

# 报告

章逸冰

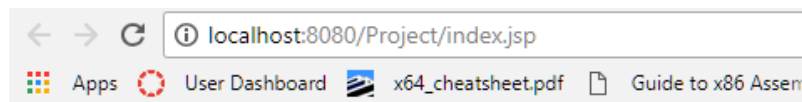
8/6/2018

## 1 概要

该项目是基于 Java 的 Web 应用程序，具备 1. 读取 data.dat 文件，解析成格式化数据并在界面展现，2. 支持观测要素的选择，然后按照采集时间从远及近动态展示数据，以温度为默认要素，3. 选取一个要素为 x 轴，查看各个观测要素与之的关系图 4. 以多图像形式展现各个观测要素与时间的关系图 这四个功能。在开发过程中，即运用了 jQuery 和 jsp 完成前端设计，servlet 作为服务端程序，XMLHttpRequest 和 JSON 实现前后台数据交互；在实现数据可视化方面，即运用了 ECharts 和 DataTable 作为可视化控件。

## 2 使用说明

首先，打开浏览器，输入访问该程序的网址



### 功能1：读取文件并查看所有数据

选择想要的文件来查看数据

data.dat ▼ 查看格式化数据

### 功能2：观测要素动态展示

### 功能3：观测要素之间的关系图

### 功能4：观测要素和时间的关系图

### [查看报告](#)

进入到主界面以后可以自由选择查看各个功能，每个功能都会以新窗口的方式打开。以下是各个功能的介绍。

2.1 查看格式化数据

选择所要查看的文件名字，点击查看按钮以后即可查看对应文件的所有格式化数据。格式化数据以表格的形式呈现，运用了 **DataTable** 控件来提高可读性，左上角下拉框可以选择当前页面的显示纪录条数，右下角支持上下页切换以及右上角支持简单的查询纪录（由于数据列数太多，以下截图十分模糊，实际参考程序界面）。

AppsUser Dashboardx64\_cheatsheet.pdfGuide to x86 AssemProblems - LeetCodfailed to load selinCS 4104: HomeCS3214 ComputerSRichard J Quintin / CList of LaTeX mathW3Schools OnlineVHTML 框架HTML 图像\_w3csch»

Show10entries

Search:

接驳盒编号	仪器编号	温度 (°C)	电导率 (S/m)	深度 (米)	第一外部电压 (volts)	第二外部电压 (volts)	浊度 (NTU)	溶解氧 (umol/kg)	采集时间
DH-JB-003	10903	19.1043	4.52675	40.617	0.1258	0.7565	25.318	0.773051	18 Jul 2018 00:02:31
DH-JB-003	10903	19.101	4.52677	40.588	0.1282	1.9332	25.329	0.77312	18 Jul 2018 00:03:21
DH-JB-003	10903	19.0968	4.52622	40.682	0.1226	1.8306	25.338	0.773219	18 Jul 2018 00:04:11
DH-JB-003	10903	19.0935	4.52598	40.636	0.1296	0.9515	25.349	0.77332	18 Jul 2018 00:05:01
DH-JB-003	10903	19.0898	4.52611	40.574	0.131	1.0745	25.359	0.773402	18 Jul 2018 00:05:51
DH-JB-003	10903	19.084	4.52556	40.57	0.1276	2.3351	25.369	0.773527	18 Jul 2018 00:06:41
DH-JB-003	10903	19.0789	4.52492	40.67	0.1201	2.1986	25.382	0.773629	18 Jul 2018 00:07:31
DH-JB-003	10903	19.0763	4.5245	40.61	0.1185	1.7746	25.383	0.773717	18 Jul 2018 00:08:21
DH-JB-003	10903	19.0726	4.52421	40.585	0.1197	1.006	25.401	0.773793	18 Jul 2018 00:09:11
DH-JB-003	10903	19.0693	4.5237	40.577	0.1171	0.559	25.406	0.773868	18 Jul 2018 00:10:01

