

**T.C.
FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**

**ASP
(Active Server Pages)**

BİTİRME ÖDEVİ

HAZIRLAYAN

00220506

FATİH MEHMET ARSLAN

DANIŞMAN

Yrd. Doç. Dr. H.HASAN BALIK

ELAZIĞ-2004

GİRİŞ

ASP Microsoft firması tarafından klasik HTML sayfalarına dinamik bir yapı kazandırmak amacıyla ortaya çıkarılmış bir teknolojidir. ASP ile web sayfalarını dinamik hale getirebilmek için ASP sunucu(server) tarafında yapılmış olan kodlar çalıştırılır ve istemci tarafı ona özel görüntülenmesi istenen sayfayı görür.

ASP kodlanırken bütün script dilleri kullanılabilir. En çok kullanılanları JavaScript ve VBScript'tir. Özellikle VBScript daha kolay ve referansları daha çoktur.

Kişisel Web Server Kurulumu

Bilgisayarınız Windows 95, 98, NT4 WorkStation veya NT4 Server ile çalışıyorsa, sisteminize bir Web Server programını siz kurmak zorundasınız. Windows 2000 Professional veya Windows 2000 Server ise Kişisel Web Server programını kendiliğinden kurar. Windows 98'e bir kişisel Web Server kurmaya geçmeden önce bilgisayarımıza bir kimlik vermemiz gerekir: Bilgisayarım/Denetim Masası/Ağ'ı tıklayarak açacağınız diyalog kutusunda ikinci sekme olan Tanımlama'yı açın ve "Bilgisayar adı" kutusuna istediğiniz adı yazın. Bilgisayarın ağ ortamında olması gerekmez.

Windows 98'e Kişisel Web Server kurmak için iki yol izleyebilirsiniz. Windows 98 CD-ROM'unda Add-ons klasöründeki PWS dizininde Kur.exe'yi tıklayın veya Windows NT Option Pack CD-ROM'unda Default.htm'i açın. Bilgisayarınızın Windows 98 ile çalıştığını algılayacak olan program size Personal (kişisel) Web Server (PWS) kurmayı önerecektir. Kişisel Web Server'ı kurarken her iki durumda da ikinci diyalog kutusunda Minimum/En az veya Typical/Tipik seçeneğini değil, Custom/Özel'i seçin ve açılacak yeni diyalog kutusunda Microsoft Data Access Components (MS Veri Erişim Bileşenleri) satırına işaret koyarak, Alt Bileşenleri Göster düğmesini tıklayın. Açılacak seçme kutusunda ise ADO Documentation satırına işaret koyun. Bu belgelerden daha sonra veri-yönlendirmeli Web Uygulaması yaparken yararlanılacaktır.

Kişisel Web Server kurulduktan sonra bilgisayarın yeniden başlatılması gerekir. Windows NT4.0 Workstation veya Server'a IIS4.0 kurmak için Option Pack CDROM'undaki default.htm'i çalıştırıp ve açılacak Browser penceresinde IIS'i kurma seçeneğini tıklamak yeterlidir. Burada da ADO Documentation'i sabit diske aktarabilmek için gerekli seçenek işaretlenmelidir.

Nasıl Çalışır?

Sayfamızı hazırladık ve uzantısı **.asp** olacak şekilde kaydettik. Peki nereye kaydettik? Öyle herhangi bir dizine kaydederseniz, bu dosyaları asla çalıştıramayız. Yapmamız gereken PWS'nin **kurulu** olduğu dizine kaydetmektir. Varsayılan olarak PWS, **c:\inetpub** dizinine kurulur ve sunucumuzun kök dizini ise **c:\inetpub\wwwroot** olur. Yani dosyalarımızı kaydederken, bu dizin içerisine kaydetmeliyiz. Bu dizin içerisinde olmak koşuluyla, istediğimiz gibi yeni dizinler oluşturabilir ve sayfaları o dizinlerden çağırabiliriz. Yani sunucumuzun kalbi **c:\inetpub\wwwroot** dizinindedir

Kaydetme işi tamam, her şeyi yaptık ama sayfalar browser'da gözükeceği yerde; ya **başka** bir program (ki çoğu kere bu program Photoshop olur) tarafından açılıyor, ya da **hiç** açılmıyor. Eğer dosyayı **çift** tıklayarak çalıştırmaya çalışırsak, bu tür sorunlarla karşılaşırız. Bu sayfaları çift tıklamayla çalıştıramayız. Yapmamız gereken çok basit; açalım browser'ımızı, adres satırına gelip ya **127.0.0.1/dosya_ismi.asp** ya da **localhost/dosya_ismi.asp** yazalım ve enter tuşuna basalım. Artık sayfamız görüntülendi. Burada 127.0.0.1, bizim makinamızın IP adresidir. **dosya_ismi.asp** kısmına ise, dosyayı hangi isimle kaydettiyseniz o ismi yazıyoruz.

Bir Örnek Yapalım

Şimdi ilk ASP sayfamızı yapalım ve bununla kişisel Web Server programının çalışıp çalışmadığını sınavalım. Buraya kadar anlamış olduğunuz gibi ASP sayfası da HTML gibi düz yazı dosyasıdır; dolayısıyla beğendiğiniz bir düz yazı programı ile ASP yazabilirsiniz. Eğer kelime-işlemci kolayınıza gidiyorsa, dosyayı ASCII veya ANSI biçiminde kaydetmeyi unutmayın (10'ncu satırın sonunda nokta olduğuna dikkat edin):

```
<HTML>
<HEAD>
<TITLE>ASP İLE İLK SAYFA</TITLE>
<META http-equiv="content-type" content="text/html; charset=ISO-8859-9">
<META http-equiv="Content-Type" content="text/html; charset=windows-1254">
</HEAD>
<BODY>
<H1><CENTER>Merhaba Dünya!</H1>
```

```
<H2>Bugün:  
<% Response.Write(Date) %>.  
</CENTER  
</H2>  
</BODY>  
</HTML>
```

Bu dosyayı **merhaba.asp** adıyla kaydedin ve kendi bilgisayarınızda kişisel Web Server'da veya ISS'te sınyacaksınız, bu programları çalıştırın. Browser'ınızın URL hanesine, kişisel Server'ınızın adıyla birlikte dosyanın adını yazın; (<http://localhost/merhaba.asp>) ve ilk ASP programımız dünyaya Merhaba desin.



