计算器项目总结报告

1 项目组及分工简介：

项目名：简单计算器

开发小组：1 号开发小组

项目人员：段耀威

黄浩彤

张君琦

王志飞

开发周期：1 周

1.1 项目的意义：

根据用户按键指令做出相应动作，处理简单的四则运算 百分号，完成运算功能 杜绝输入错误

2 项目流程：

2.1 组织讨论明确项目需求及分工

2.1.1 项目需求：

（1）数字和运算按键

（2）运算结果显示

2.1.2 功能概述及模块：

（1）按键上方显示按键结果

（2）按键之后显示相应字符

（3）按键后实时运算结果显示

（4）相应按键运算功能

（5）运算符不能连接出现

（6）整数首位不能为 0

（7）小数点不能连续出现

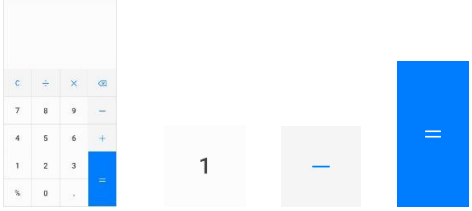
（8）按键错误提示

2.1.3 项目分工：

张君琦	UI 切片和效果图
黄浩彤	css
王志飞	js
段耀威	测试+项目文档总结

2.2 网站效果图设计以及 ui 切图

首先由项目 ui 负责制作出项目效果图以及切图。并根据情况给出项目文档。



整体分为显示部分与按键部分

本次涉及的切图为各个按键的图片

给出项目文档提供网页内容并说明各个内容间的距离和背景颜色

2.3 css+div 样式布局

2.3.1 样式书写:

Css 开发部分采用了 div 与表格布局 按键放置在表格中 使用背景模拟按键。

2.4 js 部分：

2.4.1 显示按键字符串或实现运算功能:

实现运算功能及显示，使用事件绑定:

```
<td id="data21" onclick="press(7)"></td>
<td id="data22" onclick="press(8)"></td>
<td id="data23" onclick="press(9)"></td>
<td id="data24" onclick="jian()"></td>
```

显示按键内容

```
function press(y) {
    // 点击数字在表达上添加数字 刷新显示内容
    str += y;
    answer.innerHTML = str;
    // 下面为判断表达式里面有没有运算符如果有点击数字时（也就是改变表达式时）实时在第
    if (str.indexOf("+") !== -1 || str.indexOf("-") !== -1 || str.indexOf("*")
        answer2.innerHTML = Number(eval(str));
    }
}
```

按键运算逻辑，使用双分支结构及嵌套:

举例

```
function presszero() {
    // 提出问题：如果你进行了详细的测试你会发现如果用 eval 处理 021+1 或者 1+021 之类 0 开头的表达式那么结果并不是你想象的 22
    // 那么我们就需要对 0 进行处理首先我们思考什么时候会出现错误
    // 分析得出什么时候出现错误：错误情况为第一位是 0 或者运算符之后直接是一个 0 开头的非小数
    // （1）当第一位输入是零的情况下 如果后面点击的是数字 即点击顺序为 023 之类我们希望的结果是在表达式中记录 23 而不是 023
    // 于是有了我们第一次判断 当表达式为空点击 0 时我们不对表达式（str）进行操作
        // 当然你点击 0 没有任何反应 用户可能认为你的计算器有毛病那么我们就为用户显示一个 0 并不在表达式中记录这个 0
            // 然后再点击其他数字时自然那个 0 就没有于是实现了 023 正确显示并记录为 23
    // 有的人可能已经想到了如果我第一位要输 0.3 之类的小数怎么办?
        // 没关系我们来走一走流程用户点 0 显示 0 然后点击点 进入我们点的函数此时表达式为空（刚才的 0 只是显示并没有记录到表达式）结果字符串变为了 0.显示存储正确无需处理
    // （2）如果我们表达式最后一位是运算符 此时会出错那么我们想到了解决思路与上面一样：仅显示不存储
    // 如果不是运算符那么就是点或者其他数字了允许正常输入
    if(str===""){
        answer.innerHTML = 0;
        // 为用户显示一个 0
    }else{
        if (str.substr(str.length - 1, 1) == "+" || str.substr(str.length - 1, 1) == "*" || str.substr(str.length - 1, 1) == "-" || str.substr(str.length - 1, 1) == "/")
         {
        answer.innerHTML = str+"0";
        // 仅显示不存储(不修改 str)
        }else{
```

```
            // 允许正常输入
            str += 0;
            answer.innerHTML =str;
            answer2.innerHTML = Number(eval(str));
        }
    }
}
// 下面来到百分号 好多同学将百分号设置为了取模 但是我个人认为这并不符合大部分用户对于计算器上百分号的理解 于是我将百分号设置功能如下:如果表达式是一个数字比如 999 那么直接变为 9.99 如果是 99+6*999 那么变为 99+6*9.99 (经测试大多数手机计算器均为此逻辑)
// 功能确定后就是具体实现我们依然按照上面的思路一步一步来

var a = "", b = "", c = "", d = "", i = "";
function pressbf() {
    // (1) 如果表达式为数字 那么表达式除 100 并记录
    if (!isNaN(str)) {
        str = str / "100";
        answer.innerHTML = str;
        answer2.innerHTML = str;
    }
    else {
    // (2)如果表达式最后一位是运算符这时是没有意义的我们不进行任何操作
        if (str.substr(str.length - 1, 1) == "+" || str.substr(str.length - 1, 1) == "*" || str.substr(str.length - 1, 1) == "-" || str.substr(str.length - 1, 1) == "/") {
        } else {
    // (3)如果接下来实现第二个功能 99+6*999 变为 99+6*9.99 我们需要知道最后一个运算符的位置才可以继续
            a = str.lastIndexOf("+");
            b = str.lastIndexOf("-");
            c = str.lastIndexOf("*");
            d = str.lastIndexOf("/");
            i = Math.max(a, b, c, d);
            // 取最大值即为最右面的运算符位置
            str = str.substr(0, i + 1) + (str.substr(i + 1, str.length - i) / 100);
            // 截取表达式第一位到运算符那一位连接上"运算符后面的数字/100" 存入新表达式
            answer.innerHTML = str;
            // 显示结果
            answer2.innerHTML = eval(str);
            // 功能成功实现
        }
    }
}
```

3 总结

本次项目实现过程中主要的难点在于逻辑思考，如何实现相应的逻辑需要进行思考，如何获得实现方案的思路很重要。要学会从问题分析出发找到解决方案。