

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

مقالات

ASP

منبع:

www.Dev.ir

ورود اطلاعات فارسی با استفاده از جاوااسکریپت

اگر قبلا اقدام به استفاده از زبان فارسی در کدهای خود و مخصوصا هنگام کار با Microsoft Access کرده باشید به احتمال زیاد با مشکلات موجود بر سر راه آشنا هستید. یکی از مشکلات اساسی بر سر راه استفاده از زبان فارسی در بانکهای اطلاعاتی Microsoft Access در نحوه ورود اطلاعات می باشد. برای درک بهتر این موضوع اینگونه عمل نمایید:

در ویندوز خود قابلیت فارسی نویسی را فعال نمایید و پس از آن در Access يك Table با نام test و با يك فیلد به نام test از نوع text ایجاد نمایید. سپس با استفاده از خود Access اطلاعاتی به زبان فارسی وارد نمایید. سپس از Access خارج شده و با استفاده از ASP به بانک اطلاعاتی Access متصل شده و اطلاعات وارد شده را خوانده و نمایش دهید. مشاهده خواهید نمود که اطلاعات نمایش داده شده داخل مرورگر با اطلاعات وارد شده توسط شما در داخل Access متفاوت می باشد.

برای حل این مشکل چه باید کرد؟

برای حل این مشکل باید اطلاعات از خارج از Access و با استفاده از ASP و يك تابع Javascript که صفحه کلید را در همه ویندوزها اعم از فارسی و انگلیسی، فارسی نموده و امکان تایپ فارسی را فراهم میکند، وارد شوند. روش کار به این صورت است که در يك فرم HTML و در داخل يك Input Box هنگام فشردن هر کلیدی از صفحه کلید، تابع Javascript فرا خوانده شده و معادل فارسی حرف وارد شده را نمایش میدهد. برای نمایش حروف فارسی لازم است که از Times New Roman و یا Tahoma انتخاب نمایید. این دو فونت فونتهای استاندارد ویندوز بوده و برای دیده شدن متون فارسی نیازی به نصب فونت جدید روی سیستم کاربر نخواهد بود. Javascript استفاده شده در زیر آمده است:

```
گ،(و+$%#)
<|>_پشذزیشلاهنتمئدخضقسفعرصطغظ>
arCodeAt(key-32);
else if (e)
e.which='
گ،(و+$%#"!)
<|>_پشذزیشلاهنتمئدخضقسفعرصطغظ>
arCodeAt(key-32);
}
return true;
}
//-->
</SCRIPT>
```

پس از طراحی فایل HTML نوبت به ایجاد فایل ASP می رسد. در فایل ASP باید اطلاعات وارد شده در فرم HTML را دریافت نموده و در بانک اطلاعاتی Access توسط برنامه خود بنویسید. نکته جالب این است که اگر وارد Microsoft Access شوید و اطلاعات وارد شده توسط برنامه خود را مشاهده نمایید اثری از حروف فارسی مشاهده نخواهید کرد. حال اگر دوباره توسط برنامه ASP اولیه به بانک اطلاعاتی متصل شوید و اطلاعات دریافت شده را داخل مرورگر نمایش دهید اطلاعات به صورت فارسی نمایش داده میشوند. لازم به ذکر است که باید از charset=windows-1256 در فایل ASP که اطلاعات فارسی را نمایش میدهد استفاده نمایید.

یک شمارنده ساده یا کونتور

این برنامه یک مثال خوب از نحوه باز کردن ، خواندن و نوشتن داخل یک فایل است که نیاز به هیچ نوع بانک اطلاعاتی هم ندارد . این فایل را با اسم counter.asp ذخیره کنید و در کنارش یک فایل متنی ساده به کمک NotePad و به نام database.inc ایجاد کنید و در سطر اولش هم عدد صفر یا هر عدد دیگری را که مایلید پیش فرض شمارنده شما باشد بنویسید. فراموش نکنید که سطح دسترسی لازم برای نوشتن را نیز به این فایل بدهید . با کمی ابتکار می توان این شمارنده را به نوع گرافیکی تبدیل کرد، حتی می توانید به تعداد صفحات سایت ، فایل ایجاد کنید تا تعداد مشاهده هر صفحه را به تنهایی ردیابی کنید.

```
<%  
Dim objFSO , objFile  
set objFSO = createobject("scripting.filesystemobject")  
set objFile = objFSO.opentextfile(server.mappath("database.inc"))  
'get the current count and store in counter  
counter = cLng(objFile.readline)  
'add one to the current count  
counter = counter + 1  
'close file object  
objFile.close  
'write a new text object with the same name and new count  
Set objFile = objFSO.CreateTextFile(server.mappath("database.inc"), true)  
objFile.writeline(counter)  
'close file object  
objFile.close  
,  
  
counter_length = len(counter)  
'loop through the hit count and display the image  
for i = 1 to counter_length  
%>  
<%= mid(counter,i,1) %>  
<%  
next  
%>
```

ارسال ایمیل در ASP ساده و کار آمد

با این کد ساده می توان صفحه ای برای ارسال ایمیل ایجاد کرد . این کُد را یک هندی به نام مستعار Prince Joseph نوشته است و آنطور که گفته ملزومات سیستمی آن برای اجرا عبارتند از : یک ویندوز نسخه سرویس دهنده و SMTP Service بر روی IIS . این دو صفحه با استفاده از کامپوننت یا ابزار CDO که توسط شرکت مایکروسافت ارائه شده کار می کنند .

: mailSend.htm

```
<html><body>
<form id=frmMail method=post action=mailSend.asp>
TO:<input type=text name=txtTo size=30><br>
FROM:<input type=text name=txtFrom size=30><br>
SUBJECT:<input type=text name=txtSubject size=30><br>
BODY:<textarea rows="5" name=txtBody cols="28">
</textarea><br>
<input type=submit value="Send the Mail">
</form>
</body></html>
```

: mailSend.asp

```
<%@ Language=VBScript %>
<%
Dim objMail
Set objMail = CreateObject("CDONTS.NewMail")
objMail.From = Request.Form("txtFrom")
objMail.To= Request.Form("txtTo")
objMail.Subject= Request.Form("txtSubject")
objMail.Body= Request.Form("txtBody")
objMail.Send
Set objMail = Nothing
Response.Write "Mail should have been sent"
%>
```

درج تاریخ شمسی

تا به حال الگوریتم‌های زیادی برای نمایش تاریخ جاری بر اساس تقویم جلالی ارائه شده‌اند . در زیر یک نمونه ساده ولی کارآمد آن ارائه می شود . این نمونه تاریخ Server را نشان می دهد . برای اینکه تاریخ از روی کامپیوتر بازدیدکننده محاسبه شود کافیست runat=server را از کد حذف کنید .

```
<script language="vbscript" runat="server">
Function IranDate()
Dim ifday,ifmonth,ifyear,ifdayofyear
Dim iyear,idayofyear
Dim inumdayofyear
Dim aifmonthdays
aifmonthdays = Array(31,31,31,31,31,31,30,30,30,30,30,29)
inumdayofyear = 365
iyear =year(date)
idayofyear = datepart("y", date)
If isleapyear(iyear-1) then
inumdayofyear = 366
aifmonthdays(11) = 30
End if
If (idayofyear > 79 ) then
ifyear = iyear -621
ifdayofyear = idayofyear - 79
Else
ifyear = iyear - 622
ifdayofyear = (inumdayofyear - 79 ) + idayofyear
End if
ifday = ifdayofyear
While ( ifday > aifmonthdays(ifmonth) )
ifday = ifday - aifmonthdays(ifmonth)
ifmonth = ifmonth + 1
Wend
ifmonth = ifmonth + 1
IranDate = ifyear & "/" & ifmonth & "/" & ifday
End Function
Function isleapyear(nyear)
isleapyear = ( ( nyear mod 4) = 0 And (nyear mod 100) <> 0 ) or (nyear mod 400) = 0 )
End Function
Sub Showfdate
End sub
</script>

<%= (IranDate) %>
```

