实验十　JavaScript 程序结构

【实验目标】

1. 掌握 JavaScript 分支结构语法，并学会使用分支结构编程。
2. 掌握 JavaScript 循环结构语法，学会使用多种循环编写应用程序。
3. 掌握多分支 if...else 结构与 switch 结构互换编程。

【实验内容】

1. 用分支结构实现简易代码编程。
2. 用循环结构实现简易代码编程。
3. 用两种多分支结构编写相关程序。
4. 用自定义函数实现相关程序功能。
5. 使用 CSS＋DIV 综合编程。

【实验项目】

1. 成绩百分制转换为五级制。
2. 计算∑N!

项目 1　成绩百分制转换为五级制

1. 实验要求

五级制成绩表示法，将百分制成绩转换成“优秀、良好、中等、及格、不及格”共 5 个等级。要求能够使用多分支 if…else 和 switch 两种结构并通过编写直接执行 JavaScript 脚本、自定义判定成绩等级的函数两种方法来实现程序功能，程序运行页面如图 1-10-1～图 1-10-3 所示。

图 1-10-1　运行初始页面

图 1-10-2　运行中页面

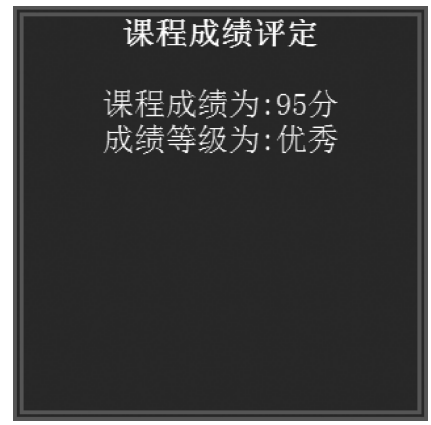

图 1-10-3　运行结束后页面

2. 实验内容

(1) 设置图层的属性。

(2) 熟悉两种多分支结构(if...else if...else、switch)。

(3) 熟悉 JavaScript 脚本编程。

(4) 学会定义 JavaScript 等级转换函数 conversion(score)。

3. 实验中所需标记语法

(1) 图层 div 标记。

```
<div id = "div1">…</div>
```

(2) 样式 style 标记。

```
<style type = "text/css">
    #div1{
        background:#006600; color:white;
        width:300px; height:300px;
        margin:100px auto; font-size:24px;
        border:8px double #009933;text-align:center;
      }
</style>
```

(3)脚本 script 标记。

```
<script type = "text/javascript" src = "*.js">…</script>
```

4. 程序结构

(1) if(){}else{}结构。

- 单分支结构:

```
if (x>10) alert("单分支结构")
```

- 双分支结构:

```
if (x>=10) {alert("x 大于等于 10")}else {alert("x 小于 10"); }
```

• 多分支结构：

```
if (x>=90){alert();}
else if (x>=80) { alert();}
else if (x>=70) { alert();}
…
else { alert();}
```

（2）switch 结构。

```
1  switch (level) /* level 是变量 */
2  {
3      case 1:{result="优秀";break;}       /* 每一 case 语句均必须以 break; 结束 */
4      case 2:{result="良好";break;}
5      case 3:{result="中等";break;}
6      case 4:{result="及格";break;}
7      default:{result="不及格";}          /* 最后一个为默认 default */
8  }
```

5. JavaScript 自定义函数结构

自定义函数格式：

```
function 函数名(参数 1,参数 2,...,参数 n){函数体}
```

```
1  <script type="text/javascript">
2      //成绩与等级转换函数
3      /* 参数: score,表示输入成绩
4      返回值: result,是等级
5       */
6      function conversion(score){
7          var result="",level=0;
8          if (score>=90){level=1};
9          if (score<90 && score>=80){level=2};
10         if (score<80 && score>=70){level=3};
11         …
12         return result;
13     }
14 </script>
```

