

Lesson #10



ASP

(Active Server Pages)

ASP

- Active Server Pages (ASP) is Microsoft's server-side script engine for dynamically-generated web pages. It is marketed as an add-on to Internet Information Services (IIS).
- Several scripting languages may be used in ASP. However, the default scripting language (in classic ASP) is VBScript.

ASP or PHP?

- Fans of either technology are usually very camped in their positions. In reality, both solutions are somewhat similar.
- PHP is based on Unix and will be your preferred choice if you are already proficient in JavaScript. ASP is based on Microsoft platforms and uses VisualBasic.
- Most choice are history-based. For example, bulletin boards are mostly written in PHP and e-commerce sites mostly in ASP.

A Simple Program

- ASP (like PHP) is included code inside the XHTML code. Its inclusion looks more like SSI than Perl.
- Like SSI and PHP, ASP programs need a special extension (.asp).
- ASP code appears between the special tags <% and %> called wrappers.

<%

```
response.write ("This is my first ASP program.")
```

%>

A Simple Program

- ASP (like PHP) is included code inside the XHTML code. Its inclusion looks more like SSI than Perl.
- Like SSI and PHP, ASP programs need a special extension (.asp).
- ASP code appears between the special tags <% and %> called wrappers.

<%

```
response.write ("This is my first ASP program.")
```

%>

ASP Operators

- The mathematical operators in ASP are similar to many other programming languages. However, ASP does not support shortcut operators like ++, --, +=.
- Comparison operators are =, <, >, <=, >=, <>.
- Logical operators are and, or, not.
- The only string operator is the string concatenation operator "&" that takes two strings and slams them together to form a new string.

Using Variables

- In ASP, there is no ; after each statement. No \$ for variable names (unlike Perl or PHP).
- Elements inside response.write are separated by &.

```
<%
```

```
days = 5
```

```
sentence = "It is hot today."
```

```
temp = 30.0
```

```
response.write (sentence & " The temprature has  
been over " & temp & " degrees for " & days & "  
days.")
```

```
%>
```

```
It is hot today. The temprature has been over 30 degrees for 5 days.
```

Using Variables

- Once again, ASP uses VBScript by default and so it also uses VBScripts variable naming conventions. These rules are:
 - 1. Variable name must start with an alphabetic character (A - Z or a - z).
 - 2. Variables cannot contain a period.
 - 3. Variables cannot be longer than 255 characters
 - 4. Variables must be unique in the scope in which it is declared.

Using Variables inside HTML

```
<%
```

```
days = 5
```

```
sentence = "It is hot today."
```

```
temp = 32.5
```

```
%>
```

```
<% response.write (sentence) %> The temprature has  
been over <% response.write (temp) %> degrees for  
<% response.write (days) %> days.
```

It is hot today. The temprature has been over 32.5 degrees for 5 days.

ASP - Selection (if)



Note: equality
in ASP is =, not
==. Not equal is
≠ not !=

```
<%  
distance = 850  
if distance <= 500 then  
    response.write ("Easy in one day.")  
else  
    if distance <= 800 then  
        response.write ("Feasible in one day.")  
    else  
        response.write ("I should reserve a  
hotel room.")  
    end if  
end if  
%>
```

ASP - Select Case

- ASP uses the Select Case statement to check for multiple equality conditions of a single variable. The Select statement resembles a switch statement that other programming languages use.

```
<%  
number = 3  
select case number  
  case 2  
    Response.Write("number is Two")  
  case 3  
    Response.Write("number is Three")  
  case else  
    Response.Write("number is " & number)  
end select  
>%
```


ASP - Select Case

- Select case can use strings as well as integers for control variable.

```
<%  
myPet = "cat"  
select case pet  
  case "dog"  
    Response.Write("I own a dog")  
  case "cat"  
    Response.Write("I own a cat")  
  case Else  
    Response.Write("I do not have a pet")  
end select  
%>
```

ASP - Loops

```
<%  
count = 100  
do while count <> 0  
    response.write (count & " ")  
    count = count - 1  
loop  
>%
```



--count not supported in ASP!

```
100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76  
75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50  
49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24  
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

ASP - Arrays

- Arrays in ASP follow the exact same form and rules as those arrays in VBScript. You can create an array of specific size or you can create a dynamic sized array. Below we have examples of both types of arrays.

```
<%  
Dim myFixedArray(3) 'Fixed size array  
Dim myDynArray() 'Dynamic size array  
%>
```

Notice the
comments
syntax!

ASP - Fixed Arrays

Size is maximum index in ASP!

