

第3章 网站注册模块设计与开发



- 1) 掌握 Form 表单、表单域的创建方法和属性设置方法，Form 表单的 Action 属性和 Method 属性。
- 2) 掌握 IIS 环境的配置方法，理解虚拟目录的含义。
- 3) 掌握 ASP 的变量、常量、数组、流程语句等语法知识，理解过程、函数的定义和调用方法。
- 4) 掌握 Request、Response 对象的使用方法。

早期的网页都是静态网页，即网页制作完成后，所有用户浏览的页面效果都是一样的。因此，Web 服务器也只能简单地响应浏览器发送过来的 HTTP 请求，并将存储在服务器上的 HTML 文件返回给浏览器。随着信息化程度的提高，人们更需要搜索满足个人需求的信息、处理个人事务等功能，从而动态生成 HTML 技术逐渐登上历史舞台。

最早实现动态网页效果的是 CGI 技术，随后 PHP、ASP、JSP 逐渐发展起来。ASP 基于普遍使用的 Windows 操作系统，且使用 VBScript 或 JScript 进行脚本编程，简单易学，因此被广泛使用。ASP 应用程序把代码放置在服务器端，根据用户的请求把代码翻译成静态网页，然后把静态网页发送给浏览器进行显示，动态网页的执行过程如图 3-1 所示。客户端的 HTTP 请求除了提供访问的动态网页名称外，还需要提供相应的处理数据（一般通过表单域指定），因此，动态网页开发需要和 HTML 的 Form 表单结合起来。本章以网站常用的注册模块与开发为例，学习表单的使用，ASP 开发环境的配置以及 ASP 基本编程等知识。

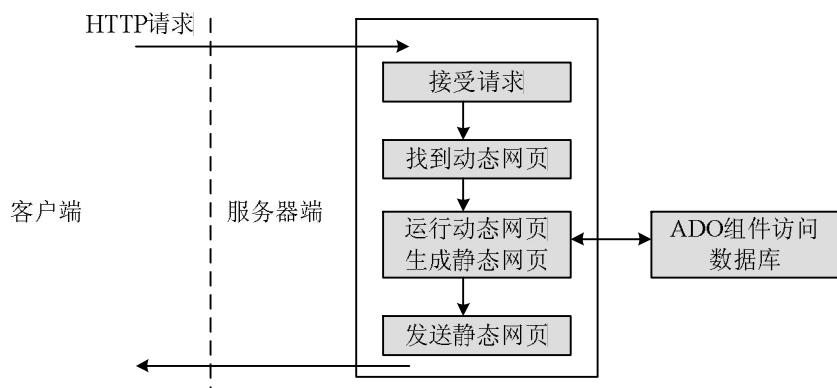


图 3-1 动态网页的运行过程

3.1 需求分析

大部分网站都针对会员提供特殊的功能，会员的注册和登录成为必不可少的功能模块。表单是实现用户注册和登录的前提，输入正确格式的信息是成功注册和登录的保证。

由于学习者初次接触动态网页编程，本实例不涉及数据库操作或文件操作。在该实例中，当用户在表单中填入相应信息并单击“提交”按钮后，转入表单处理程序，检测用户输入的信息是否合乎要求，如果检测通过，网页显示用户的注册信息，否则，网页显示错误提示并提供继续输入接口。用户名是否被注册过以及注册信息的保存等功能在学习数据库操作后才能实现。

3.2 任务分解

该实例的实现需要学习者具有以下 4 个方面的知识：

- (1) 表单的创建。
- (2) ASP 运行环境的搭建。
- (3) 表单数据的获取、向浏览器输出内容。
- (4) ASP 脚本编程。

接下来本章将把每个知识点作为一个任务进行详细讲解。

3.2.1 任务一：表单与表单域的创建

深层次的 Web 应用需要更好地实现人机交互，而不仅仅是信息在服务器和浏览器之间单向传递。即 Web 浏览器能够采集用户的输入信息，经过服务器处理后，能把处理结果通过浏览器反馈给用户。从 HTML2.0 开始，表单就被引入，为浏览器用户提供输入界面，使用户能够主动地向服务器发送请求，以此让服务器理解用户的意图并实现信息的处理。

1. 表单

表单可以收集用户的信息和反馈意见，是网站管理者与浏览者之间沟通的桥梁。表单的应用包括两个部分：一部分是 HTML 源代码用于描述表单；另一部分是脚本或应用程序用于处理提交的信息。不使用处理脚本就不能搜集表单数据。

表单在 HTML 代码中用标签对<form></form>表示。表单定义了数据采集的范围、数据采集的对象即表单控件(例如文本框、下拉列表、按钮等)，都放置在表单里面。在 Dreamweaver CS3 中可以通过单击“插入”面板中“表单”选项卡中的“表单”按钮实现表单的插入，表单插入后的效果和 HTML 源代码如图 3-2 所示。

表单的使用有 3 个要点：method 方法、action 行为和表单控件。

method 表示发送表单信息的方式。method 有两个值：get 和 post。get 方式是将表单控件的 name/value 信息经过编码之后，通过 URL 发送。而 post 则将表单的内容通过 http 发送。

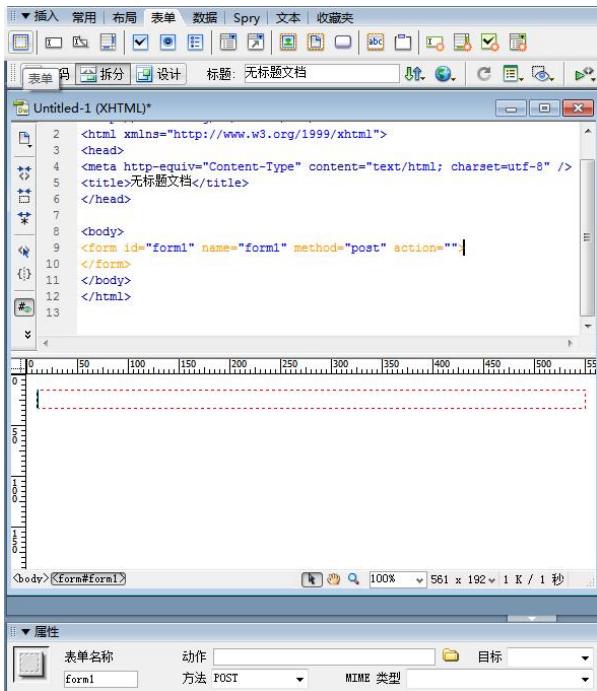


图 3-2 插入表单效果图

提示:①get 方法仅仅是将查询内容与网页的 URL 进行拼接,然后直接向服务器请求数据,post 方法会将 Form 的数据集包装在 HTTP 请求的 body 中,发送给服务器,然后向服务器请求数据。

②get 只能向服务器发送 ASCII 字符,而 post 则可以发送整个 ISO10646 中的字符(如果同时指定 enctype="multipart/form-data")。

③IE 将请求的 URL 长度限制为 2083 个字符,从而限制了 get 提交的数据长度。而对于 post,因为提交的数据不在 URL 中,所以通常可以简单地认为数据长度仅受限于服务器的设置。

④如果用 get 提交一个验证用户名和密码的 form,一般认为是不安全的。因为用户名和密码将出现在 URL 上,进而出现在浏览器的历史记录中。显然,在对安全性有要求的情况下,应该使用 post。

⑤如果提交“请求”只是从服务器端获取数据而不进行其他操作,并且多次提交不会有明显的副作用,应该使用 get。例如,搜索引擎的查询或分页显示。如果提交“请求”会产生其他操作和影响,就应该使用 post。例如:修改服务器上数据库中的数据,发送一封邮件或删除一个文件。

action 指明了处理用户填入表单信息的方法。其值通常有 2 种:(1)程序路径,如 ASP、PHP、JSP 等动态页面的相对路径或绝对路径,由指定路径的动态网页(程序)来处理表单输入的数据;(2)mailto:邮件地址,把表单输入的数据直接发送到邮箱,此方法一般不用。

action 值可以为空,表示用当前网页来处理提交的数据。动态网页和 form 表单合二为一时,可以保留空值或输入当前网页的名称。

除了 method 和 action 外,Form 表单的 enctype 属性指定了表单数据向服务器提交时所采用的编码类型,缺省值是“application/x-www-form-urlencoded”。

提示：①在向服务器发送大量的文本（包含非 ASCII 字符的文本或二进制数据）时，例如上传图片、文件时，应设置 enctype 属性值为 multipart/form-data。

②multipart/form-data 编码类型只能用于表单的 post 方法。

2. 表单控件

表单控件即表单域，是表单中收集用户各种输入信息的对象。常用的表单域有：单行文本框、多行文本框（文本区域）、密码框、单选按钮、复选框、下拉列表、提交按钮、重置按钮、普通按钮、文件域等。

表单中插入表单域的 HTML 代码为（注意，不是每个表单域都有如下属性）：

- name 属性表示表单域的名称，由编程者自定义；
- type 属性用于标识表单域类型；
- value 属性表示表单域的初始值，该值提交到服务器后被服务器提取以进行处理；
- size 属性表示表单域的宽度，单行文本框和密码框使用；
- maxlength 属性表示表单域的输入区域所能容纳的最大字符数，单行文本框和密码框使用；
- tabindex 属性表示不同表单域获得焦点的顺序；
- checked 属性表示表单域的选项的初始值为“选中”，单选按钮和复选框使用。

除了上述属性外，还有 src、align、alt 等属性，以及 onfocus、onblur、onselect、 onchange 等事件，事件主要用在表单域的客户端脚本编程中，以及时响应用户的交互操作，例如，在用户提交表单的时候，检查文本框的邮箱地址是否满足规范。

（1）单行文本框。

文本框是一种让用户输入内容的表单域，通常被用来填写单个字或者简短的回答，如姓名、地址等。

示例代码：

```
请输入字符: <input type="text" name="example1" size="20" maxlength="15">
运行效果如图 3-3 所示，注意，可以显示 20 个字符，但是最多只能输入 15 个字符。
```

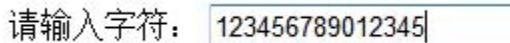


图 3-3 文本框运行效果图

（2）多行文本框。

多行文本框也是一种让用户输入内容的表单域，只不过能让用户填写较长的内容，支持文本换行。

cols 属性和 rows 属性分别定义多行文本框的宽度和高度，单位是单个字符宽度。wrap 属性定义输入内容大于文本域时显示的方式，可选值如下：

- 默认值是文本自动换行，当输入内容超过文本域的右边界时会自动转到下一行，而数据在被提交处理时自动换行的地方不会有换行符出现；
- Off（关），用来避免文本换行，当输入的内容超过文本域右边界时，文本将向左滚动，必须用 Return 才能将插入点移到下一行；

- Virtual（虚拟），允许文本自动换行。当输入内容超过文本域的右边界时会自动转到下一行，而数据在被提交处理时自动换行的地方不会有换行符出现；
- Physical（实体），让文本换行，当数据被提交处理时换行符也将被一起提交处理。

示例代码：

```
个人简介: <textarea name="grjj" cols="20" rows="2" wrap="physical"></textarea>
代码的运行效果如图 3-4 所示，注意，每行可以显示 20 个字符，自动换行。
```

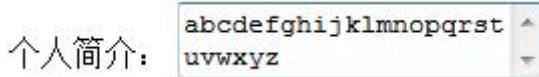


图 3-4 文本框运行效果图

（3）密码框。

密码框是一种特殊的单行文本框，仅仅是 type 变成 password，用于实现保密信息的输入，如密码。当用户输入文字时，文字会被星号或其他符号代替。

示例代码：

```
密 码: <input type="password" name="mima" size="20" maxlength="15">
代码的运行效果如图 3-5 所示，注意，输入的字符都用黑点代替。
```



图 3-5 密码框运行效果图

（4）隐藏域。

隐藏域是用来收集或发送信息的不可见元素，对于用户来说，隐藏域是看不见的。当表单被提交时，隐藏域就会将信息以名称和值对的形式发送到服务器上。隐藏域可以用来判断页面是第一次显示，还是提交后显示。

示例代码：

```
<input type="hidden" name="ExPws" value="已提交">
```

（5）单选按钮。

单选按钮让用户在一组选项中只能选择一个。name 属性定义单选按钮的名称，为保证数据的准确采集，单选按钮都是以组为单位使用的，在同一组中的单选按钮都必须用同一个名称，且 value 值必须不同。

示例代码：

```
性别: <input type="radio" name="sex" value="male" checked>男 &nbsp; &nbsp;
      <input type="radio" name="sex" value="female" >女 <br>
```

代码的运行效果如图 3-6 所示，注意，“男”和“女”只能选中一个。



图 3-6 单选按钮运行效果图

（6）复选框。

与单选按钮相反，复选框允许用户选择一个及其以上的选项。与单选按钮类似，同一组

的复选框同名，且各个复选框的 value 值不同。

示例代码：

```
喜欢吃: <input type="checkbox" name="fruit" value="apple" >苹果 &nbsp;
<input type="checkbox" name="fruit" value="orange" >橘子 &nbsp; &nbsp;
<input type="checkbox" name="fruit" value="mongo" >芒果<br>
```

