

互联网程序设计前端部分实验报告

姓名：李文驰

学号：2015118140

日期：2017年10月29日 15:24:52

一、计算器

| | | | | | |
|---|---|---|-----|----|---|
| 1 | 2 | 3 | + | - | (|
| 4 | 5 | 6 | * | / |) |
| 7 | 8 | 9 | ^ | √(| 1 |
| 0 | . | C | del | = | 1 |

计算器要求实现一个比较简单的计算器要求有加减乘除和乘方，开根号等基础功能。代码比较简单，核心代码就是获取到计算表达式以后，用JavaScript自带的eval()函数进行表达式的计算，这一点相比传统语言来说可以说是相当方便了，想当年有一道ACM现场赛的题目就是要求计算一个表达式，如果当时支持Python或JavaScript语言可以说是一道签到题了。

```
if(this.value == '=') {  
    var exp = (expression.value).replace(/√/g, 'Math.sqrt')  
    exp = exp.replace(/\^/g, '**')  
    try {  
        expression.value = eval(exp)  
    } catch (EvalError){  
        expression.value = 'error'  
    }  
}
```

二、日历



关于日历我并没有动态生成那么多的HTML代码块（将近40个），反正是静态div我直接复制粘贴了，而且我并不认为在这种需求下直接写HTML是低级行为。灰色文字代表无法选择黑色的文字代表可以选择，且绿色方框代表当前选择，橘色方框表示鼠标悬停在上方。且日历最上端的年份和月份也可以随时调整，年份范围为当前年份正负30年。

我觉得在这个小实例里面，根据老师的要求，比较重要的就是用户对当前日期的选择，同时要记住过去的时间不能选择，还有一些良好的交互，比如鼠标放到上面，方框变成橘色。

```

1  //每个月有多少天
2      month_day = [31,28,31,30,31,30,31,31,30,31,30,31]
3  //获得当前日期
4      var date = new Date()
5  //当前的年月日以及礼拜几
6      var initdata = {
7          year: date.getFullYear(),
8          month: date.getMonth() + 1,
9          day: date.getDate(),
10         week_day: date.getDay() == 0 ? 7 : date.getDay(),
11     }
12 //目前选择的日期
13     var nowdata = {
14         year: initdata.year,
15         month: initdata.month,
16         day: initdata.day,
17     }
18 //获得用户选择的年份
19     var div_year = document.getElementsByClassName('year')[0]
20     div_year.innerText = initdata.year
21 //获得用户选择的月份
22     var div_month = document.getElementsByClassName('month')[0]
23     div_month.innerText = initdata.month

```

那么如何记录用户的选择是很重要的，还有比较重要的一点就是知道这个月的第一天是星期几

```

date = new Date()
date.setFullYear(parseInt(div_year.innerText), parseInt(div_month.innerText)-1, 1)

```

还有特殊年份的判断

```

var isRunYear = div_year.innerText
if(((isRunYear%4==0)&&(isRunYear%100!=0))||isRunYear%400==0) {
    month_day[1] = 29
}

```

三、面向对象的选课

题目要求很模糊，只说了每个按钮的功能，其实对面向对象接触不多，大二前暑假用C#基于Unit3D写了一个五子棋的游戏，当时是结合网上的教程一点点码出来的，完全的面向对象，当时搞得我真的头疼，当时算是颠覆了我的编程观念.....因为我一直写的是ACM的竞赛算法题，基本就是30~150行的算法题，完全的面向过程。所以我现在的“面向对象”其实就是把对象当成结构体用了.....（这里真的说实话）。

```
1  var teachers = []
2  var students = []
3
4  class Person {
5      constructor(name) {
6          this.name = name
7      }
8  }
```

题目要求人→老师/学生的继承关系，我直接用了新特性中的class关键字，我一直秉承着既然有方便、高效的新东西，为什么不用呢？但是这些新东西的原理还是原型那一套，所以源码还是要吃透的。

