

# 1、Vue 快速入门

---

## 1.1、Vue的介绍

---

- Vue是一套构建用户界面的渐进式前端框架。
- 只关注视图层，并且非常容易学习，还可以很方便的与其它库或已有项目整合。
- 通过尽可能简单的API来实现响应数据的绑定和组合的视图组件。
- 特点
  - 易用：在有HTMLCSSJavaScript的基础上，快速上手。
  - 灵活：简单小巧的核心，渐进式技术栈，足以应付任何规模的应用。
  - 性能：20kbmin+gzip运行大小、超快虚拟DOM、最省心的优化。

## 1.2、Vue的快速入门

---

- 开发步骤

1. 下载和引入vue.js文件。
2. 编写入门程序。

视图：负责页面渲染，主要由HTML+CSS构成。

脚本：负责业务数据模型（Model）以及数据的处理逻辑。

- 代码实现

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>快速入门</title>
</head>
<body>
  <!-- 视图 -->
  <div id="div">
    {{msg}}
  </div>
</body>
<script src="js/vue.js"></script>
<script>
  // 脚本
  new Vue({
    el:"#div",
    data:{
      msg:"Hello Vue"
    }
  });
</script>
</html>
```

## 1.3、Vue快速入门详解

---

- Vue 核心对象：每一个 Vue 程序都是从一个 Vue 核心对象开始的。

```
let vm = new Vue({
  选项列表;
});
```

- 选项列表
  - el选项：用于接收获取到页面中的元素。(根据常用选择器获取)。
  - data选项：用于保存当前Vue对象中的数据。在视图中声明的变量需要在此处赋值。
  - methods选项：用于定义方法。方法可以直接通过对象名调用，this代表当前Vue对象。
- 数据绑定
  - 在视图部分获取脚本部分的数据。
  - {{变量名}}

## 1.4、Vue快速入门的升级

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>快速入门升级</title>
</head>
<body>
  <!-- 视图 -->
  <div id="div">
    <div>姓名: {{name}}</div>
    <div>班级: {{classRoom}}</div>
    <button onclick="hi()">打招呼</button>
    <button onclick="update()">修改班级</button>
  </div>
</body>
<script src="js/vue.js"></script>
<script>
  // 脚本
  let vm = new Vue({
    el:"#div",
    data:{
      name:"张三",
      classRoom:"黑马程序员"
    },
    methods:{
      study(){
        alert(this.name + "正在" + this.classRoom + "好好学习!");
      }
    }
  });

  //定义打招呼方法
  function hi(){
    vm.study();
  }

  //定义修改班级
  function update(){
    vm.classRoom = "传智播客";
  }
```

```
</script>
</html>
```

## 1.5、Vue小结

- Vue是一套构建用户界面的渐进式前端框架。
- Vue的程序包含视图和脚本两个核心部分。
- 脚本部分
  - Vue核心对象。
  - 选项列表
    - el: 接收获取的元素。
    - data: 保存数据。
    - methods: 定义方法。
- 视图部分
  - 数据绑定: {{变量名}}

## 2、Vue 常用指令

### 2.1、指令介绍

- 指令: 是带有 v- 前缀的特殊属性, 不同指令具有不同含义。例如 v-html, v-if, v-for。
- 使用指令时, 通常编写在标签的属性上, 值可以使用 JS 的表达式。
- 常用指令

| 指令        | 作用                             |
|-----------|--------------------------------|
| v-html    | 把文本解析为HTML代码                   |
| v-bind    | 为HTML标签绑定属性值                   |
| v-if      | 条件性的渲染某元素, 判定为true时渲染, 否则不渲染   |
| v-else    |                                |
| v-else-if |                                |
| v-show    | 根据条件展示某元素, 区别在于切换的是display属性的值 |
| v-for     | 列表渲染, 遍历容器的元素或者对象的属性           |
| v-on      | 为HTML标签绑定事件                    |
| v-model   | 在表单元素上创建双向数据绑定                 |

### 2.2、文本插值

- v-html: 把文本解析为 HTML 代码。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>文本插值</title>
</head>
<body>
  <div id="div">
    <div>{{msg}}</div>
```

