

【实验目标】

1. 理解 JavaScript 脚本放置与运行的方法。
2. 掌握 JavaScript 基本构成和基础语法。
3. 掌握自定义函数定义与引用。

【实验内容】

1. JavaScript 放置与运行。
2. JavaScript 变量、标识符、表达式定义与使用。
3. 自定义函数编写与调用。
4. 消息对话框与系统内部函数应用。
5. 事件调用与简约化调用。
6. 样式表定义与使用。
7. 表单、表单元素的使用。

【实验项目】

1. 改变网页字号大小。
2. 计算圆的面积。
3. 消息对话框的使用。
4. 系统内部函数的使用。

项目 1 改变网页字号大小

1. 实验要求

很多网站的新闻版块均设有个性化的“选择字号【大中小】”链接的功能,主要是给不同的网络访问者带来视觉上的方便。例如江苏省人民政府网站就有这样的功能,可以改变该页字号大小,如图 1-9-1 所示。

根据“江苏省人民政府网站”中的这一个人性的功能设计如图 1-9-2 所示的界面,要求当网络访问者选择字号中的“大、中、小”时能实现页面字号大小变化,选择“中”时,页面效果如图 1-9-3 所示。

2. 实验内容

- (1) 创建 div、设置 div 的样式并指定 id。
- (2) 通过 CSS 设置层、段落、标题字、文字与排版属性。
- (3) div、p 样式的引用方法。



选择字号

图 1-9-1 江苏省人民政府网站



图 1-9-2 未单击前初始状态页面效果

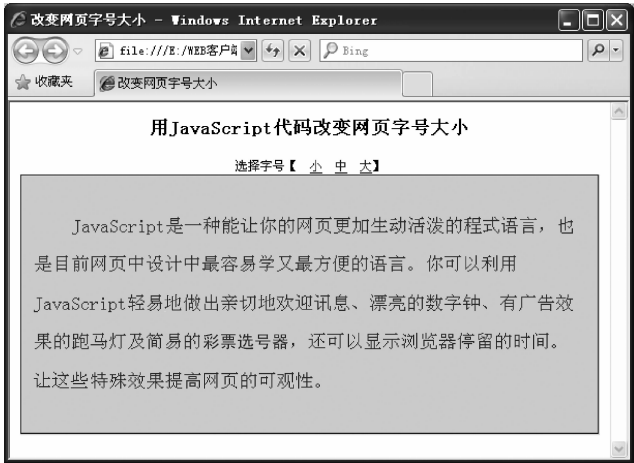


图 1-9-3 单击“中”链接后的页面效果

(4) 自定义实现设置字体大小的 JavaScript 函数 setFont(size)。

(5) 使用超链接实现自定义函数的调用。

3. 实验中所需标记语法

(1) 图层 div 标记。

```
<div id = "content">...</div>
```

(2) 样式 style 标记。

```
1 <style type = "text/css">
2     #div1{text-align:center;font-size:12px;}
3     #content{font-size:12px;line-height:200%;padding:10px;
background: # 000099;color:white;border:2px groove # 0000cc;}
4     p{text-indent:2em;}
5 </style>
```

(3) 脚本 script 标记。

```
1 <head>
2     <script type = "text/javascript">...</script>
3 </head>
```

(4) 超链接 a 标记。

```
<a href = "javascript:setFont('12px')">小</a>
```

(5) 标题字 h3 标记。

```
<h3 align = "center">用 JavaScript 代码改变网页字体大小</h3>
```

4. JavaScript 脚本的放置与函数

(1) 在 head 标记中。

```
1 <head>
2     <script type = "text/javascript">...</script>
3 </head>
```

如果脚本放在 head 标记中,script 标记中脚本代码必须定义成函数形式,格式如下:

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

(2) 在 body 标记中。

```
1 <body>
2     <script type = "text/javascript">...</script>
3 </body>
```

如果脚本放在 body 标记中, script 标记中脚本代码既可以是函数,也可以是代码段。但在 body 标记中可以调用已经定义过的脚本函数,调用方式可以事件调用,也可以是简约化调用,还可以动态调用。

(3) JavaScript 自定义函数结构(setFont(size))

格式:

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

```
1 function setFont(size){
2     /* 定义设置字体大小函数 */
3     //size:大小,单位 px
4     var obj = document.getElementById("content");    //根据 id 获取文档对象
5     obj.style.fontSize = size;                        //设置对象的字体大小
6     //obj.style.color = "#ff0000";                  //设置对象的颜色
7 }
```

