

--说明：本文档相当于课件，把要讲授的内容（包括参考代码等）、每天安排学生做的事都记录进来；

【补充-课前准备】

- 1、需要软件：idea、vscode、jdk17、tomcat、maven、mysql、navicat；
- 2、创建qq群、课堂派班级等；
- 3、

第1次课：

0、大家先进qq群（实名）、课堂派班级群（姓名、学号要准确）；

1、介绍本次可视化项目要求：

1) 至少实现8个可视化页面，至少使用5种图表进行可视化展示，如：柱状图、折线图、饼图、散点图、雷达图、热力图、关系图、K线图、河流图、地图等。哪方面的数据不限，使用技术不限（纯前端、Java、Python都可以，但是要求编码实现。练习编码，不要直接使用可视化工具）；建议是对同一个项目的相关数据的可视化（实在没有同一个项目的数据，使用分散的数据也可以）；

2) 要求界面效果布局合理且美观、功能齐全。

3) 对于每个可视化图表，要有至少一段文字对可视化结果进行分析说明（重要，必须有）；

4) 最后一天交成果物（要求在最下面，先带大家看看）、进行答辩；

5) 成绩评定标准：

企业实训：数据可视化实训成绩评定标准								
	平时成绩(30)		项目综合成绩(70)					
	出勤情况(10)	课堂表现(20)	项目成果物(50)		答辩表现(20)			
			源代码	实训报告	项目演示效果	答辩基本礼仪	项目表述能力	回答问题
权重	10%	20%	30%	20%	5%	3%	5%	7%
实训成绩的评定过程为学生各项成绩的累加结果，综合评价学生的技术能力、业务理解能力、团队合作及沟通能力，保证成绩评定过程的公平公正。								

2、完善编码环境：

1) 建议使用我的这套编码环境，如果大家习惯自己的环境的话，也可以使用大家自己的；

2) 配置环境：

a、安装jdk；

b、解压缩tomcat，并进行基本配置

c、解压缩maven，并进行基本配置；

d、安装vscode；

e、解压缩idea，破解；

f、如果需要使用数据库的话，安装mysql并配置，安装navicat并破解；-----这个不在课堂上统一讲了，大家课后自行完成，有问题可以找我。

3、我讲解echarts的基本使用：

1) echarts的基本使用;

a、菜鸟教程: <https://www.runoob.com/echarts/echarts-tutorial.html>

b、在vscode中创建项目, 引入那两个js文件, 简单介绍教程中的基本例子并演示;

b1、第一个例子

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>第一个 ECharts 实例</title>
  <script src="../js/jquery.min.js"></script>
  <script src="../js/echarts.min.js"></script>
</head>
<body>
  <!-- 为ECharts准备一个具备大小（宽高）的Dom -->
  <div id="main" style="width: 600px;height:400px;"></div>
  <script type="text/javascript">
    // 基于准备好的dom, 初始化echarts实例
    var myChart = echarts.init(document.getElementById('main'));

    // 指定图表的配置项和数据
    var option = {
      title: {
        text: '第一个 ECharts 实例'
      },
      tooltip: {},
      legend: {
        data:['销量']
      },
      xAxis: {
        data: ["衬衫", "羊毛衫", "雪纺衫", "裤子", "高跟鞋", "袜子"]
      },
      yAxis: {},
      series: [{
        name: '销量',
        type: 'bar',
        data: [5, 20, 36, 10, 10, 20]
      }]
    };

    // 使用刚指定的配置项和数据显示图表。
    myChart.setOption(option);
  </script>
</body>
</html>
```

b2、其它基本例子, 参考菜鸟教程的简单演示及说明;

2) 纯前端可视化: 创建html页面, 直接使用jQuery读取本地数据、使用echarts进行可视化;

```
<!DOCTYPE html>
<html>
<head>
```

```

<meta charset="utf-8">
<title>第3个 ECharts 实例</title>
<script src="../../js/jquery.min.js"></script>
<script src="../../js/echarts.min.js"></script>
</head>
<body>
  <!-- 为ECharts准备一个具备大小（宽高）的Dom -->
  <div id="main" style="width: 600px;height:400px;"></div>
  <script type="text/javascript">
    // 基于准备好的dom，初始化echarts实例
    var myChart = echarts.init(document.getElementById('main'));
    myChart.showLoading();

    $.get("../data/test03.json", function(data){
      myChart.hideLoading();
      myChart.setOption({
        series:[
          {
            name:'访问来源',
            type:'pie',
            radius:'55%',
            data:data.data_pie
          }
        ]
      })
    })
  </script>
</body>
</html>

```

3) 留作业一

a、参考课堂例子，自己编一个数据（或者找一个数据），使用jQuery读取本地数据的方式，完成一个简单可视化代码；

b、交word文档（命名要求：学号-班级-姓名-作业1.docx）；

b1、把html源代码拷贝到word文档中（注意是直接拷贝源代码，不要源代码截图）；

b2、把浏览器页面执行结果，截图粘贴到word文档中；

4、在idea中创建maven的web项目，前端使用使用jQuery访问后端获取数据；

1) 参考依赖如下