代码的运行效果如图 3-7 所示，注意，可以同时选中多个水果。

喜欢吃： 苹果 橘子 芒果

图 3-7 复选框运行效果图

(7) 下拉列表。

下拉列表既可以单选，也可以多选，即可以实现单选按钮和复选框的功能。下拉列表插入后，属性面板如图 3-8 所示。如果选择“菜单”，就只能多选，如果选择“列表”，允许多选。单击“列表值”按钮，弹出如图 3-9 所示的对话框，进行列表项值的设置。其中，“项目标签”表示下拉列表中显示的列表项，“值”表示上传到服务器，能被提取并进行处理的值。单击“确定”按钮，完成列表值的设置，在“初始化时选定”区域中设置默认被选中的列表项，通过“selected”属性标识。



图 3-8 下拉列表属性面板

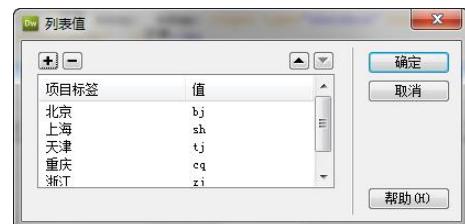


图 3-9 列表值的设置

示例代码：

```
<select name="comefrom" id="comefrom">
    <option value="bj" selected="selected">北京</option>
    <option value="sh">上海</option>
    <option value="tj">天津</option>
    <option value="cq">重庆</option>
    <option value="zj">浙江</option>
</select>
```

代码的运行效果如图 3-10 所示，注意，“北京”是默认选中的。



图 3-10 下拉列表运行效果图

(8) 按钮。

按钮共分为3种：提交按钮、重置按钮和普通按钮，type分别为“submit”、“reset”和“button”。按钮的name属性是按钮的名称，value属性是按钮上显示的提示文字。提交按钮将表单的数据提交到服务器，重置按钮将表单中各个表单域的值设置为初始值，普通按钮一般用来添加相应的事件，实现客户端脚本编程。

示例代码：

```
<input type="submit" name="submit" id="submit" value="提交">
<input type="reset" name="reset" id="reset" value="重置">
<input type="button" name="common" id="common" value="普通按钮" onclick="javascript:alert('你单击了普通按钮')">
```

代码的运行效果如图3-11所示，当单击了“普通按钮”后，会弹出对话框，显示“你单击了普通按钮”的提示信息。



图3-11 各种按钮运行效果图

(9) 文件域。

文件域用于用户上传文件。文件域和其他文本域差不多，只是它还包含了一个“浏览”按钮。用户可以通过输入需要上传的文件的路径或者单击“浏览”按钮选择需要上传的文件。

提示：使用文件域上传文件时，enctype编码类型应设置为“multipart/form-data”，提交方式应设置为“post”。

除了前面所介绍的表单域外，还有图像域等表单域。表单域的创建一般通过可视化集成开发工具，如Dreamweaver、Expression实现，这样可以提高开发效率。

3.2.2 任务二：ASP运行、编程环境的配置

ASP是一种服务器端脚本语言，只能在Web服务器环境下才能运行。把自己的计算机（无论专业版或服务器版操作系统）配置成Web服务器只需安装Microsoft的IIS组件即可，而客户端只需要一个普通的浏览器即可。ASP的工作流程如图3-12所示。

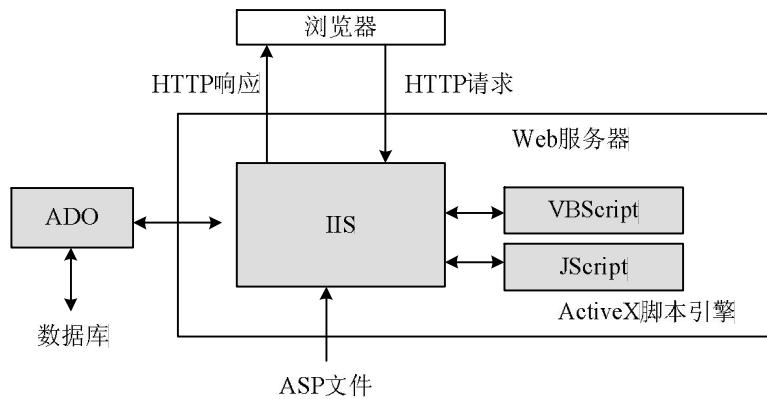


图3-12 ASP的工作流程

服务器在响应 HTTP 请求后，找到并读取 ASP 文件的内容，判断是否有 ASP 服务器端的代码需要执行。判断方法是查看源文件中是否包含 ASP 特有的`<%...%>`脚本标记或查看源文件中是否包含在服务器端运行的脚本，如`<script runat="server">`。假如源文件中包含 ASP 脚本，服务器将会解释这些代码。服务器在解释完脚本代码后，将解释结果与静态 HTML 代码进行合并，形成一个最终的网页页面，发送给客户端浏览器进行显示。

1. ASP 运行环境配置

进行 ASP 页面开发之前，必须配置好解释 ASP 脚本代码的软件环境。服务器版操作系统一般在安装时就会默认安装 IIS 组件，而专业版操作系统，如 Windows XP、Windows 7，需要手动安装 IIS，本书以专业版操作系统为例，讲解 IIS 的安装和配置。

【练习 3-1】在 Windows XP 系统中安装 IIS。

(1) 打开“控制面板”，双击“添加或删除程序”图标，打开“添加或删除程序”对话框。单击“添加/删除 Windows 组件”按钮，弹出如图 3-13 所示的“Windows 组件向导”对话框。



图 3-13 “Windows 组件向导”对话框

(2) 选中“Internet 信息服务”复选框。如果不想安装邮件服务器，可以在图 3-13 所示的对话框中单击“详细信息”按钮，在弹出的对话框中去掉“SMTP 邮件服务器”的选中状态即可。

(3) 单击“下一步”按钮，提示插入系统光盘复制相应的文件。安装完成后，系统分区中将出现如图 3-14 所示的 Inetpub 目录，网站的所有页面放在 wwwroot 子目录中。

IIS 安装完成后，Web 服务器就创建完成。为了正确浏览网页，设置 ASP 的运行环境，需要运行“Internet 信息服务”管理器对 IIS 进行设置。

【练习 3-2】在 Windows XP 系统中设置 IIS。

(1) 在“控制面板”中打开“计算机管理”对话框，双击“Internet 信息服务管理器”选项，打开“Internet 信息服务”控制台，如图 3-15 所示。

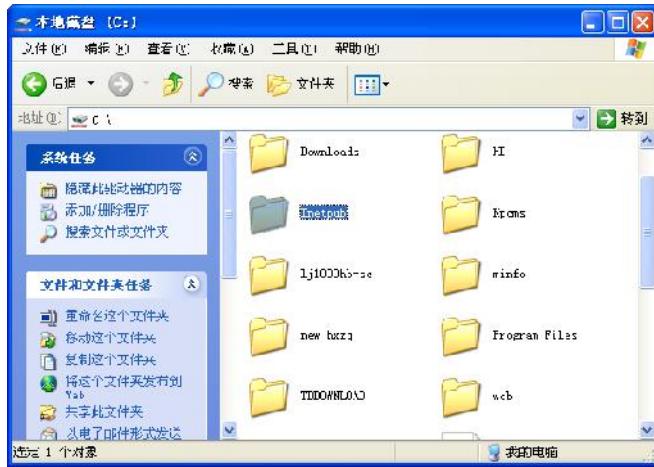


图 3-14 完成安装后自动生成 Inetpub 目录

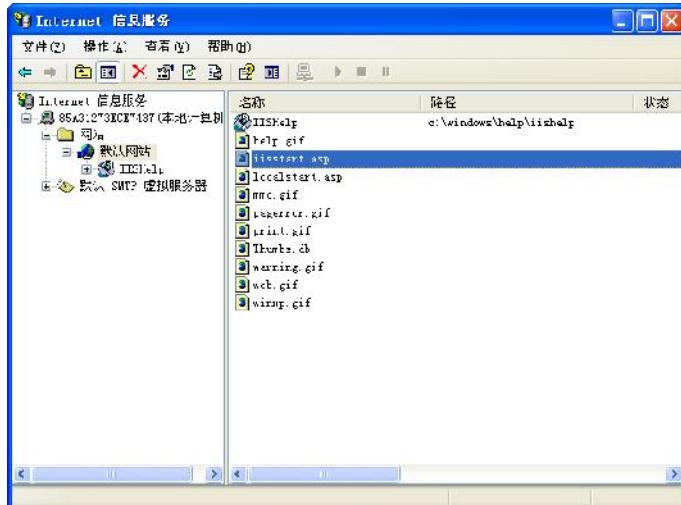


图 3-15 “Internet 信息服务”界面

(2) 在“默认网站”名称上单击右键，从弹出的快捷菜单中选择“属性”命令，打开如图 3-16 所示的属性设置对话框。

(3) 单击“网站”标签，设置网站的 IP 地址、端口号、连接超时等属性，这里保持默认值。

提示：①“IP 地址”设置网站浏览时需要输入的 IP 地址，该地址可以是局域网 IP，也可以是公网 IP。如果保持默认值，可以通过本机 IP 地址、localhost、127.0.0.1、计算机名等 4 种方式访问网站中的网页，如果设置了 IP 地址，则只能通过该 IP 地址访问。本机调试时，一般不设置 IP 地址。

②IIS 运行在一台计算机上创建多个 Web 站点，且使用同一 IP 地址。为了区分不同的网站，需要对不同的站点指定不同的 TCP 端口号。当访问网站时，需要在 IP 地址后面附上端口号（通过冒号附加端口号），如 http://172.16.11.212:8080，但默认网站除外，它使用的端口号是 80。



图 3-16 “默认网站 属性”对话框

(4) 单击“主目录”标签，打开“主目录”选项卡，如图 3-17 所示，这里保持默认值。

提示：①每个站点都有主目录，主目录是 Web 站点上发布文件的中心位置，是访问者的起点，主目录及子目录中的所有文件将自动对访问者可用。在浏览器中输入网站的 IP 地址后，将自动定位到该主目录进行文件的查找并显示。如果网站放在 D:\myweb 文件夹下面，可以在“本地路径”中输入“D:\myweb”。

②单击“配置”按钮，可以对 ASP 的 Session 和默认脚本进行设置，如图 3-18 所示。



图 3-17 “主目录”选项卡



图 3-18 “应用程序配置”对话框

(5) 单击“文档”标签，对网站的默认文档进行设置，如图 3-19 所示。默认文档指当输入网站的域名或 IP 地址而没有指明访问哪个网页的时候，网站自动搜索并且显示的网页的名称。单击“添加”按钮，输入“index.asp”，并且通过上、下箭头把该名称移到最上面。



图 3-19 “文档”选项卡

提示：默认文档的搜索顺序是从上至下的。即找到网站中存在相应名称的网页就运行，而不会继续寻找后面名称的网页。

若要从主目录外的其他目录发布信息，则可以通过创建虚拟目录实现。虚拟目录指物理上未包含在主目录中的目录，但浏览器却认为该目录包含在主目录中。例如，主目录下的 test 目录中包含 index.asp 文件，则访问该页面的方式为 http://localhost/test/index.asp；如果主目录中没有 test 目录，而虚拟目录 test 指向到 D:\myweb 目录，该目录下有 index.asp 页面，则访问该网页的方式仍然为 http://localhost/test/index.asp。

【练习 3-3】建立虚拟目录。

(1) 在“Internet 信息服务”控制台左侧的“默认网站”上单击鼠标右键，在弹出的菜单中选择“新建/虚拟目录”命令，如图 3-20 所示，打开“虚拟目录创建向导”对话框。

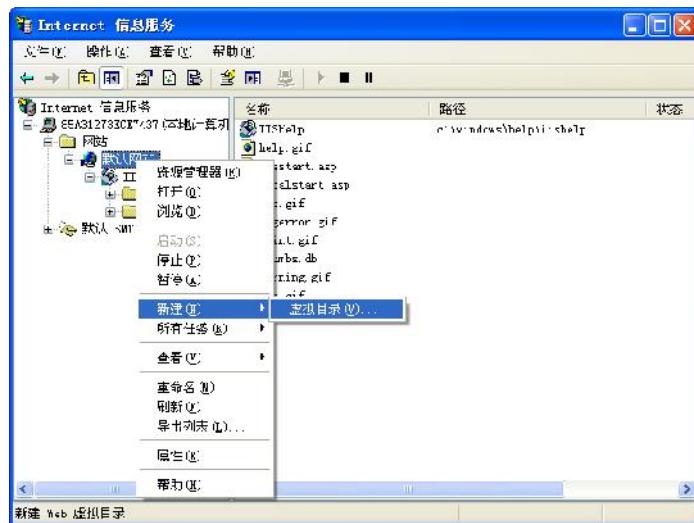


图 3-20 新建“虚拟目录”快捷菜单

(2) 单击“下一步”按钮，输入一个虚拟目录别名，也就是映射到主目录中的子目录名（访问网页的时候需要输入的名称），如图 3-21 所示。



图 3-21 设置“虚拟目录”别名

(3) 单击“下一步”按钮，输入虚拟目录对应的物理路径。

(4) 单击“下一步”按钮，进入访问权限设置步骤，如图 3-22 所示。为了保证网站的安全，只须选择前 3 个选项就可以了。



图 3-22 设置“虚拟目录”访问权限

(5) 单击“下一步”按钮，完成虚拟目录创建。

考慮到部分用户使用的是 Windows 7 操作系统，接下来简要介绍下 Windows 7 系统中 IIS 的安装和配置。

【练习 3-4】Windows 7 中 IIS 的安装和配置。

(1) 打开“控制面板”，双击“程序和功能”图标，打开“程序和功能”对话框，如图 3-23 所示。单击“打开或关闭 Windows 功能”链接，弹出如图 3-24 所示的“Windows 功能”对话框。