فرستادن فایل یا Upload از صفحه وب

در اینجا به ذکر نمونه ای از يك كد آماده برای بارگزاری فایل از طریق صفحه وب (Upload) به يك سرویس دهنده می پردازیم . این کدها از مجموعه موجود در سایت گرفته شده اند . در ابتدا باید هر دو فایل زیر را بسازید :

: upload_page.asp

```
<!-- #include file="upload_class.asp" -->
<%
Server.ScriptTimeout = 300 'now i can upload and save files upto ~8Mb
Dim intLevel, intUpload, intSave, strError, strContentType, strFilename, lngFileSize
Dim objUpload
Dim lngTime, lngUploadTime, lngSaveTime
intLevel = Request.QueryString("level")
'-----
wrHead
If intLevel = 1 Then
    Set objUpload = New FileUpload
    With objUpload
        .Path = "D:\Inetpub\wwwroot\test"
        lngTime = Timer()
        intUpload = .Upload
        lngUploadTime = Round((Timer() - lngTime) * 1000,3)
        lngTime = Timer()
        intSave = .Save(true)
        lngSaveTime = Round((Timer() - lngTime) * 1000,3)
        strError = .Error
        strFilename = .Filename
        lngFilesize = .Size
        strContentType= .ContentType
    End With
    Set objUpload = Nothing
End If

wrForm
wr "<hr style='\"'height:1px;width:100%;\"' />"
wr "Upload = " & intUpload & "<br />"
wr "Save = " & intSave & "<br />"
wr "Error = " & strError & "<br />"
wr "Filename = " & strFilename & "<br />"
wr "Filesize = " & lngFilesize & "<br />"
wr "Content-Type = " & strContentType & "<br />"
wr "Upload time = " & lngUploadTime & " ms<br />"
wr "Save time = " & lngSaveTime & " ms <br />"
wr "<hr style='\"'height:1px;width:100%;\"' />"
wrFoot
'-----
```

```

Sub wrForm
    wr "<form method=""post"" enctype=""multipart/form-data""
action=""?level=1"">"
    wr "<input type=""file"" name=""file""></input>"
    wr "<input type=""submit""></input>"
    wr "</form>"
End Sub

```

```

Sub wrHead
    wr "<html>"
    wr "<head>"
    wr "<title>upload</title>"
    wr "</head>"
    wr "<body>"
End Sub

```

```

Sub wrFoot
    wr "</body>"
    wr "</html>"
    Response.End
End Sub

```

```

Sub wr(byval sText)
    If sText <> "" Then Response.Write sText & vbNewLine
End Sub
%>

```

: upload_class.asp

```

<%
'+-----+
'|Class:   FileUpload   |
'|Date:    11:01 PM 7/23/2002|
'|By:      M.Meijer      |
'|Version: 1.0           |
'+-----+
'
'To upload and save a file submitted within a html form
'
'**Remarks:
'Uploading files with this class is not recommended for huge files,
'it takes alot of time saving the file to a textstream (as it the function 'save' does).
'It takes 7.812ms to upload a file from 'localhost', with a size of 40,000 bytes.
'Saving this file however costs 1078.125ms, and it takes 145828.1ms to save a file of
5.5Mb.
'Conclusion don't save big files, use the maxfilesize property to limit the filesize.

```

'The class can only handle one file on a submission.

'The file will be saved in the specified 'Path', if there is no 'path' set, it can't save the file. (doh!)

'Properties:

'-----

'	ContentType	string	read	Content-Type of the file
'	Filename	string	read/write	Name of the file
'	Path	string	read/write	A path to a directory with permissions to write the file
'	Size	long	read	The size of the file in bytes
'	AllowedFiles	string	read/write	Allowed file extension(s), multiple separated with a comma
'	Maxfilesize	long	read/write	Maximum allowed size of the file
'	Error	string	read	The explanation of an error if occurred

'Methods

'-----

'	Upload()	= Status		
'				Copies the result of Request.Binaryread to a file
'				
'	Status	integer	0	Upload success
'			1	A file has not been posted
'			2	File exceeds the maximum allowed filesize
'			3	Type is not allowed
'				
'	Save(Overwrite)	= Status		
'				Slaat de bytearray op in een bestand met de in Filename gedefinieerde bestandsnaam,
'				in de in Path gedefinieerde directory.
'				
'	Overwrite	boolean	true	If the file exists it will be overwritten
'			false	If the file exists it will not be overwritten
'				
'	Status	integer	0	The file has been saved
'			1	The binary value could not be written to a file
'			2	There is no binary value
'			3	The filename is empty
'			4	An error already occurred, can't continue

'Code:

'-----

Class FileUpload

```
Private strContentType  
Private bytData  
Private strFilename  
Private strPath  
Private lngTotalbytes  
Private strAllowedFiles  
Private lngMaxFileSize  
Private strError
```

```
Private Sub Class_initialize()  
    strContentType = ""  
    bytData = ChrB(10)  
    strFilename = ""  
    strPath = ""  
    lngTotalbytes = 0  
    strAllowedFiles = ""  
    lngMaxFileSize = 0  
    strError = ""  
End Sub
```

```
Private Sub Class_Terminate()  
    bytData = Null  
End Sub
```

```
Public Property Get Size  
    Size = lngTotalbytes  
End Property
```

```
Public Property Let MaxFileSize(byval vData)  
    If IsNumeric(vData) > 0 Then  
        lngMaxFileSize = vData  
    End If  
End Property
```

```
Public Property Get MaxFilesize  
    MaxFilesize = lngMaxFileSize  
End Property
```

```
Public Property Let AllowedFiles(byval vData)  
    If Len(vData) > 0 Then  
        strAllowedFiles = vData  
    End If  
End Property
```

```
Public Property Get AllowedFiles  
    AllowedFiles = strAllowedFiles  
End Property
```

```
Public Property Get Error
```

```
    Error = strError
```

```
End Property
```

```
Public Property Get ContentType
```

```
    ContentType = strContentType
```

```
End Property
```

```
Public Property Let Path(byval vData)
```

```
    If Len(vData) > 0 Then
```

```
        strPath = vData
```

```
    End If
```

```
End Property
```

```
Public Property Get Path
```

```
    Path = strPath
```

```
End Property
```

```
Public Property Let Filename(byval vData)
```

```
    If Len(vData) > 0 Then
```

```
        strFilename = vData
```

```
    End If
```

```
End Property
```

```
Public Property Get Filename
```

```
    Filename = strFilename
```

```
End Property
```

```
Public Function Upload()' as integer
```

```
    Dim bytAllData
```

```
    lngTotalbytes = Request.Totalbytes
```

```
    If lngTotalbytes > 0 Then
```

```
        If lngMaxFileSize <> 0 Then
```

```
            If lngTotalBytes > lngMaxFileSize Then
```

```
                strError = "The file exceeds the allowed capacity."
```

```
                Upload = 2
```

```
                Exit Function
```

```
            End If
```

```
        End If
```

```
        bytAllData = Request.BinaryRead(lngTotalbytes)
```

```
        strContentType = GetContentType(bytAllData)
```

```
        strFilename = GetFilename(bytAllData)
```

```
        If strAllowedFiles <> "" Then
```

```
            If Not AllowedFile(strFilename) Then
```

```
                strError = "Filetype is not allowed."
```

```
                Upload = 3
```

```

                                Exit Function
                            End If
                        End If
                    bytData = GetData(bytAllData)
                    Upload = 0
                Else
                    Upload = 1
                    strError = "No data recieved."
                End If
            End Function

Public Function Save(byval bOverwrite)
    If strError <> "" Then
        Save = 4
        Exit Function
    End If
    If strPath <> "" Then
        If Mid(strPath,Len(strPath)-1,1) <> "\" Then strPath = strPath & "\"
        If strFilename <> "" Then
            If LenB(bytData) > 1 Then
                If SaveBinaryData(bytData,strPath &
strFilename,bOverwrite) Then
                    Save = 0
                Else
                    Save = 1
                End If
            Else
                Save = 2
                strError = "No data."
            End If
        Else
            Save = 3
            strError = "Not a valid filename specified."
        End If
    Else
        Save = 4
        strError = "No path specified."
    End If
End Function

Private Function AllowedFile(byval sFilename)'as boolean
    Dim arrAllowedFiles, intCount
    Dim strExtension
    If Len(sFilename) > 0 Then
        If InStr(sFilename,".") > 0 Then
            strExtension = Mid(sFilename,Len(sFilename) -
inStr(strReverse(sFilename),".")+2)
            arrAllowedFiles = Split(strAllowedFiles,",")