```

<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.23</version>
</dependency>

<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>4.0.1</version>
  <scope>provided</scope>

```

```

</dependency>

<dependency>
    <groupId>jstl</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>

    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>1.18.30</version>
        <scope>provided</scope>
    </dependency>

<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.11.0</version>
</dependency>
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.14.0</version>
</dependency>

```

2) Java相关代码:

先创建两个实体类:

```

package com.neuedu.po;

import lombok.Data;

@Data
public class Unit {
    private String name;
    private Integer[] values;
}

```

```

package com.neuedu.po;

import lombok.Data;

import java.util.List;

@Data
public class BodyHealth {
    private List<String> legends;
    private List<String> indicators;
    private List<Unit> units;

}

```

创建Servlet, 读取本地数据 (或者数据库表中数据) , 返回给前台页面;

```
package com.neuedu.servlet;

import com.fasterxml.jackson.databind.ObjectMapper;
import com.neuedu.po.BodyHealth;
import com.neuedu.po.Unit;
import org.apache.commons.lang3.StringUtils;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.ArrayList;

@WebServlet("/body_health")
// http://localhost:8888/temp_web/body_health
public class TestServlet_01 extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        resp.setContentType("text/html;charset=utf-8");
        PrintWriter out = resp.getWriter();

        // 读取文件 (这里读取本地文件了, 大家也可以把数据存储到数据库中, 然后使用jdbc访问数据
        库获取数据)
        FileReader fileReader = new
FileReader("d:/io/data/data_01_bodyhealth.txt");
        BufferedReader bufferedReader = new BufferedReader(fileReader);
        BodyHealth bh = new BodyHealth();

        ArrayList<String> LegendList = new ArrayList<>();
        ArrayList<String> IndicatorList = new ArrayList<>();
        ArrayList<Unit> units = new ArrayList<>();

        String line = bufferedReader.readLine();
        String[] items = line.split(",");
        for (String legend : items) {
            LegendList.add(legend);
        }
        bh.setLegends(LegendList);

        line = bufferedReader.readLine();
        items = line.split(",");
        for (String indic : items) {
            IndicatorList.add(indic);
        }
        bh.setIndicators(IndicatorList);
    }
}
```

```

        line = bufferedReader.readLine();
        while (!StringUtils.isEmpty(line)) {
            Integer[] data = new Integer[7];
            Unit unit = new Unit();
            items = line.split(",");

            unit.setName(items[0]);
            data[0] = Integer.parseInt(items[1]);
            data[1] = Integer.parseInt(items[2]);
            data[2] = Integer.parseInt(items[3]);
            data[3] = Integer.parseInt(items[4]);
            data[4] = Integer.parseInt(items[5]);
            data[5] = Integer.parseInt(items[6]);
            data[6] = Integer.parseInt(items[7]);
            unit.setValues(data);
            units.add(unit);

            line = bufferedReader.readLine();
        }
        bh.setUnits(units);
        bufferedReader.close();
        fileReader.close();

        ObjectMapper om = new ObjectMapper();
        out.print(om.writeValueAsString(bh));
        out.close();
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {
        doGet(req, resp);
    }
}

```

3) 创建html页面，通过ajax访问后台取得数据并可视化;

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>ECharts</title>

    <script type="text/javascript" src="js/jquery.min.js"></script>
    <script type="text/javascript" src="js/echarts.min.js"></script>
</head>

<body style="width: 100%;height:800px;margin-left:0px;margin-top:0px">

    <!-- 为ECharts准备一个具备大小（宽高）的Dom -->
    <div id="main" style="height: 100%"></div>

</body>
</html>