图 3-23 “程序和功能”对话框



图 3-24 “Windows 功能”对话框

(2) 单击“Internet 信息服务”左边的“”按钮，展开列表，勾选“Web 管理工具”和“万维网服务”，单击“确定”按钮，Windows 会更新相应功能，直至 IIS 成功安装。

提示：展开“万维网服务”确保“应用程序开发功能”下的“ASP 选项”被选中。Windows 7 集成了.NET Framework，支持 ASP.NET 的开发。

(3) 在“控制面板/管理工具”窗口中启动“Internet 信息服务管理器”，如图 3-25 所示。

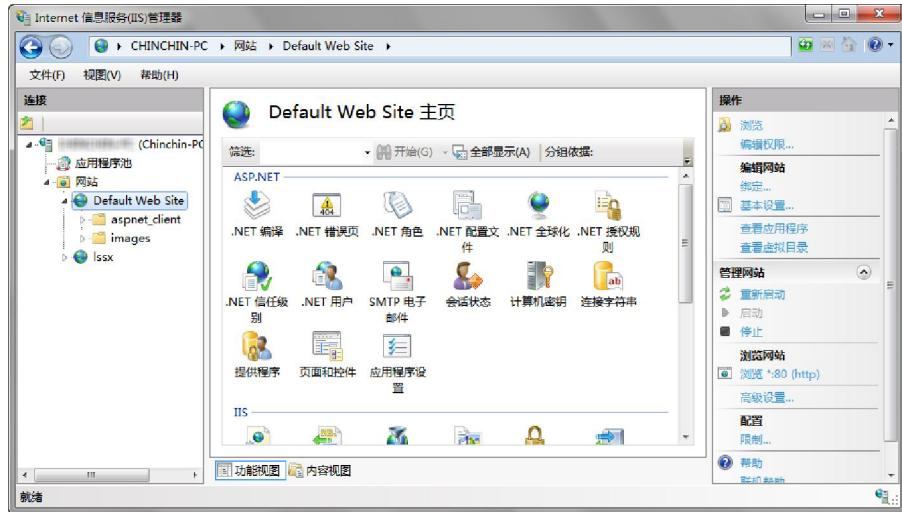


图 3-25 “Internet 信息服务 (IIS) 管理器”窗口

(4) 在图 3-25 所示的窗口右侧单击“绑定”按钮，弹出如图 3-26 所示的对话框，设置网站的 IP 地址和口号。

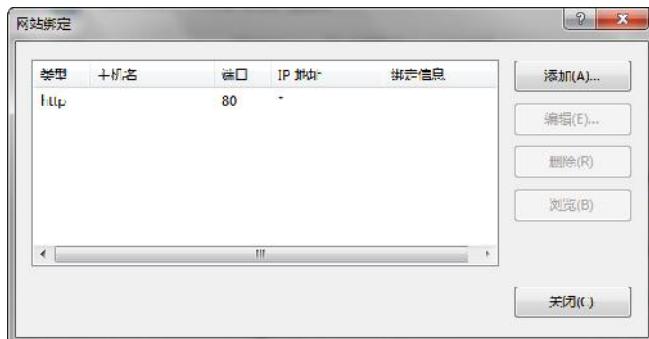


图 3-26 设置网站 IP 地址

(5) 在图 3-25 所示的窗口右侧单击“基本设置”按钮，弹出如图 3-27 所示的对话框，设置网站的主目录。

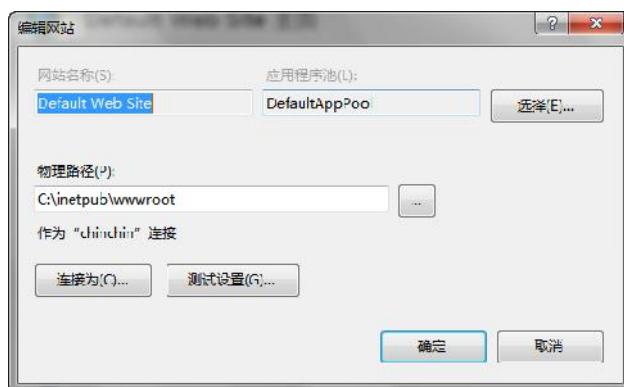


图 3-27 设置网站主目录

(6) 在图 3-25 所示的窗口中间滑动滚动条，双击“ASP”图标，如图 3-28 所示，设置 ASP 的运行参数，如图 3-29 所示，把“将错误发送到浏览器”设置为“True”。

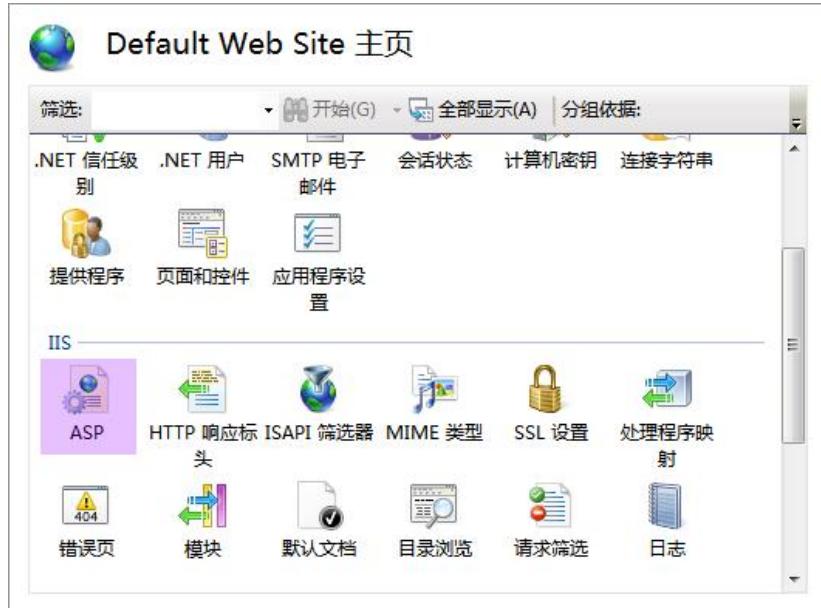


图 3-28 选中“ASP”图标以设置参数

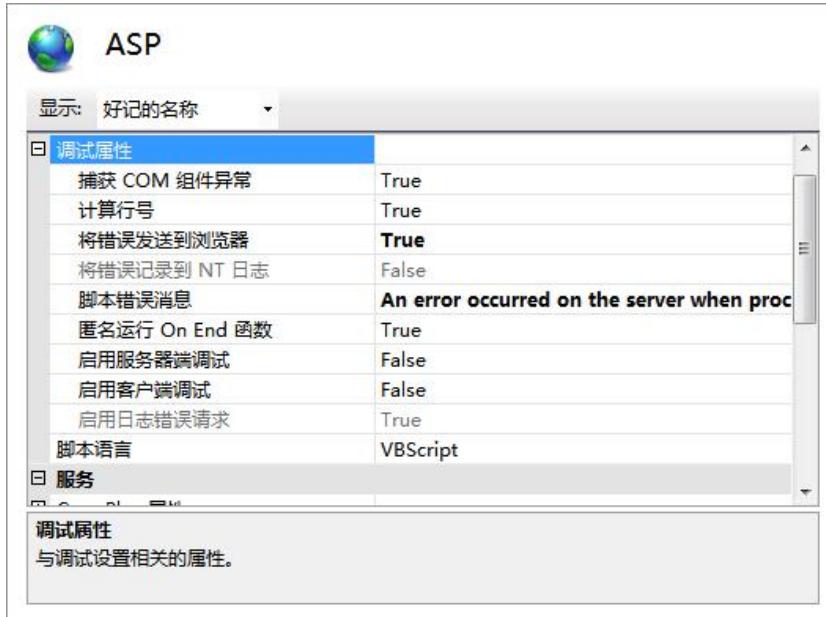


图 3-29 ASP 运行参数设置

(7) 在图 3-25 所示的窗口左边的“Default Web Site”上单击鼠标右键，打开如图 3-30 所示的对话框，添加虚拟目录，参数设置如图所示。单击“确定”按钮，完成虚拟目录的添加。注意，虚拟目录的名字前面有个文件夹的快捷方式图标，如图 3-31 所示。

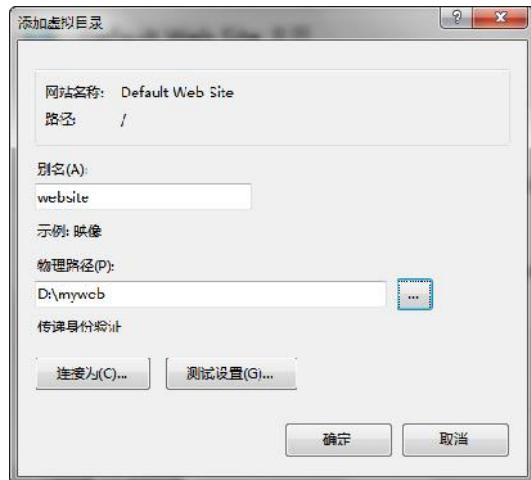


图 3-30 添加虚拟目录

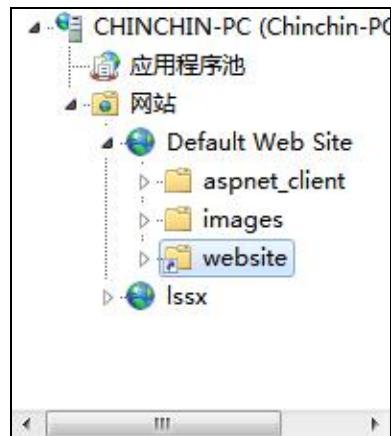


图 3-31 虚拟目录的图标效果

2. ASP 页面编辑

ASP 文件实际上是将标准的 HTML 文件扩展了一些附加特征，其中包含实现动态功能的 VBScript 或 JScript 语句。ASP 文件的后缀名是.asp，可以使用任何文本编辑软件进行编辑。常见的 ASP 集成开发环境包括 Dreamweaver、Visual Studio、ASP Express、UltraEdit 等，记事本也可以用来开发 ASP 程序。

【练习 3-5】创建 ASP 程序并浏览显示。

(1) 打开记事本程序，输入以下代码：

```
<% for i=1 to 5%>
<font size="<% response.write(i) %>"> 使用 ASP 语句控制字体大小 </font>
<% next %>
```

(2) 将文件另存为“index.asp”，存放到网站主目录中（c:\inetpub\wwwroot），如图 3-32 所示。

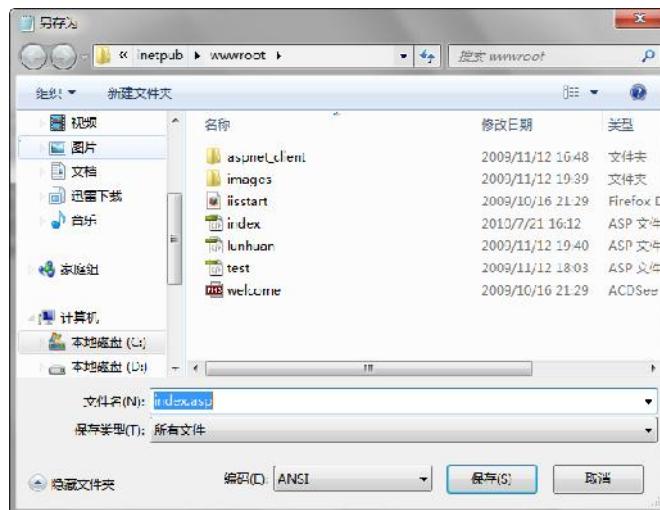


图 3-32 ASP 页面的保存

- 提示：**①ASP 代码嵌入到 HTML 标签中，使用`<%...%>`作为 ASP 代码的起止标记。
 ②在编写 ASP 程序时，字符不区分大小写，标点符号要求英文状态。
 ③ASP 代码必须一行写一条语句，且语句结束不加任何标点符号，如果一行写不完，可以加续行符“_”。
 ④ASP 代码中，使用 REM 或“!”符号来标记注释语句。