```

```

        AllowedFile = False
        For intCount = 0 To Ubound(arrAllowedFiles)
            If arrAllowedFiles(intCount) <> "" Then
                If Lcase(strExtension) =
Lcase(Trim(arrAllowedFiles(intCount))) Then
                    AllowedFile = True
                    Exit For
                End If
            End If
        Next
    Else
        AllowedFile = False
    End If
Else
    AllowedFile = False
End If
End Function

Private Function SaveBinaryData(byval bData, byval sFilename, byval bOverwrite)
'as boolean
    Dim objFs, objTextFile
    Dim intCount, strFile
    If LenB(bData) < 2 Then
        strError = "No data."
        SaveBinaryData = False
        Exit Function
    End If

    Set objFs = Server.CreateObject("scripting.filesystemobject")
    If Not objFs.FolderExists(strPath) Then
        strError = "Directory does not exists."
        SaveBinaryData = False
        Exit Function
    End If

    If Not bOverwrite And objFs.FileExists(sFilename) Then
        strError = "File already exists."
        SaveBinaryData = False
        Exit Function
    End If

    Set objTextFile = objFs.CreateTextFile(sFilename,True,False)

    For intCount = 1 To LenB(bData)
        objTextFile.Write Chr(AscB(MidB(bData,intCount,1)))
    Next

    objTextFile.Close

```

```

Set objTextFile = Nothing
Set objFs = Nothing
Session("file") = Null
SaveBinaryData = True
End Function

Private Function GetData(byval bFile) as bytearray
    Dim intStart, intEnd

    If LenB(bFile) < 1 Then
        GetData = ChrB(10)
        Exit Function
    End If
    intStart = InStrB(bFile, ChrB(13) & ChrB(10) & ChrB(13) & ChrB(10)) + 4
    intEnd = InStrB(bFile, ChrB(13) & ChrB(10) & ChrB(45) & ChrB(45) &
ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) &
ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) &
ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) &
ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45) & ChrB(45))
    If intStart > 0 Then
        If intStart < intEnd Then
            GetData = MidB(bFile, intStart, intEnd - intStart)
        Else
            GetData = ChrB(10)
        End If
    Else
        GetData = ChrB(10)
    End If
End Function

Private Function GetFilename(byval bFile) as string
    Dim bytFilename, bytChar, strFilename
    Dim intStart, intCount

    If LenB(bFile) < 1 Then
        GetFilename = ""
        Exit Function
    End If

    If LenB(bFile) > 0 Then
        If InStrB(bFile, ChrB(102) & ChrB(105) & ChrB(108) & ChrB(101) &
ChrB(110) & ChrB(97) & ChrB(109) & ChrB(101) & ChrB(61)) Then
            intStart = InStrB(bFile, ChrB(102) & ChrB(105) & ChrB(108) &
ChrB(101) & ChrB(110) & ChrB(97) & ChrB(109) & ChrB(101) & ChrB(61)) + 10
            For intCount = intStart To LenB(bFile)
                bytChar = MidB(bFile, intCount, 1)
                If bytChar = ChrB(34) Then
                    Exit For
                End If
            Next intCount
            strFilename = LeftB(bFile, intCount - 1)
        End If
    End If
End Function

```

```

                                End If
                                bytFilename = bytFilename & bytChar
                            Next
                        End If
                    End If
                    For intCount = 1 To LenB(bytFilename)
                        strFilename = strFilename &
Chr(AscB(MidB(bytFilename,intCount,1)))
                    Next
                    strFilename = Mid(strFilename,Len(strFilename) -
inStr(strReverse(strFilename),"\")+2)
                    GetFilename = strFilename
                End Function

Private Function GetContentType(byval bFile)
    Dim bytContentType, strContentType, bytChar
    Dim intStart, intCount

    If LenB(bFile) < 1 Then
        GetContentType = ""
        Exit Function
    End If

    If inStrB(bFile,ChrB(67) & ChrB(111) & ChrB(110) & ChrB(116) & ChrB(101)
& ChrB(110) & ChrB(116) & ChrB(45) & ChrB(84) & ChrB(121) & ChrB(112) &
ChrB(101) & ChrB(58)) > 0 Then
        intStart = inStrB(bFile,ChrB(67) & ChrB(111) & ChrB(110) &
ChrB(116) & ChrB(101) & ChrB(110) & ChrB(116) & ChrB(45) & ChrB(84) & ChrB(121)
& ChrB(112) & ChrB(101) & ChrB(58)) + 14
        For intCount = intStart To LenB(bFile)
            bytChar = MidB(bFile, intCount,1)
            If bytChar = ChrB(13) Then
                Exit For
            End If
            bytContentType = bytContentType & bytChar
        Next
    End If
    For intCount = 1 To LenB(bytContentType)
        strContentType = strContentType &
Chr(AscB(MidB(bytContentType,intCount,1)))
    Next
    GetContentType = strContentType
End Function
End Class
'-----

```

%>

حالا کافیسٲ در فایل اول در خط ۱۵ مسیر را برای ذخیره فایلها روی سرویس‌دهنده خود مشخص کنید :

```
;"Path = "D:\Inetpub\wwwroot\test.
```

مسیر تعیین شده باید حتماً موجود باشد.

ارسال ایمیل با کمک ASP

استفاده از آبجکت CDONTS که یکی از کامپوننتهای ویندوز سرور ۲۰۰۰ و ان تی می باشد به راحتی می توان در اسکریپتهای سمت سرور ایمیل ارسال کرد.
طریقه استفاده به شکل زیر می باشد

```
<%  
Dim objNewMail  
Set objNewMail = Server.CreateObject("CDONTS.NewMail")  
%>
```

مقادیر قابل تنظیم برای این آبجکت به شرح جدول زیر می باشد.

From

یک مقدار رشته ای که آدرس پست الکترونیکی فرستنده را مشخص می کند
(به طور مثال Me@somewhere.com)

To

یک مقدار رشته ای که آدرس پست الکترونیکی گیرنده را مشخص می کند.
(به طور مثال, someone@somewhere.com)
آدرس گیرنده های مختلف با استفاده از علامت روبرو امکان پذیر میباشد ";"

Subject

عنوان پیام

Body

متن پیام

Cc

آدرس پست الکترونیکی شخصی که قرار است یک کپی از نامه را دریافت کند

Bcc

آدرس الکترونیکی شخصی که قرار است یک کپی از نامه را دریافت کند (بدون اطلاع دریافت کننده نامه)

Importance

یک مقدار عددی که نشان دهنده درجه اهمیت نامه می باشد
(برای مثال خیلی مهم , مهم , عادی)

BodyFormat

یک مقدار عددی که نشان دهنده فرمت اطلاعات متن نامه می باشد
ObjMail.MailFormat = 0 (Mime format)
ObjMail.BodyFormat = 1 (متن عادی)

MailFormat

یک مقدار عددی که نشان دهنده فرمت اطلاعات نامه می باشد
(ObjMail.MailFormat = 0 (Mime format)

ObjMail.MailFormat = 1 (متن عادي)

متودهاي اين آبجکت

AttachFile

این متود یک فایل را به نامه پیوست می کند

AttachURL

این متود آدرس فایل پیوست شده را تخصیص می دهد

Send

این متود نامه را ارسال می کند

بعد از ارسال ایمیل مقادیر در حافظه باقی می ماند و شما باید حافظه را خالی کنید. همچنین برای ارسال یک ایمیل جدید باید یک آبجکت جدید ایجاد کرده و از آن استفاده کنید. کد زیر نحوه استفاده از این آبجکت را در asp نشان می دهد.

```
<%
```

```
Option Explicit
```

```
Dim objNewMail
```

```
'ابتدا یک آبجکت از نوع cdonts ایجاد می کنیم
```

```
Set objNewMail = Server.CreateObject("CDONTS.NewMail")
```

```
'بعد از ایجاد آبجکت ما باید مقادیر مربوط به آن را تنظیم کنیم'
```

```
objNewMail.From = "webmaster@iran.com"
```

```
objNewMail.To = "test@yahoo.com"
```

```
objNewMail.Subject = "این یک ایمیل تستی است"
```

```
objNewMail.Body = "این متن نامه است"
```

```
objNewMail.Send
```

```
'بعد از ارسال ایمیل با استفاده از کد زیر فضای حافظه را خالی می کنیم'
```

```
Set objNewMail = Nothing
```

```
Response.Write "ایمیل ارسال شد"
```

```
%>
```

شي RESPONSE و استفاده از فايل هاي include

براي پاسخ گويي به مرورگر ، از متدهاي شيء Response استفاده مي شود . در واقع شيء Response راهي براي برقراري ارتباط با سرويس گيرنده است .

