T.C.
FIRAT ÜNIVERSITESI
MÜHENDISLIK FAKÜLTESI
ELEKTRIK-ELEKTRONIK MÜHENDISLIGI

ASP
(Active Server Pages)

HAZIRLAYANLAR
Gülden AKAY
Ülkü YILDIRIM

DANISMAN
Yrd.Doç. HASAN H. BALIK

LISANS TEZI

ELAZIG - 2001

ÖNSÖZ

Internet günlük yasantimizin vazgeçilmez unsurlarindan biri haline gelmistir. Bilisim ve internet teknolojilerinin bas döndürücü bir sekilde degismesi bilgisayar kullanicilarina çok daha yeni kolayliklar saglamaktadir. Günlük hayatta kullandigimiz otomasyonlari bunlara örnek olarak gösterebiliriz.

Internetin yaygin kullanimi, web sayfalarinin önemini de beraberinde getirmistir. Bu nedenle is yerlerinde, kamu kuruluslarinda, evlerde vs. kullanilan büyük otomasyon programlari web sayfalari üzerine tasinmasi zorunlu hale gelmistir. Bu sekilde dinamik olarak hazirlanan web sayfalari için ASP (Active Server Pages) en çok kullanilan teknolojiler arasindadir.

Microsoft Active Server Pages (ASP) ile resim, ses, film ve benzeri çoklu ortam kavramlarini destekleyen yeni bir platform hazirlayarak, programcilara hem mükemmel bir ortam sunmus hem de tasarim yeteneklerini ön plana çikarabilecekleri araçlari piyasaya beraberinde kazandirmistir.

Lisans tezimizde de ASP dosyalarinin nasil hazirlandigi, çalisma prensibi ve web ortaminda nasil yayimlandigi, ASP'de kullanilabilecek olan script dilleri, ASP'nin nesneleri vs. hakkinda bilgi verilmistir. Buna paralel olarak da telefon rehberi örnegi uygulama olarak yapilmistir.

Gülden AKAY

Ülkü YILDIRIM

TESEKKÜR

Hazirlamis oldugumuz lisans tezimizde faydali olacagina inandigimiz günümüz teknolojisi olan ASP (Active Server Pages) anlatilmis ve hazirlamis oldugumuz uygulama ödevi lisans tezimizle birlikte sunulmustur.

Lisans tezimizin uygulama çalismalarinda ve hazirlanmasinda her türlü yardimlarini ve imkanlarini esirgemeyen, daima bize rehber olan degerli hocamiz Sayin **Yrd.Doç. Dr. Hasan H. BALIK** bey'e tesekkür ederiz. Ayrica çalismalarimiza tavsiye ve yönlendirmeleriyle her zaman yardimci olan Sayin **Resul DAS** hocamiza da tesekkürü bir borç biliriz.

Faydali ve basarilara vesile olmasi dilegimizle....

Gülden AKAY

Ülkü YILDIRIM

IÇINDEKILER

1.GIRIS

ASP (Active Server Pages =Etkin Sunucu Sayfalari) teknigi, Web Sayfalarini canlandiran bir tekniktir. Bu teknik, sil bastan bir bilgisayar programlama dili ögrenmeye gerek olmadan uygulanabilir. Fakat HTML bilgisi gereklidir. Ayrica Web'in nasil çalistigini, Server (Sunucu) ve Client (Istemci) iliskisinin nasil yürüdügü de bilinmelidir.

ASP Microsoft firmasi tarafindan klasik HTML sayfalarina dinamik bir yapi kazandirmak amaciyla ortaya çikarilmis bir teknolojidir.

ASP ile web sayfalari dinamik hale getirilebilir.Bunu yapmak için ASP sunucu(server)tarafinda yapilmis olan kodlar çalistirilir ve istemci(client) tarafi ona özel görüntülenmesi istenen sayfayi görür.

Web programciligi,W3C tarafindan standart hale getirilen HTML ile start almis ve CGI,Java Script,ASP,VRML gibi dillerle devam etmistir.Bu dillerin bazilari sunucu tarafinda bazilari ise istemci tarafinda çalismaktadir.HTML dosyalarinin içerigi baglanilan sunucu tarafindan istemcinin bilgisayarina yollanir ve bulunan dosya web görüntüleyici(Internet Explorer,Netscape gibi)tarafindan istemciye anlamlandirilarak gösterilir.Yani tüm görüntüleme islemini istemci bilgisayari ile yapilir.HTML'den sonra çikan Java Script teknolojisi de buna yakindir.Yani dosyalar baglanilan bilgisayar tarafindan istemciye yollanilir ve istemci onlari bilgisayarinda düzenleyip görüntüler.Java Script(JS) içeren sayfalar diger sayfalara göre daha geç yüklenmektedir,çünkü tüm JS kodlari istemci bilgisayarinda çalistirilir ve bu nedenle performans kaybi olusur.

ASP ise tamamen sunucu tarafinda (server side)çalisan bir programlama dilidir.Tüm kodlar sunucu tarafinda çalistirilir ve istemciye sadece HTML kodlari gönderilir.Böylece performans olarak büyük bir avantaj elde edilmis olur,tabi bu web sunuculugu yapacak bilgisayarin hizli olmasina da baglidir.Örnegin ÖSYM girilen ösym numarasina göre bir program çalistirir ve bu program istemciye sadece girilen numaraya ait sonucu bir HTML dokümani olarak gönderir.