(3) 打开浏览器，在地址栏中输入“`http://localhost/index.asp`”访问刚才创建的网页，效果如图 3-33 所示。在网页上单击鼠标右键，在弹出的快捷菜单中选择“查看源文件”，如图 3-34 所示。

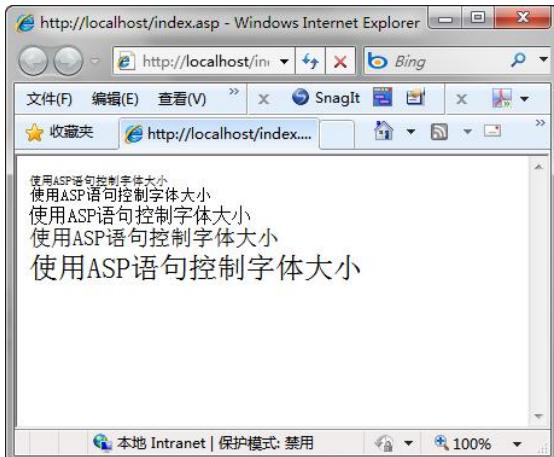


图 3-33 ASP 页面运行效果图

```

1  <font size=1> 使用ASP语句控制字体大小 </font><br>
2  <font size=2> 使用ASP语句控制字体大小 </font><br>
3  <font size=3> 使用ASP语句控制字体大小 </font><br>
4  <font size=4> 使用ASP语句控制字体大小 </font><br>
5  <font size=5> 使用ASP语句控制字体大小 </font><br>
6
7
8
9
10
11

```

图 3-34 ASP 页面的 HTML 源文件

程序通过在 HTML 代码中插入`<% for i=1 to 5 %>...<% next %>`循环语句来控制显示的次数。另外，通过`<font size=<%response.write(i)%>>`语句来控制每次显示的文本的大小随 i 值的变动而变动。同时，ASP 代码在被脚本引擎解释后，变成 HTML 代码，通过浏览器显示出来。

提示：①如果在站点配置时设置了默认文档“index.asp”，则在浏览页面时不需要在 IP 地址后面加上具体的网页名称，否则需要加上网页名称才能正确浏览。

②如果在站点配置时为网站绑定了 IP 地址，则浏览网页时必须通过 IP 地址方式实现。否则，还可以通过 `http://localhost`、`127.0.0.1`、计算机名等方式实现。

3.2.3 任务三：VBScript 基础知识

知识点：变量的定义，运算符的使用，流程控制语句的使用，系统函数的使用，自定义函数的定义与调用。

前面一节讲述了 ASP 页面编程的方法和注意事项，但是，要让 ASP 来处理用户输入的各种信息，目前所掌握的知识是远远不够的。为此，有必要详细了解 ASP 的编程脚本语言——VBScript。

VBScript 是一种弱类型高级语言，只有一种数据类型，即变体类型（Variant 类型）。它不强制变量先声明再使用。但是，它也具有高级语言的一般特征：常变量、数组、流程控制结构、函数。接下来，本节就从这几个方面展开。

1. 数据类型

VBScript 只有一种数据类型，因此，声明变量时不需要指定数据类型，也就是说变量可以接受任何类型的数据。但是，为了方便处理各种不同的数据，它提供了多种子类型，如表 3-1 所示。VBScript 代码会根据代码的上下文自动转换数据的子类型。

表 3-1 Variant 类型的子类型

子类型	说明
Boolean	逻辑值，只有 True、False 两种取值
Byte	字节型，0~255
Integer	整型，-32768~32767
Currency	精确的定点浮点数类型，常用于货币运算
Long	长整型
Single	单精度浮点数
Double	双精度浮点数
Date (Time)	日期、时间类型，常量左右两边加“#”
String	字符串类型，最大长度 20 亿个字符
Object	对象类型
Error	错误对象
Empty	变量没有初始化就为 Empty
NULL	不包含任何有效数据

在高级语言中，数据类型定义了数据在内存中存储的长度，所能表示的数据的范围以及能支持的运算。数据要么在写程序时直接给出（称为常量，程序运行期间不能改变值），要么保存在某一个内存单元中，通过一个名字来标识，内存中保存的数据可以变化（称为变量，程序运行期间可变）。常量在定义时就可以判断其类型，变量在获得相应的值后就变为相应的子类型（因为总是变体类型）。

2. 常量

常量分为直接常量（程序中直接给出的量）和符号常量（拥有固定数值的名称）。符号常量可以代表字符串、数字等不变的量，一经声明，程序执行期间不可变。例如：

```
<% Const PI=3.1415926
   Const constString="中国"
   Const dateTemp=#2010-10-12#
%>
```

提示：①常量、变量的命名规则类似于一般高级语言，可以包含数字、字母和下划线，必须以字母开始，不能包含标点符号和运算符，不能超过255个字符，不能和关键字同名，如dim、end、sub、response等。

②如果要在多个ASP文件中共享一些常量，可以把常量定义放在单独的文件中，然后使用文件包含语句把该文件包含进来即可。

③除了用const关键字声明常量外，直接给出的数值、字符串、日期等数据都是常量。

3. 变量

变量，就是存储在内存中的数据的地址的名称，和常量的最大区别是该地址的内存单元中存储的数据可以改变。

变量的定义使用Dim关键字实现，可以一次定义多个变量，使用逗号分隔。例如：

```
<% Dim strName
   Dim dateRQ, intAge
   dateTemp=NOW()
%>
```

提示：①ASP中，变量可以不定义直接使用，例如上例的变量dateTemp。

②如果在所有ASP语句之前使用了Option Explicit语句，则所有的变量必须先定义再使用，即强制变量声明。

③变量在定义的时候不能同时赋初值，这点和其他高级语言不一样。

④为了使变量的名字含义，同时了解变量的类型，一般用类型的前3个小写字母做前缀，然后使用驼峰式命名方式，如strName、intAge、blnMarried、objCurrent等，他们分别表示字符串类型、整型、布尔型和对象类型。

4. 数组

数组可以理解为把同种类型的多个变量组合在一起，形成一个整体，而每个变量（成员）的访问依据索引值（下标）来进行访问。这样，可以使用统一的方式来访问多个变量，便于循环操作。

```
<%
  Dim arrA(3)          '声明一个元素个数为4的数组arrA
  arrA(0)=1            '对第一个数组元素赋值为1
  arrA(3)=4            '对第四个数组元素赋值为4
  sum=arrA(0)+arrA(3)  '引用数组元素的值，将其和赋值给sum变量
%>
```

提示：①数组的下标是从0开始的，定义时括号里面的数值指定数组的最大元素的下标。如dim a(3)，表示数组a有4个元素。

②数组名代表数组这个整体，不能对所有元素进行整体访问，只能使用数组名+下标的形

式访问一个数组元素。

③如果事先不能确定数组的长度，可以定义一个动态数组，在运行时根据需要，使用 redim 语句重定义数组长度。定义动态数组的方法：定义数组时括号里面保留为空。

④Redim 定义数组长度时，数组原来保存的值会清空，为保留原来的值，可以使用 preserve 关键字。例如：Redim Preserve a(10)。

⑤二维数组的定义方式类似，如 dim a(3,4)，表示数组 a 共有 20 个元素。

5. 运算符

运算符是把某种类型的数据进行运算的符号。VBScript 只有 4 种类型的运算符：算术、比较、逻辑和连接运算符。当一个表达式（数据和各种运算符组成的式子）中有多种运算符时，他们是有运算的先后次序的。各种运算符的功能、优先级如表 3-2 所示。注意：括号可以改变优先级。

表 3-2 运算符的优先级

类别	运算符	功能描述	优先级
算术运算符	$^$	指数运算	1
	*	乘	
	/	除	
	\	整除	2
	Mod	求余	3
连接运算符	+	加	
	-	减	4
关系运算符	&	连接 2 个字符串	
	+	连接字符串和其他子类型数据	5
	>、>=、<、<=	大于、大于等于、小于、小于等于	
	=	等于	6
逻辑运算符	\diamond	不等于	
	IS	比较 2 个对象是否相同	
	Not	逻辑非，真变假，假变真	7
	And	逻辑与，2 个表达式同为真才为真	8
	Or	逻辑或，2 个表达式至少一个为真就为真	9
	Xor	逻辑异或，2 个表达式的逻辑值刚好不同为真	10
	Eqv	逻辑等，2 个表达式的值相同时为真	11
	Imp	逻辑蕴含，仅当第一个表达式的值为真时才为真	12

算术运算符和连接运算符是最常用的。注意：“+”号左右两边同时为字符串，才成为连接运算符，否则，是算术运算符。

```
<%
sum=a^2+b^2      'sum 的值为 a 和 b 的平方和
strTemp="Hel"&"lo!"  'strTemp 的值为 "Hello! "
strTemp="80"+5    'strTemp 的值为 85
```

```

strTemp="80"+"5"      'strTemp 的值为 "805"
%>

```

比较运算符和逻辑运算符主要用在条件表达式中，他们的结果都是逻辑值。

<%	
sum=a>b	'如果 a 大于 b，则 sum 为 True，否则为 False。关系运算符常用于数值、日期的比较。
result=a And b	'如果 a 和 b 同时为 True，则 result 为 True，否则为 False。
result=a Or b	'如果 a 和 b 至少有一个为 True，则 result 为 True，否则为 False。
%>	

6. 流程控制语句

默认情况下，脚本中的代码都是按书写的先后顺序执行的。但在实际应用中，有些代码有选择地执行或者重复执行，就要用到流程控制语句。VBScript 中有两类流程控制结构，选即选择结构和循环结构，选择结构包括 If 语句和 Select Case 语句，循环结构包括 do 循环、for 循环、for each 循环和 while 循环。

(1) If 语句。

If 语句是根据条件满足与否来决定哪部分代码被执行。在 ASP 程序中，常常需要对用户的输入信息进行判断，如用户注册、登录时，判断用户的信息是否齐全、密码是否正确等。If 语句有 3 种书写形式，如表 3-3 所示。

表 3-3 if 语句的 3 种写法

写法 1	写法 2	写法 3
If 条件表达式 then 真分支执行语句 End if	If 条件表达式 then 真分支执行语句 Else 假分支执行语句 End if	If 条件表达式 1 then 条件 1 为真执行语句 Elseif 条件表达式 2 then 条件 2 为真执行语句 Elseif ... 条件 n 为真执行语句 Else 所有条件都不满足执行语句 End if
说明：条件满足（为 True），就执行“真分支执行语句”，否则，执行 if 语句后面的语句	说明：条件满足，就执行“真分支执行语句”，否则，执行“假分支执行语句”，二选一执行	说明：如果条件 1 满足，就执行“条件 1 为真执行语句”，如果条件 1 不满足，接下来判断条件 2 是否满足，以此类推。如果所有条件都不满足，则执行 Else 后面的分支语句。 注意：条件 2 是在条件 1 不满足的情况下继续判断的

【练习 3-6】利用条件语句，根据学生的成绩给出相应的提示信息。

清单：3-1.asp

```
<%
```

```
Dim Grade  
Grade=86
```

这里为了简单，直接赋值了，一般来说应该是传过来的参数，比如从数据库中读出，或由程序计算得出。

```
If Grade>=85 Then
    Response.Write("太不可思议了，太棒了！")
Elseif Grade>=70 And Grade<85 Then
    Response.Write "不错啊，继续努力吧！"
Elseif Grade>=60 And Grade<70 Then
    Response.Write("继续努力")
Else
    Response.Write "加加油吧"
End if
%>
```

程序的运行结果如图 3-35 所示。



图 3-35 程序运行效果图

(2) Select Case 语句。

Select Case 语句是多条件判断的 If 语句的另外一种实现方式。它把条件空间映射到可数个具体的值上,如果具体的条件经过变换后和某个值相等,就执行相应的分支语句。Select Case 语句的语法如表 3-4 所示。

表 3-4 Select Case 语句的语法

写法	说明
<pre>Select Case 变量或表达式 Case 常量 1 执行语句 1 Case 常量 2 执行语句 2 Case 常量 n 执行语句 n Case else 执行语句 n+1 End select</pre>	首先对表达式进行运算,一般为数学运算或字符串运算,然后把表达式的值依次与常量 1、常量 2……常量 n 进行比较,如果与某个常量值相等,就执行后面的语句,执行完毕就跳出 Select Case 语句。否则,执行 Case else 后面那条语句

【练习 3-7】利用 Select Case 语句,根据学生的成绩给出相应的提示信息。

清单: 3-2.asp

<%

```

Dim Grade
Grade=60      '这里为了简单，直接赋值了，一般来说应该是传过来的参数，  

               '比如从数据库中读出，或由程序计算得出。
Select Case grade\10
Case 10
    Response.Write "Great!"
Case 9
    Response.Write("Well done!")
Case 8
    Response.Write "Keep on!"
Case 7
    Response.Write("Try your best!")
Case else
    Response.Write "加加油吧"
End select
%>