Internet'te ASP

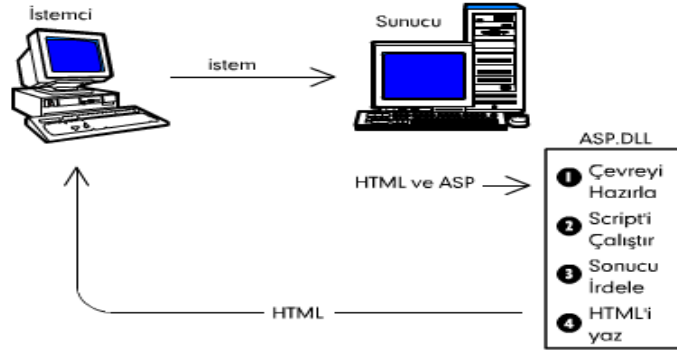
ASP teknolojisi ile oluşturduğunuz sayfaları, yani içinde ASP teknolojisinin gerektirdiği kodlar bulunan HTML dosyalarını, .asp uzatmasıyla kaydederiz. Bu dosyalar talep edildiğinde, Web Server programı, eğer ASP teknolojisini tanıyorsa, talep edilenin ASP sayfası olduğunu dosya adının uzatmasından anlar; ve bu sayfayı ziyaretçiye yollamadan önce kendisi işleme tabi tutar.

ASP sayfamızdaki kodlar büyük bir ihtimalle bir takım dinamik işler yapacak, örneğin hesaplamalar, araştırmalar, veri tabanından veri çekme, dosya sisteminden dosya isteme gibi görevlerini yerine getirecek, ve ortaya çıkacak olan HTML dosyasını ziyaretçiye gönderecektir. Diyelim ki bu kitapçıktaki örnekleri yaptınız ve ücretsiz Site yeri edinmiş olduğunuz Hosting firmasının bilgisayarındaki sitenize yerleştirdiniz. Sonra Browser'ınızın URL hanesine bu ASP dosyasının adını yazdınız ve sayfayı talep ettiniz. Karşınıza, ya ASP sayfasının içindeki kodları görüntüleyen veya sadece sayfanın HTML unsurlarına yer veren bir sayfa geldi! Bu Server'ınızın ASP anlamadığını gösterir.

ASP sayfalarınızı ASP-uyumlu Web sunucularında çalıştırabilirsiniz. Sitenizde ASP sayfaları bulunacaksa ve ücretli evsahibi firmalardan site alanı edinecekseniz, sunucunun ASP-uyumlu olmasına dikkat edin.

Şimdi, ilk ASP'mizi çalıştırdığımızı göre, biraz teknikten söz edebiliriz. HTML'in ziyaretçinin bilgisayarında çalıştığını biliyorsunuz; istemci Browser, sizin URL hanesine adını yazdığımız HTML dosyasını yine adresteki sunucu Web Server'dan ister. Web Server da bu dosyayı bulur ve içinde kayıtlı resim ve diğer unsurlarla birlikte istek sahibine gönderir. Fakat kimi zaman Server'a bize sadece bir dosyayı göndermesini değil, fakat bu dosyanın içinde kayıtlı komutlar varsa onları icra etmesini de bildirebiliriz. Bunu yapmanın bir yolu CGI programlarıdır. Bir diğer yolu ise ASP'dir.

Web Server, kendisinden bir ASP belgesi talep edildiğinde, kendi kendine "Dur bakalım! ASP istendiği zaman hemen alıp göndermek yok, önce bunu ASP.DLL programına gönderelim.. Ondan sonra işimize devam ederiz!" der.



ASP.DLL, kendisine bir .asp dosyasının geldiğini görünce, hemen ASP'lerin Anayasası olan **global.asa**'nin çalışıp çalışmadığına bakar. **global.asa**, tıpkı diğer ASP dosyaları gibi bir düz yazı dosyasıdır ve ASP programlarının çalışma koşullarını düzenleyen kuralları içerir. (Bu dosyayı sırası gelince ele alacağız.) Yukarıdaki örnekte gördüğümüz gibi ASP dosyası hem HTML kodları içerir, hem de içinde bir Script diliyle yazılmış kodlar vardır. ASP'ye "program" özelliği kazandıran bu Script dili ile yazılmış kodlardır. ASP.DLL, önce gelen .asp dosyasında hangi Script dilinin kullanıldığına bakar ve bunun için gerekli ortamı oluşturur; yani bu Script dilini yorumlayacak programı çalıştırır; bu program Script'i yorumlar ve icra edilecek komutları icra eder; ASP.DLL, icra edilen komutlar, işletim sisteminin yardımını istiyorsa (örneğin bir veritabanından veri çekmek gibi, veya dosya sistemine bir dosya açtırmak, yazdırmak, sildirmek gibi) bu yardımın edinilmesini de sağlar.

Bütün bu işlerin sonunda sizin yazdığınız HTML kodlarına ek yapmak (örneğin bir tablonun içeriğini, çekilen verilerle doldurmak veya dosya sisteminden edinilen bir dosyanın içeriğini sayfaya aktarmak gibi) gerekiyorsa bu ekleri ASP.DLL yapar.

