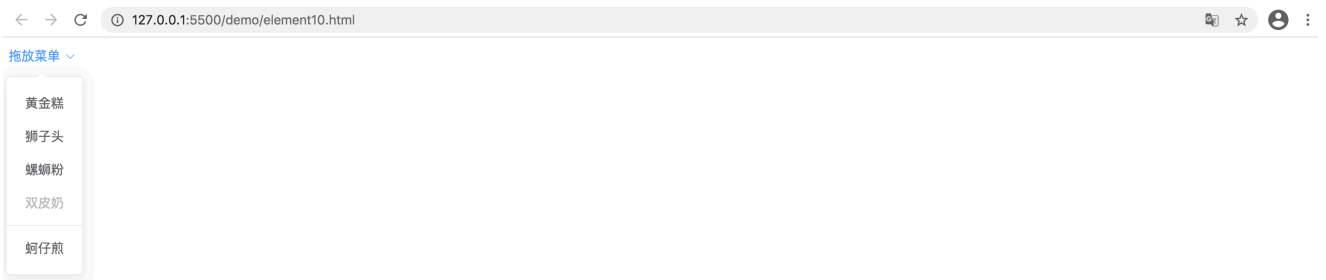
        this.$message('click on item ' + command);

      }
    }
  })
</script>
</body>

</html>

```

效果：



第4节：综合案例

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
  <!-- ElementUI依赖样式-->
  <link rel="stylesheet" href="element-ui-2.13.0/lib/theme-chalk/index.css" />

  <!-- vue依赖 -->
  <script src="vue.js"></script>

  <!-- ElementUI依赖 -->
  <script src="element-ui-2.13.0/lib/index.js"></script>
  <style>
    .el-header {
      background-image: url(img/bal.png);
      background-repeat: no-repeat;
      background-size: 100%;
      background-color: #fff;
      color: #333;
      text-align: center;
      line-height: 60px;
    }

    .el-aside {
      background-color: #545c64;
      color: #333;
      text-align: center;
      line-height: 200px;
      height: 700px;
    }

    .el-main {
      background-color: #e9eef3;
      color: #333;
      text-align: right;
      line-height: 160px;
    }

    body > .el-container {
      margin-bottom: 40px;
    }
  </style>
</html>
```

```

.el-container:nth-child(5) .el-aside,
.el-container:nth-child(6) .el-aside {
  line-height: 260px;
}

.el-container:nth-child(7) .el-aside {
  line-height: 320px;
}
</style>
</head>
<body>
<div id="app">
  <el-container>
    <el-header></el-header>
    <el-container>
      <el-aside width="200px">
        <el-menu
          default-active="2"
          class="el-menu-vertical-demo"
          @open="handleOpen"
          @close="handleClose"
          background-color="#545c64"
          text-color="#fff"
          active-text-color="#ffd04b"
        >
          <el-submenu index="1">
            <template slot="title">
              <i class="el-icon-location"></i>
              <span>导航一</span>
            </template>
            <el-menu-item-group>
              <template slot="title">分组一</template>
              <el-menu-item index="1-1">
                <a href="student.html" target="right">学生信息</a>
              </el-menu-item>
              <el-menu-item index="1-2">
                <a href="#" target="right">老师信息</a>
              </el-menu-item>
            </el-menu-item-group>
            <el-menu-item-group title="分组2">
              <el-menu-item index="1-3">选项3</el-menu-item>
            </el-menu-item-group>
            <el-submenu index="1-4">
              <template slot="title">选项4</template>
              <el-menu-item index="1-4-1">选项1</el-menu-item>
            </el-submenu>
          </el-submenu>
          <el-menu-item index="2">

```

```

        <i class="el-icon-menu"></i>
        <span slot="title">导航二</span>
      </el-menu-item>
      <el-menu-item index="3" disabled>
        <i class="el-icon-document"></i>
        <span slot="title">导航三</span>
      </el-menu-item>
      <el-menu-item index="4">
        <i class="el-icon-setting"></i>
        <span slot="title">导航四</span>
      </el-menu-item>
    </el-menu>
  </el-aside>
  <el-main>
    <iframe name="right" width="100%" height="100%" frameborder="0">
</iframe>
  </el-main>
</el-container>
</el-container>
</div>
<script>
  new Vue({
    el: "#app",
    methods: {
      handleOpen(key, keyPath) {
        console.log(key, keyPath);
      },
      handleClose(key, keyPath) {
        console.log(key, keyPath);
      },
    },
  });
</script>
</body>
</html>

```

student.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Document</title>
  <!-- ElementUI依赖样式-->
  <link rel="stylesheet" href="element-ui-2.13.0/lib/theme-chalk/index.css" />

```



```
el: "#app",
data() {
  return {
    tableData: [
      {
        date: "2016-05-02",
        name: "王小虎",
        address: "上海市普陀区金沙江路 1518 弄",
      },
      {
        date: "2016-05-04",
        name: "王小虎",
        address: "上海市普陀区金沙江路 1517 弄",
      },
      {
        date: "2016-05-01",
        name: "王小虎",
        address: "上海市普陀区金沙江路 1519 弄",
      },
      {
        date: "2016-05-03",
        name: "王小虎",
        address: "上海市普陀区金沙江路 1516 弄",
      },
    ],
    search: "",
  };
},
methods: {
  handleEdit(index, row) {
    console.log(index, row);
  },
  handleDelete(index, row) {
    console.log(index, row);
  },
},
});
</script>
</body>
</html>
```

导航一

分组一

学生信息

老师信息

分组2

选项3

选项4

导航二

导航三

导航四

Date

Name

输入关键字搜索

| | | |
|------------|-----|-----------------------|
| 2016-05-02 | 王小虎 | <div>EditDelete</div> |
| 2016-05-04 | 王小虎 | <div>EditDelete</div> |
| 2016-05-01 | 王小虎 | <div>EditDelete</div> |
| 2016-05-03 | 王小虎 | <div>EditDelete</div> |

<

1

2

3

4

5

6

...

100

>