Sunucu tarafli programlamanin diger bir avantaji ise güvenliktir.Örnegin js kodlari kaynagi görüntüle denildiginde görülebilmektedir.Eger güvenlik açisindan kullanicilarin görmemesi istenilen bir program çalistiriliyorsa sunucu tarafli programlama yapilmalidir.Bu tip programlamanin ilk örnegi CGI ile gerçeklesmistir .Perl dili kullanilarak üretilen bu kodlar pek çok kisi tarafindan rahatça kullanilamamistir.Çünkü CGI çalistirmak için hosting yapilan yerde bu destegin verilmesi gerekmektedir.Esas sorun bu kodlar gelistirilirken hiçbir sekilde denenememesidir.Çalistigindan emin olmak için programin sunucuya yüklenmesi gerekmektedir.

Sonuç olarak ASP sunucu tarafli bir programdir.En yaygin olarak kullanilan script dili VBScript'tir.Bunun yani sira Java Script de kullanilabilir,fakat günümüzde JS'yi kullanarak program gelistirmeyi saglayan yeni bir dil olan JSP(Java Server Pages) bulunmaktadir.**Bizde bitirme ödevimizde VBScript kullandik**.

2. ASP Nasil Çalisir?

Bir ASP dosyasi ek özelliklere sahip standart bir HTML dosyasidir. Standart bir HTML dosyasi gibi ASP dosyalari da server tarafindan yorumlanacak HTML taglarina sahiptir. HTML dosyasi içinde bulunan her sey (java appletleri, yanip sönen metinler, istemci tarafli scriptler ya da istemci tarafli ActiveX kontroller gibi) ASP dosyalari içinde de olabilir. ASP dosyalarinin üç önemli özelligi vardir.

1. ASP dosyalari sunucu tarafli script içerir. Sunucu tarafli bu dosyalar dinamik içerikli Web sayfasi yapmayi saglar. Örnegin istediginiz kisinin puanlarini ögrenmek gibi.

2. ASP dosyasinin içinde çok sayida yerlesik nesne vardir. Bu nesneler ASP dosyalarinin script olarak programlanmasini saglar. Ömegin **Request** nesnesi kullanilarak kullanicidan bilgi alinir.

3. ASP dosyalari ayrica birtakim bilesenlerle zenginlestirilebilir. Sunucu tarafli ActiveX bilesenleri veritabanlariyla çalismayi, elektronik posta göndermeyi ya da dosya sistemine erismeyi saglar. Bu anlamda ASP dosyalariyla yapilabilecekleri gelistirmenin siniri yoktur.

ASP dosyalarinin isleyisinin daha iyi anlasilabilmesi için HTML dosyalarinin çalismasiyla karsilastirilabilir.

HTML dosyalarinin isleyisi:

?? Kullanici bir Internet adresinin URL adresini Internet Explorer'in adres çubuguna yazar.
?? Tarayici bu istegini Web sunucusuna gönderir. Bu sirketteki bir Web Server ya da Internet üzerindeki bir Web sunucudur.
?? Web sunucusu HTML dosyayi diskinden alir ve tarayiciya gönderir.
?? Tarayici HTML dosyayi yorumlayarak kullaniciya gösterir.

ASP dosyalarinin isleyisi:

1.Kullanici dosyanin adresini yazar.Örnegin http://localhost/giris.asp adresini girince istemci sunucudan istekte bulunmus olur.

2. Tarayici Internet Information Server sunucusuna ASP dosyasi için bir istek gönderir.

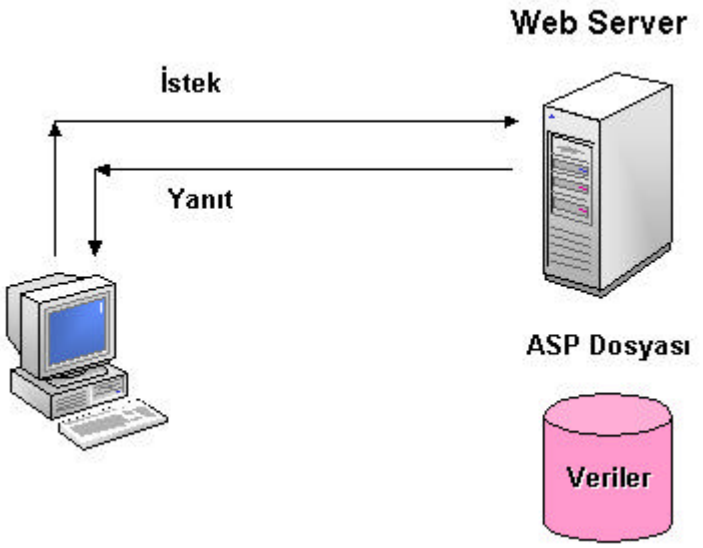
3. Web sunucusu (IIS), istenilen ASP dosyasini disk ya da bellekten alir. Bu asamada dosyanin HTML kismi ve ASP kod kismi ayrilir.