شيء Scripting Context :

اين شيء امکان دسترسى اشياء ActiveX خارجي به اشياء دروني ASP را فراهم مي کند. البته بهتر است به جاي اين شيء از شيء Object Context استفاده کنید . براي گرفتن شيء Object Context نيز بايد متد Object Context() را فراخواني کنید . شيء Scripting Context ، تمام اشياء ASP را درون خود جمع مي کند . هنگامي که يك صفحه ASP شامل رفرنس هايي به اشياء ActiveX مي باشد اجراء ميشود ، ابتدا موتور ASP متد On Start Page هر يك از آن اشياء ActiveX را فراخواني کرده و شيء Scripting Context را به عنوان پارامتر به آن مي فرستد. اشياء ActiveX با استفاده از اين پارامتر مي توانند اشاره گر هايي به اشياء Server ، Application ، Request و Response به دست آورند. در اين جا ليستي از ويژگي هاي شيء Scripting Context را ملاحظه مي کنید :

- Server
- Application
- Session
- Request
- Response

هر يك از اين ۵ ويژگي يك اشاره گر (Pointer) به يكي از اشياء دروني ASP بر مي گردانند .
• توجه : Microsoft توصيه مي کند که به جاي استفاده از شيء Scripting Context از متد get Object Context اگر چه امکان استفاده از شيء Scripting Context کماکان وجود دارد، ولي بايد بدانيد که اين شيء از رده خارج شده و در برنامه هاي جديد خود نبايد از آن استفاده کنید .

شيء Object Context :

شيء Object Context به عنوان پل ارتباطي با MTS (Microsoft Transaction Server) تلقي مي شود. با MTS شما مي توانيد کاري کنید که اشياء ActiveX در تراکنش هايي که صفحه ASP آغاز شده اند شرکت کنند. همچنين با استفاده از شيء Object Context در هنگام نوشتن اشياء ActiveX مي توانيد به اشياء دروني ASP دسترسى پيدا کنید .
آشنائي با ساختار يك برنامه ASP

از نظر موتور ASP ، يك برنامه کاربردي عبارت است از مجموعه اي از تمام فايل و زير دايرکتوري هاي (Subdirectory) موجود در يك دايرکتوري که شامل فايل asa . Global باشد . بيشتر برنامه هاي کاربردي ASP شامل فايل هاي ASP و فايل هاي Include (با پسوند .inc) هستند که هر دوي آنها مي توانند ترکيبي از HTML ، کد و فايل هاي گرافيکي باشند. البته شما مي توانيد فايل هاي ASP را با فايل هاي HTML يا هر فايل ديگري که براي سرويس دهنده قابل شناسايي باشد ترکيب کنید .

بالا ترين دايرکتوري در اين ساختار شامل فايل asa . Global است . اين فايل دايرکتوري ريشه (Root Directory) برنامه کاربردي را مشخص مي کند .

اولين باري که يك کاربر يکي از فايل هاي ASP موجود در يکي از دايرکتوري هاي برنامه را درخواست مي کند ، موتور ASP ساختار درختي درختي دايرکتوري ها را به سمت بالا طي ميکند تا فايل asa . Global را پيدا کند و يا به ريشه وب (Root Web Directory) برسد . اين نکته از اهميت زيادي برخوردار است ، چون اگر شما چند برنامه کاربردي را درون هم قرار دهيد (که به آنها برنامه هاي کاربردي تودرتو - Nested Applications - مي گويند) ، فايل asa . Global که

برای هر درخواست کاربر اجرا می شود بستگی دارد به این که کاربر ابتدا کدام فایل را درخواست نماید . چون نمی توان اولین درخواست کاربر را کنترل نمود ، بهتر است برنامه های ASP را درون هم قرار ندهید ؛ مگر این که دلیل موجهی برای این کار داشته باشید .

ساختن یک برنامه کاربردی وب تحت IIS 4.0 به یک مرحله دیگر نیز دارد : شما باید به IIS بگویید که دایرکتوری مجازی (Virtual Directory) در برگیرنده فایل Global . asa ، دایرکتوری ریشه برنامه کاربردی است .

صرف نظر از این که کدام فایل درخواست شود، فایل Global . asa همواره اولین فایل خواهد بود که اجرا میشود. بنابراین اگر در نظر دارید کاربر برنامه کاربردی شما را با صفحه خاصی که مورد نظر شماست آغاز کند ، این فایل بهترین جایی است که میتوانید مرورگر را به آن صفحه هدایت کنید .

برنامه کاربردی که در شکل قبل ملاحظه فرمودید ، شامل چندین فایل ASP و دو زیردایرکتوری به نام های images و include می باشد. البته از نظر تعیین ساختار دایرکتوری های یک برنامه کاربردی هیچ محدودیتی وجود ندارد ؛ شما می توانید همه فایل ها را درون یک دایرکتوری قرار دهید . ولی در عمل ، بهتر است که فایل ها را بر اساس کارایی آنها دسته بندی کنید چون این کار ساختن و نگهداری برنامه را بسیار آسان تر می کند .

• توجه : دایرکتوری ریشه وب و دایرکتوری ریشه برنامه ASP لزوما نباید یکسان باشند. به عنوان مثال ،

می توانید برای چندبرنامه کاربردی ASP از یک فایل Global . asa استفاده کنید. در این شرایط می توانید فایل Global . asa را درون یک دایرکتوری قرار داده و سپس هر یک از زیر دایرکتوری های درون آن را به عنوان یک دایرکتوری مجازی تعریف کنید که بر اساس برنامه کاربردی مربوطه نامگذاری شده اند . ساختار دایرکتوری موجود در شکل زیر ، شامل ۴ برنامه کاربردی است : k۴۰۱، Paycheck، Retirement، Timesheet .

بالاترین دایرکتوری ، که HR Applications نام دارد، شامل فایل Global . asa است . هر زمان که یک کاربر به یکی از ۴ برنامه کاربردی موجود متصل شود ، موتور ASP ساختار درختی دایرکتوری هارا بالا می رود تا به دایرکتوری

HR Applications که در برگیرنده فایل Global . asa است برسد . یکی از دلایلی که می توان برای مرتب کردن برنامه ها به این شکل در نظر گرفت ، این است که ممکن است هر چهار برنامه از یک روش امنیتی استفاده کنند و یا ممکن است تمام برنامه ها از یک نوع روش برای برقراری ارتباط با پایگاه داده و یا از یک پایگاه داده مشترک استفاده کنند و شما بخواهید اطلاعات اولیه را در فایل Global . asa تنظیم کنید .