```

        <div v-html="msg"></div>
    </div>
</body>
<script src="js/vue.js"></script>
<script>
    new vue({
        el: "#div",
        data: {
            msg: "<b>Hello vue</b>"
        }
    });
</script>
</html>

```

## 2.3、绑定属性

- v-bind: 为 HTML 标签绑定属性值。

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>绑定属性</title>
    <style>
        .my{
            border: 1px solid red;
        }
    </style>
</head>
<body>
    <div id="div">
        <a v-bind:href="url">百度一下</a>
        <br>
        <a :href="url">百度一下</a>
        <br>
        <div :class="cls">我是div</div>
    </div>
</body>
<script src="js/vue.js"></script>
<script>
    new vue({
        el: "#div",
        data: {
            url: "https://www.baidu.com",
            cls: "my"
        }
    });
</script>
</html>

```

## 2.4、条件渲染

- v-if: 条件性的渲染某元素, 判定为真时渲染, 否则不渲染。
- v-else: 条件性的渲染。

- v-else-if: 条件性的渲染。
- v-show: 根据条件展示某元素, 区别在于切换的是display属性的值。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>条件渲染</title>
</head>
<body>
  <div id="div">
    <!-- 判断num的值, 对3取余 余数为0显示div1 余数为1显示div2 余数为2显示div3 -->

    <div v-if="num % 3 == 0">div1</div>
    <div v-else-if="num % 3 == 1">div2</div>
    <div v-else="num % 3 == 2">div3</div>

    <div v-show="flag">div4</div>
  </div>
</body>
<script src="js/vue.js"></script>
<script>
  new Vue({
    el: "#div",
    data: {
      num: 1,
      flag: false
    }
  });
</script>
</html>
```

## 2.5、列表渲染

- v-for: 列表渲染, 遍历容器的元素或者对象的属性。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>列表渲染</title>
</head>
<body>
  <div id="div">
    <ul>
      <li v-for="name in names">
        {{name}}
      </li>
      <li v-for="value in student">
        {{value}}
      </li>
    </ul>
  </div>
</body>
```

```

<script src="js/vue.js"></script>
<script>
  new Vue({
    el:"#div",
    data:{
      names:["张三","李四","王五"],
      student:{
        name:"张三",
        age:23
      }
    }
  });
</script>
</html>

```

## 2.6、事件绑定

- v-on: 为 HTML 标签绑定事件。

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>事件绑定</title>
</head>
<body>
  <div id="div">
    <div>{{name}}</div>
    <button v-on:click="change()">改变div的内容</button>
    <button v-on:dblclick="change()">改变div的内容</button>

    <button @click="change()">改变div的内容</button>
  </div>
</body>
<script src="js/vue.js"></script>
<script>
  new Vue({
    el:"#div",
    data:{
      name:"黑马程序员"
    },
    methods:{
      change(){
        this.name = "传智播客"
      }
    }
  });
</script>
</html>

```

## 2.7、表单绑定

- 表单绑定  
v-model: 在表单元素上创建双向数据绑定。

- **双向数据绑定**

更新data数据，页面中的数据也会更新。

更新页面数据，data数据也会更新。

- **MVVM模型(ModelViewViewModel): 是MVC模式的改进版**

在前端页面中，JS对象表示Model，页面表示View，两者做到了最大限度的分离。

将Model和View关联起来的就是ViewModel，它是桥梁。

ViewModel负责把Model的数据同步到View显示出来，还负责把View修改的数据同步回Model。



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>表单绑定</title>
</head>
<body>
  <div id="div">
    <form autocomplete="off">
      姓名: <input type="text" name="username" v-model="username">
      <br>
      年龄: <input type="number" name="age" v-model="age">
    </form>
  </div>
</body>
<script src="js/vue.js"></script>
<script>
  new Vue({
    el: "#div",
    data: {
      username: "张三",
      age: 23
    }
  });
</script>
</html>
```