ASP.DLL, kendisine sevk edilen dosyayı satır satır okur ve önce dosyadaki ASP kodlarının gerektirdiği HTML değerlerini bulur; yani önce ASP icra edilir, gereği yerine getirilir. Sonra HTML bölümleri ile birleştirilip sonuçta sunucuya saf ve temiz bir HTML sayfası gönderilir. Bu sayfanın içinde bir satır bile ASP kodu bulunmaz. Eğer sayfaya ziyaretçinin Browser'ında çalışması amacıyla Javascript veya VBScript kodları koydu iseniz, elbette bu kodlar HTML'in parçası olarak ziyaretçiye gidecektir. Fakat giden sayfada artık ASP'ye ilişkin hiç bir şey kalmamış olacaktır.

Bilindiği gibi, mevcut sürümleri itibariyle Browser programları içinde Netscape VBScript dilini anlamaz. ASP sayfalarımızda istediğimiz Script dilini kullanabiliriz; VBScript de kullanabiliriz. Netscape'in VBScript anlamamasıyla, ASP sayfalarımızda VBScript kullanmamızın bir ilgisi yoktur; çünkü ASP sayfasının ortaya çıkartacağı HTML kodunda ASP dolayısıyla VBScript bulunmayacaktır; dolayısıyla ASP sayfalarımız, Netscape tarafından da anlaşılıp, görüntülenebilecektir. Tabii ortaya çıkacak HTML sayfasının Netscape tarafından arzu ettiğiniz gibi görüntülenebilmesi için Netscape'in anlayabildiği dinamik HTML unsurlarına yer verme zorunluğunuz hâlâ devam ediyor.

ASP tekniğinin nasıl çalıştığını anladığımıza göre şimdi biraz terminoloji serpiştirelim ki, Webmaster meslektaşlarımızla biraraya geldiğinizde, gerçekten ASP ile mesela Javascript arasındaki farkı bildiğiniz belli olsun. ASP bir **Server-Side Script** teknolojisidir. İnternet'te istemci ile sunucu arasındaki çizginin sunucu tarafına **Server-Side** (Sunucu Tarafı), istemci tarafına da **Client-Side** (evet, doğru tahmin ettiniz: İstemci Tarafı) denir. Server-Side'da çalışan Script programları da "**Server-Side Script**" (Sunucu Tarafı Script'i) adını alır. Dolayısıyla şöyle bir sayfa yaparsanız (**merhaba.htm**), bu sayfada **Client-Side** Script teknolojisi kullanmış olursunuz:

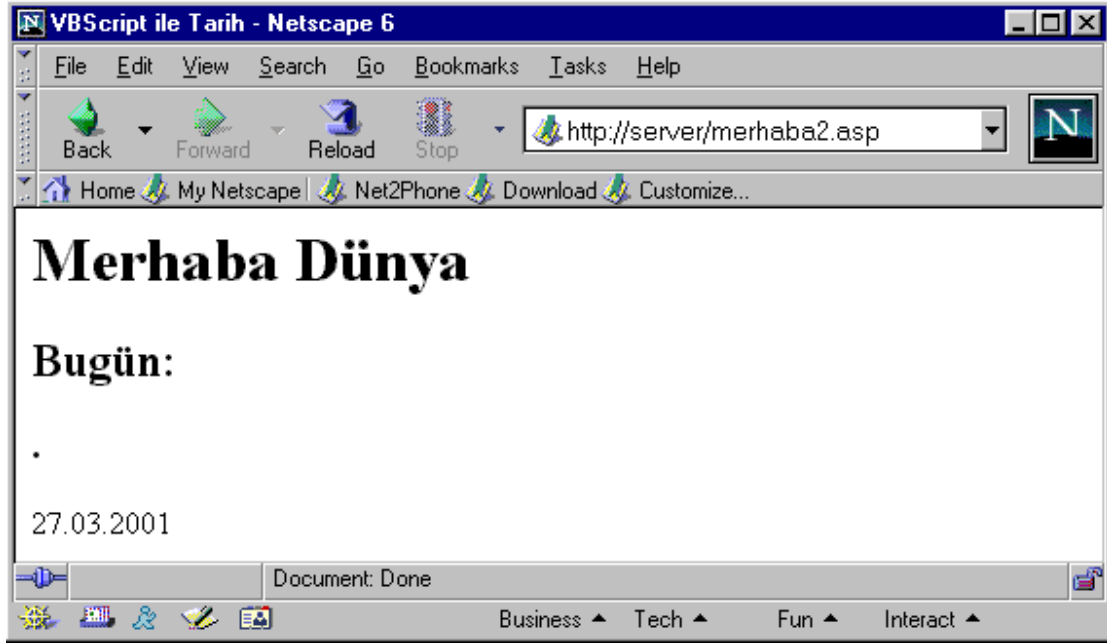
```
<HTML>
<HEAD>
<TITLE>JavaScript ile Tarih</TITLE>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
</HEAD>
<BODY BGCOLOR=WHITE>
<H1>Merhaba Dünya</H1>
<H2>Bugün:</H2>
<H3>
<SCRIPT LANGUAGE=JAVASCRIPT>
<!--
tarih = new Date();
document.write(tarih);
//-->
</SCRIPT>
.</H3>
</BODY>
</HTML>
```