(4) 超链接中调用 JavaScript。

```
<a href = "javascript:setFont('12px')">小</a>    //简约化调用
<a href = "# " onclick = "javascript:setFont('18px');">中</a>    //事件调用
<a href = "javascript:setFont('24px');">大</a>    //简约化调用
```

5. 页面结构分析及编程要求

(1) 整个页面由 1 个标题、2 个图层构成。第一个图层中放置 3 个超链接;第二个图层放置 1 个段落。然后选择第一个图层中的“大”、“中”、“小”链接实现改变该网页字号大小。

(2) 编写实现改变字号大小的自定义函数 setFont(size)。实现这个函数最关键的是如何获取页面上特定的元素,可以使用 document.getElementById("id")来获取一个 HTML 文档中指定 ID 对象,如获取 id 为 content 的图层 DIV 的方法是 var obj = document.getElementById("content");,然后可以获取或设置该对象属性,如通过 obj.style.fontSize=size 和 obj.style.color="#ff0000"两个赋值语句完成对字体大小、颜色等属性值重新设置。

6. 实验步骤

(1) 建立 prj_9_1_js_setfont.html 文档基本结构。

(2) 在 head 标记中插入内部样式表,并在样式表中定义图层、段落的样式。

(3) 在 head 标记中插入 script 标记,完成 setFont(size)函数的定义。

(4) 参照图 1-9-2 进行页面布局,在 body 标记中插入标题、2 个图层;在第一个图层中插入 3 个超链接,并给超链接设置 href、onclick 等属性,完成自定义函数的调用;在第二个图层中插入 1 个段落,给上述元素应用所定义的样式。

需要说明的是通过 DOM 的 document.getElementById("id")方法获取 HTML 页面元素后,在对该对象的 style 对象的属性进行设置时,需要注意,DOM 中 style 对象的属性与 CSS 样式中常用的属性名称未必相同。HTML DOM Style 对象的属性有背景、边框和边距、布局、列表、杂项、定位、打印、滚动条、表格、文本、规范。详细请参照网站 http://www.w3school.com.cn/html/dom/dom_obj_style.asp。此处仅列举与文本相关的部分属性(如表 1-9-1 所示),便于编程使用。

表 1-9-1 Text 部分属性表

属 性	描 述
color	设置文本的颜色
font	在一行设置所有的字体属性
fontFamily	设置元素的字体系列
fontSize	设置元素的字体大小
fontStyle	设置元素的字体样式
fontVariant	用小型大写字母字体来显示文本
fontWeight	设置字体的粗细
letterSpacing	设置字符间距
lineHeight	设置行间距
textAlign	排列文本
textIndent	缩紧首行的文本
textTransform	对文本设置大写效果

项目 2 计算圆的面积

1. 实验要求

(1) 掌握外部 JavaScript 脚本的编程方法,学会在 HTML 文档中引用自定义脚本程序,完成计算圆的面积 compute(radius)、显示圆的面积 show() 等函数的定义,实现页面布局如图 1-9-4 所示。

(2) 学会使用 Document 文档对象模型获取 HTML 页面元素的方法。

2. 实验内容

- (1) 定义 JavaScript 外部函数。
- (2) 使用事件调用和简约化调用 JavaScript 函数。
- (3) 定义图层。
- (4) 定义内部样式表。
- (5) 定义表单和设置表单元素。

3. 实验所需标记语法

(1) 图层 div 标记。

```
<div id="div1" class=""> ... </div>    <!-- 用于放置表单 -->
```

(2) 样式 style 标记。

```
1 <style type="text/css">
2   /* 图层样式 */
3   div{ margin:0 auto; background: #66ffcc; border:12px inset #ff0000;
4     width:300px; height:200px; text-align:center; }
5   /* 表单样式 */
6   form{ margin:0 auto; padding:30px; }
7 </style>
```

(3) 表单 form 标记。

```
1 <form method="post" action="">
2     <label>半径:</label>
<input type="text" name="radiustext" id="radiustext"><br>
3     <label>面积:</label>
<input type="text" name="areatext" id="areatext" readonly><br>
4     <input type="button" value="计算" onclick="show();" />
5     &nbsp;&nbsp;&nbsp;<input type="reset" name="reset" value="清空" />
6 </form>
```

(4) 脚本 script 标记。

有两种格式：分别使用 type、language 属性定义。

```
<script type="text/javascript" src="area.js"></script>
<script language="javascript" src="area.js"></script>
```