```

程序的运行结果如图 3-36 所示。



图 3-36 程序运行效果图

(3) Do 循环。

Do...Loop 循环是一种条件型循环，当条件为真时一直循环或当条件变为真之前一直循环。

Do 循环有 3 种书写形式，如表 3-5 所示：

表 3-5 Do 循环的语法

写法 1	写法 2	写法 3
Do while 条件表达式 重复执行的语句 Loop	Do 重复执行的语句 Loop While 条件表达式	Do 重复执行的语句 Loop Until 条件表达式
先判断条件再决定是否循环，如果条件为真，就执行循环体，否则，退出整个循环。	先执行循环体再决定是否继续执行循环体，如果条件为真，就执行循环体，否则，退出整个循环	先执行循环体再决定是否继续执行循环体，如果条件为假，就执行循环体，否则，退出整个循环。

【练习 3-8】利用 Do 循环，输出循环执行的具体过程。

清单：3-3.asp

```
<%
intNum=1
```

```

do while intNum<7
    response.write "循环语句正在执行第"&intNum&"次循环<br>"
    intNum=intNum+1
Loop
%>

```

程序的运行结果如图 3-37 所示。

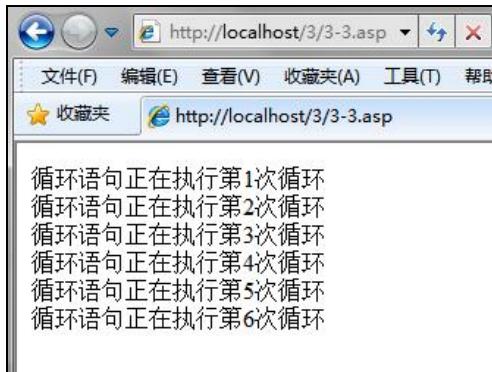


图 3-37 程序运行效果图

提示：使用 Exit Do 语句可以提前结束循环，该语句一般和 if 语句结合使用。

(4) For 循环。

For...Next 循环是一种固定次数的循环。For 循环中有一个循环控制变量，循环控制变量值的增加或减少控制循环的执行。For 循环的语法如表 3-6 所示。

表 3-6 For 循环的语法

语法	说明
For 循环变量=初始值 to 结束值 step 步进值 重复执行的语句 Next	初始值和结束值都可以是：常量、变量或表达式。 步进值可以为正，可以为负，默认为 1
执行步骤：	
(1) 将循环控制变量设置为初始值。	
(2) 测试循环控制变量是否大于结束值（若步进值为负，则判断是否小于结束值）。若是，退出循环，否则执行循环体 1 次。	
(3) 执行完重复执行的语句，到 next 时，将循环控制变量的值与步进值相加。	
(4) 跳转到第 2 步，继续执行	

【练习 3-9】利用 For 循环，求 1~100 的奇数的和。

清单：3-4.asp

```

<%
    Dim intNum,i,intSum
    intNum=1
    intSum=0
    For i=1 to 100 Step 2
        Response.Write "循环语句正在执行第" & intNum & "遍循环" & "<br>"

```

```

Response.Write "第" & intNum & "个欲相加的 i 值为" & i & "<br>"
Response.Write "SUM=" & intSum & "+" & i & "=" & intSum+i & "<br>"
intSum=intSum+i
intNum=intNum+1

```

Next

%>

程序的运行结果如图 3-38 所示。

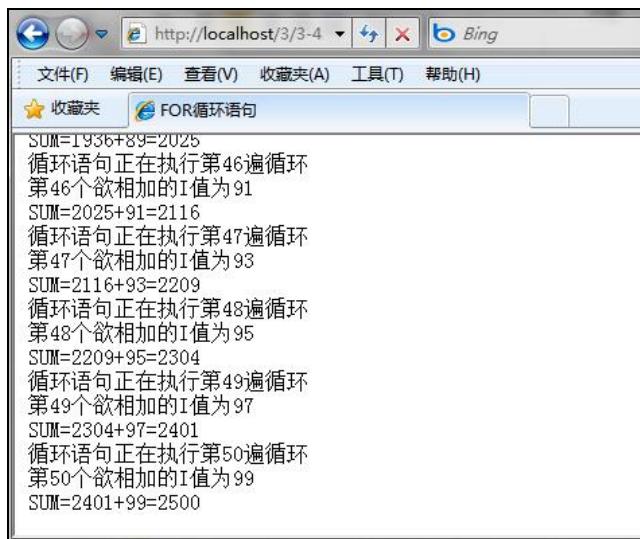


图 3-38 程序运行效果图

提示：使用 Exit For 语句可以提前结束循环，该语句一般和 if 语句结合使用。

(5) For each 循环。

For each 循环是一种特殊的循环方式，只针对数组或集合使用，把数组中或集合中的每个元素都遍历一下，因此，不需要指定循环的次数。For each 循环的语法如表 3-7 所示。

表 3-7 For each 循环的语法

语法	说明
For each 循环变量 in 集合或数组 重复执行的语句 Next	只要集合中或数组中还有元素没有遍历，循环体就要重复执行。 使用 exit for 语句可以提前退出循环

输出数组中的每个元素的示例代码如下：

```

<%
Dim a(6)
For i=0 to 6
a(i)=i
for each item in a
    Response.write a(i) & "<br>"      '在输出完一个元素后，换行
next
%>

```

(6) While 循环。

While 循环不太常用，功能与 Do...Loop 循环类似。例如，求 $1+2+\dots+9$ 的示例代码如下：

```
<% dim i,sum
    i=1
    sum=0
    while i<10
        sum=sum+i
        i=i+1
    wend
%>
```

(7) 循环嵌套。

循环嵌套是指循环体中再包含一个循环的程序结构。循环嵌套用在二维结构的数据组织方式中，如二维数组、矩阵。

【练习 3-10】利用循环嵌套，输出九九乘法表。

清单：3-5.asp

```
<%
    Dim row,col
    For row=1 to 9
        For col=1 to 9
            Response.write row & "x" & col & "=" & row*col & "&nbsp;&nbsp;"'
            if eval("col=row") then      '如果列数等于行数，此行不需要再输出了
                exit for
            end if
        next
        response.write "<br>"           '一行输出完后要换行
    next
%>
```

程序的运行结果如图 3-39 所示。

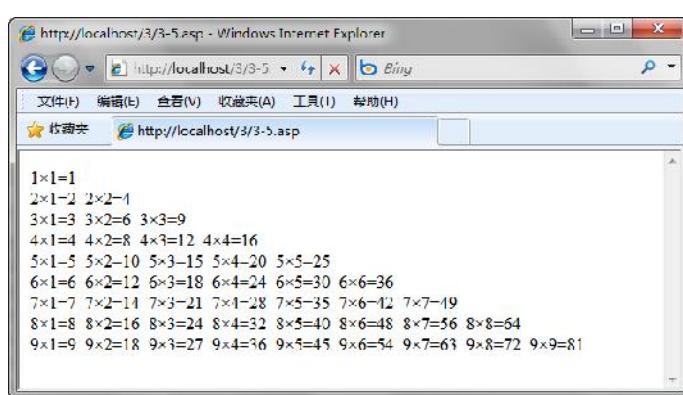


图 3-39 程序运行效果图

7. 过程

过程是用来执行特定任务的独立的程序代码。使用过程可以将程序划分为一个个较小的逻辑单元，过程中的代码能够被反复调用，这样可以减少不必要的重复。可以将过程定义放在

调用该过程的同一个 ASP 文件中，也可以将常用过程放在共享的文件中，并使用#include 命令将该文件包含在调用该过程的 ASP 文件中。

根据是否具有返回值，可以将过程分为 sub 子过程（子程序）和函数。子过程没有返回值，而函数具有返回值。

（1）Sub 子过程。

Sub 子过程是一种可以获取参数，执行一系列语句以及可改变其参数值的独立过程。Sub 过程可以使用参数（由调用过程传递的常数、变量或表达式），参数用于在调用过程和被调用过程之间传递信息。如果 Sub 过程无任何参数，则 Sub 过程必须包含空括号，其语法和调用方法如表 3-8 所示。

表 3-8 Sub 子过程的定义和调用

定义	调用
Sub 子过程名（参数 1, 参数 2, …） … End sub	调用方法： ● Call 子过程名（参数 1, 参数 2, …） ● 子过程名 参数 1, 参数 2, …
Sub 子过程名（） … End sub	调用方法： ● Call 子过程名（） ● 子过程名

提示：使用 Exit Sub 可以提前退出 Sub 子过程。

【练习 3-11】使用 Sub 过程求 a (a=3) 和 b (b=4) 的平方和。

清单：3-6.asp

```
<%
    Dim m,n                               'm 和 n 为实际参数，见下面说明
    m=3
    n=4
    Call pingfanghe(m,n)                  '调用子程序，显示结果
    '下面是子程序，用来计算 a 和 b 的平方和
    Sub pingfanghe (a,b)                  'a 和 b 是形式参数，见下面说明
        Dim sum
        sum=a^2+b^2
        Response.Write "a 和 b 的平方和是：" & CStr(sum)
    End Sub
%>
```

程序的运行结果如图 3-40 所示。



图 3-40 程序运行效果图

程序说明：

- 1) 此处的 m 和 n 是实际参数，而过程中的 a 和 b 是形式参数。运行过程中，由实际参数将具体的值传给形式参数，这里由 m 和 n 传给 a 和 b。
- 2) 实际参数和形式参数按位置对应，而不是按名字对应。实际参数和形式参数可以相同，也可以不同。
- 3) 实际参数可以是常量、变量和表达式，但要注意数据类型。
- 4) 关于变量的作用域（可以使用的范围）：m 和 n 在整个 ASP 文件中都可以赋值和引用；而在过程中定义的变量 sum，只能在过程中使用，过程外无效。

（2）Function 函数。

函数和子过程一样，也是用来完成特定功能的独立的程序代码，可以读取参数、执行一系列语句并改变参数的值。函数在调用时，可以返回一个值，类型为 Variant 类型，是通过函数体赋值给函数名实现的。函数的定义和调用方法如表 3-9 所示。函数的参数传递的方法和过程、sub 子过程一致。

表 3-9 Function 函数的定义和调用

定义	调用
Function 函数名（参数 1, 参数 2, …） 语句 函数名=表达式 End Function	调用方法： 变量=函数名（参数 1, 参数 2, …）
Function 函数名（） 语句 函数名=表达式 End Function	调用方法： 变量=函数名（）

提示： 使用 Exit Function 可以提前退出 Function 函数。

【练习 3-12】 使用 Function 函数求 a 和 b 的平方和。

清单：3-7.asp

```
<%
    Dim m,n,sum                                'm 和 n 为实际参数
    m=3
    n=4
    sum=pingfanghe(m,n)                         '调用函数
    Response.write "a 和 b 的平方和是：" & CStr(sum)
    '下面是函数，用来显示 a 和 b 的平方和
    Function pingfanghe (a,b)                    'a 和 b 是形式参数
        Dim sum
        sum=a^2+b^2
        pingfanghe=sum
    End Function
%>
```

程序的运行结果如图 3-41 所示。

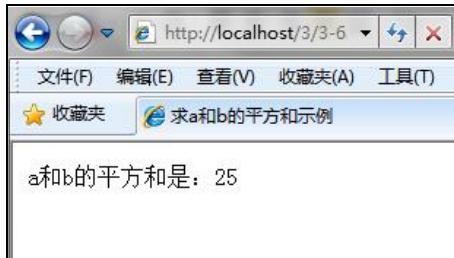


图 3-41 程序运行效果图

为了便于运算，系统提供了大量的函数，称为系统函数，而自己定义的 Function 函数称自定义函数。常见的系统函数的功能和用法参见附录。

3.2.4 任务四：Request 和 Response 对象

表单为服务器和客户端的动态交互提供了可能，并且信息传递到服务器有两种方法：Get 和 Post。服务器要获得客户端传递过来的信息必须通过 Request 对象实现，处理完成后通过 Response 对象返回客户端。

1. Request 对象

Request 对象可以收集和处理用户通过 HTTP 请求传递的信息，包括表单提交的信息（Get 或 Post 方法）、Cookies 数据和用户认证等。

Request 对象有 3 类成员：集合、属性和方法，如表 3-10 所示，Request 可以基于 3 类成员进行数据的获取。

表 3-10 Request 对象成员的功能说明

类别	名称	功能
集合	QueryString	从 HTTP 查询字符串变量中获取值，HTTP 查询字符串位于网址中间号后面，以“变量名=变量的值”的形式提供。查询字符串可以人为指定，也可以使用 form 表单的 Get 方法提交实现
	Form	从 HTTP 请求正文获取变量的值，可以使用 form 表单的 post 方法提交实现
	Cookies	从 HTTP 请求中发送的 cookie 值中获取变量的值
	ServerVariables	获取服务器端环境变量信息
属性	ClientCertificate	获取客户端浏览器的身份验证信息
	TotalBytes	获得客户端提交数据的字节数
方法	BinaryRead	以二进制方式获取客户端 post 数据

Request 对象的使用方法是：Request.[数据集合|属性|方法]（变量或字符串）。

例如：

```
<% unamea=equest.Form("username")
   unameb=Request.Querystring("username")
%>
```