Showing 1 to 10 of 1,614 entries

Previous12345...162Next

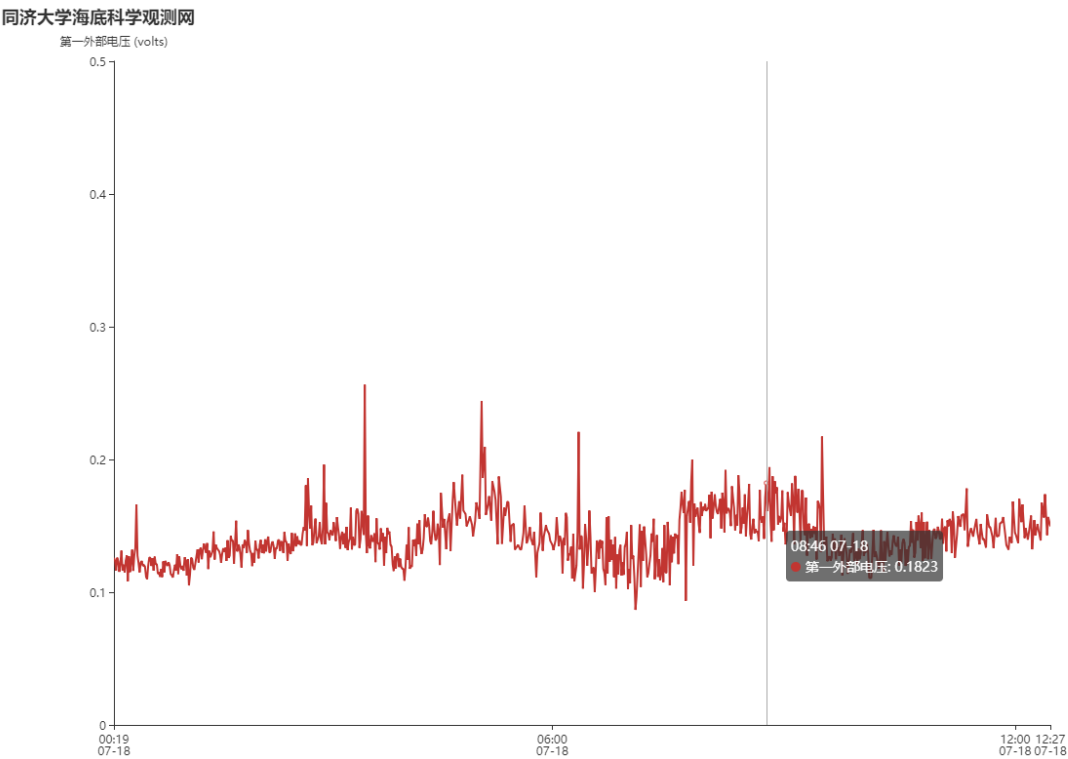
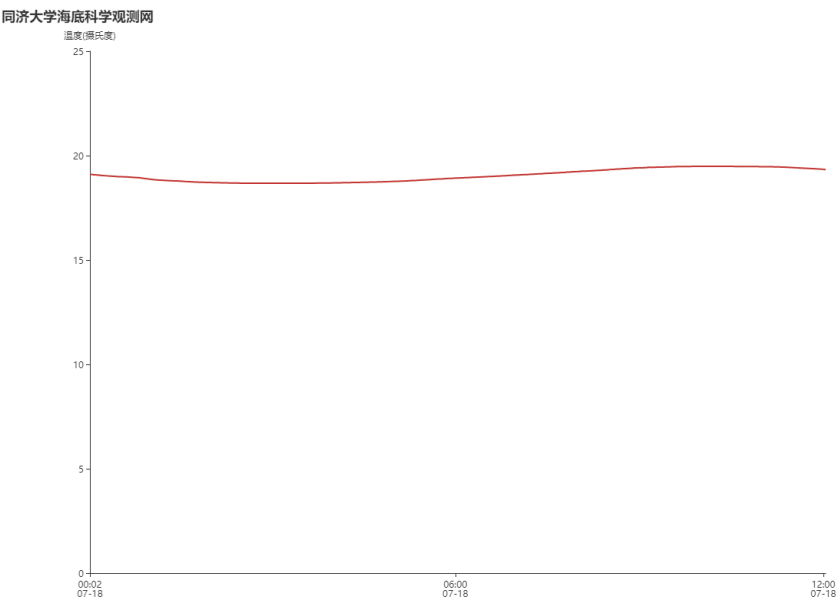
2.2 观测要素动态展示