## 2.8、小结

- **指令**: 是带有v-前缀的特殊属性，不同指令具有不同含义。

- **文本插值**

v-html: 把文本解析为HTML代码。

- **绑定属性**

v-bind: 为HTML标签绑定属性值。

- **条件渲染**

v-if: 条件性的渲染某元素，判定为真时渲染,否则不渲染。

v-else: 条件性的渲染。

v-else-if: 条件性的渲染。

v-show: 根据条件展示某元素, 区别在于切换的是display属性的值。

- **列表渲染**

v-for: 列表渲染, 遍历容器的元素或者对象的属性。

- **事件绑定**

v-on: 为HTML标签绑定事件。

- **表单绑定**

v-model: 在表单元素上创建双向数据绑定。

## 3、Element 基本使用

### 3.1、Element介绍

- Element: 网站快速成型工具。是饿了么公司前端开发团队提供的一套基于Vue的网站组件库。
- 使用Element前提必须要有Vue。
- 组件: 组成网页的部件, 例如超链接、按钮、图片、表格等等~
- Element官网: <https://element.eleme.cn/#/zh-CN>
- 自己完成的按钮

我是按钮

- Element 提供的按钮



### 3.2、Element快速入门

- **开发步骤**

1. 下载 Element 核心库。
2. 引入 Element 样式文件。
3. 引入 Vue 核心 js 文件。
4. 引入 Element 核心 js 文件。
5. 编写按钮标签。
6. 通过 Vue 核心对象加载元素。

- **代码实现**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>快速入门</title>
  <link rel="stylesheet" href="element-ui/lib/theme-chalk/index.css">
  <script src="js/vue.js"></script>
  <script src="element-ui/lib/index.js"></script>
```



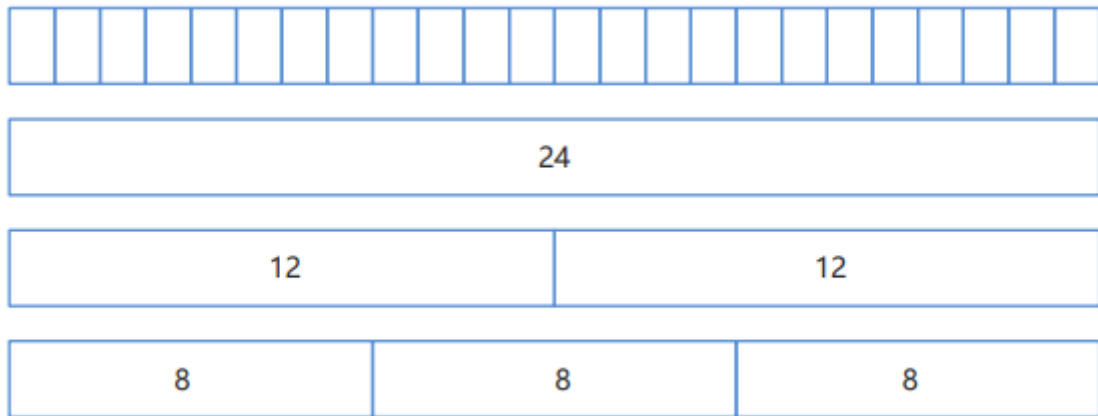
```