HTML sayfanıza <SCRIPT>..
</SCRIPT> etiketleri arasına yerleştireceğiniz bu kodun çalışması için Server'ın hiçbir şey yapması gerekmez; kodu Browser çalıştıracak ve günün tarihini size (Türkçeleştirmediğiniz için İngilizce olarak) bildirecektir. Şimdi, bir de Server tarafında çalışan Script içeren sayfaya örnek verelim

merhaba2.asp Bu kodu yazarken kapanan Script etiketinden sonra nokta olduğuna dikkat edin:

```
<HTML>
<HEAD>
<TITLE>VBScript ile Tarih</TITLE>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
</HEAD>
<BODY BGCOLOR=WHITE>
<H1>Merhaba Dünya</H1>
<H2>Bugün:</H2>
<H3>
<SCRIPT LANGUAGE=VBScript RUNAT=SERVER>
Response.write(Date)
</SCRIPT>.
</H3>
</BODY>
</HTML>
```

Bu sayfada kullandığınız Script'i VBScript ile yazdığınızı farkettiler. Yani? Bu HTML sayfası, Netscape'de görüntülenemez! Diyemeyiz. Görüntülenir, çünkü Script'i Netscape değil, Server çalıştıracak. Bunu <SCRIPT> etiketinin içindeki "RUNAT" özelliği ile belirtiyoruz. ("Run," çalıştır; "at" ise "içinde, üzerinde" anlamına gelir. "RUNAT" "...de çalıştır" gibi bir anlama sahiptir.) Burada RUNAT'in karşısına yazdığımız SERVER ifadesi ile, Script'in Browser'a gönderilmeden önce Server'da çalıştırılmasını sağlıyoruz; Server'a "Bu Script'i Server'da Çalıştır!" demiş oluyoruz.



Netscape bu sayfayı görüntüleyecektir; ama galiba bir terslik var. **merhaba.asp** ile **merhaba2.asp**'nin ekran görüntülerini ve kodlarını karşılaştırırsanız, birinde noktanın olması gereken yerde olduğu halde diğerinde noktanın yanlış yerde olduğunu görüyoruz. Oysa iki kodu da karşılaştırırsanız, ikisinde de nokta işareti, kodun bittiği yerde yer alıyor. Yukarıda ASP'nin icrasında HTML sayfa Server'a verilmeden önce ASP.DLL tarafından içindeki ASP kodlarının icra edildiğini söylemiştik. Nitekim, burada gördüğümüz gibi, ASP.DLL, HTML sayfayı içinde Server'ın çalıştırması gereken bölümle birlikte Server'a gönderdi; Server kendisine gelen dosyada kendi icra etmesi gereken (RUNAT=SERVER özelliği verilmiş olan Script'i) bölümü icra etti ve sonucu Browser'a aktardı. Yani, Server'ın çalıştırdığı Script'in sonucu sayfaya ASP'den sonra eklendi. Dolayısıyla, nokta, yanlış yerde kaldı.

Bundan çıkartacağımız sonuç şu olacak: ASP sayfalarımıza, Browser, Server ve ASP.DLL tarafından çalıştırılacak kodları yerleştirirken, sırasına dikkat edeceğiz ve hangi sırada icra edilmelerini istiyorsak, o sırada koyacağız.

Server ile ASP.DLL'in ilişkisi sadece Script dilini çalıştırmaktan ibaret değildir. ASP, istemciden gelen HTTP İstemi (Request) ve HTTP'ye giden Karşılık (**Response**) unsurlarından tutun, ActiveX Data Objects (ADO, ActiveX Veri Nesneleri) aracılığıyla, işletim sisteminin sunacağı veritabanına erişim imkanını ve işletim sisteminin sunduğu dosya yönetimine kadar bir çok imkanı kullanır.

ASP'nin Unsurları

ASP tasarımcısı olarak, biz gerçekte ASP'nin Nesneleri ile birşeyler yaparız; başka bir deyişle ASP kodlarımız bu nesnelere yöneliktir, onları kullanma ve onlardan bir sonuç alma veya onlara bir sonuç aktarma amacına yöneliktir. ASP'nin Nesneleri altı grupta toplanır:

Application/Uygulama:

Bir ASP sitesi, gerçekte bir Uygulama Programı olarak görülür. Bu, HTML/CGI geleneğine aşina tasarımcı için yeni bir kavram. ASP'yi icad edenler; bir ziyaretçi bir ASP sayfasından girerek, bir sitede surfing'e başladığında, onu bir programı işleten bilgisayar kullanıcısı olarak görüyorlar. Böylece, sitemiz, her ziyaretçinin karşısına çıktığında “bir program çalışmış” gibi sayılıyor. Bu yaklaşımın Web tasarımcısı olarak bize kazandırdığı imkanları ele alacağız.

Session/Oturum:

Bir ziyaretçi sitemize geldiğinde, hangi sayfamızı talep ederse etsin, bu bağlantı ASP açısından bir oturum sayılır. Her oturumun belirli bir süre devam eden özellikleri, değişkenleri ve değerleri vardır. Site tasarımında oturum özelliklerinden geniş ölçüde yararlanacağız.

Request/Talep:

Browser'dan Server'a ulaşan bütün bilgiler, Request (Talep) nesnesinin öğeleridir. Bu nesneyi kullanarak, istemciden gelen her türlü HTTP bilgisini kullanırız.

Response/Karşılık:

Server'dan ziyaretçinin bilgisayarına gönderdiğimiz bütün bilgiler, çerezler (cookie) ve başlıklar (Header) Response (Karşılık) nesnesinin öğeleridir. Bu nesneyi kullanarak ziyaretçiye göndermek istediklerimizi göndeririz.

Server/Sunucu:

ASP, Web Server programını bir nesne olarak ele alır ve onun bize sağladığı araçları ve imkanları kullanmamızı sağlar.

