# 一、概述

ASP是**服务器端的脚本语言环境**。

ASP(Active Server Pages)是微软推出的一种用来取代CGI(Common Geteway Interface)，它可以将HTML页面、脚本命令和ActiveX组件组合起来，创建动态的、交互的Web网页和基于Web的功能强大的应用程序。

IIS内含ASP，ASP程序其实是以扩展名为.asp的纯文本形式存在于WEB服务器上的，你可以用任何文本编辑器打开它，ASP程序中可以包含纯文本、HTML标记以及脚本命令。

ASP包含了以下三方面的含义：

1. Active：ASP使用微软公司的ActiveX技术。ActiveX技术是现在微软公司软件的重要基础。它采用了封装对象、程序调用对象的技术，简化了编程，加强了程序见的合作，病毒运行的时候都要调用这个组件ASP本身就封装了一些常见的组件。只要在服务器上安装这些组件，通过访问组件，就可以快速简易地建立Web应用程序。
2. Server：ASP运行在服务器端。这样就不必担心浏览器是否支持ASP所使用的编程语言。ASP 的编程语言可以是VBScript和JavaScript。
3. Pages：ASP返回标准的HTML页面，可以在常用的浏览器中显示。浏览者查看页面文件的时候，看到的是ASP生成的HTML代码，而不是ASP程序代码。这样就可以防止ASP源代码泄露了。

特点

1. 使用VBScript、JavaScript等简单易懂的脚本语言，结合HTML代码，就可以快速的完成网站应用程序的开发。
2. 容易编写，不需要编译就可以在服务端直接执行。
3. 使用普通的文本编辑器，比如Windows下的记事本就可以完成代码的编辑。
4. 与浏览器无关，只要客户端浏览器能够执行HTML代码，就可以浏览利用ASP设计的网页内容。ASP所使用的脚本语言(VBScript、JavaScript)均在WEB服务器端执行，用户端的浏览器不需要能够执行这些脚本语言。
5. ASP能与任何ActiveX scri pting语言相容。除了可使用VBScript或JavaScript语言来设计外，还通过plug-in的方式，使用由第三方所提供的其他脚本语言，譬如REXX、Perl、Tcl等。脚本引擎是处理脚本程序的COM(Component Object Model)物件。
6. ASP的源程序，不会被传到客户浏览器，因而可以避免所写的源程序被他人剽窃或篡改，提高了程序的安全性。
7. 可使用服务器端的脚本来产生客户端的脚本。
8. ActiveX Server Components(ActiveX服务器组件)具有无限可扩充性。可以使用VisualBasic、Java、Visual C++等各种编程语言来编写你所需要的ActiveX Server Component。

文档结构

ASP文件同普通的文本文件的区别是ASP文件的后缀名必须为“asp”。另外在访问方式上，ASP文件和HTML文件也有区别，即HTML文件可以使用物理路径来访问，也可以通过虚拟路径来访问；而ASP文件只能通过虚拟路径访问。

1. **ASP文件组成**

一个完整的ASP文件应该包含下面的这些内容：

1. 普通的HTML文件，也就是普通的Web的页面内容。
2. 服务器端的脚本程序代码，位于<%......%>内的程序代码。
3. 客户端的脚本程序代码，位于<script>......</script>内的程序代码。

ASP是依赖于脚本语言的，但是它同普通的脚本语言又有很大的不同。普通的脚本语言如VBScript、JavaScript等都是在客户端执行的，即HTML文档被下载以后通过浏览器解释执行，它的格式就是代码被限制在“<script>......</script>”标记对中。

而脚本语言用做ASP编程的时候，就不是在客户端执行了，必须通过服务器如IIS的解释之后，再发到客户端，由浏览器来解释执行。虽然服务端的脚本和客户端脚本的语法是一样的，但是为了能够让服务器区分哪些是服务器端哪些是客户端，在ASP编程的时候引入了“<%......%>”标记对，只要被包含在该标记对中的代码都会被认为是服务器端脚本语言。

服务器端脚本语言的定义格式如下：

<%

'脚本代码

'VBScript或者是JavaScript

%>

还有定义服务端脚本语言的格式并不是唯一的，还可以采用下面的形式来定义：

<script language="VBScript" runat="Server">

'脚本代码

</script>

这种服务器端脚本语言定义的方法和我们客户端的基本类似，只是必须在“<script>”标记后加上“runat="Server"”的关键字，因为默认的情况下是在客户端执行的。