4. Web sunucusu dosyayi özel bir program olan ASP.DLL'e gönderir.

5. ASP dosyasi içinde komutlar islenir. Bu islemin sonucu bir HTML dosyasidir.

6. HTML dosyasi tarayiciya geri gönderilir.

7. HTML dosyasi kullanicinin tarayicisi tarafindan islenir ve sonuç kullaniciya gösterilir.



Sekil-1: ASP'nin temel isleyisi

2.1. Kisisel Web Server Kurulumu

Bilgisayar Windows 95, 98, NT4 WorkStation veya NT4 Server ile çalisiyorsa, sisteme bir Web Server programi kurulmak zorundadir. Windows 2000 Professional veya Windows 2000 Server ise Kisisel(Personal) Web Server(PWS) programi kendiliginden kurulur. Windows 98'e bir kisisel Web Server kurmadan önce bilgisayara bir kimlik verilmelidir. Bilgisayarim/Denetim Masasi/Ag'i tiklayarak açilan diyalog kutusunda ikinci sekme olan Tanimlama seçilerek ve "Bilgisayar adi" kutusuna bir isim yazilir. Bilgisayarin ag ortaminda olmasi gerekli degildir.

Windows 98'e PWS kurmak için iki yol izlenebilir. Windows 98 CD-ROM'unda Add-ons klasöründeki PWS dizininde Kur.exe tiklanir veya Windows NT Option Pack CD-ROM'unda Default.htm açildiginda bilgisayar Windows 98 ile çalistigini algilayacak olan program Personal (kisisel) Web Server kurmayi önerir. Kisisel Web Server'i kurarken her iki durumda da ikinci diyalog kutusunda Minimum/En az veya Typical/Tipik seçenegi degil, Custom/Özel seçilip ve yazilan diyalog kutusunda Microsoft Data Access Components (MS Veri Erisim Bilesenleri) satirina isaret konur,Alt Bilesenleri Göster dügmesi tiklanir. Açilacak seçme kutusunda ise ADO Documentation satirina isaret konulmalidir.Bu belgelerle veri-yönlendirmeli Web Uygulamasi yaparken yararlanilir.

Kisisel Web Server kurulduktan sonra bilgisayarin yeniden baslatilmasi gerekir.

Windows NT4.0 Workstation veya Server'a IIS4.0 kurmak için Option Pack CD-ROM'undaki default.htm'i çalistirip ve açilacak Browser penceresinde IIS'i kurma seçenegini tiklamak yeterlidir. Burada da ADO Documentation'i sabit diske aktarabilmek için gerekli seçenek isaretlenmelidir

Windows 98'e Kisisel Web Server kuruldugunda Masaüstü'nde Yayimla (Publish) adli bir simge belirir. NT sistemlerinde ise Baslat menüsünden Programlar bölümüne IIS Manager satiri eklenir. Bu yollardan biriyle PWS veya IIS'i çalistirilabilir.

Kisisel Web Server'da Personel Web Server Manager (Yönetici) kutusu açildiginda soldaki araç çubugunda Yönetici'nin çesitli bölümlerine gitmek için gereken gezinme simgeleri görülür. Açilan ana pencerede iki unsura dikkat edilmelidir.

1. Kisisel Web Server'in adi: Bilgisayarin adi buraya Server adi olarak yazilmalidir. Internet'e koyulmadan önce sinanacak ASP sayfalari çagirilirken, Browser'in adres kutusuna burada görülen isim yazilir.

2. Kisisel Web Server'in bilgisayarda sabit diskteki gerçek adresi: Bu, Kisisel Web Server'in kök (root) dizinidir. Genellikle C:\inetpub\wwwroot klasörüdür. Kisisel Web sitesi yapilirken, sitenin gerektirdigi bütün dizinler ve dosyalar burada görülen dizinin içinde olmalidir. Yapilan ASP dosyalari bu dizinin içine konulur.

Bu iki unsur dikkate alindiktan sonra, soldaki araç çubugunda Gelismis simgesi tiklanir; ortadaki pencerede sanal dizinler görülür. Bu asamada Home seçilip ,sagdaki "Özellikleri düzenle" dügmesi tiklanir.

Bu islemler IIS'te degisik araçlar ve diyalog kutulariyla, fakat temel ilkeler itibariyle ayni sekilde yapilabilir. NT4 sistemlerine IIS'i kurmadan önce, Service Pack 3'ü uygulanmali; Internet Explorer 5 kurulmali, varsa Service Pack 4, 5 veya 6'yi en son uygulanmalidir.

ASP sayfalari sinanirken bilgisayarda Microsoft Internet Explorer programi kurulu bulunmasi sart degildir. ASP sayfalari Netscape ile de sinanabilir.

Örnek:

Kisisel Web Server programinin çalisip çalismadigini sinamak için bir ASP sayfasi hazirlanacak olursa,;

ASP sayfasi da HTML gibi düz yazi dosyasidir; dolayisiyla istenilen bir düz yazi programi ile ASP yazilabilir. . Eger kelime-islemci kullanilirsa dosya ASCII veya ANSI biçiminde kaydedilmelidir.

```
<HTML>
<HEAD>
<TITLE>ASP ILE ILK 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>ASP Ugrasan Siz Sevgili Arkadaslarimiza
 Çalismalarinizda Basarilar Diler,Kaynak Olarak Lisans Tezimizden faydalandiginiz Için
Tesekkür Ederiz.  </H1>
```

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

Bu dosya bitirme.asp adiyla kaydedilip ve bilgisayarda PWS'de veya ISS'te sinanacagi zaman, Browser'in URL hanesine, kisisel Server'in adiyla birlikte dosyanin adi yazilir. Bu ASP programiyla ekranda, açiklama satirinda yazmis oldugumuz metin yazisi görülür.