6. 编程要求

（1）采用 if (){} else if ()else{}结构和脚本直接编程。要求利用提示对话框函数输入成绩，并将评定等级保存在变量 result 中，最后输出到页面上。

```
1  <script type="text/javascript">
2  //五级制成绩表示法
3  //采用分支嵌套结构
4      document.write("<b>课程成绩评定</b><br><br>");
5      //利用函数输入一个成绩
```

```
6       var result = "";                                        //定义保存评定等级结果的变量
7       var x = prompt("请输入你的成绩: ",0);                    //利用提示对话框输入成绩
8       if (x! = null)
9       {   document.write("课程成绩为:" + x + "分");
10          if (x>=90){ result = "优秀"; alert("1 -- 成绩等级为" + result); }
11          else if (x>=80){ result = "良好"; alert("2 -- 成绩等级为" + result); }
12          else if (x>=70){ result = "中等"; alert("3 -- 成绩等级为" + result); }
13          else if (x>=60){ result = "及格"; alert("4 -- 成绩等级为" + result); }
14          else{ result = "不及格"; alert("5 -- 成绩等级为" + result); }
15      }
16      else{ alert("请重新输入成绩!"); }
17      document.write("<br>成绩等级为:" + result);          //最后输出评定等级
18  </script>
```

(2) 采用 switch 结构和脚本直接编程。要求利用提示对话框函数输入成绩,将成绩分为 5 个等级分别对应"1-优秀、2-良好、3-中等、4-及格、5-不及格",用数字等级作为 case 常量,再根据输入成绩所属范围转换成相应的等级,保存在 level 变量中,然后采用 Switch 结构编程,参照上例代码将结果输出到页面上。

```
1   <script type = "text/javascript">
2       //五级制成绩表示法
3       //采用分支嵌套结构
4       document.write("<b>课程成绩评定</b><br><br>");
5       var x = prompt("请输入你的成绩: ",0); //利用函数输入一个成绩
6       if (x! = null)
7       {
8           document.write("课程成绩为:" + x + "分");
9           var level = 0,result = "";
10          if (x>=90){level = 1};
11          if (x<90 && x>=80){level = 2};
12          if (x<80 && x>=70){level = 3};
13          if (x<70 && x>=60){level = 4};
14          if (x<60){level = 5};
15          switch (level)
16          {
17              case 1:{result = "优秀";break;}
18              case 2:{result = "良好";break;}
19              case 3:{result = "中等";break;}
20              case 4:{result = "及格";break;}
21              default:{result = "不及格";}
22          }
23          document.write("<br>成绩等级为:" + result);
24      }
25      else{
26          alert(请重新输入成绩!);
27      }
28  </script>
```

（3）采用函数编程。要求编写独立的成绩与等级转换函数，将下列代码是放置在 head 标记中，在 body 标记中通过 conversion(score)函数实现调用，将函数值赋给指定的变量。在函数体内可以使用两种多分支结构编程，即 if(){}else if(){}或多分支结构编程。

```
<script type="text/javascript">
    //成绩与等级转换函数
    /*参数:score,表示输入成绩
      返回值:result,是等级
    */
    function conversion(score){
        var result="",level=0;
        if (score>=90){level=1};
        if (score<90 && score>=80){level=2};
        if (score<80 && score>=70){level=3};
        if (score<70 && score>=60){level=4};
        if (score<60){level=5};
        switch (level)
        {
        case 1:{result="优秀";break;}
        case 2:{result="良好";break;}
        case 3:{result="中等";break;}
        case 4:{result="及格";break;}
        default:{result="不及格";}
        }
        return result;
    }
</script>
```

7. 实验步骤

（1）建立 prj_10_1_js_function.html、prj_10_1_js_if_else.html、prj_10_1_js_switch.html 等多文档框架。

（2）在 head 标记中插入内部样式表，运用 CSS 定义图层样式。

（3）在 head 标记中插入 script 标记，在 script 标记里定义 conversion(score)函数。

（4）参照图 1-10-1～图 1-10-3 所示的页面效果，在 body 标记中插入 div 和 script 等相关元素。

（5）完成相关代码编写。

项目 2　计 算 ∑N!

1. 实验要求

（1）掌握 For、while、Do...while、For in 等循环语法结构。

（2）能熟悉运行多种循环结构解决实际工程问题。

（3）运用 For 循环实现计算∑N!，页面效果如图 1-10-4 和图 1-10-5 所示。

图 1-10-4　计算∑N！初始页面

2. 实验内容

(1) 定义内部样式表。

(2) 掌握 JavaScript 放置与运行方式。

(3) 定义域及域标题。

(4) 学会利用 For 循环编写应用程序。

3. 实验中所需标记语法

(1) 域 fieldset 标记、域标题 legend 标记。

```
<fieldset>
    <legend>计算 1! + 2! + ... + N!</legend>…
</fieldset>
```