</head>
<body>
  <button>我是按钮</button>
  <br>
  <div id="div">
    <el-row>
      <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>
    </el-row>
    <br>
    <el-row>
      <el-button plain>朴素按钮</el-button>
      <el-button type="primary" plain>主要按钮</el-button>
      <el-button type="success" plain>成功按钮</el-button>
      <el-button type="info" plain>信息按钮</el-button>
      <el-button type="warning" plain>警告按钮</el-button>
      <el-button type="danger" plain>危险按钮</el-button>
    </el-row>
    <br>
    <el-row>
      <el-button round>圆角按钮</el-button>
      <el-button type="primary" round>主要按钮</el-button>
      <el-button type="success" round>成功按钮</el-button>
      <el-button type="info" round>信息按钮</el-button>
      <el-button type="warning" round>警告按钮</el-button>
      <el-button type="danger" round>危险按钮</el-button>
    </el-row>
    <br>
    <el-row>
      <el-button icon="el-icon-search" circle></el-button>
      <el-button type="primary" icon="el-icon-edit" circle></el-
button>
      <el-button type="success" icon="el-icon-check" circle></el-
button>
      <el-button type="info" icon="el-icon-message" circle></el-
button>
      <el-button type="warning" icon="el-icon-star-off" circle></el-
button>
      <el-button type="danger" icon="el-icon-delete" circle></el-
button>
    </el-row>
  </div>
</body>
<script>
  new Vue({
    el: "#div"
  });
</script>
</html>

```

### 3.3、基础布局

将页面分成最多 24 个部分，自由切分。



- 代码实现

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>基础布局</title>
  <link rel="stylesheet" href="element-ui/lib/theme-chalk/index.css">
  <script src="js/vue.js"></script>
  <script src="element-ui/lib/index.js"></script>
  <style>
    .el-row {
      /* 行距为20px */
      margin-bottom: 20px;
    }
    .bg-purple-dark {
      background: red;
    }
    .bg-purple {
      background: blue;
    }
    .bg-purple-light {
      background: green;
    }
    .grid-content {
      /* 边框圆润度 */
      border-radius: 4px;
      /* 行高为36px */
      min-height: 36px;
    }
  </style>
</head>
<body>
  <div id="div">
    <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>
</body>
<script>
    new vue({
        el: "#div"
    });
</script>
</html>

```

## 3.4、容器布局

将页面分成头部区域、侧边栏区域、主区域、底部区域。



- 代码实现

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>容器布局</title>
  <link rel="stylesheet" href="element-ui/lib/theme-chalk/index.css">
  <script src="js/vue.js"></script>
  <script src="element-ui/lib/index.js"></script>
  <style>
    .el-header, .el-footer {
      background-color: #d18e66;
      color: #333;
      text-align: center;
      height: 100px;
    }
    .el-aside {
      background-color: #55e658;
      color: #333;
      text-align: center;
      height: 580px;
    }
    .el-main {
      background-color: #5fb1f3;
      color: #333;
      text-align: center;
      height: 520px;
    }
  </style>
</head>
<body>
  <div id="div">
    <el-container>
      <el-header>头部区域</el-header>
      <el-container>
        <el-aside width="200px">侧边栏区域</el-aside>
        <el-container>
          <el-main>主区域</el-main>
          <el-footer>底部区域</el-footer>
        </el-container>
      </el-container>
    </div>
  </body>
</html>
```

```

        </el-container>
      </el-container>
    </el-container>
  </div>
</body>
<script>
  new Vue({
    el: "#div"
  });
</script>
</html>

```

## 3.5、表单组件

由输入框、下拉列表、单选框、多选框等控件组成，用以收集、校验、提交数据。

- 代码实现

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>表单组件</title>
  <link rel="stylesheet" href="element-ui/lib/theme-chalk/index.css">
  <script src="js/vue.js"></script>
  <script src="element-ui/lib/index.js"></script>
</head>
<body>
  <div id="div">
    <el-form :model="ruleForm" :rules="rules" ref="ruleForm" label-
width="100px" class="demo-ruleForm">
      <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-time-picker>
          </el-form-item>
        </el-col>
      </el-form-item>
    </el-form>
  </div>

```

```

<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>
  <el-button type="primary" @click="submitForm('ruleForm')">立即
创建</el-button>
  <el-button @click="resetForm('ruleForm')">重置</el-button>
</el-form-item>
</el-form>
</div>
</body>
<script>
  new Vue({
    el: "#div",
    data: {
      ruleForm: {
        name: '',
        region: '',
        date1: '',
        date2: '',
        delivery: false,
        type: [],
        resource: '',
        desc: ''
      },
    },
    rules: {
      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' }
      ],
      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) {
        this.$refs[formName].validate((valid) => {
            if (valid) {
                alert('submit!');
            } else {
                console.log('error submit!!');
                return false;
            }
        });
    },
    resetForm(formName) {
        this.$refs[formName].resetFields();
    }
}
});
</script>
</html>