استفاده از فایل های Include :

برای جلوگیری از تکرار کد یا دستورات HTML در صفحات ASP ، می توانید از فایل های خارجی استفاده کرده و آنها را در صفحات ASP خود بگنجانید . یک فایل Include در یک فایل ASP از دستور INCLUDE استفاده می شود . دستور INCLUDE مشابه این است که محتویات فایل include مورد نظر را درون یک صفحه دیگر قرار دهید. در این جا مثالی از یک دستور INCLUDE را مشاهده می کنید :

```
< -- "INCLUDE = "C:\include\my Include .inc #--!>
```

این دستور سبب می شود که سرویس دهنده آن را با محتویات فایل مشخص شده جایگزین کند. دو نوع دستور INCLUDE موجود است : یکی INCLUDE FILE # و دیگری INCLUDE VIRTUAL # . نوع اول برای مشخص کردن فایل به یک آدرس فیزیکی نیاز دارد در حالی که نوع دوم به آدرس مجازی به فایل اشاره می کند . به هر حال از هر روشی که استفاده کنید، موتور ASP قبل از شروع پردازش دستورات ، آن را با محتویات فایل مورد نظر جایگزین می کند . بنابر این برای

گنجاندن يك فايل نمي توانيد از دستورات شرطي استفاده كنيد . به عنوان مثال ، كد زير به گونه‌اي كه انتظار مي رود عمل نخواهد كرد ؛ موتور ASP هر دو فايل را خواهد گنجاند :

```
< %  
if my Var = True than  
% >  
< ! -- # INCLUDE FILE = " C: \ include \ my Include . inc" -- >  
< %  
else  
%>  
< ! -- # INCLUDE FILE = " C: \ include \ your Include . inc" -- >  
< %  
end if  
% >
```

هنگامي كه موتور ASP اين فايل را تحليل مي كند ، دو دستور INCLUDE را با فايل هاي نامبرده جايگزين مي كند و اين كار را قبل از اجراي دستور if...end if انجام مي دهد . پس باز هم تأكيد مي كنم : شما با استفاده از كد نمي توانيد براي موتور ASP تعيين كنيد كه کدام فايل را بگنجاند و يا اين كه اصلا يك فايل را بگنجاند يا خير.

برنامه کاربردی تحت وب چیست ؟

يك برنامه کاربردی تحت وب بايك وب سایت تفاوت دارد . يك وب سایت شامل اطلاعات ثابتی است كه از پیش آماده شده و به صورت فایل های HTML ذخیره شده اند . در این حالت جهت حرکت اطلاعات از سرویس دهنده به

سرویس گیرنده می باشد . در واقع وب سایت اطلاعات را به کاربر ارائه می دهد . بین درخواست ها ، سرویس دهنده از عملکرد سرویس گیرنده اطلاعی ندارد . سرویس گیرنده می تواند از يك محل به محل دیگر ببرد و این کار روی وب سایت اولیه هیچ تأثیری ندارد ، چون هر يك از صفحات به صورت يك واحد كاملاً جدا می باشند . در واقع وب سایت شامل گروهی از فایل های HTML از هم جدا می باشد .

در مقابل ، يك برنامه کاربردی تحت وب ، اطلاعات را برای يك کاربر یا گروهی از کاربران به شکلی خاص فرمت کرده و ارائه می دهد . در این حالت ، جهت حرکت اطلاعات دو طرفه می باشد ؛ محتوایاتی كه توسط مرور گر

نمایش داده می شوند ، بر حسب اطلاعات ورودی کاربر یا هویت وی تعیین می گردند . بنابر این يك برنامه کاربردی تحت وب ، اطلاعات را به صورت دینامیکی ارائه می دهد نه به صورت فایل های HTML ثابت . در واقع وظیفه برنامه ، فراهم کردن اطلاعات بر حسب شرایط است كه این اطلاعات معمولاً از پایگاه داده (Database) بدست می آید . برنامه کاربردی نه تنها اطلاعات می دهد ، بلكه می تواند از کاربر اطلاعات بگیرد و بکارهای مختلف پاسخ مناسب دهد . به عبارت دیگر ، يك برنامه کاربردی ، عملکرد کاربر را از يك درخواست به

درخواست دیگر ردیابی می کند . بنابراین کاربر می تواند آن گونه كه در يك وب سایت به جستجو و گردش می پردازد ، به شكل دلخواه تمام صفحات يك برنامه کاربردی را بررسی و مشاهده کند . بلكه این برنامه کاربردی است كه صفحات مناسب را در شرایط مقتضی به طور دینامیکی تولید کرده و به کاربر ارائه می دهد .

برای نوشتن برنامه های کاربردی تحت وب از زبان های مختلفی می توان استفاده کرد ، مثل : Visual Basic , C , C ++ , Perl , Python , Java حتي Quick Basic یا Script های Unix .

در هر صورت مواردی كه در زیر به آنها اشاره می شود جزء ضروریات يك برنامه کاربردی هستند :

- ارتباط با يك پایگاه داده
- سرعت
- کار با رشته ها
- امنیت (Security)
- تراکنش ها (Transactions)
- تراکنش چیست ؟

يك برنامه کاربردی تحت وب اغلب باید کارهای مختلفی انجام دهد كه هر يك از آنها برای تداوم اعتبار اطلاعات باید به طور كامل انجام شوند . هنگام کار با يك پایگاه داده ، معمولاً این کارها را بصورت يك تراکنش در نظر می گیریم . يك تراکنش قرار دادي است كه موفقیت یا عدم موفقیت مجموعه ای از عملیات را تضمین می کند كه در صورت عدم موفقیت ، اطلاعات باید دست نخورده باقی بمانند . به عنوان يك نمونه کلاسیك ، می توانیم يك حساب چك (Checking Account) را در نظر بگیریم .

هنگامی كه يك چك را به حساب خود می گذارید ، بانك مبلغ مشخص شده را از حساب مربوطه برداشته و به حساب شما واریز می کند . كل این عملیات در صورتی موفقیت آمیز خواهد بود كه برداشت و واریز پول به طور كامل انجام شود . در این حالت عمل دو گانه برداشت از يك حساب و واریز به يك حساب دیگر را يك تراکنش می گوئیم .

تفاوت میان برنامه های IIS و DHTML

برنامه های IIS ، برنامه هایی هستند كه بر روی سرویس دهنده و تحت Microsoft Internet Information Server اجرا می شوند . در مقابل ، برنامه های DHTML برنامه Microsoft Internet Explorer استفاده می کنند و به این ترتیب به ورودی های کاربر پاسخ سریع می دهند .

این برنامه ها برای نوشتن بازی ها و شبیه سازی محیط و گرفتن اطلاعات از کاربر مناسب هستند . متأسفانه این نوع برنامه ها به نسخه چهارم IE یا نسخه های بالاتر آن نیاز دارند . با توجه به این که در این کتاب به تفصیل در مورد ASP و مسائل حاشیه ای آن بحث خواهیم کرد ، بد نیست کمی بیشتر در مورد برنامه های IIS توضیح دهیم . چون هدف از یادگیری ASP ، نوشتن برنامه های IIS می باشد که بر روی سرویس دهنده اجرا می شوند .

برنامه های IIS با چندین مرورگر مختلف و نسخه های متفاوت آنها سازگاری دارند . همچنین در محیط های مختلف Mak یا Unix نیز اجراء می شوند . بنابر این این نوع برنامه ها برای نوشتن برنامه های تجاری که مخاطبین زیادی دارند مناسب هستند ؛ چون دلیلی ندارد همه کاربران از ویندوز استفاده کرده و یا نسخه IE را در اختیار داشته باشند.

با توجه به این که برنامه های IIS بر روی سرویس دهنده اجرا می شوند ، برای به روز در آوردن آنها تنها کافی است کد موجود بر روی سرویس دهنده را تغییر داد. این نوع برنامه ها از هر محلی قابل دسترسی می باشند . هنگامی که یک برنامه IIS بر روی سرویس دهنده نصب شده و به اجرا در آمد ، هر کامپیوتری که بتواند به آن سرویس دهنده متصل شود می تواند از آن برنامه استفاده کند .

شما می توانید برای شناسایی کاربران ، به آنها نام و کلمه عبور اختصاص دهید و هنگام برقراری ارتباط با برنامه کاربردی ، نام و کلمه عبور را در خواست کنید . با توجه به این که تمام اطلاعات بر روی یک کامپیوتر ذخیره می شوند ، هنگامی که کاربر برنامه کاربردی را ترک کند ، هیچ نوع اطلاعاتی را از دست نمی دهد .