ObjectContext/Nesne Bağlamı:

Microsoft'un Transaction Server (MTS) programının sunduğu hizmetlere erişmemizi sağlar. MTS, ASP sayfaları içinden, uygulama programlarından yararlanmamızı sağlar. ASP uzmanlığınızı ileri düzeylere ulaştırdığınız zaman MTS ve ObjectContext nesnesinden yararlanabilirsiniz.

ASP'nin Dili

ASP, bir teknolojidir. Kendi başına bir yazım kuralı yoktur. ASP tekniğini kullanabilmek için, ASP sayfasının talep edilmesi halinde ziyaretçiye gönderilmeden önce ASP.DLL'ye teslim edilmesi bu teknolojinin kullanılabilmesi için hemen hemen tek şarttır. Bunu, dosya uzantısını .asp yaparak sağlarız.

ASP.DLL ise, dünyada mevcut bütün Script dilleri ile verilecek komutları kabul edebilir. Sadece ASP.DLL'e sayfadaki kodların hangi dilde olduğunu söylemeniz gerekir. Bunu, ASP sayfasının birinci satırında yaparız. Örneğin ASP'ye VBScript dilini kullanmasını belirtmek için bu satırı şöyle yazarız:

```
<% @Language=VBScript %>
```

ASP sayfalarında genellikle VBScript, JavaScript ve JScript kullanılır. Ancak örneğin Perl dilinden türetilen PerlScript, PHP'den türetilen PHPScript de giderek ilgi çeken ASP dilleri arasına giriyor. Bir ASP sayfası içinde farklı Script dilleri kullanılabilir.

VBScript'e Giriş

Visual Basic dilini biliyorsanız, VBScript biliyorsunuz sayılır. VBScript, güçlü bir dildir; ancak Netscape firmasının hiç bir zaman Browser'ında istemci tarafında çalıştırılabilecek diller arasında kabul etmemesi sebebiyle VBScript, Web'in istemci tarafında kendisinden bekleneni yapamadı. MS'un Browser'ı Internet Explorer ise VBScript ile yazacağınız İstemci-Tarafı kodları okuyabilir ve icra edebilir.

Ne var ki ASP kodlarımız hiç bir zaman ziyaretçinin Browser'ının yüzünü göremeyeceği ve sadece Server'da çalışacağı için Server'da VBScript desteği bulunduğu sürece, ASP sayfalarınızı VBScript ile yazabilirsiniz. Bir Server'da ASP desteği varsa, VBScript desteği de var demektir.

VBScript'in hemen hemen bütün komutlarını ve yöntemlerini ASP'de kullanabilirsiniz. Ancak bunun bir kaç kısıtlaması vardır. VB veya VBScript'e ASP dışında aşına iseniz, mesaj kutusu (MsgBox) ve girdi kutusu (InputBox) aracılığı ile programlarınıza kullanıcının bilgi girmesini sağlayabileceğinizi biliyorsunuz demektir. Bu iki komutu ASP içindeki VBScript kodunda kullanamayız. Ayrıca ASP teknolojisi zaten VBScript'in bütün komutlarını ve deyimlerini kullanmanızı da gerekli kılmayacaktır. Mükemmel ASP sayfaları oluşturmak için bile bir avuç VBScript komutu kullanılabilir.

ASP sayfalarımızdaki HTML kodları ile VBScript (veya diğer Script dillerinin) kodlarını birbirinden ayırmamız gerekir. Bu ASP.DLL'ye, HTML'in nerede bittiğini, Script diliyle yazılmış kodun nerede başladığını gösterebilmemiz için gerekli. Bunu sağlamak için Script diliyle yazılmış herşeyi “<%” ve “%>” işaretleri arasına alırız. ASP.DLL bu işaretleri görünce, içindekileri “yazmak” yerine “yapar.” Bir ASP sayfanızda HTML'in klasik “<” ve “>” işaretleri arasındaki unsurlar, ASP.DLL tarafından ziyaretçiye gönderilecek olan sayfaya aynen aktarılır; ancak “<%” ve “%>” arasındaki herşey, başta belirttiğiniz LANGUAGE etiketinde yazılı Script dilinin yorumlayıcısına verilir; yorumlatılarak, gereği yerine getirilir.

“<%” ve “%>” işaretlerine “sınırlayıcı” denir. Sınırlayıcının içinde bir veya daha çok satır kod bulunabilir. Sınırlayıcılar ve içindeki Script, HTML etiketlerinin içinde veya dışında yer alabilir. Sınırlayıcının içindeki kodlarımızı açıklamak için koyacağımız yorum satırlarının başına tek tırnak işareti (') koyarız. İşte bu kuralları uyguladığımız bir ASP sayfası örneği:

```
<% @LANGUAGE=VBscript %>
<html>
<head>
<title>Hoşgeldiniz!</title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
</head>
<body>
<center>
<%
' Yazı tipi boyutunu tutacağımız bir değişken tanımlayalım
Dim fontBoyut
%>
<%
' yazı tipi boyutunu 1'den 7'ye kadar değiştirelim
For fontBoyut = 1 To 7
%>
```

```
<font size = <%=fontBoyut%>>
Hoşgeldiniz!<br>
<% Next %>
</center>
<h3>Bugün <%=WeekdayName(Weekday(Date)) %>, <%= Date %>.
Şu anda Server'da saat: <%= Time %>.<p>
</h3>
</body>
</html>
```

Burada sınırlayıcı arasında tek veya çok satırlı VBScript kodları ile başında tek tırnak olan icra edilmeyen, yorum satırlarını görüyorsunuz. HTML etiketinin içine gömülmüş VBScript kodu ise HTML'in etiketinde yer alıyor: <font size = <%=fontBoyut%>>. Burada karşılaştığımız “<%=” ifadesi, ASP’ye, “Bu değişkenin değerini bul ve tam buraya yaz!” dememizi sağlıyor. Bu ifade daha sonra yakından ele alacağımız Response.Write metodunun kısaltılmış halidir. HTML etiketinin içine yazdığımız VBScript bölümünün kendi sınırlayıcı işaretlerinin yine de kullanıldığına dikkat edin.