```

## 3.6、表格组件

用于展示多条结构类似的数据，可对数据进行编辑、删除或其他自定义操作。

- 代码实现

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>表格组件</title>
    <link rel="stylesheet" href="element-ui/lib/theme-chalk/index.css">
    <script src="js/vue.js"></script>
    <script src="element-ui/lib/index.js"></script>
</head>
<body>
    <div id="div">
        <template>
            <el-table
                :data="tableData"
                style="width: 100%">
                <el-table-column

```

```

        prop="date"
        label="日期"
        width="180">
    </el-table-column>
    <el-table-column
        prop="name"
        label="姓名"
        width="180">
    </el-table-column>
    <el-table-column
        prop="address"
        label="地址">
    </el-table-column>

    <el-table-column
        label="操作"
        width="180">
        <el-button type="warning">编辑</el-button>
        <el-button type="danger">删除</el-button>
    </el-table-column>
    </el-table>
</template>
</div>
</body>
<script>
    new Vue({
        el:"#div",
        data:{
            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'
            }
        ]
    });
</script>
</html>

```

### 3.7、顶部导航栏组件

处理中心    我的工作台    消息中心    订单管理

- 代码实现



```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>顶部导航栏</title>
  <link rel="stylesheet" href="element-ui/lib/theme-chalk/index.css">
  <script src="js/vue.js"></script>
  <script src="element-ui/lib/index.js"></script>
</head>
<body>
  <div id="div">
    <el-menu
      :default-active="activeIndex2"
      class="el-menu-demo"
      mode="horizontal"
      @select="handleSelect"
      background-color="#545c64"
      text-color="#fff"
      active-text-color="#ffd04b">
      <el-menu-item index="1">处理中心</el-menu-item>
      <el-submenu index="2">
        <template slot="title">我的工作台</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 index="2-3">选项3</el-menu-item>
        <el-submenu index="2-4">
          <template slot="title">选项4</template>
          <el-menu-item index="2-4-1">选项1</el-menu-item>
          <el-menu-item index="2-4-2">选项2</el-menu-item>
          <el-menu-item index="2-4-3">选项3</el-menu-item>
        </el-submenu>
      </el-submenu>
      <el-menu-item index="3" disabled>消息中心</el-menu-item>
      <el-menu-item index="4"><a href="https://www.ele.me" target="_blank">订
单管理</a></el-menu-item>
    </el-menu>
  </div>
</body>
<script>
  new Vue({
    el:"#div"
  });
</script>
</html>

```

### 3.8、侧边导航栏组件



- 代码实现

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>侧边导航栏</title>
  <link rel="stylesheet" href="element-ui/lib/theme-chalk/index.css">
  <script src="js/vue.js"></script>
  <script src="element-ui/lib/index.js"></script>
</head>
<body>
  <div id="div">
    <el-col :span="6">
      <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">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-col>
</div>
</body>
<script>
  new Vue({
    el:"#div"
  });
</script>
</html>

```

## 3.9、小结

- Element：网站快速成型工具。是一套为开发者、设计师、产品经理准备的基于Vue的桌面端组件库。
- 使用Element前提必须要有Vue。
- 使用步骤
  - 1.下载Element核心库。
  - 2.引入Element样式文件。
  - 3.引入Vue核心js文件。
  - 4.引入Element核心js文件。
  - 5.借助常用组件编写网页。
- 常用组件
 

网页基本组成部分，布局、按钮、表格、表单等等~~~

常用组件不需要记住，只需要在Element官网中复制使用即可。

## 4、综合案例 学生列表

### 4.1、案例效果和分析

EXPORT

处理中心我的工作台首页

学工部

学生列表

添加学生

在校学生管理

学生升级/留级

学生就业情况

咨询部

教研部

| 日期         | 姓名 | 地址     | 操作              |
|------------|----|--------|-----------------|
| 2088-08-08 | 张三 | 北京市昌平区 | <div>编辑删除</div> |
| 2088-08-08 | 李四 | 北京市昌平区 | <div>编辑删除</div> |
| 2088-08-08 | 王五 | 北京市昌平区 | <div>编辑删除</div> |

