师资培训课件（echarts 题目 ）

1. 十三届省赛和手机相处的时光

```
var chartDom = document.getElementById("main");
    var myChart = echarts.init(chartDom);
    var option = {
      title: {
        text: "一周的手机使用时长",
      },
      xAxis: {
        type: "category",
        data: ["周一", "周二", "周三", "周四", "周五", "周六", "周日"],
      },
      yAxis: {
        type: "value",
      },
      series: [
        {
          data: [2.5, 2, 2.6, 3.2, 4, 6, 5],
          type: "line",
        },
      ],
    };
    myChart.setOption(option);
```

2. 十三届国赛 天气趋势

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>天气统计</title>
    <meta
      name="viewport"
      content="width=device-width,initial-scale=1,minimum-scale=1,maximum-
scale=1,user-scalable=no"
    />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
    <script src="./js/axios.js"></script>
    <script src="js/vue.min.js" type="text/javascript" charset="utf-8">
</script>
    <script
      src="js/echarts.min.js"
      type="text/javascript"
      charset="utf-8"
    ></script>
  </head>

  <body>
    <div id="app">
      <div class="top-bar">2022年Y城全年温度统计图</div>
      <!-- 主体 -->
      <div class="container">
        <!-- 月份 -->
        <div class="month">
          <ul>
            <li
              v-for="(val, key, index) in monthList"
              :class="activeMonth===index+1?'active':''"
              @click="changeMonth(index)"
            >
              {{val}}
            </li>
          </ul>
        </div>
        <!-- 未来七天和本月tab切换框，只有当前月才显示 -->
        <div class="chart">
          <div id="currentMonth" v-if="currentMonth==activeMonth">
            <div class="title">
              <h3>{{typeTitle}}</h3>
              <div class="type">
                <span
                  id="seven"
                  :class="type==1?'active':''"
                  @click="changeType(1)"
                  >未来7天</span
                >
                <span
                  id="current"
```

```
                :class="type==2?'active':''"
                @click="changeType(2)"
                >本月</span
              >
            </div>
          </div>
        </div>
        <div id="chart"></div>
      </div>
    </div>
  </body>
</html>
<script>
  var vm = new Vue({
    el: "#app",
    data: {
      type: 2, //本月和未来七天type，默认显示本月
      chart: null, // 图表
      chartOptions: null, // 图表配置项
      typeTitle: "本月天气",
      monthList: {
        January: "1月",
        February: "2月",
        March: "3月",
        April: "4月",
        May: "5月",
        June: "6月",
        July: "7月",
        August: "8月",
        September: "9月",
        October: "10月",
        November: "11月",
        December: "12月",
      },
      currentMonth: 0, // 当前月份
      activeMonth: 1, // 点击的月份
      currentDay: null, // 今天
      weatherData: [], // 天气数据
      seriesData: [], // Y 轴坐标数据
      xAxisdata: [], // x轴坐标数据
      months: [], // 把月份放进一个数组方便取数据
    },
    async created() {
      this.weatherData = await axios
        .get("./js/weather.json")
        .then((res) => res.data);
      //设置月份数组
      for (let key in this.monthList) {
        this.months.push(key);
      }
      let date = new Date();
      // 设置当前月份
      this.currentMonth = date.getMonth() + 1;
      // 获取今天的日期
      this.currentDay = date.getDate();
      // 设置默认显示一月份的天气
```

```
      this.chartOptions.series[0].data = this.weatherData[0]
[this.months[0]];
      this.chart.setOption(this.chartOptions);
    },
    mounted: function () {
      // 初始化 echarts
      this.$nextTick(() => {
        this.initChart();
      });
    },
    methods: {
      initChart() {
        // 初始化图表
        this.chart = echarts.init(document.getElementById("chart"));
        // 配置项
        this.chartOptions = {
          grid: {
            top: 35,
            bottom: 5,
            left: 10,
            right: 10,
            containLabel: true,
          },
          tooltip: {
            trigger: "axis",
            axisPointer: {
              lineStyle: {
                color: {
                  type: "linear",
                  x: 0,
                  y: 0,
                  x2: 0,
                  y2: 1,
                  colorStops: [
                    {
                      offset: 0,
                      color: "rgba(255,255,255,0)", // 0% 处的颜色
                    },
                    {
                      offset: 0.5,
                      color: "rgba(255,255,255,1)", // 100% 处的颜色
                    },
                    {
                      offset: 1,
                      color: "rgba(255,255,255,0)", // 100% 处的颜色
                    },
                  ],
                  global: false, // 缺省为 false
                },
              },
            },
          },
          xAxis: [
            {
              type: "category",
              boundaryGap: false,
              axisLabel: {
```