VBScript komutları, anahtar kelimeleri ve değişken adlarının büyük harf-küçük harf olması önemli değildir. Yani yukarıdaki ifadelerden birini şu biçimlerden birinde yazabilirdik; kodumuz yine de çalışırdı:

```
For fontBoyut = 1 To 7
FOR FONTBOYUT = 1 TO 7
for fontboyut = 1 to 7
```

Fakat... Bu demek değildir ki, VBScript ile kodlamanın kendi gelenekleri yok! VBScript’çiler genellikle komutların birinci harfini büyük yaparlar: For gibi. Değişken adlarında ise kendinize anlamlı gelen bir biçim tutturabilir ve öyle devam edebilirsiniz.

VBScript'te İşlemciler (Operatörler)

Bilgisayar Operatörü ile program Operatörü arasında fazla bir fark yoktur. Her ikisi de verdiğiniz değerleri ya karşılaştırır bir sonuç bulurlar; ya da bu değerlerle aritmetik işler yapar ve bir sonuç ortaya çıkartırlar. Bilgisayar Operatörü daha sonra bu iş için sizden para ister; program operatörü istemez! VBScript'in operatörleri ve yaptıkları işler şöyle sıralanır:

Operatör	İşlev	Sınıfı
+	Toplama	Aritmetik
-	Çıkartma	
*	Çarpma	
/	Bölme	
^	Üssünü alma	
\	Tamsayı bölme	
<u>Mod</u>	Modüler aritmetik	
=	Bir değişkenin diğerine eşit olduğunu sınar	Karşılaştırma
<>	Bir değişkenin diğerine eşit olmadığını sınar	
>and<	Bir değişkenin diğerinden büyük veya küçük olduğunu sınar (<u>and</u> kelimesi var)	
>= and <=	Bir değişkenin diğerinden büyük veya eşit, veya küçük veya eşit olduğunu sınar (<u>and</u> kelimesi var)	
<u>Is</u>	Bir ifadedeki iki referansın aynı Nesne'ye yapılıp yapılmadığını sınar	
<u>And</u>	Bir veya daha fazla değişkeni test olarak karşılaştırır	Mantıksal
<u>Or</u>	Bir işlemin devamı için hangi koşulun oluşması gerektiğini sınar	
<u>Not</u>	Bir ifadeyi negatif hale getirir	
<u>XoR</u>	Sadece bir koşulun doğru olup olmadığını sınar	
<u>Eqv</u>	İki değişkenin eşitliğini sınar	
<u>Imp</u>	İki ifadede mantıksal implikasyon işlemi yapar.	

VBScript ile yazacağımız ASP sayfalarında bu işlemcileri beklediğimiz sonucu verebilmesi için kullanım sıraları önemlidir. Bunu bir örnekle açıklayalım. 100'den 6'yı çıkarmak ve sonucu 2'ye bölmek istiyorsunuz; sonuç 47 olacaktır. Yani: $100-6/2$. Değil mi? Bu işlemin VBScript'teki sonucu 97 olacaktır. Çünkü, VBScript önce 6'yı 2'ye bölecek ve elde ettiği sonucu 100'den çıkartacaktır. VBScript'te, aritmetik işlemlerin yapılma sırası şöyledir:

<u>Operatör</u>	<u>İşlev</u>	<u>Öncelik</u>
+	Toplama	3
-	Çıkartma	3
*	Çarpma	2
/	Bölme	2
^	Üssünü alma	1

VBScript ile hesap işlemi yaparken, aritmetik işlem sırasını karıştırarak hatalı sonuç almamak için sık sık parantez kullanmak yerinde olur. Yukarıdaki örnek şöyle yazılabilirdi: $(100-6)/2$. Tabii amacınız 100'ü $6/2$ 'ye bölmek ise bu işlemi şöyle yazmalısınız: $100-(6/2)$.

ASP'nin Nesneleri

Masa, nesnedir. Notebook bilgisayarınız da nesnedir! Arkadaşınız da bir nesnedir. Tabii bunu onun yüzüne karşı söylemeseniz, iyi olur. Her nesnenin özellikleri vardır. Masa serttir. Su, içine konulduğu nesnenin biçimi alır.

ASP tekniğinde de amacımız nesnelere özelliklerini kullanarak, ya bu özellikleri belirlemek, ya da değiştirmektir. Nesne Yönelimli Programlama (Object Oriented Programming, OOP) kavramını daha önce çok duymuş olmalısınız. Bir önceki bölümde fonksiyonlarla, Sub'larla, değişkenlerle tanıştık. Diyelim ki öğretmensiniz (ah, nerede o günler!) ve ASP programınızda her öğrencinin notunu veritabanına işleyen, veritabanından notları alarak geçeni-kalanı belirleyen veya öğrencilerle ilgili daha yapılması gereken bir çok işi yapan fonksiyonlarınız ve Sub'larınız var; bunların kullandığı bir çok değişkeniniz var: demek ki sizin ASP programınızda "öğrenci" diye bir nesneniz var. Ve siz bu nesneye yönelimli program yapmışsınız!

Her "program nesnesi" iki unsura sahiptir:

Özellik (Property, Attribute): Bir nesnenin özellikleri, onun değişkenleridir. "Öğrenci" nesnesinin "Öğrencinin Adı," "Notları," "Adresi" gibi değişkenleri, yani özellikleri vardır.