```
<%  
Dim myFixedArray(3) 'Fixed size array  
myFixedArray(0) = "Albert Einstein"  
myFixedArray(1) = "Carl Sagan"  
myFixedArray(2) = "Michio Kaku"  
myFixedArray(3) = "Louis de Broglie"  
For Each item In myFixedArray  
    Response.Write(item & "<br />")  
Next  
%>
```

Another type of loop!

ASP - Dynamic Arrays

- To create an array whose size can be changed at any time simply do not put a number within the parenthesis when you declare the array. When you know what size you want the array to be use the ReDim keyword. You may ReDim as many times as you wish.
- If you want to keep your data that already exists in the array then use the Preserve keyword.

<%

```
Dim myDynArray() 'Dynamic size array
```

```
ReDim myDynArray(1)
```

```
myDynArray(0) = "Albert Einstein"
```

```
myDynArray(1) = "Carl Sagan"
```

```
ReDim Preserve myDynArray(3)
```

```
myDynArray(2) = "Michio Kaku"
```

```
myDynArray(3) = "Louis de Broglie"
```

%>

ASP - Sessions

- The Session Object in ASP allows you to keep information specific to each of your site's visitors. Information like username, shopping cart, and location can be stored for the life of the session. The most important thing to know about ASP's Session Object is that it is only created when you store information into the Session Contents collection.
- To store a Session Variable you must put it into the Contents collection. Here we are saving the Time when someone visited the page into the Session Contents collection and then displaying it .

<%

key

value

```
Session("TimeVisited") = Time()
```

```
Response.Write("You visited this site at: " &  
Session("TimeVisited"))
```

%>

ASP - Sessions

- The ASP Session ID is the unique identifier that is automatically created when a Session starts for a given visitor. The Session ID is a property of the Session Object and is rightly called the SessionID property.
- A Session will not last forever, so eventually the data stored within the Session will be lost. There are many reasons for a Session being destroyed. The user could close their browser or they could leave their computer for an extended amount of time and the Session would time out. You can set how long it takes, in minutes, for a session to time out with the Timeout property.

<%

```
mySessionID = Session.SessionID
```

```
Session.Timeout = 240
```

%>

ASP - Cookies

- Creating an ASP cookie is exactly the same process as creating an ASP Session. Once again, you must create a key/value pair where the key will be the name of our "created cookie". The created cookie will store the value which contains the actual data.

```
<%  
Response.Cookies("language") = "Russian"  
%>
```

- To retrieve the cookie we use the Request.Cookies method.

```
<%  
favlang = Request.Cookies("language")  
Response.Write("You set the language to " &  
favlang)  
%>
```

ASP - Cookies

- In ASP you can set how long you want your cookies to stay fresh and reside on the user's computer. A cookie's expiration can hold a date; this date will specify when the cookie will be destroyed.

```
<%
```

```
'create a 10-day cookie
```

```
Response.Cookies("language") = "Persian"
```

```
Response.Cookies("language").Expires = Date() + 10
```

```
'create a static date cookie
```

```
Response.Cookies("name") = "Suzy Q."
```

```
Response.Cookies("name").Expires = #January 1,2009#
```

```
%>
```

ASP - Cookie Collections

- in ASP you can store data in a collection of cookies referring to a single object.

<%

```
Response.Cookies("brownies")("numberEaten") = 13
```

```
Response.Cookies("brownies")("eater") = "George"
```

```
Response.Cookies("brownies")("weight") = 400
```

```
For Each key In Request.Cookies("Brownies")
```

```
    Response.Write("<br />" & key & " = " & _
```

```
        Request.Cookies("Brownies")(key))
```

Next

%>

ASP - Using Forms

- Like Perl and PHP, you can use forms with an HTML part and a server-side (ASP) part.

```
<form action="process.asp" method="post">
```

```
<select name="item">
```

```
<option>Paint</option>
```

```
<option>Brushes</option>
```

```
<option>Erasers</option>
```

```
</select>
```

```
Quantity: <input name="quantity" type="text" />
```

```
<input type="submit" />
```

```
</form>
```

ASP - Using Forms

- When the POST method is used to send data you retrieve the information with the Request Object's Form collection.

```
<%
```

```
item = Request.Form("item")
```

```
quantity = Request.Form("quantity")
```

```
Response.Write("Item: " & item & "<br />")
```

```
Response.Write("Quantity: " & quantity & "<br />")
```

```
%>
```

ASP - Query Strings

- To extract data from query strings (or forms using the GET method), we use the Request Object's QueryString collection.
- For example, with a url like www.domain.com/query.asp?Name=Lucy&Age=22

```
<%
```

```
name = Request.QueryString("Name")
```

```
age = Request.QueryString("Age")
```

```
%>
```

Using JavaScript with ASP

- VBScript is the default scripting language that ASP is coded in, so if you want to specify a different scripting language you have to state which scripting language you will be using at the very beginning of your code. Below is the line of code that must be your first line of ASP code or else your page will break and you'll get an error message.

```
<%@ Language="javascript"  
'The rest of your ASP Code....  
%>
```

Using JavaScript with ASP

- VBScript is the default scripting language that ASP is coded in, so if you want to specify a different scripting language you have to state which scripting language you will be using at the very beginning of your code. Below is the line of code that must be your first line of ASP code or else your page will break and you'll get an error message.

```
<%@ Language="javascript"  
'The rest of your ASP Code....  
%>
```

End of lesson