1. **声明脚本语言**

在ASP编程中，可以使用的脚本语言很多种，但常用的是VBScript和JavaScript，要在程序中声明这些脚本语言，大致可以分为三种方法：

**①** 直接在IIS中设置，这种设置方法在第一章中已经有了介绍。如果采用了这种方法，那么在ASP编程中，若不特别声明，那么默认表示所有的ASP程序都采用这种脚本语言。IIS默认的脚本语言是 VBScript。

**②** 第二种方法就是直接在ASP文件开始设置，采用这种方法只作用于该页面，对其他的页面不产生影响。如果在一个ASP文件的开始处设置成JavaScript脚本语言，那么即使在IIS中设置成VBScript，在该页面也会默认为 JavaScript。第二种脚本语言的声明方法如下：

<%@ language=JavaScript %>或者<%@ language=VBScript %>

举个例子给大家看看：

<%@ language=JavaScript %>

<%

For(i=1;i<5;i++)

{

%>

Hello world!<br>

<%

}

%>

这个例子会在浏览器中输出5次Hello world!语句。这种声明方法必须将声明语句放在文件的第一行，同时要注意在@和保留字language之间有一个空格。

**③** 第三种声明方式的作用范围最小，这种声明方式，就是在标记对“<script>”的后面通过language关键字来声明，同客户端的声明是一样的，不过要记得在后面加上“runat="Server"”关键字。下面举个例子：

<script language=JavaScript runat="Server">

Function hell()

{

Response.write("hello world!") //Response为输出语句

}

</script>

<%

Hello() //注意：调用hello()函数的语句是用<%....%>标志的

%>

1. **ASP脚本的书写格式**

HTML文本、服务器脚本以及客户端脚本是ASP文档中的三种元素，在编码的时候他们互相不会影响。通常HTML文本和客户端脚本之间相互不会影响，但是服务器端脚本能够控制HTML文本和客户端脚本的输出。下面列举几种情况。

**1）**第一种是三种元素互相不影响：

在这种情况下，原则上服务器端脚本是可以在ASP文件的任意位置，不过为了提高阅读性可以放在HTML标记的“<head>....</head>”之间。最简单的一种书写格式ASP文件中的三种脚本互相不受影响，代码如下：

<html>

<head>

<title>ASP 示例</title>

<% response.write("hello world!<br>") %>

<SCRIPT LANGUAGE="VBScript">

<!--

document.write "这是第2个输出的!<br>"

-->

</SCRIPT>

</head>

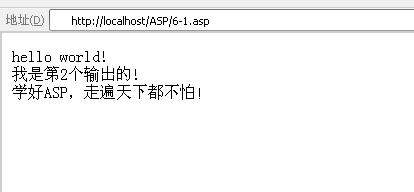
<body>

学好ASP基础，就可以找ASP程序漏洞了！

</body>

</html>

将上面的代码用记事本编辑之后保存为6-1.asp，然后把这个文件放入到服务器的虚拟路径中，最后在浏览器中输入6-1.asp文件的虚拟路径，这里必须为虚拟路径，代码执行的结果如图所示。



服务器在解释6-1.asp文件的时候首先遇到服务器端脚本，那么服务器将它首先解释成普通的脚本，而如果代码中是客户端脚本或是普通的HTML文本，那么服务器会原封不动地将它们发往客户端，因此在客户端浏览器实际解释的文件代码如下(在浏览器的浏览器的浏览区域单击右键，在弹出菜单中选择【查看源文件】查看代码)：

<html>

<head>

<title>ASP 示例</title>

hello world!<br>

<SCRIPT LANGUAGE="VBScript">

<!--

document.write "我是第2个输出的!<br>"

-->

</SCRIPT>

</head>

<body>

学好ASP，走遍天下都不怕！

</body>

</html>

**2）**第二种是服务器端脚本控制HTML文本的输出：

因为ASP是属于动态网页，所以可以通过ASP的服务器端脚本，根据不同的情况输出不同的内容。下面例子是一个服务器端脚本控制HTML文本输出的实例，演示了根据不同输入的年龄，判断一个人的年龄层。代码如下：