Metod (Method): Bir nesnenin işlemesi, çalışması için, kısaca kendisinden bekleneni yerine getirebilmesi için çalışma yöntemlerine ihtiyacı vardır. Dolayısıyla bir ASP nesnesinin fonksiyonları, onun metodlarıdır.

Fakat ASP’de nesnelere sadece sizin öbekler halinde toplayacağınız fonksiyonlar ve değişkenlerden ibaret değildir. Bir kere, ASP programında kullandığınız Script dilinin getirdiği nesnelere sahiptir. ASP sayfasını Javascript ile yazarsanız başka, VBScript ile yazarsanız başka dil nesnelere sahiptir; ancak her ikisinde de ortak olan “Scripting” nesnelere sahiptir. Bunlara birazdan ayrıntılı ele alacağız. Sonra Web Server’ın size hazır sunduğu nesnelere sahiptir. Bunları daha sonraki bölümde göreceğiz. Ve tabii, Browser’ın bir HTML sayfasının bölümlerini nesne sayarak oluşturduğu nesnelere sahiptir.

Nesneler nasıl oluşmuş olursa olsunlar, daima size bir değer verirler:
Nesne.Özellik = Değer

Bir nesnenin bir özelliğinin değeri, bizim için bir değişken değeri gibi önem taşır:

If Nesne.Özellik > Değer Then ...

Nesnelerin özelliklerinin değerlerini değişkenlere atayabiliriz; ancak bunu yaparken Nesne’nin bir metoduna (fonksiyonu) göndermede bulunmamız ve gerekiyorsa bu fonksiyona kullanması için veri göndermeliyiz (bir fonksiyona kullanması için gönderilen değere argüman/argument denir):

Değişken = Nesne.Metod(argüman1, argüman2...)

Daha sonra bu değişkeni istediğimiz yerde kullanırız. Bu kadar teoriden sonra bir örnek nesne oluşturursak, belki işin içinden daha kolay çıkabiliriz.

Nesneler, diğer yararlarının yanı sıra, birbiri ile ilgili Sub’larımızı, fonksiyonlarımızı ve değişkenlerimizi birarada tutmamızı sağlar. Şimdi VBScript ile bir nesne oluşturabiliriz:

```
<%  
Class Ogrenci  
Public Adi, Soyadi, No  
Function AdiSoyadi  
AdiSoyadi = Adi & “ “ & Soyadi  
End Function  
End Class%>
```


Nesne (object) adını verdik fakat Class (sınıf) oluşturduk! Bu grubun Object..End Object olması gerekmiyor mu? Belki haklısınız; ama bir “sınıf” ancak VBScript tarafından kullanılmaya başlanırsa Nesne olur. Dolayısıyla biz “sınıf” yazarız; VBScript onu Nesne haline getirir. Peki, nesneyi oluşturduk; şimdi bunu nasıl kullanacağız. Çok kolay; önce bu nesneden yeni bir olgu (instance) oluşturacağız; sonra.. Ama önce bunu nasıl yapacağımızı yazalım:

```
<%  
Dim Ogr1  
Set Ogr1 = New Ogresci  
Ogr1.Adi = “Necip”  
Ogr1.Soyadi = “Fazıl”  
Ogr1.No = “181”  
  
Response.Write Ogr1.AdiSoyadi  
%>
```

Her nesne, New (yeni) komutu ile yeni bir değişkene bütün özelliklerini verir. Burada Ogr1 değişkeni, yukarıda oluşturduğumuz Ogresci nesnesinin bütün özelliklerini kazanmış oluyor. Ogresci nesnesinin “.Adi”, “.Soyadi” ve “.No” özellikleri olması gerekir; nitekim Ogr1’e bu özelliklerini burada veriyoruz. Ogresci nesnesinin bir de metodu (fonksiyonu) var; Ogr1 bunu da kazanabilir mi? Tabii; hem kazanır, hem de bunu ziyaretçinin Browser penceresine yazdırabilir. Nesne kavramını kavramak zordur. Fakat yukarıdaki örnekte görüldüğü gibi, ASP sayfalarınızda nesne oluşturarak çalışmaya alıştığınız zaman bunun sağladığı kolaylıktan kolayca vazgeçmeyeceksiniz.

Cookie (Çerez)

Daha önce ne kadar çok olursa olsun tek-tek Web sayfalarından oluşan siteler yaptıysanız bile, Cookie (Çerez) ile ilginiz olmamış olabilir. Ya da sadece Internet ziyaretçisi olarak başkalarının size gönderdiği Cookie’lere sinirlenmekle yetinmiş olabilirsiniz. Fakat şimdi ASP ile Web Programı oluşturmaya hazırlanan kişi olarak Cookie’lerle daha yakından ilgilenmeniz gerekiyor.

Tabii bunun için önce “Cookie neden gereklidir?” sorusunu yanıtlamak gerekir. HTTP ile yapılan iletişim, belirgin olmayan durum bağlantısına dayanır: yani ne istemci sunucunun, ne de sunucu istemcinin o anda hatta (on-line)

olduğunu bilmek zorunda değildir; birbirlerinden istedikleri ve gönderdikleri şeyleri karşı tarafından almaya hazır olduğunu bilmeden gönderirler. Oysa sözgelimi elektronik alışveriş gibi ziyaretçinin bir yerlere birşeyler kaydettiği, geçici değişkenler oluşturduğu durumlarda sitemizde kimin ne yaptığını bilmek zorundayız. Ziyaretçinin bir sayfada yaptığı tercihler diğer sayfalarda ona sunacağımız içeriği etkileyebilir, belirleyebilir. Oysa aynı ziyaretçi bir sayfadan diğerine geçerken Server ile ilişkisini kaybedebilir. Bunun için ziyaretçinin Internet'ten kopması gerekmez; sadece TCP/IP protokolü gereği bizimle bağlantısı kesilebilir. Bunu ziyaretçi farketmeyebilir; ama Server etmek zorundadır. Heryeni ilişkiye yeni bir "application" (uygulama programı) başlatamayız; ziyaretçinin bir önceki sayfada yaptığı tercihlerin devam etmesi gerekir. Bu devamlılığı ziyaretçiyi işaretleyerek yaparız; bu işareti de Cookie sağlar.