### 4.2、头部区域的实现

- 实现思路
  - 头部效果实现。
  - 侧边栏和主区域效果实现。
- 代码实现

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>学生列表</title>
  <link rel="stylesheet" href="element-ui/lib/theme-chalk/index.css">
  <script src="js/vue.js"></script>
  <script src="element-ui/lib/index.js"></script>
  <style>
    .el-header{
      background-color: #545c64;
    }
    .header-img{
      width: 100px;
      margin-top: 20px;
    }
  </style>
</head>
<body>
  <div id="div">
    <el-container>
      <!-- 头部 -->
      <el-header class="el-header">
        <el-container>
          <div>
            <el-image src="img/export.png" class="header-img"></el-image>
          </div>
          <el-menu
            :default-active="activeIndex2"
            mode="horizontal"
            @select="handleSelect"
            background-color="#545c64"
            text-color="white"
            active-text-color="#ffd04b"
            style="margin-left: auto;">
            <el-menu-item index="1">处理中心</el-menu-item>
            <el-submenu index="2">
              <template slot="title">我的工作台</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 index="2-3">选项3</el-menu-item>
            </el-submenu>
            <el-menu-item index="3"><a href="学生列表.html" target="_self">首
页</a></el-menu-item>
          </el-menu>
        </el-container>
      </el-header>
    </el-container>
  </div>
</body>
<script>
  new Vue({
    el:"#div"
  });
</script>
</html>

```

## 4.3、侧边栏区域的实现

```
<!-- 侧边栏区域 -->
<el-container style="height: 580px; border: 1px solid #eee">
  <el-aside width="200px" style="background-color: rgb(238, 241, 246)">
    <el-menu :default-openeds="['1']">
      <el-submenu index="1">
        <template slot="title"><i class="el-icon-menu"></i>学工部
      </template>
      <el-menu-item-group>
        <el-menu-item index="1-1"><i class="el-icon-help"></i>在校学生
        管理</el-menu-item>
        <el-menu-item index="1-2"><i class="el-icon-help"></i>学生升
        级/留级</el-menu-item>
        <el-menu-item index="1-3"><i class="el-icon-help"></i>学生就业
        情况</el-menu-item>
      </el-menu-item-group>
    </el-submenu>
    <el-submenu index="2">
      <template slot="title"><i class="el-icon-menu"></i>咨询部
    </template>
    <el-menu-item-group>
      <el-menu-item index="2-1"><i class="el-icon-help"></i>意向学生
      管理</el-menu-item>
      <el-menu-item index="2-2"><i class="el-icon-help"></i>未报名学
      生管理</el-menu-item>
      <el-menu-item index="2-3"><i class="el-icon-help"></i>已报名学
      生管理</el-menu-item>
    </el-menu-item-group>
  </el-submenu>
  <el-submenu index="3">
    <template slot="title"><i class="el-icon-menu"></i>教研部
  </template>
  <el-menu-item-group>
    <el-menu-item index="3-1"><i class="el-icon-help"></i>已有课程
    管理</el-menu-item>
    <el-menu-item index="3-2"><i class="el-icon-help"></i>正在研发
    课程管理</el-menu-item>
    <el-menu-item index="3-3"><i class="el-icon-help"></i>新技术课
    程管理</el-menu-item>
  </el-menu-item-group>
</el-submenu>
</el-menu>
</el-aside>

</el-container>
```

## 4.4、主区域的实现

主区域和侧边栏区域放在一起

```
<!-- 主区域 -->
<el-container>
  <el-main>
    <b style="color: red;font-size: 20px;">学生列表</b>
    <div style="float:right">
```

```

        <el-button type="primary">添加学生</el-button>
    </div>
    <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-column
            label="操作"
            width="180">
            <el-button type="warning">编辑</el-button>
            <el-button type="danger">删除</el-button>
        </el-table-column>
    </el-table>
</el-main>
</el-container>

```

## 在vue中定义data

```

<script>
    new Vue({
        el: "#div",
        data: {
            tableData: [
                {
                    date: "2088-08-08",
                    name: "张三",
                    address: "北京市昌平区"
                }, {
                    date: "2088-08-08",
                    name: "李四",
                    address: "北京市昌平区"
                }, {
                    date: "2088-08-08",
                    name: "王五",
                    address: "北京市昌平区"
                }
            ]
        }
    });
</script>

```