该功能默认用户选择温度进行观测，也可后期用户自由选择。在进入页面之后，点击左上角的下拉框，选取所要查看的要素，再次点击确

定后即可动态查看指定数据。

选取一个要素为横坐标

温度



### 2.3 选取任一要素作为横坐标，查看其他要素与之的关系图

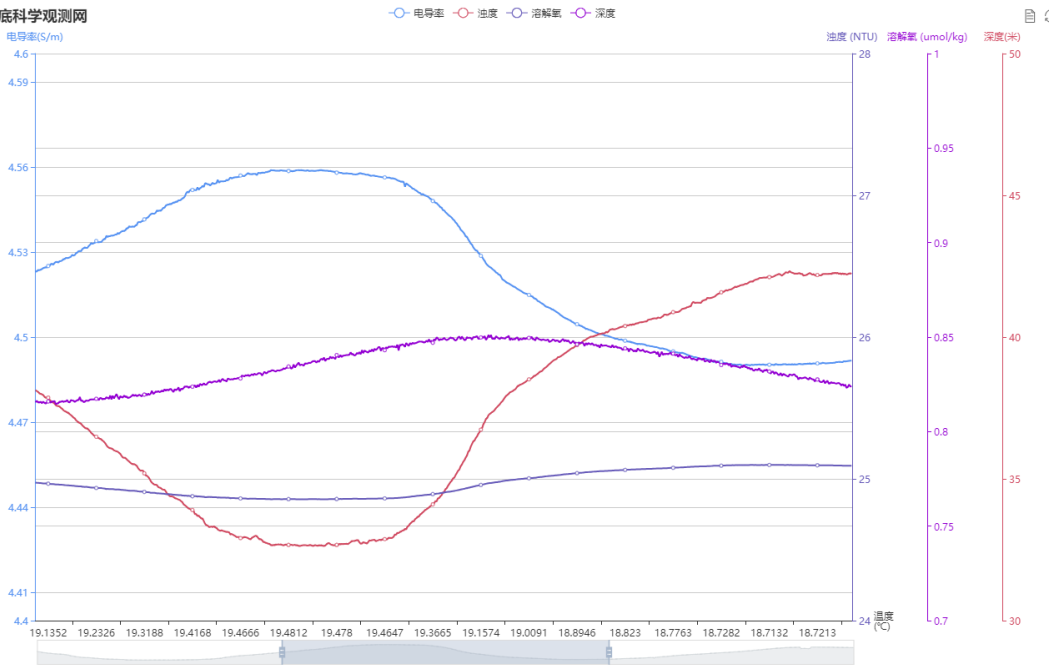
该功能要求用户必须选择两个观测要素，选择单个或者多个皆不可，选择单个会提醒用户，选取两个要素后会自动禁选剩余要素（jQuery 应用），下一步点击按钮即可查看。

在观测过程中，用户可以选择隐藏某一系列数据，点击图像上方的图例进行隐藏，再次点击即可恢复，也可以点击右上角的还原按钮让图像恢复成初始状态。也可以点击下载按钮保存成图像或直接查看格式化数据。图像下方还配备了数据缩放功能，可自由缩放数据。