图 1-10-5　计算∑N！结果页面

(2) 样式 style 标记。

```
<style type = "text/css">
    fieldset{
        margin:20px auto; background: # 0033ff; color:white;
        width:300px; padding:0 40px; border:2px outset # 009966;
    }
    legend{ font - size:28px; color: # 99ffff; font - weight:bolder; }
</style>
```

(3) 脚本 script 标记。

有两种格式：

```
<script type = "text/javascript" [src = "外部文件名称.js"]></script>
<script language = "javascript" [src = "外部文件名称.js"]></script>
```

4. 编程要求

(1) 主程序为 prj_10_2_js_ for. html。

(2) 使用域、域标题标记,将页面信息进行分组,定义 fieldset、legend 标记样式。

(3) 掌握 JavaScript 脚本的放置与运行方式,会使用多种方式进行编程;学会使用提示对话框给变量赋值,并对变量的取值进行判断。

(4) 利用 For 循环、While、Do... while 等循环结构解决实际问题,并进行比较,总结在哪些情况下循环结构可以相互替换使用,不断积累编程经验。

(5) 该实验仅以 For 循环为例编程实现计算∑N!,其他循环结构可参照编写,在此略去。

5. JavaScript 脚本调用

(1) 事件调用。

```
<input id = "button" type = "button" value = "计算∑N!" onclick = "show();">
```

(2) 直接调用。

将脚本放置在 body 标记中直接执行。

```
<script type = "text/javascript">
    /* 这是直接调用 JS */
    document.write("这是直接调用 JS");
</script>
```

6. 实验步骤

(1) 建立 prj_10_2_js_for. html、prj_10_2_js_ circulation_function. html 文档框架。

(2) HTML 文档 head 标记中插入样式 style 标记。

(3) 在 style 标记中分别定义域样式、域标题样式。

(4) 分别在 head、body 标记中插入 script 标记,并编写相应的 JavaScript 代码。

(5) 在 body 标记中插入域、域标题标记、脚本等标记。

(6) 利用多种循环编程计算∑N!,也可以编写通用自定义函数计算∑N!,函数名 compute_sum(n)。

7. 拓展与提高

(1) 采用 While 循环实现计算∑N!,程序文件名 prj_10_2_js_while. html。

(2) 采用 Do... while 循环实现计算∑N!,程序文件名 prj_10_2_js_do_while. html。

(3) 采用自定义函数编程实现计算∑N!,程序文件名 prj_10_2_js_circulation_function. html。函数名为 compute_sum(n)。

- 页面布局要求——采用表格和表单布局,效果如图 1-10-6 所示,其中表格采用 4 行 4 列格局,定义表格 table、td 和 # button 的样式属性,使其界面达到如下效果:输入整数 N 的值后,单击“计算∑N!”按钮后,页面效果如图 1-10-7 所示,单击“清空”按钮可以将页面上的所有文本框清空。
- 页面元素获取——利用 Document 对象模型的 getElementById(“ID 名称”)来获取 HTML 页面元素,然后通过对该元素 value 属性获取或赋值来实现页面文本框数据

图 1-10-6　计算ΣN! 初始页面

图 1-10-7　输入 N 并单击“计算ΣN!”按钮后的页面

的读取和重置。

- 编写外部 js 文件：文件名为 sum_factorial.js，定义的函数为 compute_sum(n)，然后通过单击“计算ΣN!”按钮的 onclick 事件来实现 js 函数的调用。

外部 JavaScript 脚本代码如下：

```
/*功能：计算ΣN!
函数名：sum_factorial.js*/
//采用 For 循环实现
function compute_sum(n){
    var result = 1,sum = 0;                    //定义保存阶乘累加和、N 阶乘结果的变量
    for (i = 1;i <= n;i++)
    {
        result = result * i;                   //计算 i 的阶乘
        sum = sum + result;                    //计算累加到 i 阶乘的和
    }
    return sum;                                //返回阶乘累加和
}
```

```
13 //显示累加和的函数
14 function show(){
15     var n = parseFloat(document.getElementById("n_text").value);     //拿到文本框的值并
                                                                        //转换成实数
16     var sum = compute_sum(n);              //阶乘计算累加和
17     document.getElementById("sum_text").value = sum;                //向累加和文本框赋值
18     return;                                //结束函数
19 }
```