(1) QueryString 集合。

使用 Get 方法提交表单时，表单数据不是被当做一个单独的包发送，而是附加在 URL 的查询字符串中一起被提交到服务器。查询字符串中以“变量名=变量值”的形式包含提交的数据，多个数据通过“&”符号分隔，因此，QueryString 集合中的数据也以“变量名=变量值”的形式组织。

【练习 3-13】 使用 QueryString 集合实现简易计算器。

清单：3-8.htm

```
<html>
<head>
<title> Form 获取方法示例 1 </title>
</head>
<body>
<form name="test" method="get" action="3-8.asp">
    var1      <input name="var1" type="text" id="var1" size="10">
    +
    var2      <input name="var2" type="text" id="var2" size="10">
    <p>
        <input type="submit" value="确定">
    </form>
</body>
</html>
```

清单：3-8.asp

```
<%
    Dim a,b,c
    a=Request.QueryString("var1")
    'a=Request("var1")
    b=Request.QueryString("var2")
    c=CInt(a)+CInt(b)
    Response.Write a&"+"&b&"=" & CStr(c)
%>
```

程序的运行结果如图 3-42、图 3-43 所示。



图 3-42 简易加法表单

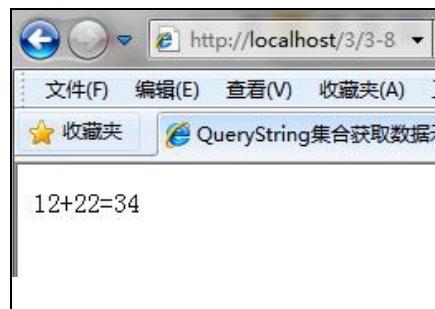


图 3-43 使用 QueryString 集合获取数据

程序说明：

- 1) Request 对象获取数据的语法“Request.数据集合（变量或字符串）”中，变量或字符串代表的是 QueryString 集合中的变量名（变量名可以以变量的值或字符串常量的形式提供），由于 QueryString 集合的数据组织与查询字符串匹配，因此，变量名对应的是表单中表单域的名字。
- 2) 获取数据时，集合名可以省略，如注释中的代码：a=Request("var1")。
- 3) 如果省略集合名，ASP 会自动按照 QueryString、Form、Cookies、ServerVariables、ClientCertificate 的顺序在集合中进行检查。
- 4) 3-8.asp 中的页面并不是一定要在表单数据提交后才能运行，在浏览器地址中输入“<http://localhost/3/3-8.asp?var1=12&var2=22>”也能得到如图 3-43 所示的结果。因此，人为提供查询字符串与 Get 方法提交表单数据产生的查询字符串的效果一致，编程时可以在超级链接后面添加查询字符串以在页面间传递数据。

在上面的例子中，客户端在单击 3-8.htm 中的“确定”按钮后，就可以将信息发送到 3-8.asp，在 3-8.asp 页面中完成信息的处理。在实际编程中一般不这样处理，因为网站中网页的数量会成倍增加，需要把表单和处理文件合二为一，也就是表单提交后交由自身处理，而不是由另外一个单独的页面处理。实现的语法很简单，表单页面的名称由“.htm”变为“.asp”，且 Form 的 action 属性设置为本页面的名称或保留为空。实现的关键点：判断表单（网页）是单纯显示还是在数据提交到该页面后显示。

【练习 3-14】表单与表单处理页面的融合。

清单：3-9.asp

```
<html>
<head>
    <title> 简单计算器 </title>
</head>
<body>
    <form name="test" method="Get" action="">
        第一个数 <input name="var1" type="text" size="5"> +
        第二个数 <input name="var2" type="text" size="5">
        <input name="tijiao" type="submit" id="tijiao" value="确定">
    </form>
    <% If Request.QueryString("var1")<>"" and Request.Querystring("var2")<>"" Then
    '判断表单是否提交方法 1
    if Request.QueryString("tijiao")<>""      '判断表单是否提交方法 2
    if Request.QueryString.count<>0            '判断表单是否提交方法 3
        Dim a,b,c
        a=Request.QueryString("var1")      '获取操作数 1
        b=Request.QueryString("var2")      '获取操作数 2
        c=CInt(a)+CInt(b)                '把操作数 1 和 2 转换成整型并相加
        Response.Write a+"&"&b+"&=" & CStr(c)
    Else
        Response.Write "请输入两个整数后按确定按钮"
    End If %>
</body>
</html>
```

程序的运行结果如图 3-44、图 3-45 所示。

图 3-44 第一次运行效果图

图 3-45 表单提交后效果图

程序说明：

- 1) 方法 1 是通过判断能否获取文本框中的数据实现该页面是否提交后显示的判断。
- 2) 表单一般都包含提交按钮，如果能获取提交按钮的值，就表示该页面是提交后显示。
- 3) 方法 3 是通过判断 `QueryString` 集合中变量值对的个数实现该页面是否提交后显示的判断。只要表单中包含表单域，且是 Get 方法提交，`QueryString` 集合就非空。
- 4) 集合一般具有 `Count` 属性，以此判断集合中元素或变量值对的个数。

(2) Form 集合。

`Get` 方法的优点是方便地向服务器传递信息，缺点是不能传递长而复杂的数据，且安全性不高，对于多媒体数据上传也无能为力，此时应使用 `Post` 方法。`Post` 方法在 HTTP 请求内发送数据，服务器从 `Request` 对象的 `form` 集合中获取数据。

使用 `Request` 对象的 `Form` 集合获取数据的方法与 `QueryString` 集合一致，仅针对不同的表单的提交方法。下面以简单个人信息注册表为例，讲述数据获取后的处理方法。

【练习 3-15】简单个人信息注册的实现。

清单：3-10.asp

```
<html>
<head>
    <title> Request 的 Form 获取方法用法示例</title>
    <link href="css.css" rel="stylesheet" type="text/css">
</head>
<body>
<%  if request.Form.Count=0 then  '如果该页面不是提交后显示  %>
<h2 align="center">请填写个人信息</h2>
<form action="" method="post" name="personal">
<table width="526" border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#CCCCCC">
<tr> <td width="131" height="30" bgcolor="#FFFFFF" class="txt">姓    名: </td>
        <td width="392" height="30" bgcolor="#FFFFFF">
            <input name="name" type="text" class="txt" id="name" size="25"></td>
        </tr>          <tr>
        <td height="30" bgcolor="#FFFFFF" class="txt">密    码: </td>
        <td height="30" bgcolor="#FFFFFF">
```

```

<input name="pwd" type="password" class="txt" id="pwd" size="28"></td>
</tr>          <tr>
<td height="30" bgcolor="#FFFFFF" class="txt">性 别: </td>
<td height="30" bgcolor="#FFFFFF" class="txt2">
<input name="sex" type="radio" id="radio" value="male" checked> 男
<input type="radio" name="sex" id="radio2" value="female"> 女</td>
</tr>          <tr>
<td height="30" bgcolor="#FFFFFF" class="txt">爱 好: </td>
<td height="30" bgcolor="#FFFFFF" class="txt2">
<input name="hobby" type="checkbox" id="hobby" value="movie"> 电影
<input name="hobby" type="checkbox" id="hobby" value="music"> 唱歌
<input name="hobby" type="checkbox" id="hobby" value="internet"> 上网
<input name="hobby" type="checkbox" id="hobby" value="dance"> 跳舞</td>
</tr>          <tr>
<td height="30" bgcolor="#FFFFFF" class="txt">来 自: </td>
<td height="30" bgcolor="#FFFFFF">
<select name="come" class="txt" id="come">
<option value="bj">北京</option>
<option value="tj">天津</option>
<option value="sh">上海</option>
<option value="cq">重庆</option>
<option value="sc">四川</option>
<option value="ah">安徽</option>
<option value="zj">浙江</option>
</select>          </td>          </tr>          <tr>
<td height="30" bgcolor="#FFFFFF" class="txt">简 述: </td>
<td height="30" bgcolor="#FFFFFF">
<textarea name="intro" cols="40" rows="5" class="txt" id="intro"></textarea></td>
</tr>          <tr>
<td height="30" colspan="2" bgcolor="#FFFFFF" class="txt"><input name="submit" type="submit" value="提交">
<input name="reset" type="reset" value="重置"></td></tr>
</table>  </form>
<%else  '如果该页面是提交后显示，获取用户输入的姓名、密码、性别等信息
name=request.Form("name")
pwd=request.Form("pwd")
sex=request.Form("sex")
hobby=request.Form("hobby")
come=request.Form("come")
intro=request.Form("intro")
name=trim(name)
pwd=trim(pwd)
hobby=split(hobby,",")  '复选框提交过来的数据是以","为分隔的，通过 split 函数把多个值分割开,
                           '保存在数组中
intro=replace(intro,vbCrLf,"<br>")  '文本区域提交过来的数据中的换行符为 vbCrLf，而 vbCrLf 在浏览

```

```
'器中显示为空格，需要替换成 html 标签中的换行标签<br>
response.write "<h1>您输入的个人信息如下：</h1><br>"
response.write "姓名: "& name &"<br>"
response.write "密码: "& pwd &"<br>"
response.Write "性别: "& sex &"<br>"
response.write "爱好: "
for each item in hobby  '把选择的爱好项都输出，通过空格分隔
    response.write item & " "
next
response.Write "<br>"
response.write "来自: " & come &"<br>"
response.write "个人介绍: " & intro
end if%>
</body>
</html>
```

程序的运行结果如图 3-46、图 3-47 所示。

图 3-46 个人信息注册表单

图 3-47 个人信息注册表单提交后数据输出

(3) ServerVariables 集合。

有时候非常希望知道服务器端或客户端的一些信息，如客户端的 IP 地址，客户端的浏览器版本，服务器端页面的路径信息等，以实现相关信息的输出或者根据浏览器的不同显示不同的页面。这时就需要用 ServerVariables 集合获取所需信息。

列举 ServerVariables 集合里所包含的所有成员及其值的代码如下，各个成员的具体含义可以通过查阅相关资料学习。

```
<OL>
<% for each key in Request.ServerVariables %>
<Li>
<% =key %> <% =Request.ServerVariables(key) %> <br></Li>
<% next %>
</OL>
```

2. Response 对象

Response 对象用来向客户端输出信息，Response 对象的属性和方法如表 3-11 所示。

表 3-11 Response 对象的方法、属性描述

类别	名称	功能
方法	Write	常用方法，输出信息到客户端
	Redirect	引导客户端浏览器到新的页面
	BinaryWrite	以二进制形式输出信息，如生成图片
	Clear	清除缓冲区里面的数据 Buffer 属性应为 True
	End	终止 ASP 程序的执行
	Flush	强制输出缓冲区里面的数据，Buffer 属性应为 True
属性	Buffer	设置缓冲区是否打开，默认为 True
	Expires	设置网页的过期时间
	ContentType	控制文件的输出类型
	Charset	设置网页使用的字符集
	Status	传递 HTTP 响应的状态

Write 方法是 Response 对象的最常用方法，它把变量的值传递到客户端。Write 方法几乎可以输出所有的对象和数据，包括 HTML 标签。

```
<% Response.Write "<br>"           '输出一个换行符，方法的括号可以省略
   Response.Write(name)            '输出变量 name 的值
   Response.Write "姓名:" & name    '输出字符串常量和变量连接后的字符串的值
   Response.Write "<tr></td>" & Request.form("varname") & "</td></tr>"
   '以表格的形式输出获取到的数据
%>
```

Write 方法中可以使用字符串连接运算符，输出的内容是连接后的更长的字符串。如果内容只输出一次，可以用“=”进行简写。

```
<% =varname %>
```

提示：①一个<% %>里面只能使用一个“=”。

②使用该简写可以方便生成查询字符串，如<a href="test.asp?name=<% = name%>&age=<% =age%>">显示。

③输出内容时，可以应用预定义的样式。如预定义了样式“.txt”，在表格中输出内容的方法为：<% response.write "<td class='txt'>" & varname & "</td>" %>。

3. Cookies 的使用

Cookie 是 Web 服务器嵌入客户机以标识用户的标记，它是服务器或脚本可以维护用户信息的一种方式。

当用户访问一个 Web 站点时，Web 服务器可以在用户的计算机上建立一个 Cookie 文件，包含用户的识别信息，如 SessionID、密码、访问时间、次数等。以后，无论用户何时连接到服务器，Web 服务器都可以访问这个 Cookie 文件，查询和修改其中的信息。

Cookie 分为会话（临时）Cookie 和永久 Cookie，前者在浏览器打开时存在，后者保存在客户机的硬盘上，在指定日期之前一直有效。

(1) Cookie 值的保存。

```
<% Response.Cookies("varname")="变量值"
   Response.Cookies.Expires=#2011-1-1#
   Response.Cookies("user")("name")="张三"
   Response.Cookies("user")("age")=16
%>
```

(2) Cookie 值的读取。

```
<% =Request.Cookies("varname")
   =Request.Cookies("user")("name")
   =Request.Cookies("user")("age")
%>
```