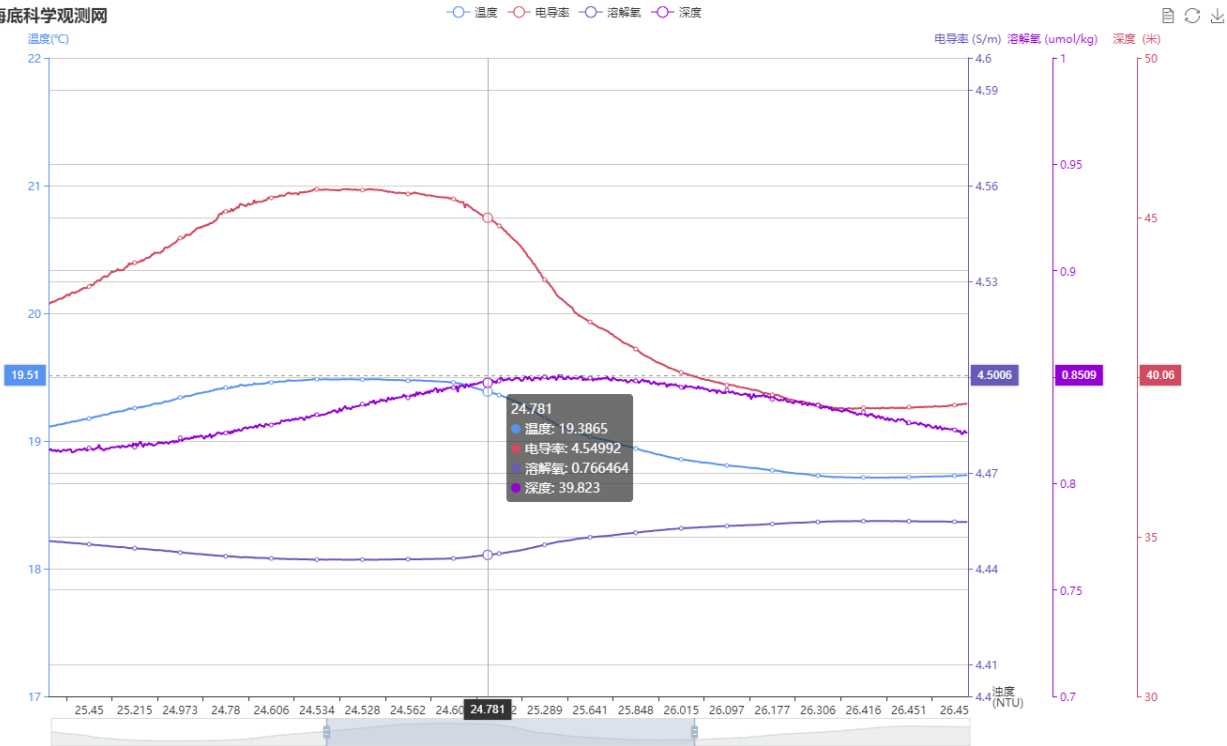
选取一个要素作为X轴

温度 提交并查看数据

同济大学海底科学观测网

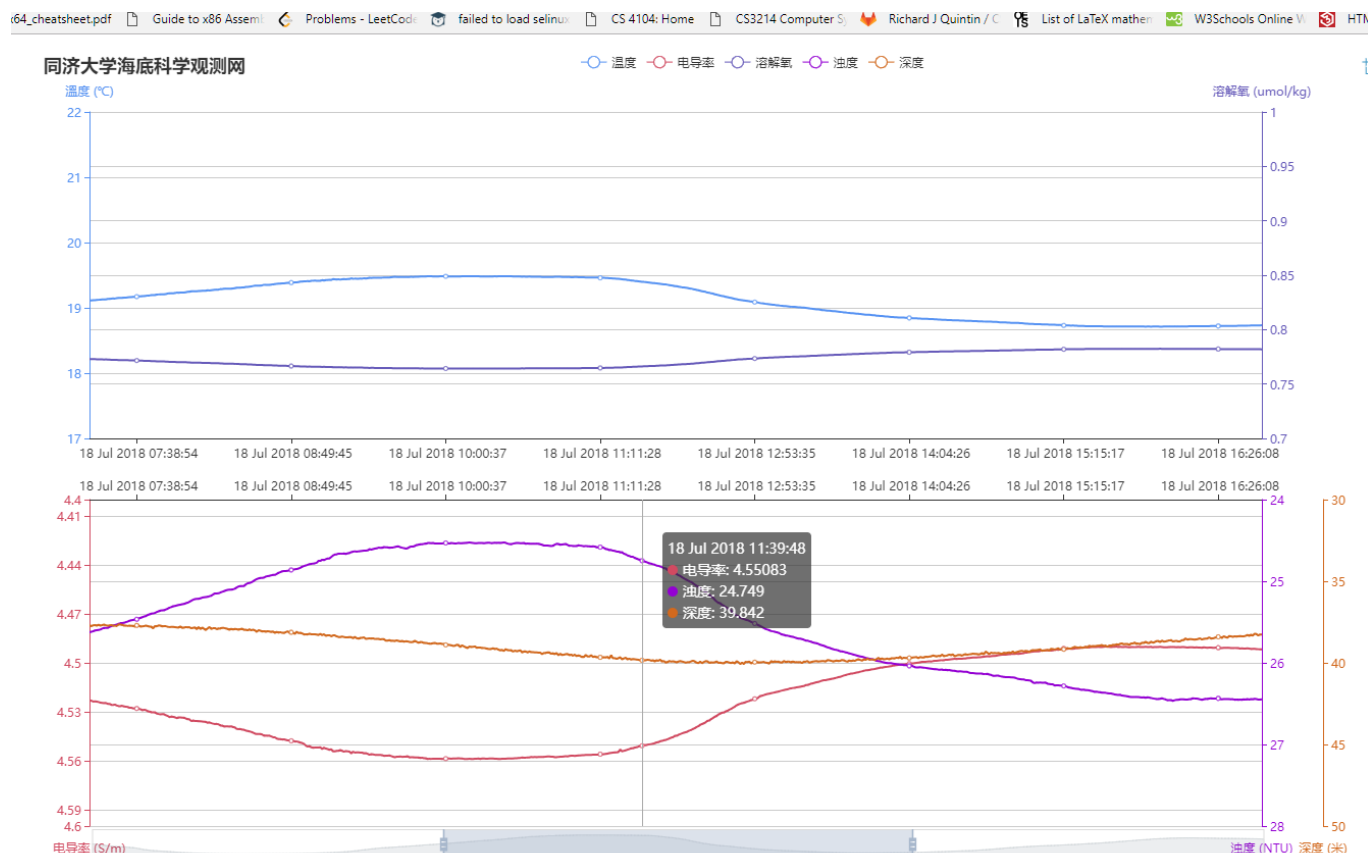


同济大学海底科学观测网



## 2.4 以多图像形式展现各个要素与时间关系

此功能无需用户选择，点击即可查看所有要素与时间的关系。功能三支持的数据缩放，下载图像，隐藏数据等功能也都具备。



## 3 历程

刚拿到这个项目的时候，发现要开发一个 Web 应用程序，我其实挺陌生的，对 Web 应用程序并不了解，一开始还以为是要写个 Web 网站，于是我就去网上查 Web 应用程序。在查阅过程中，我才了解到 Web 应用程序和 Web 是不同的，还看到一位 CSDN 博主 life is wonderful 写的一系列“JAVA Web 学习路线”[1]。他上面所用的是 Tomcat 服务器，我没用过 Tomcat，只用过 maven，就按照上面的指示安装[1]（这一内容其实很简单，可以忽略不计）。在安装好 Tomcat 以后，我就开始着手整个 project 的头绪工作。

### 3.1 阅读 data 文件，思考如何解剖数据

通过阅读文件，发现每一串数据占一行，后面跟随了一行短一点的 `<Executed>` 代号，我要提取数据，可以直接使用 java 里的 `split` (分隔符为 " } " )，并根据切分出来的 `array` 的长度 (大于 2) 来判断这是否是完整的数据 (像 `<Executed/>{DH-JB-003,10903,180718002926}` ts 切出来的 `array` 长度是 2，我们就可以立马排除了，有的残缺的数据也无法通过这一判断)。再要把每一个要素对应的数据提取出来就很容易了，可以对 `array` 里的第一个和最后一项再进行分隔符为 “ , ” 的 `split`。我就在 `eclipse` 上面又自己写了一个简单的程序来验证我的想法，在 `console` 上面打印提取出来的数据来和原文件里的数据做对比。验证完成后，第一步准备工作就完成了 (另外还需要考虑如何将数据存储和以什么样的 `type` 存储，我的存储方式是 `arrayList<DataDemo>`，`type` 是 `DataDemo`，里面包含了各个要素)。

### 3.2 设计前端界面

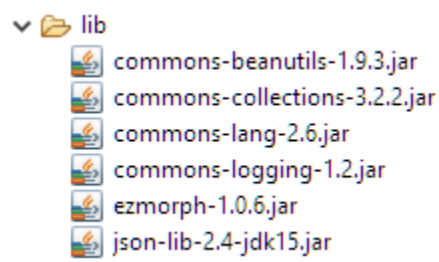
完成前端的界面，Java Web 主要是写 `jsp` 文件。通过阅读前面提到的几篇文章下来，我发现写 `jsp` 文件和写 `html` 可以说是一样的，两者几乎没有区别，唯一的区别就是开头的 `<%@page import` 等。由于我以前自学过 `HTML`，完成前端界面样式的设计是没有问题的。