4. 页面设计及编程要求

(1) 主程序为 prj_9_2_js_circle_area.html, 外部 JavaScript 程序为 area.js。

(2) 学会使用图层来进行页面布局, 会设置图层边框样式, 理解 CSS 盒模型含义, 理解边界、边框、填充、内容之间的关系, 页面布局如图 1-9-4 所示。

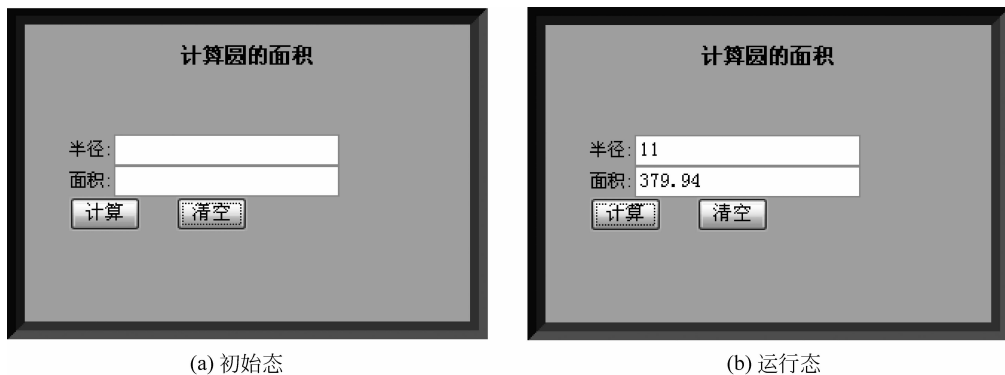


图 1-9-4 计算圆的面积页面效果图

(3) 利用 Document 文档对象模型获取 HTML 页面元素的方法。

- 通过元素的 ID 获取指定元素: document.getElementById("元素 ID")。
- 通过元素的名称获取一组元素: document.getElementsByName("元素名称"), 然后再通过数组去访问其中的每一个元素。

图 1-9-4 所示页面的表单中含有 2 个文本框、2 个按钮。文本框分别用于输入圆的半径和显示圆的面积, 其中半径文本框允许输入数值, 采取指定 ID 获取该元素输入的数值; 面积文本框设置成“只读”文本框(设置文本框的 readonly 属性), 通过元素名称获取元素, 并将计算出来的面积动态赋值给面积文本框, 由于通过元素名称获取的是同类元素的数组, 即一次可以同时获取多个同类元素, 但该实验中与面积文本框同名的文本框只有唯一的一个, 而且数组的下标总是从 0 开始编号, 所以可能通过 document.getElementsByName("areatext")[0].value 方

法来获取或设置圆的面积。如果同名元素有多个,则可以通过下标依次访问其中的任意一个元素。当然也可以通过元素的 ID 获取页面元素,方法与获取半径文本框中的数值相同。

(4) 编写外部 JavaScript 脚本函数 area.js,需要自定义 2 个函数,分别是 compute(radius)和 show(),代码如下:

```
1  /* 计算圆的页面 area.js */
2  //计算圆的面积
3  function compute(radius){
4      var pi = 3.14;
5      var area = pi * radius * radius;
6      return area;
7  }
8  //显示圆面积
9  function show(){
10     var radius = parseFloat(document.getElementById("radiustext").value);
11     var area = compute(radius);
12     document.getElementsByName("areatext")[0].value = area;
13     //通过 ID 获取面积文本框中的数值方法
14     //document.getElementById("areatext").value = area;
15     return;
16 }
```

(5) JavaScript 脚本调用方式。

- 事件调用。

```
<p onclick = "show();" style = "background: # 00ff00 ">单击我,这是事件调用!</p>
```

- 简约化调用。

```
<a href = "javascript:alert('Hello World!!');">这是行内调用 -- 简约化</a>
```

- 直接调用。

将脚本放置在 body 标记中直接执行。

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

5. 实验步骤

- (1) 建立 prj_9_2_js_circle_area.html 文档框架。
- (2) 在 HTML 文档 head 标记中分别插入样式 style 标记和脚本 script 标记。
- (3) 在 style 标记中分别定义图层样式、表单样式。
- (4) 在 script 标记里设置 src 属性,引用外部 js 文件。

(5) 在 body 标记中插入标题、图层,在 DIV 中插入 1 个表单、2 个文本框、1 个普通按钮、1 个重置按钮。

(6) 编写名为 area.js 的外部 JavaScript 程序,完成计算圆的面积函数 compute(radius)、显示圆的面积函数 show() 这 2 个自定义函数。