2.1.1. Global.asa dosyasi:

ASP.DLL'e bir .asp dosyasi geldiginde global.asp'nin çalisip çalismadigina bakar. global.asa tipki diger ASP dosyalari gibi bir düz yazi dosyasidir ve ASP programlarinin çalisma kosullarini düzenleyen kurallari içerir.ASP'ye "program" özelligi kazandiran HTML kodlari degil Script dili ile yazilmis kodlardir. ASP.DLL, önce gelen .asp dosyasinda hangi Script dilinin kullanildigina bakar ve bunun için gerekli ortami olusturur; yani bu Script dilini yorumlayacak programi çalistirir; bu program Script'i yorumlar ve icra edilecek komutlari icra eder; ASP.DLL, icra edilen komutlar, isletim sisteminin yardimini istiyorsa (örnegin bir veritabanindan veri çekmek gibi, veya dosya sistemine bir dosya açtirmak, yazdirmak, sildirmek gibi) bu yardimin edinilmesini de saglar. Bütün bu islerin sonunda yazilan HTML kodlarina ek yapmak (örnegin bir tablonun içini, çekilen verilerle doldurmak veya dosya sisteminden edinilen bir dosyanin içerigini sayfaya aktarmak gibi) gerekirse bu ekler ASP.DLL tarafindan yapilir.

```
<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 sayfasinda <SCRIPT>..</SCRIPT> etiketleri arasina yerlestirilen bu kodun çalismasi için Server'in hiçbir sey yapmasi gerekmez; kodu Browser çalistirir ve günün tarihini bildirir. Server tarafinda çalisan Script içeren bir örnek deneme.asp adiyla asagida verilmistir .(Script etiketinden sonra nokta olduguna dikkat edilmelidir):

```
<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 sayfa VBScript ile yazilmistir Bu HTML sayfasi, Netscape'de görüntülenir, çünkü Script'i Netscape degil, Server çalistirir. Bunu saglayan <SCRIPT> etiketinin içindeki "RUNAT" özelligidir. ("Run," çalistir; "at" ise "içinde, üzerinde" anlamina gelir. "RUNAT" "...de çalistir" gibi bir anlama sahiptir.) Burada RUNAT'in karsisina yazilan SERVER ifadesi ile, Script'in Browser'a gönderilmeden önce Server'da çalistirilmasi saglanir.

Netscape bu sayfayi görüntüler. ASP sayfalarina, Browser, Server ve ASP.DLL tarafindan çalistirilacak kodlar yerlestirilirken, sirasina ve hangi sirada icra edilmeleri gerektigine dikkat edilmelidir.

Server ile ASP.DLL'in iliskisi sadece Script dilini çalistirmaktan ibaret degildir. ASP, istemciden gelen HTTP Istemi (Request) ve HTTP'ye giden Karsilik (Response) unsurlari, ActiveX Data Objects (ADO, AcvtiveX Veri Nesneleri) araciligiyla, isletim sisteminin sunacagi veritabanina erisim imkanini ve isletim sisteminin sundugu dosya yönetimi imkanlarini saglar.Bu " imkanlar" ASP de "nesne" (Object) sayilirlar .

2.2. ASP'nin Unsurlari

Gerçekte ASP'nin Nesneleri ile bir seyler yapilir.ASP kodlari bu nesnelere yöneliktir, onlari kullanma ve onlardan bir sonuç alma veya onlara bir sonuç aktarma amacina yöneliktir.ASP'nin Nesneleri alti grupta toplar.

Application/Uygulama: Bir ASP sitesi, gerçekte bir Uygulama Programi olarak görülür. Bu, HTML/CGI gelenegine asina tasarimci için yeni bir kavramdir. Ziyaretçi bir ASP sayfasindan girerek, bir sitede surfing'e basladiginda programi isleten bir bilgisayar kullanicisi olur. Böylece, site, her ziyaretçinin karsisina çiktiginda "bir program çalismis" gibi sayilir.
Session/Oturum: Ziyaretçi siteye geldiginde, hangi sayfayi talep ederse etsin, bu baglanti ASP açisindan bir oturum sayilir. Her oturumun belirli bir süre devam eden özellikleri, degiskenleri ve degerleri vardir. Site tasariminda oturum özelliklerinden genis ölçüde yararlanilir.
Request/Talep: Browser'dan Server'a ulasan bütün bilgiler, Request (Talep) nesnesinin ögeleridir. Bu nesneyi kullanarak, istemciden gelen her türlü HTTP bilgisini kullanilir.
Response/Karsilik: Server'dan ziyaretçinin bilgisayarina gönderilen bütün bilgiler, çerezler (cookie) ve basliklar (Header) Response (Karsilik) nesnesinin ögeleridir. Bu nesneyi kullanarak ziyaretçiye göndermek istenilenler gönderilir.
Server/Sunucu: ASP, Web Server programini bir nesne olarak ele alir ve onun kullanicilara sagladigi araçlari ve imkanlari kullanmalarini saglar.
ObjectContext/Nesne Baglami: Microsoft'un Transaction Server (MTS) programinin sundugu hizmetlere erisilmesini saglar. MTS, ASP sayfalari içinden, uygulama programlarindan yararlanilmasini saglar. MTS ve Object/Context nesnesinden ASP uzmanligini ileri düzeylere ulastirilanlar yararlanabilirler.