```
            formatter: "{value}",
            fontSize: 12,
            margin: 20,
            textStyle: {
              color: "#bfbfbf",
            },
          },
          axisLine: {
            lineStyle: {
              color: "#e9e9e9",
            },
          },
          splitLine: {
            show: true,
            lineStyle: {
              color: "#f7f7f7",
            },
          },
          axisTick: {
            show: false,
          },
          // x 轴显示的数据，即天数
          data: [
            1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
19,
            20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
          ],
        },
      ],
      yAxis: [
        {
          boundaryGap: false,
          type: "value",
          axisLabel: {
            textStyle: {
              color: "#bfbfbf",
            },
            formatter: `{value}\u2103`,
          },
          nameTextStyle: {
            color: "# fff ",
            fontSize: 12,
            lineHeight: 40,
          },
          splitLine: {
            lineStyle: {
              color: "#f7f7f7",
            },
          },
          axisLine: {
            show: true,
            lineStyle: {
              color: "#e9e9e9",
            },
          },
          axisTick: {
            show: false,
```

```
            },
          },
        ],
        series: [
          {
            name: "天气",
            type: "line",
            smooth: false,
            showSymbol: false,
            symbolSize: 0,
            zlevel: 3,
            itemStyle: {
              color: "#ff6600",
              borderColor: "#a3c8d8",
            },
            lineStyle: {
              normal: {
                width: 3,
                color: "#ff6600",
              },
            },
            areaStyle: {
              normal: {
                color: new echarts.graphic.LinearGradient(
                  0,
                  0,
                  0,
                  1,
                  [
                    {
                      offset: 0,
                      color: "#ff6600",
                    },
                    {
                      offset: 0.8,
                      color: "#ff9900",
                    },
                  ],
                  false
                ),
              },
            },
            //  Y 轴显示的数据，即温度数据
            data: [
              23, 19, 30, 31, 18, 20, 16, 15, 23, 27, 29, 30, 32, 23, 25,
20,
              22, 24, 34, 24, 21, 26, 23, 24, 25, 23, 25, 28, 32, 20,
            ],
          },
        ],
      };

      // 调用此方法设置 echarts 数据
      this.chart.setOption(this.chartOptions);
    },
    // 切换月份
    changeMonth: function (item) {
```

```
    this.typeTitle = "本月天气";
    this.type = 2;
    // 点击月的数据
    this.activeMonth = item + 1;
    this.xAxisdata = [];
    this.seriesData = this.weatherData[item][this.months[item]];