3.3 开发流程

3.3.1 实例功能分析与设计

该实例实现网站用户注册表单的制作和数据验证功能。用户名、密码、邮箱、电话号码、QQ 号码的有效性验证都是通过相应的过程实现。由于未学习数据库操作知识，该表单提交的数据最后仅显示出来，不会把注册用户名与已有用户名进行冲突检测，也不能把注册信息写入数据库。同时，数据的有效性检测是放在服务器端实现的。在学习与了解 JavaScript 后有效性检测应放在客户端实现。

整个实例的运行过程如图 3-48 所示。

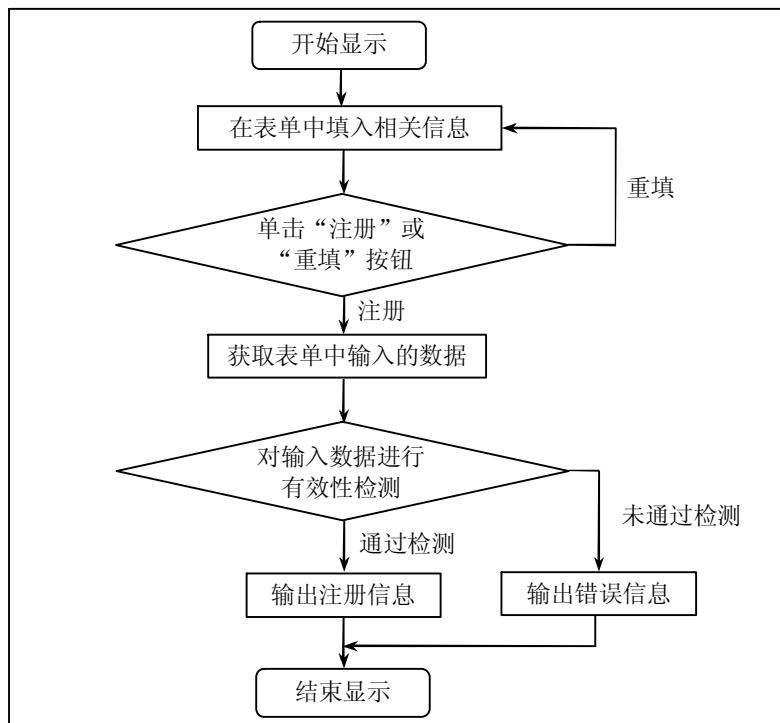


图 3-48 实例运行流程图

3.3.2 表单制作

本实例把表单和表单处理程序独立开来，先制作表单，然后再书写表单处理程序。

1. 定义样式

步骤 1：运行 Dreamweaver CS4，新建一脚本为 VBScript 的 asp 动态页面，命名为 reg.asp。

步骤 2：在如图 3-49 所示的 CSS 面板中单击 “” 按钮，定义文本样式。

步骤 3：在弹出的窗口中，选择“仅对该文档”，使样式仅对该文档生效，“选择器类型”为“类”，样式名为“.txttitle”，如图 3-50 所示。该样式用于设置标题。



图 3-49 CSS 样式面板

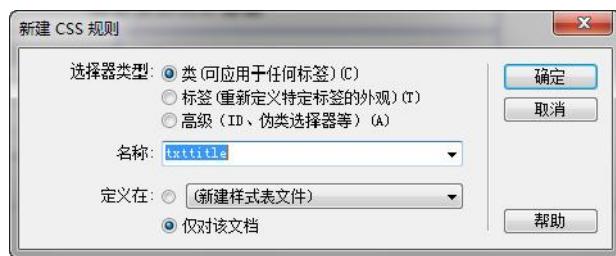


图 3-50 “新建 CSS 规则”对话框

步骤 4：在弹出的“.txttitle”的 CSS 规则定义”窗口中，设置文本的字体为“黑体”，文本大小为“16”，文本样式为“无”，文本粗细为“粗体”，在“区块”类别中设置文本的水平对齐方式为“居中对齐”，如图 3-51 所示。

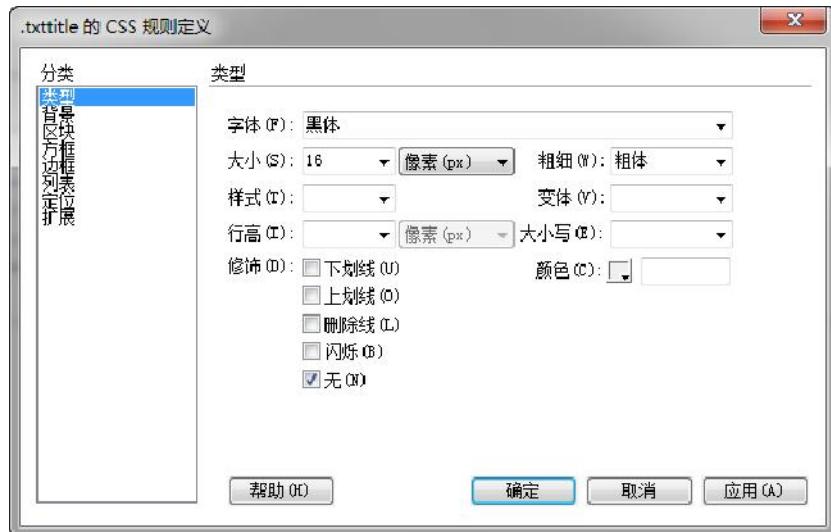


图 3-51 “.txttitle”的 CSS 规则定义”对话框

步骤 5：按照类似方法添加“.txtcontent”样式和“.txtform”样式用于设置表单中正文的格式和设置表单域中的文本格式。在“CSS 规则定义”对话框中设置文本大小为“12”，“修饰”为“无”，其中“.txtcontent”样式的文本对齐方式为“右对齐”。样式添加完成的最后效果如图 3-52 所示。



图 3-52 样式定义列表

2. 添加表单

步骤 6：切换到设计视图。在“表单”工具栏中单击“”按钮插入表单。

步骤 7：在表单中添加一个 9 行 3 列的表格，合并表格的第一行和最后一行。设置表格的背景色为“#abcdef”，对齐方式为“居中对齐”，单元格间距为 1，单元格填充为 0，表格的边框为 0。通过鼠标选中所有的单元格，设置单元格的背景颜色为白色，制作的效果如图 3-53 所示。

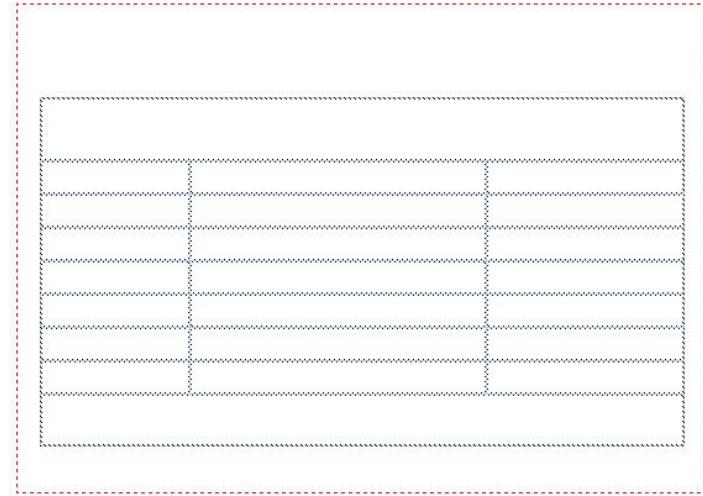


图 3-53 插入表单和表格后效果图

步骤 8：在第一行中输入文本“会员注册”。选中该单元格，在 CSS 面板中的“.txttitle”样式上单击右键，选择“套用”（方法 2：鼠标放在该单元格中，在“属性”面板上方的快速选择器中单击“<td>”标签，单击右键，在快捷菜单中指向“设置类”，为该单元格套用标题样式，如图 3-54 所示）。

步骤 9：在第二行第一列中输入文本“用 户 名*：”，为单元格套用“.txtcontent”样式。在第二行第二列中插入一个文本框，为文本框（注意，不是单元格）套用“.txtform”样式。在第二行第三列中输入提示文字“(6~12 个字符)”，为单元格套用“.txtform”样式。同理，在其他行中插入如图 3-55 所示的文字和相应的文本框，并套用相应的样式。



图 3-54 为单元格套用样式

会员注册	
用户名*	<input type="text"/> (6-12个字符)
密 钥*	<input type="text"/> 必填
验证密码*	<input type="text"/> 两次输入的密码必须一致
真实姓名*	<input type="text"/>
电子邮箱*	<input type="text"/> 正确的邮箱格式
Q Q*	<input type="text"/> 5-13位数字
联系方式*	<input type="text"/> 手机号码或座机

图 3-55 插入文本和文本框后效果图

步骤 10：对第二行第二列的文本框进行属性设置，如图 3-56 所示。第 5~8 行的文本框的属性与第二行第二列的属性类似，文本域的命名分别为“realname, email, qq, telephone”。



图 3-56 普通文本框的属性设置

步骤 11：第三行第二列、第四行第二列的文本框的属性与其他文本框的属性类似，“类型”设置为“密码”，如图 3-57 所示，文本域的命名分别为“pwd, pwd_confirm”。



图 3-57 密码框的属性设置

步骤 12：在最后一行插入提交按钮和重置按钮。提交按钮的动作设置为“提交表单”，名称为“reg”，值为“注册”，如图 3-58 所示。重置按钮的动作设置为“重设表单”，名称为“reset”，值为“重新输入”。按钮之间插入多个空格（全角状态），为按钮套用“.txtform”样式，并设置按钮为居中对齐。表单制作完成后的效果如图 3-59 所示。



图 3-58 提交按钮的属性设置

会员注册		
用 户 名*:	<input type="text"/>	(6~12个字符)
密 码*:	<input type="password"/>	必填
验证密码*:	<input type="password"/>	两次输入的密码必须一致
真实姓名*:	<input type="text"/>	
电子邮箱*:	<input type="text"/>	正确的邮箱格式
Q Q*:	<input type="text"/>	5-13位数字
联系方 式*:	<input type="text"/>	手机号码或座机
<input type="button" value="注 册"/> <input type="button" value="重新输入"/>		

图 3-59 表单制作的效果图

步骤 13: 把鼠标放在表格的空白地方, 在如图 3-60 所示的快速选择栏中单击 “<form>”, 选中整个表单, 对表单的属性进行设置。

```
<body><form><table><tr><td><span.txtform>
```

图 3-60 快速选择表单

步骤 14: 设置表单名为“regform”, “动作”属性为“reg_check.asp”, “方法”属性为“POST”, 如图 3-61 所示。至此, 整个表单页面的制作就完成了。



图 3-61 表单的属性设置

3.3.3 表单提交数据处理

1. 获取表单数据

步骤 1: 新建 VBScript 脚本的动态网页, 命名为 reg_check.asp。

步骤 2: 把 reg.asp 页面的样式重新定义一下, 保存在 css.css 文件中, 并把样式文件链接到该文件。效果如图 3-62、图 3-63 所示。



图 3-62 链接样式文件



图 3-63 样式文件链接后效果图

步骤3：切换到“代码视图”模式，书写代码实现表单数据的获取。

```
<%>
    ' 定义多个变量保存相应表单域的信息
    Dim username, pwd, pwd_confirm, realname, email, qq, telephone
        '由于表单采用的是 POST 方法提交数据，因此使用 request 对象的 form
        '集合，同时，变量名和表单域的名字相同，便于写代码
        username=trim(request.form("username"))
        pwd=trim(request.form("pwd"))
        pwd_confirm=trim(request.form("pwd_confirm"))
        realname=trim(request.form("realname"))
        email=trim(request.form("email"))
        qq=trim(request.form("qq"))
        telephone=trim(request.form("telephone"))
%>
```

2. 数据有效性检测

步骤4：定义用户名长度检测 username_check 子过程（注：子过程的定义和数据获取放在同一个<% %>内）。

```
Sub username_check ( user_name )
    dim gs          '定义字符个数变量
    gs= len(user_name)      '计算用户名的长度
    if(gs<6 or gs>12) then    '如果长度小于 6 或者大于 12，输出包含超级链接的错误提示信息
        response.write "<p class='txttitle'><a href='reg.asp'>用户名长度必须为 6~12 个字
符!</a></p>"
        response.end      '结束程序的执行
    end if
end sub
```

步骤5：定义密码检测 pwd_check 子过程。

```
Sub pwd_check ( user_pwd1,user_pwd2 )
    if(len(user_pwd1)<6) then    '如果密码长度小于 6，输出包含超级链接的错误提示信息
        response.write "<p class='txttitle'><a href='reg.asp'>密码太弱，最好是包含数组、大写字母
和小写字符，且长度大于 6!</a></p>"
        response.end      '结束程序的执行
    elseif(user_pwd1<>user_pwd2) then
        response.write "<p class='txttitle'><a href='reg.asp'>两次输入的密码不一致，请重新输
入!</a></p>"
        response.end      '结束程序的执行
end sub
```

```

    end if
end sub