ASP tekniğiyle tasarladığımız sitede, ziyaretçilerimize Cookie göndermek zorunluktur diyebiliriz. ASP açısından Cookie'ler bir nesne oluştururlar. Aslında, Cookie'ler ASP için iki ayrı grup Cookie nesnesi oluştururlar: verilenler, ve hakkında bilgi alınan varolan Cookie'ler. Birinci grup Request (talep) nesnelere, ikinci grup ise Response (karşılık) nesnelere içinde ele alınabilir.

KİŞİSEL WEB SAYFAMIZIN (index.asp) KODLARI

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body background="Fire%20Sidebar.jpg" text="#FF6600">
<div align="center">
<div align="justify">
<div align="center">
<pre>                                <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0
,29,0" width="404" height="96" align="absbottom">
    <param name="movie" value="Film2.swf"> <param name="quality" value="high">
    <embed src="Film2.swf" width="404" height="96" align="absbottom" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-
flash"></embed></object><font face="Times New Roman, Times, serif"><strong>
<%
var1 = date
Response.Write Day(var1)
```

```
Response.Write " "  
' Ayları sayalım  
if Month(var1) = 1 then  
response.Write "Ocak"  
end if  
if Month(var1) = 2 then  
response.Write "Şubat"  
end if  
if Month(var1) = 3 then  
response.Write "Mart"  
end if  
if Month(var1) = 4 then  
response.Write "Nisan"  
end if  
if Month(var1) = 5 then  
response.Write "Mayıs"  
end if  
if Month(var1) = 6 then  
response.Write "Haziran"  
end if  
if Month(var1) = 7 then  
response.Write "Temmuz"  
end if  
if Month(var1) = 8 then  
response.Write "Ağustos"  
end if  
if Month(var1) = 9 then  
response.Write "Eylül"  
end if  
if Month(var1) = 10 then  
response.Write "Ekim"  
end if  
if Month(var1) = 11 then  
response.Write "Kasım"  
end if  
if Month(var1) = 12 then  
response.Write "Aralık"  
end if  
'Ayları sayma bitti.  
Response.Write " "  
Response.Write Year(var1)  
Response.Write " "  
'Günleri sayalım  
if WeekDay(var1, 1) = 1 then  
Response.Write "Pazar"  
end if  
if WeekDay(var1, 1) = 2 then
```

```

Response.Write "Pazartesi"
end if
if WeekDay(var1, 1) = 3 then
Response.Write "Sali"
end if
if WeekDay(var1, 1) = 4 then
Response.Write "Çarşamba"
end if
if WeekDay(var1, 1) = 5 then
Response.Write "Persembe"
end if
if WeekDay(var1, 1) = 6 then
Response.Write "Cuma"
end if
if WeekDay(var1, 1) = 7 then
Response.Write "Cumartesi"
end if
'Günleri sayma bitti.

```

```

%> </strong></font></pre>
<hr align="right" width="89%">
<div align="justify">
<div align="center">
<div align="left">
<div align="right">
<pre> 
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0
,0,0" width="147" height="34">
<param name="movie" value="button1.swf"> <param name="quality" value="high">
<param name="base" value=".">
<param name="bgcolor" value="#000000">
<embed src="button1.swf" base="." quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=S
hockwaveFlash" type="application/x-shockwave-flash" width="147" height="34"
bgcolor="#000000"></embed>
</object></pre>
</div>
</div>
</div>
</div>
<div align="right">
<pre> <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0
,0,0" width="147" height="34">
<param name="movie" value="button2.swf"> <param name="quality" value="high">

```

```

    <param name="base" value=".">
    <param name="bgcolor" value="#000000">
    <embed src="button2.swf" width="147" height="34" base="." quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=S
hockwaveFlash" type="application/x-shockwave-flash" bgcolor="#000000"></embed>
    </object></pre>
    <div align="center">
    <div align="right">
    <pre><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0
,0,0" width="147" height="34">
    <param name="BGCOLOR" value="#000000">

    <param name="movie" value="button4.swf"> <param name="quality" value="high">
    <param name="base" value=".">
    <embed src="button4.swf" width="147" height="34" base="." quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=S
hockwaveFlash" type="application/x-shockwave-flash" bgcolor="#000000" ></embed>
    </object></pre>
    <div align="center">
    <div align="right">
    <pre><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0
,0,0" width="147" height="34">
    <param name="movie" value="button7.swf"> <param name="quality" value="high">
    <param name="base" value=".">
    <param name="bgcolor" value="#000000">
    <embed src="button7.swf" base="." quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=S
hockwaveFlash" type="application/x-shockwave-flash" width="147" height="34"
bgcolor="#000000"></embed>
    </object></pre>
    <div align="center">
    <div align="right">
    <pre><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0
,0,0" width="147" height="34">
    <param name="movie" value="button11.swf"> <param name="quality" value="high">
    <param name="base" value=".">
    <param name="bgcolor" value="#000000">
    <embed src="button11.swf" base="." quality="high"
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?P1_Prod_Version=S
hockwaveFlash" type="application/x-shockwave-flash" width="147" height="34"
bgcolor="#000000"></embed>
    </object></pre>
    </div>
    </div>

```