در قسمت بعد ، در مورد چگونگی کارکرد برنامه های IIS بحث خواهیم کرد .
چگونگی در خواست فایل ها توسط مرورگر :

هنگامی که شما یک آدرس URL در فیلد آدرس مرورگر خود تایپ می کنید ، چند چیز اتفاق می افتد . مرورگر ، آدرس وارد شده را تحلیل کرده و پیغامی به سرویس دهنده نام (Name Server) می فرستد تا نام مشخص شده

(به عنوان مثال ، Microsoft . com) را به یک آدرس (Internet Protocol) IP – (به عنوان مثال ، ۲۰۷,۸۴,۲۵,۳۲) ترجمه کند . سپس مرورگر با استفاده از آن آدرس IP به سرویس دهنده مربوطه متصل شده و فایل را درخواست می کند . سرویس دهنده فایل را می خواند و محتویات آن را به مرورگر بر می گرداند . در این مرحله مرورگر ، دستورات HTML آن فایل را تحلیل و فرمت فایل را تعیین می کند . بیشتر فایل های HTML درون خود به تصاویر گرافیکی اشاره می کنند. این اشاره ها به صورت رفرنس (Reference) بوده که هریک از آنها نیز بایک آدرس URL مشخص می شوند .

بنابراین کل این عملیات برای تکه تکه تصاویر و یا فایل های دیگری که با رفرنس به آنها اشاره شده است تکرار می شود.

به این ترتیب پروسه نمایش یک فایل HTML شامل تعدادی تراکنش کوچک بین سرویس گیرنده (یعنی مرورگر) و سرویس دهنده وب می باشد .

برنامه های IIS نیز تقریباً به همین صورت عمل می کنند. ابتدا یک مرورگر یک صفحه را از سرویس دهنده درخواست می کند . این درخواست همیشه برای یک فایل خاص می باشد . پاسخ سرویس دهنده به نوع فایل درخواستی بستگی دارد. اگر فایل مورد نظر از نوع HTML (یا پسوند .html یا .htm) باشد ، سرویس دهنده به سادگی محتویات فایل را می خواند و محتویات آنرا به صورت یک رشته در آورده و براساس قوانین URL کد گذاری می کند و سپس رشته حاصله را به مرورگر برمی گرداند . کل این پروسه ، از درخواست تا پاسخ ، یک تراکنش مابین سرویس گیرنده و سرویس دهنده تلقی می شود . سرویس گیرنده همواره تراکنش را شروع می کند و سپس تا زمان رسیدن پاسخ از سوی سرویس دهنده منتظر می ماند که در این زمان تراکنش کامل شده و خاتمه می یابد .

درخواست فایل از سرویس دهنده وب مشابه هنگامی است که در Windows Explorer روی یک فایل شبکه ای (فایلی که بر روی کامپیوتر سرویس دهنده قرار دارد) دوبار – کلیک می کنید ؛ البته با دو تفاوت :

- سرویس دهنده وب هیچگاه اجازه نمی دهد که برنامه محلی شما (یعنی مرورگر) فایل درخواستی را باز کرده یا در آن چیزی بنویسد؛ در عوض خودش فایل را باز می کند و محتویات آن را برمیگرداند.

- ارتباط مابین سرویس گیرنده و سرویس دهنده گذرا (Transient) است. برای برقراری ارتباط با سرویس دهنده وب نیازی به مشخص کردن نام درایوی که فایل بر روی آن قرار دارد نیست. پس از این که سرویس دهنده وب به درخواست شما پاسخ داد، ارتباط را قطع می کند. هنگامی که تراکنش کامل شد، سرویس دهنده وب، سرویس گیرنده را به طور کلی از یاد می برد. اگر بلا فاصله کلید Refresh (موجود بر روی نوار ابزار IE) را کلیک کنید. سرویس دهنده وب، آن تراکنش را تکرار می کند و اصلاً به خاطر نمی آورد که ۵ ثانیه قبل این فایل توسط شما درخواست شده است.

اگر ارتباط براساس پروتکل استاندارد HTTP برقرار شود، بیشتر فایل ها به چندین تراکنش نیاز دارند، یکی برای فایل Html اصلی و بقیه برای یکایک فایل های گرافیکی که در فایل اصلی به آنها اشاره شده است. بنابراین برای نمایش يك فایل که شامل ۵ تصویر گرافیکی است، مرورگر درخواست جداگانه به سرویس دهنده وب می فرستد.

فرم ها در ASP

مثالها

يك فرم با متد "get"

چگونه با استفاده از دستور Request.QueryString به کاربر عكس العمل نشان دهيد.

يك فرم با متد "post"

چگونه با استفاده از دستور Request.Form به کاربر عكس العمل نشان دهيد.

يك فرم با Radio Button ها

چگونه با استفاده از دستور Request.Form و از طريق Radio Button ها به کاربر عكس العمل نشان دهيد.

ورودی کاربر

آبجكت Request می تواند برای گرفتن اطلاعات از کاربر از فرمها استفاده گردد.

```
<form method="get" action="simpleform.asp">  
First Name: <input type="text" name="fname">  
<br />  
Last Name: <input type="text" name="lname">  
<br /><br />  
<input type="submit" value="Submit">  
</form>
```

ورودی های کاربر می توانند از دو طریق استفاده گردد: با Request.QueryString یا Request.Form

Request.QueryString

دستور Request.QueryString برای جمع کردن مقادیر موجود در يك فرم با متد "get" استفاده می شود. اطلاعات فرستاده شده از فرمی با متد "get" برای هر کسی قابل مشاهده است (این اطلاعات در قسمت آدرس Browser نشان داده می شود) و در مورد مقدار اطلاعات محدودیت دارد.

اگر کاربر در فرمی با متد "get" مقدار "Bill" و "Gates" را وارد کنید، URLی که به سرور فرستاده می شود شبیه به این خواهد بود:

<http://www.irandevlopers.com/simpleform.asp?fname=Bill&lname=Gates>

فرض کنید فایل "simpleform.asp" حاوی اسکریپت زیر است:

```
<body>  
Welcome  
<%  
response.write(request.querystring("fname"))  
response.write(" " & request.querystring("lname"))  
%>
```

</body>

Browser خط زیر در قسمت Body سند نشان خواهد داد:

Welcome Bill Gates

Request.Form

دستور Request.Form برای جمع کردن اطلاعات از يك فرم با متد "post" استفاده می‌شود. اطلاعات فرستاده شده از فرمی با متد POST برای بقیه غیرقابل مشاهده است و هیچ محدودیتی در مقدار اطلاعات فرستاده شده ندارد.

اگر کاربر در فرمی با متد "post" کلمه "Bill" و "Gates" را تایپ کند، URLی که به سرور فرستاده می‌شود شبیه به این خواهد بود:

<http://www.w3schools.com/simpleform.asp>

فرض کنید که فایل "simpleform.asp" حاوی اسکریپت زیر باشد:

```
<body>
Welcome
<%
response.write(request.form("fname"))
response.write(" " & request.form("lname"))
%>
</body>
```

Browser خط زیر در قسمت Body سند نشان خواهد داد:

Welcome Bill Gates

تصدیق فرم یا تست معتبر بودن (درست بودن) مقادیر ورودی

ورودی کاربر باید در Browser در هر جایی که امکان داشت (با استفاده از اسکریپت‌های client) تصدیق گردد. تصدیق Browser سریعتر است و زمان load شدن سرور را نیز کاهش می‌دهد.

در صورتی که مقادیر ورودی کاربر وارد Database می‌شود لازم است که معتبرسازی یا تصدیق ورودی‌های کاربر بر روی سرور انجام شود. يك راه خوب برای تصدیق ورودی‌های کاربر، به جای فرستادن اطلاعات به يك صفحه دیگر، پست کردن آنها به خود فایل است. در آن صورت کاربر می‌تواند خطاها را در خود صفحه‌ای که فرم در آن قرار دارد مشاهده نماید. این پیدا کردن خطا را برای وی ساده‌تر می‌کند.

طرز ساخت کوکی ها

يك Cookie اغلب برای شناسائی کاربر استفاده می‌شود.

مثالها

Cookie خوش‌آمدگویی
چگونه يك Cookie خوش‌آمدگویی درست کنیم.