```

```

<script type="text/javascript">
    var myChart = echarts.init(document.getElementById('main'));
    window.addEventListener('resize', function () {
        myChart.resize();
    });
    // 显示标题，图例和空的坐标轴
    myChart.setOption({
        title: {
            text: 'Body part of Someone',
            textStyle: {
                color: '#333',
                fontSize: 30
            }
        },
        series: [{
            name: 'Something',
            type: 'radar',
            data: []
        }]
    });

    myChart.showLoading();    //数据加载完之前先显示一段简单的loading动画

    $.ajax({
        type: "post",
        async: true,          //异步请求（同步请求将会锁住浏览器，用户其他操作必须等待请
求完成才可以执行）
        url: "body_health",    //请求发送到TestServlet处
        data: {},
        dataType: "json",      //返回数据形式为json
        success: function (result) {
            //请求成功时执行该函数内容，result即为服务器返回的json对象
            if (result) {
                myChart.hideLoading();    //隐藏加载动画
                myChart.setOption({      //加载数据图表
                    legend: {
                        data: result.legend
                    },
                    radar: [{
                        //indicator:["Head","Thorax","Left arm","Right
arm","Stomach","Left leg","Right leg"]
                        indicator: result.indicators.map(function (a) {
                            return {name: a}
                        })
                    }],
                    series: [{
                        // 根据名字对应到相应的系列
                        type: 'radar',
                        name: 'Something',
                        data: result.units.map(function (a) {
                            return {value: a.values, name: a.name}
                        }),
                        showBackground: true,
                        backgroundStyle: {
                            color: 'rgba(220, 220, 220, 0.8)'
                        }
                    }
                ]
            }
        }
    });

```

```

        },
        emphasis: { //鼠标悬停线条变宽
            lineStyle: {
                width: 8
            }
        },
        label: { //显示图表元素
            show: true,
            formatter: function (params) {
                return params.value; //显示图表元素对应数值的值
            }
        }
    }
}
]
});
}
},
error: function (errorMsg) {
    //请求失败时执行该函数
    alert("图表请求数据失败!");
    myChart.hideLoading();
}
});
</script>

```

3) 留作业二

- a、参考课堂例子，自己编一个数据（或者找一个数据），完成一个web项目的简单可视化代码；
- b、交word文档（命名要求：学号-班级-姓名-作业2.docx）；
 - b1、把Servlet源代码、html源代码分别拷贝到word文档中（注意是直接拷贝源代码，不要源代码截图）；
 - b2、把浏览器页面执行结果，截图粘贴到word文档中；

第2次课：

- 1、继续完成之前内容讲解；
- 2、大家做实验，我答疑及处理问题；

第3次课：

- 1、大家做实验，我答疑及处理问题；

第4次课：

- 1、大家做实验，我答疑及处理问题；
- 2、检查大家的进度，至少该完成2个页面了；

第5次课：

- 1、大家做实验，我答疑及处理问题；

第6次课：

- 1、大家做实验，我答疑及处理问题；
- 2、检查大家的进度，至少该完成6个页面了；
- 3、建议大家今晚完成的任务：
 - a、完成实验报告文档；
 - b、完成答辩ppt文档；
 - c、源代码、演示视频可以明天整理；

第7次课：

- 1、大家做实验，我答疑及处理问题；
- 2、看人数，如果人数较多的话，就开始答辩演示：
 - 1) 演示过程（事先把环境准备好，后端服务器、前端服务器都打开）：
 - a、先介绍ppt；
 - b、再演示各个可视化页面，并做简单说明；
 - 2) 演示方式：看实际情况，可以腾讯会议共享桌面，也可以到前面连接投影仪演示；
 - 3) 我进行提问（也可能不问）；
- 3、收集成果物（大家把最终压缩包文件发到qq群，我好及时处理），成果物要求如下：

- a、项目源代码（打压缩包交，命名规则：学号-班级-姓名-个人源代码.zip，如 12345-大数据2班-王大瑞-个人源代码.zip）；
- b、演示录屏（命名规则：学号-班级-姓名-演示录屏.mp4，如 12345-大数据2班-王大瑞-演示录屏.mp4）
 - 1) 要求必须是mp4格式的，不要声音，文件大小控制在30M以内；
 - 2) 建议录制内容参考：
 - a、启动后端项目（如果需要的话，如果没有后端项目可以略过此步骤）
 - b、启动前端项目（如果需要的话，如果没有前端项目或中前端项目不需要专门启动，可以略过此步骤）
 - c、访问各个可视化页面（稍微停顿一下，要给人看的时间）
- c、答辩ppt（命名规则：学号-班级-姓名-答辩ppt.pptx，如 12345-大数据2班-王大瑞-答辩ppt.pptx）
 - 1) 参考内容如下（不是必须这样）：
 - a、封面页；
 - b、目录页；
 - c、项目或者需求或者实验内容介绍；
 - d、使用技术简介；
 - e、完成的主要页面截图（页面效果截图，本次至少8页可视化页面截图）；
 - f、技术亮点介绍；
 - g、收获与总结；
- d、实验报告（使用学校的模板，命名规则：学号-班级-姓名-实验报告.doc，如 12345-大数据2班-王大瑞-实验报告.doc）；
- e、以上4个成果物再放到一个文件夹中，打压缩包交（参考命名：学号-班级-姓名-所有成果物.zip）；

第8次课:

- 1、答辩演示;
- 2、收集成果物 (大家把最终压缩包文件发到qq群, 我好及时处理) ;
- 3、注意: 不能提前下课;