### 3.3 完成第一功能和解决数据传输

第一个功能中的难点就在于数据传输，别的基本都没有问题。数据传输问题的解决其实也算是运气好吧，我没有花特别多的时间去查阅资料，我就写到往前台传数据的时候，卡壳了。正好有别的事情要忙，忙完回来，我先在 `YouTube` (需翻墙) 上看了一会关于 `Java Web` 的学习资料，发现了 `Servlet` 这个“新大陆”，可以接收和回复前台发来的信息 (我留学时有养成平时在 `YouTube` 上看学习资料的习惯，也算是运气不错，看了一会，于是就想着使用 `servlet` 来把这个问题)。在 `DataServlet class` 里面把 `inner HTML` 的 `string` 先填写好，往前台一传送，一个以 `table` 形式展现所有数据的功能就完成了。因为我在手机 `app` 上的看，所以具体链接不好给出来了。

### 3.4 第二第三功能的完成和再次解决数据传输问题

第三第四功能的完成主要是依靠 ECharts 实现动态数据展示和要素观测。在这里面，ECharts 的 API 和官网上的实例很简洁明了，花点时间就可以读懂并进行后期的数据填充。一开始最主要的和最难的还是数据传输（两个功能的数据传输原理一致但与第一功能的数据传输不同）我查阅了很多 ECharts 资料，包括去查阅别人写过的项目，各个博主上面写的教程[2][3][4][5]（这里都是稍微有能给一点方向的资料，有些残缺质量差的资料我就没有收录在内）。各式各样，有的使用 database 传输数据，有的使用 ajax，有的使用 HTTP POST/GET，五花八门，甚至有些资料都是残缺的。但这些基本可以说是仅供参考，并不能起到实际的作用，可以说是徒劳。我思考过后，而且因为这个文件数据不多，遂决定使用 JSON 通过 XMLHttpRequest (GET) 进行前后台交互（观测要素数据的传输和接受）。凑 JSON String 的时候也碰到过一些格式的问题[6]，甚至是下载的 JSON jar 包不齐全的问题[7]。为了确保 JSON 这一块能完美无误，我先手动 hard code 一串简短的 JSON 要素数据发送至进行测试，调试结束后续再加入完整的要素数据。至于提到的安装包不齐全的问题，刚下载完 jar 包，build 进 path 后没有问题，但是运行的时候一直出错。我就把错误信息往网上一传，查了好一会才发现原来是缺少以下几个关键的 jar 包[7]。



解决完 JSON 传输的问题后，主要就是阅读 ECharts 的官方实例[8]和官方教程[9]，通过官方教程了解到我可以先空置图表里的数据，后期使用 setOption 来完成图表的数据填充。下一步就是选择一个图像的种类，花费时间理解官方实例，并适应自己的代码进行改写。像第二个功能我选取的是 multiple-y-axis 图像[10]，第三个功能选取的是 Dynamic Data+ Time Axis[11]。理解完他们的代码以后，我把数据，标题，坐标格式改好，就完成了。

### 3.5 第四功能的开发及思考



我在编写这个功能前，先仔细阅读了 ECharts 官网的，发现有个雨量流量关系图实例[12]，是有关降水量，流量和时间的关系图。我思考以后，觉得这个图像作为数据分析很不错，遂开发了第四功能。

### 3.6 工作重整和新的难点

我于 7.31 号拿到这个 project，8.1 号正式开始写，8.4 号完成第一份程序。但是上文使用说明书中的样式其实是最后才敲定的，在听取张老师的意见后重新改进了此程序。8.4 号完成第一份程序以后，我交给张老师查阅，张老师在审视了后于 8.5 号给出了几点意见：

1. 功能一中表格可进行美观优化，方便查询
2. 功能三，取消原定让用户选择两个元素，只让用户选择一个元素作为横坐标，显示其他要素与之的关系
3. 功能四同功能三类似，所有要素皆应显示