<html>

<head>

<title>ASP 示例</title>

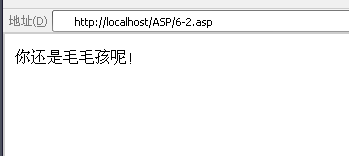
</head>

<body>

<%

dim age

age=6

if age<=6 then

%>

你还是小破孩呢！

<%

elseif age<=18 then

%>

你还是未成年哦！

<%

elseif age<=30 then

%>

你是个大人了哦！

<%

elseif age<=60 then

%>

你已经步入中年了！

<%

ele

%>

你已经进入老年了！

<%

end if

%>

</body>

</html>

从上面的代码中可以看出，并不是所有的HTML文本都显示出来了，而是根据服务器端的控制语句来输出符合条件的语句。这一点可以从客户端的源代码可以看出，代码如下所示：

<html>

<head>

<title>ASP 示例</title>

</head>

<body>

你还是毛毛孩呢！

</body>

</html>

**3）**第三种是服务器端脚本控制客户端脚本：

通过服务器端脚本来控制客户端脚本在我们的实际代码中不是很常见，下面例子根据输入的数字，通过客户端脚本动态地弹出一个对话框。代码如下：

<html>

<head>

<title>ASP 示例</title>

</head>

<body>

<%

dim myNum

myNum=9

response.write "<script language=VBScript>" & vbcrlf

response.write "<!--" & vbcrlf

response.write "msgbox ""我是数字" & myNum & """" & vbcrlf

response.write "-->" & vbcrlf

response.write "</script>" & vbcrlf

%>

</body>

</html>

**4）**第四种是在HTML中引用服务器端变量：

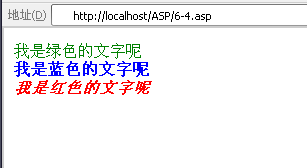
很多的时候，需要在HTML的文本中插入服务器定义的变量值。虽然通过“respone”对象能够直接输出变量，但是这样的使用比较麻烦，下面介绍一种快捷方式，如下所示：

<% =变量名%>

下面我们通过一个实例来说明，示例 6-4 演示了根据服务器定义的颜色值，客户端文字就现

实不同的颜色。代码如下：

<html>

<head>

<title>ASP 示例</title>

</head>

<body>

<%

dim color1,color2,color3

color1="green"

color2="blue"

color3="red"

%>

</body>

<font color=<% =color1 %>>我是绿色的文字呢</font><br>

<B><font color=<% =color2 %>>我是蓝色的文字呢</font><br>

<I><font color=<% =color3 %>>我是红色的文字呢</font>

</html>

上面的代码中，在HTML文本中就直接使用服务器脚本的变量，其客户端脚本如下所示：

<html>

<head>

<title>ASP示例</title>

</head>

<body>

</body>

<font color=green>我是绿色的文字呢</font><br>

<B><font color=blue>我是蓝色的文字呢</font><br>

<I><font color=red>我是红色的文字呢</font>

</html>

# 二、ASP常用的内置对象

对象的概念：

1. 对象是计算机中的一块内存区域，这块内存块是被用作可多次复制的模板，它包含数据和代码。
2. 对象中的数据只能由对象中的代码来引用。
3. 对象是通过实例来使用的，对象的实例就是对象模板的一次复制。
4. 建立实例后，用户可以像使用变量那样使用对象的数据，利用对象的代码执行某种操作。
5. 各个对象之间的通信是通过函数调用实现的。
6. 对象中可以包含“集合、属性和方法” ，集合和属性是数据，而方法就是对象的内部代码。

ASP包含的主要内置对象有Request、Response、Server、Session等等。而ActiveX组件实际上就是“外建”对象。

Request对象

在ASP中服务器利用Request对象来获取客户端提供的全部信息。包括输入的电子邮件和密码、浏览器和用户的HTTP变量、这个域名下存放在浏览器的Cookies、任何用户查询而附于URL之后的字符串或页面的<from>段内的HTML控件的值。

Request对象也提供使用Secure Socket Layer(SSL)或其他加密通信协议的授权访问，当使用Secure Socket Layer(SSL)后，就是利用https协议进行通信了，同时在浏览器的底部还会有一把小锁，网上银行都是采用这种方式的。

首先要明白[HTML的响应机制（Response mechanism）](HTML%20Response%20Mechanism.docx)：

1. **Request对象的成员**

Request对象主要是获取从客户端传递过来的各种数据，该对象的具体成员如下：

|  |  |
| --- | --- |
| Request对象成员 | 说明 |
| 属性TotalBytes | 返回由客户端发出请求的字符流的字节数量，它是一个只读属性 |
| 方法BinaryRead(conut) | 当使用POST方法发送请求时，从请求的数据中获得count字节数据，返回一个数组。(注意：如果已经引用了Request.From集合，就不能使用这个方法。同样，如果用了BinaryRead方法，就不能访问Request.From集合) |
| 集合QueryString | 读取使用URL参数方式提交的名值对数据或者以GET方法提交<from>中的数据，为只读。 |
| 集合From | 读去使用POST方法时提交<from>中的数据，为只读。 |
| 集合ServerVariables | 客户端请求的HTTP报头值，以及一些Web服务器环境变量值的集合，为只读。 |
| 集合Cookies | 用户系统发送的所有cookie值的集合，为只读。 |
| 集合ClientCertificate | 客户端访问页面或其他资源时，表明身份的客户证书所有字段或条目的数据集合，为只读。 |

