

# 一、实验目的

---

编写随机点名程序(越不来上课的人，被点中的概率越高，实现抽查、预警等功能)

## 二、算法设计与结果预览

---

选择开发语言：

C/C++ 不太适合，由于我的想法是最终要做出一个比较拿的出手的图形界面

Java(Swing窗体开发)，有想过，但觉得不能完全实现我的功能，代码量大且冗余。

Python 其实可以一试，但最终选择了html语言。对于html来说，它能够快速开发出界面友好的网页，功能强大，语法简单且容易上手

前期准备：

- 5小时 html前端开发教程视频
- 0.5小时算法设计
- 1小时界面布局设计

似乎有点本末倒置，但其实对于此次实验，算法不难,在于如何呈现，学会一门新的开发技术，也是好的。

基本框架：随机点名程序，为了方便后面编码，直接将待导入的名单设为计算机一班的所有成员名单，我选择的布局是按钮式

名单布局

arrs是我导入的课堂学生名单，先设定css样式如下：

```
.box {  
    width: 1000px;  
    height: 240px;  
    margin: 100px auto 0;  
    clear: both;  
}  
<!--其实这些参数都是摸索出来的，预览一次次调试-->
```

接着导入每一个节点元素，innerHTML为相应名字，样式与设置参数一样，类名为name

```
var boxNode = document.querySelector('#box');
for (var i = 0; i < arrs.length; i++) {
    var divNode = document.createElement('div');
    divNode.innerHTML = arrs[i];
    divNode.className = 'name';
    boxNode.appendChild(divNode);
}
```

## 点名按钮设置

按钮css样式设置

```
#btn {
    width: 100px;
    height: 30px;
    background: #fff;
    border-radius: 5px;
    margin-left: 1150px;
    margin-top: 50px;
}
```

## 导入时间类

```
var span = document.querySelector("#span");
getTime();
setInterval(getTime, 1000);
function getTime() {
    var day = new Date;
    var year = day.getFullYear();
    var month = day.getMonth() + 1;
    var dat = day.getDate();
    var hour = day.getHours();
    var mit = day.getMinutes();
    var sec = day.getSeconds();
    span.innerHTML = "Current Time:" + ' ' + year + '.' + month + '.' + dat + ' ' + hour
    + ':' + mit + ':' + sec;
}
```

## 基本的标题、超链接设置

```
<h1>随机点名系统V1.0</h1>
```

```
<div class="txt-center"><a href="https://blog.csdn.net/weixin_43798170">Copyright © 菱形
继承. All Rights Reserved</a></div>
```

## 网页编码字符集、总体style属性

```
<meta charset="UTF-8">
<title>随机点名系统</title>
<style>
    body {
        background: #3ba1c4;
    }

    h1 {
        text-align: center;
        color: white;
    }

    #btn {
        width: 100px;
        height: 30px;
        background: #fff;
        border-radius: 5px;
        margin-left: 1150px;
        margin-top: 50px;
    }

    .name {
        width: 100px;
        height: 40px;
        float: left;
        background: antiquewhite;
        margin-left: 10px;
        margin-top: 10px;
        text-align: center;
        line-height: 40px;
    }

    .box {
        width: 1000px;
        height: 240px;
        margin: 100px auto 0;
        clear: both;
    }

    #span {
        float: right;
        position: relative;
        top: 50px;
        right: 270px;
        color: white;
    }

    .txt-center{
        color:red;
        float: right;
        position: relative;
        top: 40px;
        right: 270px;
        color: white;
    }
```

```
}  
</style>
```

最终基本布局预览，还算可以吧！接着就是算法设计部分啦



目前只是一个静态网页，我们还需要做以下操作：

## 算法部分

基本的思路应该是围绕着随机数生成，做函数模拟，实际编码时应该先写鼠标事件，由于我们这里是随机点名根据题意应该不是手动停止筛选(从后文中的如果无故缺勤概率增加)因为我们**设置1500ms停止，每100ms为一次子事件**，清除所有节点元素背景颜色，控制生成0~arr.length-1的随机数,并置random随机数下标的节点块为**红色**，这样显得直观清楚，接着生成新随机数，来判定是否正常出勤还是无故出勤，我这里采用的是判断这个新随机数是否是上个随机数的cof倍，cof是一个待定系数，这个系数可以用简单的**蒙特卡洛方法模拟**一下，在取**cof=0.2**时，总共测试5次，每次测试100次缺勤次数仅为4-5次，符合正常上课时出勤率的实际情况,当然cof的合理取值也可以通过数学计算求得，不过这里我觉得通过模拟更加贴合实际;接着规定，**无故缺勤三次即警告有挂科风险，下次必点名**，对于无故缺勤下次概率上升的算法这里是设置了每个人都有个pro[i],i对应其下标，初始值为1/arr.length,当上述程序判定无故缺勤时，每次增加exp倍，其中exp为自然常数e，那么pro[i]为