    for (let index = 0; index < this.seriesData.length; index++) {
      this.xAxisdata.push(index + 1);
    }
    this.chartOptions.xAxis[0].data = this.xAxisdata;
    this.chartOptions.series[0].data = this.seriesData;
    this.chart.setOption(this.chartOptions);
  },
  // 切换未来七天天气和本月天气
  changeType: function (item) {
    this.type = item;
    console.log(this.type);
    // 本月天气数据
    let currentMonthData =
      this.weatherData[this.currentMonth - 1][
        this.months[this.currentMonth - 1]
      ];
    if (item == 1) {
      this.typeTitle = "未来7天天气";
      let seven1 = currentMonthData.slice(
        this.currentDay - 1,
        this.currentDay + 6
      );
      let seven2 = [];
      this.xAxisdata = [];
      if (seven1.length < 7) {
        let nextMonth = this.month + 1;
        let nextMonthData =
          this.weatherData[nextMonth - 1][this.months[nextMonth - 1]];
        seven2 = nextMonthData.slice(0, 7 - seven1.length);
        // 本月如果剩余小于七天，先放本月天数
        for (let index = 0; index < seven1.length; index++) {
          this.xAxisdata.push(
            `${this.currentMonth}/${this.currentDay + index}`
          );
        }
        // 本月剩余如果大于七天，剩余的放下个月天数
        for (let index = 0; index < 7 - seven1.length; index++) {
          this.xAxisdata.push(`${this.currentMonth + 1}/${index}`);
        }
      } else {
        for (let index = 0; index < 7; index++) {
          this.xAxisdata.push(
            `${this.currentMonth}/${this.currentDay + index}`
          );
        }
      }
      this.nextSevenDays = [...seven1, ...seven2];
      this.chartOptions.xAxis[0].data = this.xAxisdata;
      this.chartOptions.series[0].data = this.nextSevenDays;
      this.chart.setOption(this.chartOptions);
```

```
        } else {
          this.typeTitle = "本月天气";
          this.xAxisdata = [];
          for (let index = 0; index < currentMonthData.length; index++) {
            this.xAxisdata.push(index + 1);
          }
          this.chartOptions.xAxis[0].data = this.xAxisdata;
          this.chartOptions.series[0].data = currentMonthData;

          this.chart.setOption(this.chartOptions);
        }
      },
    },
  });
</script>
```

3. 十四届省赛新冠数据统计页面

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>全球新冠疫情数据统计</title>
    <meta
      name="viewport"
      content="width=device-width,initial-scale=1,minimum-scale=1,maximum-
scale=1,user-scalable=no"
    />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
  </head>

  <body>
    <div id="app">
      <!-- 1. 学生需要修改下面的HTML模版和JS代码，使得当用户选择某一国家时展示对应的数
据，没有则展示默认值 -->
      <!-- 2. 学生需要补充完善图表渲染部分的代码，给图表设置一个居中的标题 -->
      <!-- 3. 学生需要修改补充mounted生命周期中的代码，利用axios发起数据请求并设置相应
的data变量 -->
      <header>
        <div>全球新冠疫情数据统计</div>
      </header>
      <main>
        <!-- TODO: 请修改以下代码实现 -->
        <div class="title">
          <h2>{{ currentCountry || "请选择国家" }}</h2>
        </div>
        <div class="boxes">
          <div class="box1">
            <h3>确诊</h3>
            <div class="number">
              <span class="font-bold">新增:</span>
              {{ currentData ? currentData.NewConfirmed : 0 }}
```

```
            </div>
            <div class="number">
              <span class="font-bold">总计:</span>
              {{ currentData ? currentData.TotalConfirmed : 0 }}
            </div>
          </div>
          <div class="box2">
            <h3>死亡</h3>
            <div class="number">
              <span class="font-bold">新增:</span>
              {{ currentData ? currentData.NewDeaths : 0 }}
            </div>
            <div class="number">
              <span class="font-bold">总计:</span>
              {{ currentData ? currentData.TotalDeaths : 0 }}
            </div>
          </div>
        </div>
        <select v-model="currentCountry" @change="selectChange">
          <option value="">Select Country</option>
          <option v-for="country in countries" :value="country"
:key="country">
            {{ country }}
          </option>
        </select>
        <div id="chart" style="width: 100%; height: 50vh"></div>
      </main>
    </div>
  </body>
  <script src="js/axios.min.js"></script>
  <script src="js/vue.min.js" type="text/javascript" charset="utf-8">
</script>
  <script src="js/echarts.min.js" type="text/javascript" charset="utf-8">
</script>
  <script>
    var vm = new Vue({
      el: "#app",
      data: {
        // 所有数据
        allData: null,
        // 当前选中国家的数据
        currentData: null,
        // 当前选中的国家全称
        currentCountry: "",
        // 所有的国家列表
        countries: [],
      },
      mounted: function () {
        // TODO: 学生需要补充此部分代码
        axios.get("./js/covid-data.json").then((res) => {
          console.log(JSON.stringify(res.data));
          this.countries = res.data.map((item) => item.Country);
          this.allData = res.data;
          this.initChart();
        });
      },
      methods: {
```

```
// TODO: 学生需要实现与该函数类似的功能
selectChange() {
  if (this.currentCountry) {
    this.currentData = this.allData.find(
      (i) => i.Country === this.currentCountry
    );
  } else {
    this.currentData = null;
  }
  this.initChart();
},
initChart() {
  // 初始化图表
  this.chart = echarts.init(document.getElementById("chart"));
  this.chartOptions = {
    // TODO: 学生需要补充title部分
    title: {
      text: "全球感染人数前30国家累计确诊人数统计",
      x: "center",
      // left: 'center'
    },
    tooltip: {
      trigger: "axis",
      axisPointer: {
        type: "shadow",
        label: {
          show: true,
        },
      },
    },
    // 设置x轴数据
    xAxis: {
      // 这里显示的是国家名称缩写，因为有些国家的全称太长，会导致界面不美观
      data: this.allData.map((item) => item.CountryCode),
      axisLabel: {
        show: true,
        interval: 0,
      },
    },
    yAxis: {
      type: "value",
      name: "确诊数量",
    },
    // 设置y轴数据
    series: [
      {
        data: this.allData.map((item) => item.TotalConfirmed),
        type: "bar",
        itemStyle: {
          normal: {
            color: "#a90000",
          },
        },
      },
    ],
  };
  // 调用此方法设置 echarts 数据
```

```
          this.chart.setOption(this.chartOptions);
        },
      },
    });
  </script>
</html>
```