1. **Request对象的使用**

主要讲集合QueryString、集合From、集合ServerVariables，具体的应用格式如下所示：

Request[.集合|属性|方法](变量)

**1）** 获取GET方法提交的数据——QueryString

在Request对象中，是通过QueryString集合实现的。具体的获取格式如下：

Request.QueryString("名称")

其中，名称表单元素名称或者URL中的“name=value”数据对中的“name”。下面举个例子演示如何通过GET方法提交的数据，表单文件同6-5.html基本上是一致的，另外还有一个就是提交处理文件6-7.asp，这个文件能够获取6-7.html输入的数据，并显示在浏览器中。6-7.html的代码如下所示：

<html>

<head>

<title>GET方法提交表单</title>

</head>

<body>

<form method="GET" action="3-7.asp">

<p>姓名：<input type="text" name="name" size="20"></p>

<p>年龄：<input type="text" name="age" size="20"></p>

<p>学校：<input type="text" name="school" size="20"></p>

<p>性别：<input type="text" name="sex" size="20"></p>

<p><input type="submit" value="提交" name="B1"><input type="reset" value="重置" name="B2"></p>

</form>

</body>

</html>

6-7.html的显示结果同和之前的图是一样的，6-7.asp文件的代码如下所示：

示：

<html>

<head>

<title>3-7.asp</title>

</head>

<body>

<%

dim myName,myAge,mySchool,mySex

myName=request.querystring("name") //获得客户端提交的数据，通过GET方法

myAge=request.querystring("age")

mySchool=request.querystring("school")

mySex=request.querystring("sex")

%>

我的姓名是：<% =myName %><br> //输出变量

我的年龄是：<% =myAge %><br>

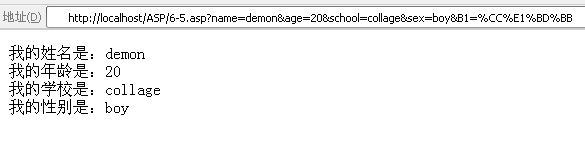
我的学校是：<% =mySchool %><br>

我的性别是：<% =mySex %><br>

</body>

</html>

代码6-7.asp执行后的结果如图所示。



**2）** 获取POST方法提交的数据——From集合

在实际的代码中，通过POST方法提交数据的时候用的最多，它能够传递大量的数据。当通过POST方法提交数据的时候，必须使用From集合来获取数据。不过这里要注意，这里的必须并不是绝对的，其实Request对象可以不通过集合来获得数据，即直接使用“Request("name")”来获取，虽然这样写代码很简单，但它的效率比较低，而且容易出错，当省略具体的集合名称时，ASP会按一下的顺序来搜索集合：

①QueryString ②FromCookie ④Servervariable。

通过From获取数据的格式如下：

Request.From("名称")

前面介绍的各种QueryString使用方法，对于From来说也是适合的，下面演示了在From下的例子，这个例子由两个文件组成，分别为6-8.html和6-8.asp。其中6-8.html是基本的表单输入文件，6-8.asp是表单处理文件，即获得客户提交的数据。6-8.html的代码如下所示：

<html>

<head>

<title>常用表单控见的数据获取</title>

</head>

<body>

<form method="GET" action="3-8.asp">

<p>你的姓名：<input type="text" name="name" size="20"></p>

<p>你的家乡在：<select size="5" name="home">

<option>江西</option>

<option>湖南</option>

<option>广东</option>

<option>北京</option>

<option>上海</option>

</select>，你喜欢哪些城市：<select size="5" name="city" multiple>

<option>深圳</option>

<option>北京</option>

<option>上海</option>

<option>南京</option>

<option>苏州</option>

</select>（可以多选）</p>

<p>你的爱好是（可以多选）：</p>

<p>足球<input type="checkbox" name="hobby" value="足球">&nbsp;&nbsp;&nbsp;篮球<input type="checkbox" name="hobby" value="篮球">&nbsp;&nbsp;羽毛球<input type="checkbox" name="hobby" value="羽毛球">&nbsp;&nbsp;&nbsp;乒乓球<input type="checkbox" name="hobby" value="乒乓球"></p>

<p>你的性别是：男<input type="radio" value="男" name="sex"checked>&nbsp;&nbsp;&nbsp;

女<input type="radio" name="sex" value="女"></p>

<p><input type="submit" value="提交" name="B1">

<input type="reset" value="重置" name="B2"></p>

</form>

</body>

</html>

6-8.html的运行结果如图：