2.2.1. ODBC (OPEN DATABASE CONNECTIVITY)'NIN TEST EDILMESI

ASP sayfasi olusturabilmek için bilgisayarda, ODBC (Açik Veritabani Baglantisi) olmasi gerekir.
Windows 98, 95 (OSR2) veya NT4.0 isletim sisteminde Denetim Masasi'nda ODBC, ODBC32 veya "ODBC Veri Kaynaklari (32 Bit)" adli simge açilir, Sistem DSN sekmesi ve açilan pencerede Ekle dügmesi tiklanir. Buradaki Access, dBase, Excel, FoxPro, Paradox sürücüleri 4.00.3711.08 veya daha büyük degilse, Microsoft'un sitesinden (http://www.microsoft.com/data/download.htm) Microsoft Data Access Components (sürüm 2.1.1.3711.11 GA, 6.2 MB) güncelleme dosyasi indirilmeli ve sistem

güncellestirilmelidir. Windows 2000 kurulu sistemlerde bunu yapmaya gerek yoktur. Böylece sistem veri-yönlendirmeli Web uygulamalari için hazir hale getirilmis olur.

2.3. ASP'nin Dili

ASP, bir teknolojidir. Kendi basina bir yazim kurali yoktur. ASP tekniginin kullanilabilmesi için, ziyaretçiye gönderilmeden önce ASP.DLL'ye teslim edilmesi bu teknolojinin kullanilabilmesi için hemen hemen tek sarttir. Bunu da dosya uzantisini .asp yaparak saglariz.

ASP.DLL dünyada mevcut bütün Script dilleri ile verilecek komutlari kabul edebilir.Ancak ASP.DLL'e sayfadaki kodlarin hangi dilde oldugunun söylenmesi gerekir. Bu da ASP sayfasinin birinci satirinda yapilir. Örnegin ASP'de VBScript dilinin kullanildigini belirtmek için su satir eklenir:

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

ASP sayfalarinda genellikle VBScript, JavaScript ve JScript kullanilir. Ancak örnegin Perl dilinden türetilen PerlScript, PHP'den türetilen PHPScript de giderek ilgi çeken ASP dilleri arasina girmektedir.

3. VBSCRIPT'E GIRIS

VBScript, güçlü bir dildir; ancak Netscape Browser'inda istemci tarafinda çalistirilabilecek diller arasinda kabul edilmedigi için Web'in istemci tarafinda bekleneni yapamaz. MS'un Browser'i Internet Explorer ise VBScript ile yazilan Istemci-Tarafi kodlari okuyup,icra edebilir.

Bir Server'da ASP destegi varsa, VBScript destegi de var demektir.
VBScript'in hemen hemen bütün komutlari ve yöntemleri ASP'de kullanilabilir. Ancak bunun bir kaç kisitlamasi vardir. VB veya VBScript'in,ASP disinda, mesaj kutusu (MsgBox) ve girdi kutusu (InputBox) komutlari ile programlara kullanicinin bilgi girmesi saglanabilir. Bu iki komut ASP içindeki VBScript kodunda kullanilamaz. ASP teknolojisi zaten VBScript'in bütün komutlarinin ve deyimlerinin kullanilmasini gerekli kilmaz. Mükemmel ASP sayfalari olusturmak için bile az sayida VBScript komutuna ihtiyaç duyulur.

ASP sayfalarindaki HTML kodlari ile VBScript (veya diger Script dillerinin) kodlarinin birbirine karistirilmamasi gerekir. Bu ASP.DLL'ye, HTML'in nerede bittigini, Script diliyle yazilmis kodun nerede basladigini gösterilebilmesi için gereklidir. Bunu saglamak için Script diliyle yazilmis her sey "<%" ve "%>" isaretleri arasina alinir. ASP.DLL bu isaretleri gördügünde, içindekileri "yazmak" yerine "yapar." Bir ASP sayfasinda HTML'in klasik "<" ve ">" isaretleri arasindaki unsurlar, ASP.DLL tarafindan ziyaretçiye gönderilecek olan sayfaya aynen aktarilir; ancak "<%" ve "%>" arasindaki hersey, basta belirtilen LANGUAGE etiketinde yazili Script dilinin yorumlayicisina verilir; yorumlatilarak, geregi yerine getirilir.

"<%" ve "%>" isaretlerine "sinirlayici" denir. Sinirlayicinin içinde bir veya daha çok satir kod bulunabilir. Sinirlayicilar ve içindeki Script, HTML etiketlerinin içinde veya

disinda yer alabilir. Sinirlayicinin içindeki kodlari açiklamak için konulacak yorum satirlarinin basina tek tirnak isareti (') konulur. Bu kurallarin uygulandigi bir ASP sayfasi örnegi asagidadir:

```
<% @LANGUAGE=VBscript %>
<html>
<head>
<title>Hosgeldiniz!</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>

<%
' Yazi tipi boyutunu tutacagimiz bir degisken tanimlayalim
Dim fontBoyut
%>

<%
' yazi tipi boyutunu 1'den 7'ye kadar degistirelim
For fontBoyut = 1 To 7
%>
<font size =<%=fontBoyut%>>
Hosgeldiniz!<br>
<% Next %>
</center>
<h3>Bugün <% =WeekdayName(Weekday(Date)) %>, <% = Date %>.
Su anda Server'da saat: <% = Time %>.<p>
</h3>
</body>
</html>
```