4. 十四届国赛 github contribution

```
function renderChart() {
  /**
   * TODO:
   *   1. 使用 ajax 完成数据请求
   *   2. 渲染 chart
   *   3. 自定义 tootip
   */
  let data = [];
  $.ajax({
    url: "./data.json",
    method: "get",
    success: function (res) {
      res.forEach((item) => {
        data.push(Object.values(item));
      });
      // console.log(data);
      render(data);
    },
  });

  function render(data) {
    const chartNode = document.getElementById("chart");
    var chart = echarts.init(chartNode);

    const option = {
      title: {
        text: "2022每日提交记录",
        left: "center",
      },
      gradientColor: ["#b6e3ff", "#54aeff", "#0969da", "#0a3069"],
      calendar: {
        range: "2022",
        top: 80,
        left: "center",
        cellSize: 15,
        itemStyle: {
          color: "#ebedf0",
          borderWidth: 1,
        },
        splitLine: {
          show: false,
        },
```

```
        yearLabel: {
          show: false,
        },
      },
      tooltip: {
        formatter: (param) => {
          // console.log(param)
          // TODO: 完成自定义 tooltip 节点,  id 请勿变更
          return `<div id="tooltip">
          <div>日期：${param.data[0]}</div>
          <div>提交次数：${param.data[1]}</div>
          </div>`;
        },
      },
      visualMap: {
        min: 0,
        max: 80,
        top: 30,
        type: "piecewise",
        orient: "horizontal",
        left: "center",
      },
      series: {
        type: "heatmap",
        coordinateSystem: "calendar",
        data,
      },
    };
    chart.setOption(option);
    // 检测需要，请勿删除
    window.chart = chart;
    window.addEventListener("resize", chart.resize);
  }
}

renderChart();
```

总结： Echarts 题目一般难度在中等或者中等偏上，有时候会和 vue 结合到一起，考题数量 1，属于必考题，高频考点数据处理，x 轴数据设置、y 轴数据设置，`myChart.setOption(option);` 方法

备赛建议：Echarts 涉及到的 API 官方都会提供，主要以数据处理为主，难度通常在中

等左右，建议优先拿分。

分数较高的题目共同特点：数据操作较为复杂，递归函数逻辑较为复杂，二维数组相关操作，函数逻辑较为复杂。

echarts 建议练习： 模拟 1 模拟 2 模拟 3