(7) 利用事件调用、简约化调用自定义 JavaScript 函数。

项目 3 消息对话框使用

1. 实验要求

- (1) 掌握 JavaScript 脚本的编程方法,学会使用事件调用自定义函数 inputName()。
- (2) 掌握 JavaScript 消息对话框语法,并根据用户的选择进行相关代码编写。
- (3) 学会使用 Document 文档对象模型获取 HTML 页面元素的方法。
- (4) 学会使用分支结构编程“if(条件) {条件成立时语句块} else {条件不成立时语句块}”。
- (5) 学会使用域标记<fieldset>和域标题标记<legend>,通过样式设置域标记的边框样式。
- (6) 按照图 1-9-5~图 1-9-8 所示的效果完成代码编写。



图 1-9-5 消息对话框实验初始页面



图 1-9-6 消息对话框实验运行后页面



图 1-9-7 提示信息框页面效果图



图 1-9-8 告警消息对话框页面效果图

2. 实验内容

- (1) 定义 JavaScript 函数 inputName(),功能是通过提示消息框完成姓名输入并验证是否为空,不为空则通过告警消息框输出。
- (2) 学会使用事件调用 JavaScript 函数的方法。
- (3) 定义域和域标题。
- (4) 定义内部样式表。
- (5) 定义表单和设置表单元素属性。

在 JavaScript 中通过 Document 文档对象模型获取该元素,以便用户输入姓名后直接赋值给姓名文本框。

(3) 参照上述实验内容的要求及样例代码,完成 inputName() 自定义函数编写。

(4) JavaScript 脚本事件调用方式如下:

```
<input type="button" value="输入姓名" onclick="inputName();">
```

5. 实验步骤

- (1) 建立 prj_9_3_js_message_dialog.html 文档框架。
- (2) 在 HTML 文档 head 标记中分别插入样式 style 标记。
- (3) 在 style 标记中分别定义 fieldset、legend、form 标记的样式。
- (4) 在 head 标记中插入 script 标记,并在该标记内定义 inputName() 函数。
- (5) 在 body 标记中插入域、域标题、表单和 1 个只读文本框、1 个普通按钮、1 个重置按钮,并设置元素的相关属性。
- (6) 在 body 标记中利用事件调用 JavaScript 函数。

项目 4 系统内部函数使用

1. 实验要求

- (1) 掌握 6 个 JavaScript 系统内部函数的语法。
- (2) 验证 6 个内部函数运行结果,函数验证页面如图 1-9-9 所示。



图 1-9-9 系统内部函数使用页面图

2. 实验内容

- (1) 定义内部样式表。
- (2) 在 HTML 中学会插入 script 标记。
- (3) 验证 JavaScript 内部函数。

3. 实验中所需标记语法

- (1) 样式 style 标记。

```

1 < style type = "text/css">
2     div{
3         background: # 000099;
4         color:white;
5         width:700px;
6         font - size:20px;
7         font - weight:bolder;
8     }
9     h4{text - align:center;}
10    b{
11        color:red;
12        font - size:18px;
13    }
14 </style>

```

- (2) 脚本 script 标记。

```
1 <script type="text/javascript">
2     //定义变量并赋值
3     var rel = eval("1000 + 3/5");
4     //用 write 方法输出结果
5     document.write("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;" + "1000 + 3/5 = " + rel);
6     document.write("<br />");
7     document.write("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;" + "x = 10, y = 20, x * y = ");
eval("x = 10; y = 20; document.writeln(x * y)")
8     document.write("<br />&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;2 + 2 = " + eval("2 + 2"));
9     document.write("<br />");
10    var x = 10;
11    document.write("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;" + "x = 10, x + 17 = " + eval(x + 17));
12    document.write("<br />");
13 </script>
```

4. 页面设计及编程要求

- (1) 主程序为 prj_9_4_js_system_function.html。利用 Document 文档对象模型的 write 方法输入 HTML 流信息,方法是 document.write(“信息”),程序主要功能验证 eval(“字符串”)、escape(“字符串”)、unescape(string)、parseFloat(string)、parseInt(numberstring,radix)、isNaN(testvalue)共 6 个系统内部函数。

- (2) 将 JavaScript 脚本放置在 body 标记中, 直接运行, 也可以放置在 head 标记中, 但需

要编写自定义验证函数,此处省略代码。

5. 实验步骤

(1) 建立 prj_9_4_js_system_function.html 文档框架。

(2) 在 HTML 文档 head 标记中插入样式 style 标记。

(3) 在 style 标记中分别定义 div、p、h4 等标记的样式。

(4) 在 body 标记中插入图层、标题字、脚本标记,编程进行函数验证,并按格式输出结果。