老师

学生

创建

由于这回要求是根据老师和学生的人数m、n来创建，所以必须是动态添加div了。

老师

3

学生

2

创建

教师0

上课

下课

学生0

说话

选课

教师1

上课

下课

学生1

说话

选课

教师2

上课

下课

哦对，忘了说我这回用了bootstrap框架的button.....因为想把更多时间放在研究面向对象方面

第一眼看到题目的时候，很不理解，因为相当于，一个teacher的div对应一个new Teacher()，我这里不太明白怎么将HTML的元素和每一个JavaScript中的对象联系起来，这个虽然代码已经写完了有了自己一定的认识，不过还是希望老师可以给一个标程来学习一下，来进一步加深对面向对象的理解。再加上HTML的方法实在太多了，看了优秀实例以后可以写出更高效、漂亮的代码，目前还是自认为我的代码有一些臃肿。

比较核心的部分就是循环为button绑定对应类的对应方法，直接执行自己对象的方法。

```

154 //添加一些动作事件
155 if(n+m > 0) {
156     //为说话按钮 添加动作监听事件
157     var btn3s = document.getElementsByClassName('btn3')
158     for (var i=0; i<btn3s.length; i++) {
159         btn3s[i].addEventListener('click', function(){
160             students[parseInt(this.parentNode.id.substring(7))].Speak()
161         })
162     }
163
164     //为选课按钮添加动作监听事件
165     var btn4s = document.getElementsByClassName('btn4')
166     for (var i=0; i<btn4s.length; i++) {
167         btn4s[i].addEventListener('click', function(){
168             students[parseInt(this.parentNode.id.substring(7))].selectCourse()
169         })
170     }
171
172     //为上课按钮添加动作监听事件
173     var btn1s = document.getElementsByClassName('btn1')
174     for (var i = 0; i < btn1s.length; i++) {
175         btn1s[i].addEventListener('click', function(){
176             teachers[parseInt(this.parentNode.id.substring(7))].beginClass()
177         })
178     }
179
180     //为下课按钮添加动作监听事件
181     var btn2s = document.getElementsByClassName('btn2')
182     for (var i = 0; i < btn2s.length; i++) {
183         btn2s[i].addEventListener('click', function(){
184             teachers[parseInt(this.parentNode.id.substring(7))].endClass()
185         })
186     }
187 }

```

不过最重要的还是，自己类里的函数内容。比如选课函数的内容

```

22     selectCourse() {
23         if(teacherlist.style.display == "block")
24             teacherlist.style.display = "none"
25         else
26             teacherlist.style.display = "block"
27         //当一个学生选择了一个老师，给这个老师的管理列表添加这个选他的学生
28         var teacherlist_child = teacherlist.childNodes
29         var thisstu = this
30         for (var i = 0; i < teacherlist_child.length; i++) {
31             teacherlist_child[i].onclick = function(){
32                 if(thisstu.myteacher != null)
33                 {
34                     var pos = thisstu.myteacher.management.indexOf(thisstu)
35                     if(pos != -1)
36                         thisstu.myteacher.management.splice(pos, 1)
37                 }
38                 teachers[parseInt(this.id)].management.push(thisstu)
39                 thisstu.myteacher = teachers[parseInt(this.id)]
40
41                 if(thisstu.mydiv.childNodes.length == 3) {
42                     thisstu.mydiv.appendChild(document.createTextNode("已选教师"+this.id))
43                 }
44                 else {
45                     thisstu.mydiv.removeChild(thisstu.mydiv.lastChild)
46                     thisstu.mydiv.appendChild(document.createTextNode("已选教师"+this.id))
47                 }
48                 if(teacherlist.style.display == "block")
49                     teacherlist.style.display = "none"
50                 else
51                     teacherlist.style.display = "block"
52             }
53         }
54     }

```

还有很多就不一一贴出了。

四、总结

经过这一次三个实例的训练，更加熟悉HTML、CSS、JavaScript的书写，更多还是JavaScript，感觉提升了很多。不过JavaScript这个语言我还是觉得有点怪怪的，好多代码呈现出的效果总是和我预想的有一些出入，但是有的时候改一改也不知道为什么又改对了。希望在接下来的课程中可以进一步提高自己。