```

步骤 6：定义邮箱检测 email_check 子过程，主要是检测邮箱地址中是否包含“@”符号。

```

Sub email_check ( user_email )
    if(instr(user_email,"@")=0) then  '如果邮件地址中不包含@，输出含超级链接的错误信息
        response.write "<p class='txttitle'><a href='reg.asp'>邮箱地址有误，请重新输入!</a></p>"
        response.end      '结束程序的执行
    end if
end sub

```

步骤 7：定义 QQ 号码检测 qq_check 子过程，主要是检测 QQ 号是否是 5~13 位数字。

```

Sub qq_check ( user_qq )
    if(len(user_qq)<5 or len(user_qq)>13) then  '如果数字位数小于 5 或者大于 13
        response.write "<p class='txttitle'><a href='reg.asp'>QQ 号码位数太少或太多，请重新输入!</a></p>"
        response.end      '结束程序的执行
    end if
    dim count
    count=1
    do while count<=len(user_qq)  '把 qq 号码表单域中输入的每个字符都检测一下
        if( not isnumeric(mid(user_qq,count,1)))  then      '在输入的 qq 号码中的 count 位置取一个字符，如果该位不是数字
            response.write "<p class='txttitle'><a href='reg.asp'>QQ 号码中包含非数字字符，请重新输入!</a></p>"
            response.end      '结束程序的执行
        end if
        count=count+1
    loop
end sub

```

步骤 8：定义联系方式检测 tel_check 子过程，主要是检测手机号码是否是 11 位数字。注意，手机号码是 1 开头的，座机不会以 1 开头。

```

Sub tel_check ( user_tel )
    '先判断电话号码中是否存在非法字符（除数字和"-"外）
    dim count
    count=1
    do while count<=len(user_tel)  '把电话号码的每个字符都检测一下
        if( not isnumeric(mid(user_tel,count,1)) and mid(user_tel,count,1)≠"-")  then
            '从输入的电话号码中的 count 位置取一个字符，如果不是数字也不是"-"
            response.write "<p class='txttitle'><a href='reg.asp'>号码中包含非数字字符，请重新输入!</a></p>"
            response.end      '结束程序的执行
        end if
        count=count+1
    loop
    '判断手机号码是否是 11 位
    if(left(user_tel,1)=1) then  '如果电话号码是 1 开头，则为手机号码
        if(len(user_tel)<11) then  '如果位数小于 11

```

```

        response.write "<p class='txttitle'><a href='reg.asp'>手机号码不对，请重新输入！</a></p>"
    end if
    response.end      '结束程序的执行
end sub

```

步骤9：在程序中调用相应的过程，实现数据的检测。

```

'调用以下过程，实现数据的检测
username_check username
pwd_check pwd,pwd_confirm
email_check email
qq_check qq
tel_check telephone

```

步骤10：网页的调试。运行 reg.asp，并输入相应的数据，如图 3-64 所示，提交表单后，检测结果如图 3-65 所示。完整的页面程序请查看 reg_check.asp 源文件。

会员注册		
用 户 名*：	aaaaaa	(6~12个字符)
密 码*：	*****	必填
验证密码*：	*****	两次输入的密码必须一致
真实姓名*：	sdag	
电子邮箱*：	aa@sina.com	正确的邮箱格式
Q Q*：	12345	5-13位数字
联系方式*：	dsg	手机号码或座机
<input type="button" value="注 册"/> <input type="button" value="重 新 输入"/>		

图 3-64 表单输入数据

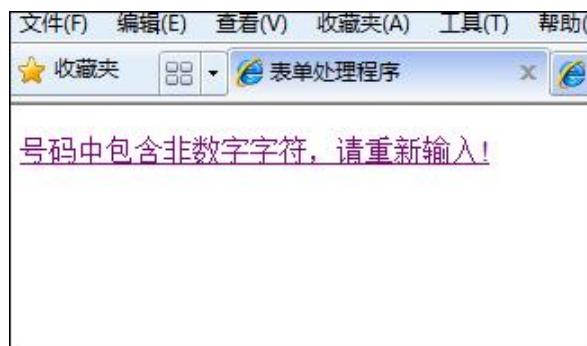


图 3-65 表单数据检测结果

3.4 拓展知识

3.4.1 子过程（子程序）与函数的区别

(1) 函数名可以带回返回值，这个值是通过在函数体中对函数名赋值实现的，返回值是 variant 类型。子过程名不可以带回返回值。

(2) 类似于 C 语言的全局变量，可以在多个函数中共享数据值。通过在子过程中为变量赋值也可以带回子过程中计算得到的值。如下的示例代码中 y 变量可以得到子过程的运行结果。

```
<%  dim x,y
x=12
y=0
sub pf(x)
y=x^2
end sub
call pf(x)
response.write y  %>
```

(3) 子程序和函数可以放在 ASP 文件的任意位置，即数据和过程没有前后依赖关系。子程序和函数也可以放在另外一个 ASP 文件中，通过文件包含实现调用，这样可以实现过程的功能复用和协同开发。

【练习 3-16】 调用另外一个 ASP 文件中的函数。

清单：3-11.asp

```
<% Option Explicit           '强制变量声明      %>
<!-- #include file="PFH.asp"-->
<% dim var1,var2,sum
var1=3
var2=4
sum=PFH(var1,var2)          'var1 和 var2 为实际参数
response.write "数 var1 和数 var2 的平方和是：" & CStr(sum)
%>
```

自定义函数 PFH.asp

```
<%
Function PFH( a,b)           'a 和 b 是形式参数
  dim sum
  sum=a^2+b^2
  PFH=sum                      '赋值给函数名，实现值的返回
End Function
%>
```

在浏览器中调试 3-11.asp 文件，得到如图 3-66 所示的效果。



图 3-66 程序运行效果图

提示：①<!-- #include file="PFH.asp" -->是 HTML 语句，实现在一个文件中包含另外一个文件中的代码。文件包含后，相当于 2 个文件中的代码是写在一起的，因此可以实现外部文件中函数的调用。

②#include 文件包含有两种写法：<!-- #include file="被包含文件的相对路径(相对于当前文件的目录关系，可以使用父目录"../" -->、<!-- #include virtual="被包含文件的虚拟路径(虚拟目录名后面加上文件的目录关系)" -->

3.4.2 参数传递方式

默认情况下，参数按地址传递，即过程按变量的内存地址去访问变量的内容。这样，将变量传给过程时，通过过程就可以改变变量的值。地址传递的关键字为 `ByRef`。

另外，参数还可以按数值来传送。按值传递参数时，传递的只是变量的副本。如果过程中改变了这个值，则所作的变动只影响副本而不会影响变量本身。按值传递的关键字是 `ByVal`。

【练习 3-17】理解按值传递和按地址传递的区别。

清单：3-12.asp

```
<%
Sub TestByVal(ByVal X)
X=X*10
End Sub

Sub TestByRef(ByRef X)
X=X/10
End Sub

intNum=60
Response.Write "原测试数值为"&intNum&"<br>"&"<br>"
Call TestByVal(intNum)
Response.Write "使用 ByVal 关键字，参数按值传递调用过程后，测试变量的值为" &intNum&
"<br>"&"<br>"
Call TestByRef(intNum)
Response.Write "使用 ByRef 关键字，参数按地址传递调用过程后，测试变量的值为"&intNum
%>
```

程序的运行结果如图 3-67 所示。从运行结果可以看出，按值传递后，变量 `intNum` 的值不变。按地址传递后，变量 `intNum` 的值改为 6。

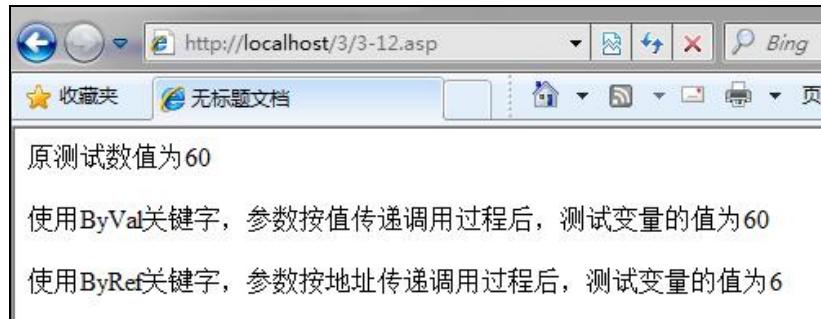


图 3-67 参数传递程序运行效果图

3.4.3 数组参数

数组作为参数传递时，与普通变量做参数有些不同，必须省略元素个数，即设置一个动态数组。在调用语句中，只需给出要传递的数组名即可，不必包括数组的长度及圆括号。

【练习 3-18】 产生 10 个随机数存放在数组中，并调用过程将数组中的元素进行降序排序。

清单：3-13.asp

```
<%
Sub Sort (arr(),arrnum)
Dim I,J,intTemp
For I=0 to arrnum
    For J=0 to arrnum
        If arr(I)>arr(J) Then
            intTemp=arr(I)
            arr(I)=arr(J)
            arr(J)=intTemp
        End If
    Next
End Sub

Dim arrTest(9),I
Response.Write "产生的随机数为：" &"<br>"
For I=0 to 9
    arrTest(I)=Int(Rnd()*100+1)
    Response.Write arrTest(I)&" "
Next
Sort arrTest,9
Response.Write "<Hr>"
For I=0 To 9
    Response.Write "排序后第"&i&"元素为：" &arrTest(I)&"<br>"
Next
%>
```

程序运行结果如图 3-68 所示。过程调用时，传递的是数组的首地址，以至于在过程中可以修改数组元素的值。

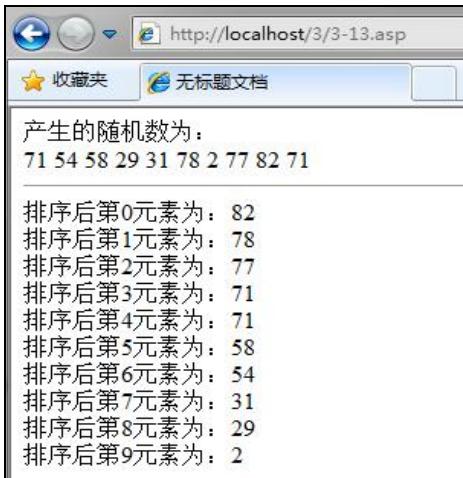


图 3-68 随机数排序程序运行效果图



本章以算术表达式计算（输入、获取、分析、转换、求解和输出）为核心，阐述了表单制作、ASP 运行环境配置、ASP 编程脚本语言、Request 对象和 Response 对象等相关基础知识点。

表单制作中，需重点掌握表单的 action、method 属性，表单域的 name 属性以及表单域的添加方法。

ASP 运行环境配置中需重点掌握主目录、虚拟目录和默认文档的设置方法，以及 ASP 缓冲区、脚本超时和默认脚本的设置。

ASP 脚本编程中需重点掌握变量及数组的定义方法，常用运算符的使用方法，选择结构和循环结构的使用方法，自定义函数和子过程的定义与调用方法，常用系统函数的使用方法。

Request 对象中需重点掌握 Form、Querystring、Cookies 和 ServerVariables 集合的区别、优先级以及获取数据的方法。掌握从表单域获取数据后的数据处理方法。

Response 对象中重点掌握 Write 方法、End 方法以及 Write 方法的简写形式。了解 Redirect 方法、Flush 方法以及 Clear 方法的使用。



一、思考题

1. IIS 配置中，主目录和虚拟目录的区别是什么？
2. 如何在程序中动态改变数组的长度及内容？
3. VBScript 脚本编程中循环结构有几种？For Each 循环的优势是什么？
4. Response 对象的 End 方法在编程和调试中有何用途？
5. 查阅相关资料，总结如何使用 Cookies 集合实现计数器？
6. 如何在网页中显示客户端和服务器端的 IP 地址？

二、单项选择题

1. 使用（ ）语句可以立即从 Sub 过程中退出，程序继续执行调用 Sub 过程之后的语句。
 - A. Exit Sub
 - B. Exit
 - C. End Sub
 - D. Loop
2. 请问下面程序段执行完毕后在网页上显示的内容是（ ）。


```
<% Response.write "<a href='http://www.sina.com.cn'>新浪</a>" %>
```

 - A. 新浪
 - B. 新浪
 - C. 新浪（超级链接）
 - D. 运行错误
3. 请问下面程序段执行完毕后，页面显示的内容是（ ）。


```
<%="祖国"  
 ="您好" %>
```

 - A. 祖国您好
 - B. 祖国（换行）您好
 - C. 祖国
 - D. 运行错误
4. 请问下面程序段执行完毕后，页面显示的内容是（ ）。


```
<% Response.write "ASP"  
Response.end  
Response.write("程序设计")  
%>
```

 - A. ASP
 - B. 程序设计
 - C. ASP 程序设计
 - D. 运行错误

三、操作题

1. 编写程序计算表达式 $11^2+12^2+13^2+14^2+15^2+16^2$ 的值。
2. 开发一页面，显示来访者的 IP 地址，并判断：如果 IP 地址以 172.16 开头，显示欢迎信息，否则显示为非法用户，并终止程序。
3. 开发一个简单的在线考试程序，可以包括若干选择、判断题，单击交卷按钮后可以根据标准答案在线评分。