يك Cookie چیست؟

يك Cookie اغلب برای شناسندن کاربر استفاده می‌شود. يك Cookie فایل كوچك است كه سرور آن را در كامپیوتر کاربر جاسازی می‌کند. هر بار كه همان كامپیوتر يك صفحه را درخواست می‌کند، Cookie نیز فرستاده می‌شود. با ASP شما می‌توانید هم مقادیر Cookie را بسازید و هم آنها را استخراج نمایید.

چگونه يك Cookie بسازید؟

Response.Cookie برای ساختن Cookie استفاده می‌شود.

توجه: Response.Cookie باید قبل از تگ درج شود.

در مثال زیر، ما يك Cookie درست خواهیم كردم و اسم آن را "firstname" می‌گذاریم و مقدار "Alex" را به آن می‌دهیم:

همچنین امکان دارد كه خصوصياتی را مثل تاريخ انقضای cookie را به آن بدهید:

چگونه مقدار cookie ها را استخراج کنیم

دستور "Request.Cookies" برای استخراج اطلاعات از cookie ها استفاده می‌شود.

در مثال زیر، ما مقدار cookie ی با نام "firstname" را استخراج کرده و در صفحه نشان دادیم:

خروجی:

Firstname=Alex

يك Cookie با کلید

اگر يك cookie حاوی مجموعه‌ای از مقادیر باشد، می‌گوییم آن cookie دارای کلیدهایی است.

در مثال زیر، ما يك مجموعه cookie با نام "user" درست می‌کنیم. cookie به نام "user" دارای کلیدهایی است كه حاوی اطلاعاتی راجع به کاربر است.

```
<%  
Response.Cookies("user")("firstname")="John"  
Response.Cookies("user")("lastname")="Smith"  
Response.Cookies("user")("country")="Norway"  
Response.Cookies("user")("age")="25"  
>%
```

خواندن همه Cookieها

به کد زیر نگاه کنید:

```
%>
Response.Cookies("firstname")="Alex"
Response.Cookies("user")("firstname")="John"
Response.Cookies("user")("lastname")="Smith"
Response.Cookies("user")("country")="Norway"
Response.Cookies("user")("age")="25"
<%
```

فرض کنیم که سرور شما همه cookieهای بالا را به کاربر فرستاده است.

حالا ما می‌خواهیم تمام cookieها را که به کاربر فرستاده شده است را بخوانیم. مثال زیر نشان می‌دهد که چگونه این کار را انجام دهیم (توجه داشته باشید که در مثال پایین با استفاده از HasKeys چک می‌کند که آیا cookie کلید دارد یا نه؟)

```
<html>
<body>
<%
dim x,y
for each x in Request.Cookies
  response.write("<p>")
  if Request.Cookies(x).HasKeys then
    for each y in Request.Cookies(x)
      response.write(x & ":" & y & "=" & Request.Cookies(x)(y))
      response.write("<br />")
    next
  else
    Response.Write(x & "=" & Request.Cookies(x) & "<br />")
  end if
  response.write "</p>"
next
%>
</body>
</html>
```

خروجی:

```
firstname=Alex user:firstname=John user:lastname=Smith user: country=Norway user:
age=25
```

اگر Browser دارای پشتیبانی Cookie نباشد چه اتفاقی می‌افتد؟

اگر برنامه شما با Browserی سرو کار داشته باشد که cookie را پشتیبانی نمی‌کند، باید از روش‌های دیگری برای انتقال اطلاعات از يك صفحه به صفحات دیگر در برنامه خود استفاده کنید. دو راه برای انجام این کار هست:

۱. اضافه کردن پارامترها به URL

شما می‌توانید پارامترها را به URL اضافه کنید:

```
<a href="welcome.asp?fname=John&lname=Smith">  
Go to Welcome Page</a>
```

و مقادیر موجود در "welcome.asp" را همچون مثال زیر استخراج کنید:

۲. استفاده از form

شما می‌توانید از فرم‌ها استفاده کنید. وقتی که کاربر روی دکمه Submit کلیک کند فرم ورودی‌های کاربر را به "welcome.asp" رد می‌کند:

```
<"form method="post" action="welcome.asp">  
<""=First Name: <input type="text" name="fname" value  
<""=Last Name: <input type="text" name="lname" value  
<"input type="submit" value="Submit">  
<form/>
```

مقادیر موجود در "welcome.asp" را همچون این مثال استخراج کنید:

```
<form method="post" action="welcome.asp">  
First Name: <input type="text" name="fname" value="">  
Last Name: <input type="text" name="lname" value="">  
<input type="submit" value="Submit">  
</form>
```

آبجکت Response در ASP

شما نمی توانید سورس کد های ASP را بوسیله انتخاب گزینه View Source در Browser مشاهده کنید، شما فقط خروجی فایل ASP را مشاهده می کنید که HTML ساده است. زیرا که اسکریپت ها قبل از این که نتایج به Browser فرستاده شود بر روی سرور اجرا می گردد.

قواعد اساسی نحو

یک فایل ASP به طور عادی دقیقا همانند HTML حاوی Tag های HTML است. اگر چه، یک فایل ASP می تواند همچنین حاوی Script ها نیز باشد. این اسکریپت ها به وسیله علامت های %< و %> احاطه شده اند. اسکریپت های سرور بر روی سرور اجرا می شود و می تواند حاوی هر گونه expressions، دستورات (Statements)، توابع، یا عملگرهایی باشد.

آبجکت Response

متد Write مربوط به آبجکت Response برای فرستادن محتویات به Browser استفاده می شود. به عنوان مثال، دستورات زیر برای فرستادن متن "سلام دنیا!" به Browser استفاده می شود.

```
<%  
response.write("سلام دنیا!")  
%>
```

VBScript

این امکان وجود دارد که از زبان های مختلف اسکریپت نویسی در فایل های ASP استفاده کنید. اگرچه زبان پیش فرض، VBScript است.

```
<html>  
<body>  
<%  
response.write("سلام دنیا!") %>  
</body>  
</html>
```

مثال بالا "سلام دنیا!" را در قسمت body متن می نویسد.

JavaScript

برای این که JavaScript را به عنوان زبان پیش فرض برای یک صفحه خاص مشخص کنید می بایست از راهنماهای زبانی در ابتدای صفحه خود استفاده کنید:

```
<%@ language="javascript"%>  
<html>  
<body>  
<%  
Response.Write("Hello World!")  
%>  
</body>  
</html>
```

نکته: JavaScript برعکس VBScript به بزرگ یا کوچک بودن حروف حساس است. شما مجبور خواهید بود که وقتی زبان احتیاج دارد، کد ASP خود را با حروف بزرگ یا کوچک بنویسید.

سایر زبان های اسکریپ نویسی

ASP همراه با VBScript و JScript (پیاده سازی میکروسافت از JavaScript) عرضه می شود. اگر شما می خواهید که با زبان دیگری همچون PERL - REXX یا Python کد بنویسید، مجبور خواهید بود که موتور Script مربوط به آنها را نصب کنید. مهم: به خاطر این که اسکریپت ها بر روی سرور اجرا می شوند، به هیچ وجه لازم نیست Browser ی که فایل ASP را نشان می دهد از Scripting پشتیبانی کند!

راه اندازی ASP

شما می توانید ASP را بدون نیاز به یک سرور خارجی اجرا کنید. برای انجام این کار شما باید Microsoft's Personal Web Server (PWS) یا Internet Information Server (IIS) را در رایانه خود نصب کنید.

چگونه ASP را در رایانه خود اجرا کنید؟

شما می توانید ASP را بدون سرور خارجی در رایانه شخصی خود اجرا کنید. برای انجام این کار باید Microsoft's (Personal Web Server (PWS یا Internet Information Server (IIS را در رایانه خود نصب کنید.

اگر در مورد اجرای ASP جدی هستید، باید حداقل ویندوز ۹۸، نسخه دوم (Second Edition) داشته باشید.

اگر در مورد اجرای ASP بسیار جدی هستید، باید از ویندوز ۲۰۰۰ استفاده کنید.

چگونه PWS را در ویندوز ۹۵ نصب کنید و ASP اجرا کنید؟

Personal Web Server (PWS) همراه ویندوز ۹۵ نیست!!

برای اجرای ASP بر روی ویندوز ۹۵، شما باید "Windows NT 4.0 Option Pack" را از مایکروسافت داون لود کنید.