$$pro[i] = pro[i] * e^{vis[i]}$$

其中vis[i]为无故缺勤次数

最后概率计算为：

$$pro = \frac{pro[i]}{\sum_{i=0}^{arr.length} pro[i]}$$

最后每次更新一下分母就好了，复杂度虽然有点高，但毕竟是小数据，还可以接受~

```

var timeID, cnt = 0, random, check, tep = 0, cof=0.2, last=0, interval=0.05*arrs.length;
btn.onclick = function() {
    if (this.value == "点名") {
        timeID = setInterval(function() {

            if (cnt == 15) {

                for (var i = 0; i < arrs.length; i++)
                {
                    if(boxNode.children[i].style.background == 'red')
                    {
                        random=i;break;
                    }
                }
                tep=parseInt(Math.random() * arrs.length);
                if(tep>random*cof>window.alert(arrs[random]+" 正常出勤!");
                else
                {

                    if(last-
vis[random]*interval<=random&&last+interval*vis[random]>=random&&last)
                    {
                        vis[last]++;
                        var ans=2*interval*vis[last]/arrs.length;
                        ans=ans*100;
                        if(vis[last]>=3>window.alert(arrs[random]+" 你已无故缺勤3次, 给予你
此门课的挂科警告!");
                        else window.alert(arrs[random]+" 无故未到, 下次被点名的概率增加了, 当
前被点中概率为"+ans+"%");
                    }
                    else
                    {
                        vis[random]++;
                        var ans=1.0/45;
                        for(var i=0;i<vis[random];i++)ans=ans*exp;
                        ans=ans+(Math.random()*0.1*ans);
                        ans=ans*100;
                        if(vis[random]>=3>window.alert(arrs[random]+" 你已无故缺勤3次, 给予
你此门课的挂科警告!" +ans+"%");
                        else window.alert(arrs[random]+" 无故未到, 当前被点中概率
为"+ans+"%");
                    }
                    last=random;
                }
                cnt = 0;

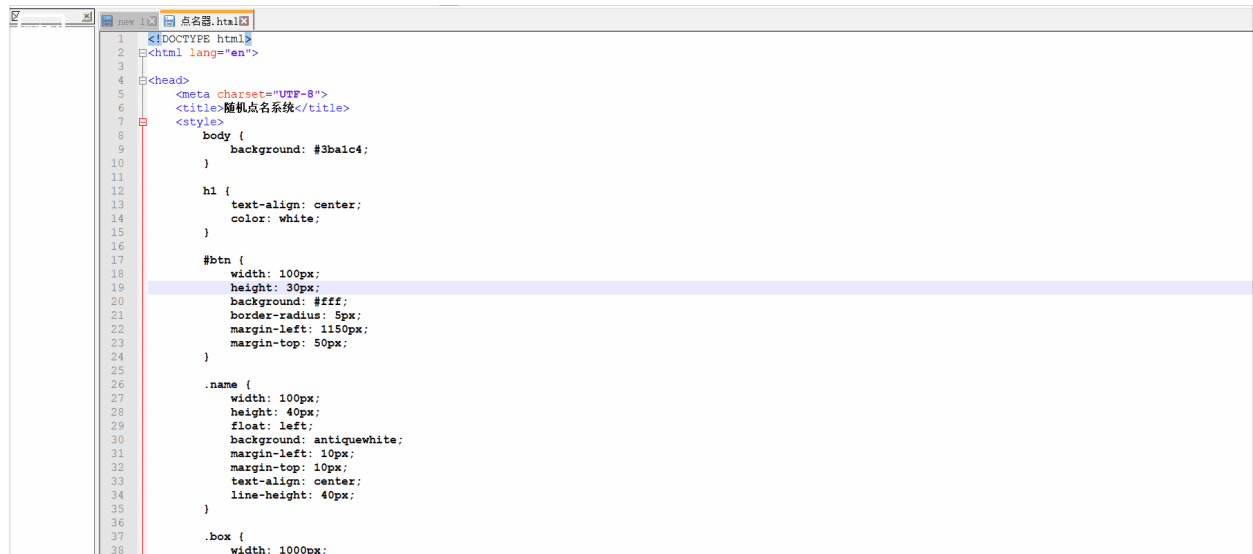
                clearInterval(timeID);
            }
            random = parseInt(Math.random() * arrs.length);
            for (var i = 0; i < arrs.length; i++) boxNode.children[i].style.background =
'';

            boxNode.children[random].style.background = 'red';
            cnt++;
        }, 100)
    }
}

```

```
}  
}
```

## 基本功能演示



```
1 <!DOCTYPE html>  
2 <html lang="en">  
3  
4 <head>  
5   <meta charset="UTF-8">  
6   <title>随机点名系统</title>  
7   <style>  
8     body {  
9       background: #3ba1c4;  
10    }  
11  
12    h1 {  
13      text-align: center;  
14      color: white;  
15    }  
16  
17    #btn {  
18      width: 100px;  
19      height: 30px;  
20      background: #fff;  
21      border-radius: 5px;  
22      margin-left: 1150px;  
23      margin-top: 50px;  
24    }  
25  
26    .name {  
27      width: 100px;  
28      height: 40px;  
29      float: left;  
30      background: antiquewhite;  
31      margin-left: 10px;  
32      margin-top: 10px;  
33      text-align: center;  
34      line-height: 40px;  
35    }  
36  
37    .box {  
38      width: 1000px;
```

## 三、实验总结

实现了要求的基本功能，基本掌握了html语法，利用此次随机点名系统的练习加强算法训练并上手了第一个html项目，不足在于，使用html有较大的局限性，很多数学函数无法调用，基本的数据结构不能使用，实现算法也比较简单，也并未实现真正的用户交互，后期可考虑开发导入csv格式的学生名单自动生成图形界面，来管理数据。