Burada görüldügü gibi sinirlayici arasinda tek veya çok satirli VBScript kodlari ile basinda tek tirnak olan icra edilmeyen, yorum satirlari vardir. HTML etiketinin içine gömülmüs VBScript kodu ise HTML'in <FONT>etiketinde yer almistir: <font size = <%=fontBoyut%>>. Burada karsilasilan "<%=" ifadesi ile ASP'nin degiskenin degerini bulup yazmasini saglar. Bu ifade Response.Write metodunun kisaltilmis halidir. HTML etiketinin içine yazilan VBScript bölümünün kendi sinirlayici isaretleri kullanilmistir.

3.1. VBScript'te Bazi Yazim Kurallari

VBscript komutlari,anahtar kelimeleri ve degisken adlarinin büyük harf-küçük harf olmasi önemli degildir.Yani örnekteki ifadeler su sekilde yazildiginda da kod çalisir.

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

VBScriptte genellikle komutlarin birinci harfi büyük yazilir:(For gibi). Degisken adlarinda ise anlamli gelen bir biçim tutturabilir ve öyle devam edilebilir.
Eger bir kod satiri çok uzun gelir ve daha sonra anlasilmasi imkansiz derecede uzarsa, bu satir alt çizgi (_) ile asagi satirda devam ettirilebilir. Örnek:

```
<%
If degisken1 > 1 And _
                degisken1 < 10 Then
%>
```

3.2. Degiskenler

Programcilikta islemler degiskenlerle yapilir. Degisken bir kap gibi düsünülebilir. örnegin "Gün," degiskenin adi ise bu degiskenin degeri Pazar, Pazartesi, Sali, vs.olabilir. Her degisken, türüne göre, ya bir ya da daha fazla deger tutar. Adindan da anlasilacagi gibi degiskenin degeri degisir.Bu degisigi programci veya programin kendisi yapabilir.

VBScript'te, bir çok baska bilgisayar programlama dilinden farkli olarak degiskenlerin tanimlanmasi veya "beyan edilmesi," "boyutlandirilmasi" gerekmez. Belirtilmemis, önceden tanimlanmamis bir degiskene deger atamaya kalkilirsa, VBScript bunu kabul eder. Fakat bu kötü bir programciliktir. Iyi programcilik degiskenlerin önceden beyan edilmesini gerektirir. Bu DIM(dimension,boyutlandir) komutuyla yapilir.Bu komut, bilgisayarin degisken yeri olarak bir bellek alaninin boyutunu belirtmesini saglar. Örnekler:

```
<%
DIM Gun, Ay, Ogrenci, Not
Gun = "Pazartesi"
```

```
Ay = “Ocak”
Ogrenci = “Necip”
Not = 5
%>
```

Burada Gun, Ay, Ogrenci, Not adiyla dört degisken olusturuldu ve bunlara sirasiyla “Pazartesi,” “Ocak,” “Necip” ve “5” degerleri atandi.Degisken isimleri, mutlaka harfle baslamalidir; içinde noktalama isaretleri bulunamaz ve uzunlugu 255 karakteri geçemez.

3.3. ASP Programcisinin Yapmamasi Gerekenler

Bir sayfada kullanilan degiskenin, daha sonraki sayfada kullanilabilmesi için, bu degiskenin degerinin yeni sayfada degismemesi gerekir. ASP programi yazilirken, bazen gelisi-güzel degiskenlere deger atanabilir.Bu degisken adi daha önce kullanilmissa ve içinde daha sonra kullanilacak bir deger varsa,bu deger degistirilmis olur. VBScript, savurgan ve daginik programciligi önlemek için OPTION EXPLICIT imkanini verir. Bir ASP sayfasinin birinci satiri olarak

```
<% OPTION EXPLICIT %>
```

yazilirsa ,VBScript DIM komutuyla belirlenmemis degisken kullanilmasina izin vermez; kullanilirsa hata verir ve durur.

VBScript yanlis kelimeyi yeni bir degisken sayar. OPTION EXPLICIT kullanilarak yanlis yazilan degiskeni yeni degisken sayar ve önceden tanimlanmamis degisken kullanildigini düsünür ve durur.

Degiskenler asagidaki gibi tanimlanirsa:

```
<%
DIM Gunler(31), Aylar(12), Ogrenciler(210), Notlar(10)
>
```

Degiskenler asagidaki gibi tanimlanirsa:

```
<%
DIM Gunler(31), Aylar(12), Ogrenciler(210), Notlar(10)
>
```

kaplar birden fazla deger tutabilirler. Yani:

```
<%
DIM Gunler(7), Aylar(12), Ogrenciler(21), Notlar(10)
Gunler(1) = “Pazartesi”
```

```
Aylar(3) = “Mart”
Ogrenciler(12) = “Necip”
Notlar(5) = 5
>
```

Böyle, birden fazla deger tutabilen degiskenlere Dizi Degisken veya Array denir.

3.4.Array Fonksiyonu

VBScript’in kullanilmaya hazir bir çok fonksiyonu vardir; bunlardan biri olan Array ile, kolayca dizi degisken olusturulabilir.