于是，在 8.5 号晚上我先开始修改第二和第三点意见。第二点意见修改很快，一会功夫就改好了。但是编写第三个碰到了两个大问题。第一个问题在于如何初始化多个 Y 轴，ECharts 需要初始化的时候需要特定的格式，一旦出错，图像就无法呈现。我查阅了很久官网的文档和实例，都没有发现踪迹，遂自己尝试，再经过多番尝试之后，终于发现我必须要把上下两个图的 Y 坐标初始化好以后才可以再添加其他的 Y 轴，哪怕是先把上一部分的 Y 轴初始化好，再管下一部分的 Y 轴都不行（因为总共有五个要素，我先添加好上图像的 2 个 Y 轴再添加下图像的 3 个 Y 轴都会报错，这一难题的解决完全是靠自己试出来的，他官网也没有有关这方面的任何介绍）。

第二个问题在于如何插入数据，插入好数据后不定义或者定义错其 x 和 Y 轴应处的位置，Y 轴的刻度不能够显示。解决这一问题的方法我也是多番尝试，找不到任何的办法，而且他位置的定义和一般正常计算机编程有悖，具体的我就不在这介绍了，也没有意义，不过也可以作为以后开发 ECharts 的参考。

在写完第四个功能以后，我想起功能三里没有数据缩放，觉得数据缩放很有必要，便往功能三里也加了它，因为起初功能三是不支持数据缩放的。后来我也觉得不同数据应该有不同的刻度，ECharts 默认的刻

度规划很难看出数据的起伏，尤其是一组小数据和一组大数据同时出现的时候，比如说浊度和电导率在同一坐标体系内的话，ECharts 会以浊度的数据为准，而电导率的数据和浊度比起来小太多，ECharts 就会把电导率的数据放的很靠近横坐标，看不出数据起伏，没有任何观测价值。于是我就给不同的数据组设置了不同的刻度，提高观测价值。

在 8.6 号，我开始改进最后一点要求（也就是第一点要求）。这一要求需要查阅大量资料，找寻可用的控件，起初我不知道改进这一要求需要多少时间，遂将其放于最后。在查询多个资料以后，我终于发现一个叫 DataTable[13]的控件，可以让网页里的表格变得简单美观，支持查询，分页。该控件功能很强大，运用起来也很方便，后续如果要开发类似程序我还会继续用它。而且他官网也可以访问，访问速度很快（U. S. MIT licensed），不像有的外国网站都被封杀了，不能打开，而且上面也有很多例子，我再阅读其大部分例子以后，很快便把这一功能写好了。其中唯一的难题就是自己如何去查阅，查阅后分析各个插件，思考它们的运用价值，从中选取一个。

## 4 总结

总的来说，该程序也算是一个我的小挑战和小测验，更像是一次大学生活的重温。和在大学里面上学的模式类似，老师只抛出一些概念和他想要的结果，实现的过程要我们自己思考去查阅相关资料，绞尽脑汁去思考如何开始程序，如何实现结果，用什么方法和什么语言去写，最后写完交给老师。

但是其中，我也收获了不少新知识，从一开始的不懂，到最后查阅各个资料完成这个程序，习得各个知识，像 JavaWeb，servlet(原理和 HTTP 的 get, post 类似，但我这方面也还欠缺，算是获得了一个新技能)等等；也有让之前自学的知识有所巩固，比如界面的设计和书写，以及 jQuery 的运用。收获颇丰，在编写过程中既提高了自己解决问题的能力，也锻炼了自己的心态，为将来工作做准备。

但是对于编写的时间，我感觉有点长了。期间总是会受各种事情打扰，没法专心投入学习和编写，如果可以专心静下来写的话，应该可以写的更快一点(不过毕竟是生活嘛，总会有各个小插曲)。还有就是，我感觉到国内编程学习环境还比不上国外，有些问题百度上给出

的答案参差不齐，而且真正有质量的能解决问题的隐藏其中，还是谷歌和 stackoverflow 能更快更准确的解决问题（像 json 的 jar 包问题以及 js, ajax 等语言的 api 百度出来的基本都不全，有的只是只言片语）。