داون لود "Windows NT 4.0 Option Pack"

چگونه PWS را در ویندوز NT نصب کنید و ASP اجرا کنید؟

Personal Web Server (PWS) همراه ویندوز NT نیست!!

برای اجرای ASP بر روی ویندوز ۹۵، شما باید "Windows NT 4.0 Option Pack" را از مایکروسافت داون لود کنید.

داون لود "Windows NT 4.0 Option Pack"

چگونه PWS را در ویندوز ۹۸ نصب کنید و ASP اجرا کنید؟

۱. شاخه Add-ons را در سی دی ویندوز ۹۸ خود باز کنید، شاخه PWS را پیدا کرده و فایل setup.exe را اجرا کنید.

۲. یک شاخه Inetpub در هارد شما ایجاد خواهد شد. آن را باز کنید و شاخه wwwroot را پیدا کنید.

۳. یک شاخه جدید درست کنید و نام آن را چیزی شبیه به "MyWeb" بگذارید.

۴. از یک ویرایشگر متن استفاده کنید و کدهای ASP را بنویسید، فایل را به نام "test1.asp" را در شاخه "MyWeb" ضبط کنید.

۵. مطمئن شوید که وب سرور شما اجرا است - برنامه نصب یک آیکن جدید در Task Bar شما اضافه کرده است (این نشانه PWS است). بر روی آن کلیک کنید و دکمه Start را در پنجره ای که آمده بزنید.

۶. Browser خود را باز کنید و تایپ کنید "http://localhost/MyWeb/test1.asp"، تا اولین صفحه ASP خود را ببینید.

چگونه PWS را در ویندوز Me نصب کنید و ASP اجرا کنید؟
Personal Web Server (PWS) در ویندوز Me وجود ندارد !!
مقاله مربوطه را در سایت مایکروسافت بخوانید

چگونه IIS را در ویندوز ۲۰۰۰ نصب کنید و ASP اجرا کنید؟

۱. از دکمه Start به Settings و سپس Control Panel بروید.
۲. در پنجره Control Panel آیکن Add/Remove Programs را انتخاب کنید.
۳. در پنجره Add/Remove Programs گزینه Add/Removes Windows Components را انتخاب کنید.
۴. در پنجره جدید گزینه Internet Information Services را انتخاب کرده، OK را بزنید.
۵. یک شاخه Inetpub در هارد شما ایجاد خواهد شد. آن را باز کنید و شاخه wwwroot را پیدا کنید.
۶. یک شاخه جدید درست کنید و نام آن را چیزی شبیه به "MyWeb" بگذارید.
۷. از یک ویرایشگر متن استفاده کنید و کدهای ASP را بنویسید، فایل را به نام "test1.asp" در شاخه "MyWeb" ضبط کنید.
۸. مطمئن شوید که وب سرور شما اجرا است - برنامه نصب یک آیکن جدید در Task Bar شما اضافه کرده است (این نشانه IIS است). بر روی آن کلیک کنید و دکمه Start را در پنجره ای که آمده بزنید.
۹. Browser خود را باز کنید و تایپ کنید "http://localhost/MyWeb/test1.asp"، تا اولین صفحه ASP خود را ببینید.

چگونه IIS را در ویندوز XP نصب کنید و ASP اجرا کنید؟
نکته: شما نمی توانید ASP را روی ویندوز XP نسخه خانگی (Home Edition) اجرا کنید.

۱. از دکمه Start به Settings و سپس Control Panel بروید.
۲. در پنجره Control Panel آیکن Add/Remove Programs را انتخاب کنید.
۳. در پنجره Add/Remove Programs گزینه Add/Removes Windows Components را انتخاب کنید.
۴. در پنجره جدید گزینه Internet Information Services را انتخاب کرده، OK را بزنید.
۵. یک شاخه Inetpub در هارد شما ایجاد خواهد شد. آن را باز کنید و شاخه wwwroot را پیدا کنید.
۶. یک شاخه جدید درست کنید و نام آن را چیزی شبیه به "MyWeb" بگذارید.

۷. از یک ویرایشگر متن استفاده کنید و کدهای ASP را بنویسید، فایل را به نام "test1.asp" را در شاخه "MyWeb" ضبط کنید.

۸. مطمئن شوید که وب سرور شما اجرا است - برنامه نصب یک آیکن جدید در Task Bar شما اضافه کرده است (این نشانه IIS است). بر روی آن کلیک کنید و دکمه Start را در پنجره ای که آمده بزنید.

۹. Browser خود را باز کنید و تایپ کنید "http://localhost/MyWeb/test1.asp"، تا اولین صفحه ASP خود را ببینید.

شروع کار با ASP

يك فایل ASP می تواند حاوی متون، Tag های HTML و اسکریپت ها باشد. اسکریپت ها در يك فایل ASP بر روی سرور اجرا می شوند.

به عنوان پیش نیاز چه چیزی را باید بدانید؟
قبل از شروع، باید با مسائل زیر را تا حدودی آشنا باشید

WWW و HTML و مسائل اولیه ساخت صفحات Web
يك زبان اسکریپت نویسی همچون JavaScript یا VBScript

ASP چیست؟

ASP حروف اختصاری Active Server Pages است.
ASP برنامه ای است که درون IIS اجرا می شود.
IIS حروف اختصاری Internet Information Services است.
IIS به عنوان يك مولفه رایگان با ویندوز ۲۰۰۰ عرضه می شود.
IIS همچنین به عنوان جزئی از Windows NT 4.0 Option Pack است.
Option Pack می تواند از سایت مایکروسافت داون لود شود.
PWS نسخه کوچکتر - ولی کاملاً عمل کننده - از IIS است.
PWS را می توانید در سی دی ویندوز ۹۵ یا ۹۸ خود پیدا کنید.

سازگاری ASP

ASP تکنولوژی مایکروسافت است.
برای اجرای IIS باید ویندوز NT نسخه ۴,۰ یا بالاتر را داشته باشید.
برای اجرای PWS باید ویندوز ۹۵ یا بالاتر را داشته باشید.
ChiliASP تکنولوژی است که ASP را بدون نیاز سیستم عامل ویندوز اجرا می کند.
InstantASP تکنولوژی دیگری است که ASP را بدون نیاز به سیستم عامل ویندوز اجرا می کند.

يك فایل ASP چیست؟

يك فایل ASP تقریباً شبیه به يك فایل HTML است.
يك فایل ASP می تواند حاوی متن، XML، HTML، و اسکریپت باشد.
اسکریپت های درون يك فایل ASP بر روی Server اجرا می شود.
يك فایل ASP پسوند ".asp" دارد.

فرق ASP با HTML چیست؟

وقتی که Browser (کاوشگر اینترنت) درخواست يك فایل HTML را بکند، Server آن فایل را بر می گرداند.
وقتی که Browser درخواست يك فایل ASP را بکند، IIS درخواست مربوطه را به موتور (هسته ای که وظیفه اصلی را بازی می کند) ASP ارجاع می دهد. موتور ASP فایل ASP را خط به خط می خواند، و اسکریپت های درون فایل را اجرا می کند. در نهایت فایل ASP به عنوان يك فایل HTML ساده به Browser برگردانده می شود.

ASP چه کاری می تواند برای شما انجام دهد؟

به طور دینامیکی محتویات يك Web page را ویرایش یا اضافه کند.
به درخواست های کاربر که از فرم های HTML ارسال شده اند پاسخ دهد.
به هر گونه اطلاعات با بانک اطلاعاتی دسترسی پیدا کرده و نتایج را به Browser برگرداند.
سفارشی نمودن (Customize) يك Web page تا برای کاربران مختلف مفیدتر باشد.
مزایای استفاده از ASP به جای CGI و Perl سرعت و سادگی است.
فراهم کردن امنیت بیشتر به گونه ای که کد ASP شما از طریق Browser قابل مشاهده نیست.
از آن جایی که فایل های ASP به شکل HTML برگردانده می شوند، میتوانند بوسیله هر Browser مشاهده گردند.
برنامه نویسی درست ASP می تواند ترافیک شبکه را به حداقل برساند.