Gunler(7) dizi-degiskenini gün adlari ile doldurulursa;

```
<%
Dim Gunler = Array (“Pazartesi” , “Sali” , “Çarsamba” , “Persembe” , “Cuma” , “Cumartesi”, “Pazar”)
%>
```

hem dizi-degiskeni olusturulabilir; hem de degerleri atanabilir.

Bu suretle olusturulan dizi degiskenin üyelerine daha sonra sira numaralari ile atifta bulunululabilir.

Örnegin:

<%=Gunler(6)%>
scripti Pazar’i verir. Çünkü Gunler dizi-degiskeni Gunler(0)’dan baslar.

```
<%
....
If Ogrenciler(OgrenciNo) = “Mehmet”
...
%>
```

Örnegin ÖgrenciNo degiskeninin degeri 12 ise yukardaki döngü ile aranilan ögrencinin Mehmet olup olmadigi sinanabilir.

Baska programlama dillerinde bir degiskenin degeri harf ve rakamlardan olusuyorsa, yani matematik islem yapmaya elverisli degilse bunlara String (Alfanümerik, karakter degerler) denir. Programlama dillerinde bir de matematik islem yapmaya elverisli degisken türü vardir: Sayi (Number). VBScript, bir degiskene alfanümerik (karakter, metin) olarak atanan deger çift tirnak içine alinmalidir. Sözgelimi Ogrenci(12) degiskeni

için Mehmet degerini atamak istendiginde, Mehmet kelimesini çift tirnak içine alinmalidir. Sayi olarak kullanilan degerler ise tirnak içine alinmaz.Çift tirnak içinde verilen bir degeri matematik islemde kullanmaya kalktiginizda karsiniza çikabilir. Rakam olmayan bir karakter-dizisi bir degiskene tirnaksiz olarak atanirsa VBScript "tanimsiz degisken" seklinde hata mesaji vererek, durur.

VBScript'in bu eksikliginin giderilmesi için degisken adlarinin önüne karakter-dizileri için "str" harfleri yazilir. StrAy, strOgrenciler, gibi.

VBScript'in kullandigi tek tür degiskene variant denir. Variant, karakter-dizini (String) de olabilir,sayi(Number)da .

3.5. Sabit Degerler

VBScript'te bir kere verildiginde degeri hiç degismeyen unsurlar vardir. Sabit deger, bütün ASP sayfasi boyunca (hatta istenirse, bütün site, yani Uygulama boyunca) degismeden kalir. Bu degerler Const (constant, sabit kelimesinden türetilme) komutuyla belirtilir.

```
Const DolarDeger = 560780
Const SirketinAdi = "Web Tasarim ve Site Onarim A.S."
Const Slogan = "Ne Mutlu Türküm Diyene"
```

3.6 VBScript'te Islemciler (Operatörler)

Operatörler verilen degerleri ya karsilastirip bir sonuç bulurlar; ya da bu degerlerle aritmetik isler yapip bir sonuç ortaya çikartirlar. VBScript'in operatörleri ve yaptiklari isler:

Operatör	Islev	Sinifi
+	Toplama	Aritmetik
-	Çikartma	
*	Çarpma	
/	Bölme	
^	Üssünü alma	
\	Tamsayi bölme	
Mod	Modüler aritmetik	
=	Bir degiskenin digerine esit oldugunu sinar	Karsilastirma
<&gt	Bir degiskenin digerine esit olmadigini sinar	
&gtand<	Bir degiskenin digerinden büyük veya küçük oldugunu sinar (and kelimesi var)	
&gt= and <=	Bir degiskenin digerinden büyük veya esit, veya küçük veya esit oldugunu sinar (and kelimesi var)	
Is	Bir ifadedeki iki referansin ayni Nesne'ye yapilip yapilmadigini sinar	

And	Bir veya daha fazla degiskeni test olarak karsilastirir	Mantiksal
Or	Bir islemin devami için hangi kosulun olusmasi gerektigini sinar	
Not	Bir ifadeyi negatif hale getirir	
XoR	Sadece bir kosulun dogru olup olmadigini sinar	
Eqv	Iki degiskenin esitligini sinar	
Imp	Iki ifadede mantiksal implikasyon islemi yapar.	

VBScript ile yazilan ASP sayfalarinda, islemcinin beklenen sonucu verebilmesi için kullanim siralari önemlidir. Bir örnekle açiklanacak olursa;100'den 6'yi çikarip ve sonun 2'ye bölünmesi durumunda,sonuç 47 olmalidir. (Yani 100-6/2) Bu islemin VBScript'teki sonucu ise 97 olur. Çünkü, VBScript önce 6'yi 2'ye bölüp elde edilen sonucu 100'den çikartir.