## 5 后续工作

鉴于当前这个项目只是实现了功能上的运用，我觉得后续工作还可以做的更多。

首当其冲，前端的界面设计和样式设计，后续需要添加 css 文件来进行前端美化，也可以使用 bootstrap 语言来进行界面设计（省去自己填写 css）。张老师也提到把所有图像集中在一张页面上，设置锚点方便内容跳转。其实，这样子写的话我代码还可以更简单点，不用考虑各个页面数据的传递，写起来更简单一点。这也算是以后改进的一个小方向。

其二，前台下拉项的文件名当前都是程序员自己设定，后续为了达到该程序的实用性，可以使用遍历磁盘内容来加入加载文件名字。

其三，后台数据传输以及文件读取数据存放可以使用 database 进行存放，以及使用 cache 来缓存某一处理过的文件内容来提高整体项目对于大型数据的处理效率。

其四，还需要添加除 localhost 以外的访问形式，如添加到服务器里，我对这个不太清楚，还需要去学习而且我也很想学习这一方面的知识。

其五，图像中加入更多的观测要素，开发更多经典或流行的数据分析的图像类型，比如说雷达图等等。还有很多功能可以开发，比如说加入异步等等，我在此就不一一列举了。

## 6 个人分享

这一内容纯属技术性内容，如果没有兴趣可以略过。我想分享这个内容也是为了方便以后其他人碰到类似的问题无从下手的时候，可以读一下以下内容。

在初稿的时候，我有用到 `checkbox` 来获取用户所有选择的值。起初我使用的是 `for object` 循环来获取，但是这一方法只适用于 `Chrome` 浏览器，`IE` 浏览器无法使用，但我将 `for object` 循环换成普通的循环并根据 `checkbox` 的长度来循环，这问题就解决了。

```

for (var k=0;k<obj.length;k++) {
    if (obj[k].checked) {
        message+=obj[k].value+",";
        switch (obj[k].value) {
            case "temperature":
                lName.push('温度');
                break;
            case "conduct":
                lName.push('电导率');
                break;
            case "depth":
                lName.push('深度');
                break;
            case "volts1":
                lName.push('第一外部电压');
                break;
            case "volts2":
                lName.push('第二外部电压');
                break;
            case "NTU":
                lName.push('浊度');
                break;
            case "oxy":
                lName.push('溶解氧');
                break;
        }
    }
}

```

```

var message="";
for (k in obj) {
    if (obj[k].checked) {
        message+=obj[k].value+",";
        switch (obj[k].value) {
            case "temperature":
                lName.push('温度');
                break;
            case "conduct":
                lName.push('电导率');
                break;
            case "depth":
                lName.push('深度');
                break;
            case "volts1":
                lName.push('第一外部电压');
                break;
            case "volts2":
                lName.push('第二外部电压');
                break;
            case "NTU":
                lName.push('浊度');
                break;
            case "oxy":
                lName.push('溶解氧');
                break;
        }
    }
}

```

## 参考文献

[1] JAVA 学习路线

<https://blog.csdn.net/erlian1992/article/list/1>

[2] <http://www.codeweblog.com/%E5%9C%A8java-web%E5%BC%80%E5%8F%91%E4%B8%AD%E5%AE%9E%E7%8E%B0echarts%E5%92%8C%E5%90%8E%E5%8F%B0%E7%9A%84%E4%BA%A4%E4%BA%92/>

[3] <https://blog.csdn.net/isea533/article/details/43225717>

[4] <https://www.cnblogs.com/hemiy/p/6228623.html>

[5] <https://github.com/Dairongpeng/echarts>

[6] <https://www.cnblogs.com/wangf-keep/p/6480019.html>

[7] [https://blog.csdn.net/gu\\_gu\\_/article/details/50551775](https://blog.csdn.net/gu_gu_/article/details/50551775)

[8] <http://echarts.baidu.com/examples/>

[9] <http://echarts.baidu.com/tutorial.html#5%20%E5%88%86%E9%92%9F%E4%B8%8A%E6%89%8B%20ECharts>

[10]

<http://echarts.baidu.com/examples/editor.html?c=multiple-y-axis>

[11] <http://echarts.baidu.com/examples/editor.html?c=dynamic-data2>

[12] <http://echarts.baidu.com/examples/editor.html?c=grid-multiple>

[13] <https://datatables.net/>