VBScript'te aritmetik islemlerin yapilma sirasi söyledir:

Operatör	Islev	Öncelik
+	Toplama	3
-	Çikartma	3
*	Çarpma	2
/	Bölme	2
^	Üssünü alma	1

VBScript ile hesap islemi yapilirken, aritmetik islem sirasini karistirarak hatali sonuç almamak için sik sik parantez kullanmak gerekir. Yukarda verilen örnekte yapismasi istenen islemin sonucunun dogru olarak elde edilebilmesi için (100-6)/2 seklinde yazilmasi gerekir

3.7. VBScript'de Program Kontrolü

Bir bilgisayar programinin varlik sebebi, ister Script diliyle, isterse gerçek bir programlama diliyle yazilsin,çesitli durumlari degerlendirerek, belirli durumlarda belirli kararlar verebilmektir. Bu, programin kontrol ögeleri kullanilarak yapilir. Programlar, bu ögeler sayesinde karsilastirma yaparlar; belirli durumlarin olusup olusmadigini sinarlar; veya belirli bir durumun olusmasina veya sona ermesine bagli olarak bir is yaparlar veya yapmazlar. Bu sinamalarla (kosullu ifadelerle) veya döngülerle saglanir. Kimi zaman da, programa (programin mantigi çerçevesinde) istenildigi anda yapmakta oldugu isi durdurup, baska bir is yapmasi istenebilir. Bunlara da Süreçler (veya Prosedürler) denir.

3.8. Mantiksal Sinamalar

VBScript'te programin karar verme mekanizmasi "eger ... ise... yap!" seklinde özetlenebilir. VBScript bu islem iki ayri ifadeyle yapilabilir.

3.8.1. If.. Else

VBScript'in verilen bir durumun bulunup bulunmadigini sinamasini saglar. Genel yazim kurali su sekildedir:

If sart Then

```
                [sart dogru ise yapilacak isler]
Else
                [sart dogru degilse yapilacak isler]
End If
```

Örnek:

Asagidaki örnek, saat 12'den önce ise sayfaya "Günaydin" ; saat 12'den sonra ise "Tünaydin" yazdirir.

Fakat saat 18'den sonra sayfaya "Iyi aksamlar!" yazdirmak için if..elseden faydalanilabilir. If döngüsü kendi içinde sinirsiz Elseif (ikinci sartli döngü) imkani vererek bize bu imkani saglar. Her Else if yeni bir If gibi düsünülebilir.Bunu gerçeklestirecek kod örnegi (welcome1.asp) asagidaki gibidir:

```
<HTML>
<HEAD>
<TITLE>ASP ILE SAATE GORE SELAM</TITLE>
<META http-equiv="content-type" content="text/html; charset=ISO-8859-9">
&lt;META http-equiv="Content-Type" content="text/html; charset=windows-1254"&gt
</HEAD>
<BODY>
<H2>
<CENTER>
<%
If Hour(Now) <12 Then
                Response.Write "Günaydin! "
ElseIf Hour(Now) >= 18 Then
                Response.Write "Iyi aksamlar! "
Else
                Response.Write "Tünaydin! "
End If
Response.Write "<BR>"
Response.Write "Site Onarim Sitesine Hosgeldiniz"
%>
</CENTER>
</H2>
</BODY>
```

```
</HTML>
```

Bu programi çalistirildiginda , çalistirilan saate göre sayfadaki selam degisir.Now() fonksiyonu o anda saati ve tarihi bildiren fonksiyondur. Bu fonksiyondan dönen degerle,o andaki saat ögrenilip 12 ile karsilastirilir. Fonksiyondan dönen deger, eger 12'den küçükse, program Response (Karsilik) Nesnesi'nin .Write Metodu'nu kullanarak (Nesneler ve Metodlar meselesi üzerinde de durmayin!) ziyaretçinin Browser penceresine "Günaydin" yazdirir.

Dönen degerler:Fonksiyonlar, kendilerini göreve çagiran VBScript komutlarina ve islemlerine bir deger sunarak karsilik verirler. Buna fonksiyondan dönen deger denir.Örnegin Now() fonksiyonunu göreve çagirildiginda,bu fonksiyon derhal isletim sisteminden saati ve tarihi ögrenerek kendisini göreve çagiran isleme bildirir.

Eger bu ilk sinamanin sonucu dogru degilse, VBScript If satirindan sonraki birinci deyimi atlar ve ikinci deyimi icra ettirir. Yani eger saat 12'den küçük degilse, ElseIf satiri icra ettirilir. ElseIf de If de oldugu gibi isleyerek bu kez saatin 18'e esit veya büyük olup olmadigi sinar. Eger saat 18'e esit veya büyükse,ilk satir icra ettirilir ve ziyaretçinin Browser penceresine "Iyi aksamlar!" yazdirir. Ancak ,saat 18'den küçük ise, ElseIf'in ikinci satiri icra edilir. Bu satirda i Else bulunur. Else, If ve ElseIf gibi çalismaz; ne olursa olsun, kendisinden sonra gelen deyimi yerine getirir. Yani saat 12'den küçük degilse, 18'den küçük veya 18'e esit degilse, yani 12 ile 17 arasinda ise, ekrana "Tünaydin" yazdirir.

3.8.2. Select Case

VBScript'in bir diger duruma bakarak karar verme ifadesi, Select Case (Durum Seç) yapisidir. Bu kontrol ögesinin çalismasi söyle özetlenebilir:

Durum Seç (Durumlarin listesi veya durumlari belirten bir degisken)
    Durum 1 : Yapilacak isler
    Durum 2: Yapilacak isler
    Durum 3: Yapilacak isler
    Durum n: Yapilacak isler
Seçmeyi Bitir.

VBScript, verilen durum listesine veya içinde çesitli degerler bulunan degiskene bakarak, bu degiskenin her bir degerini bir "durum" sayarak verilen durumlardan hangisini sagliyorsa, ona ait komut dizisini icra ettirir. Yukaridaki örnegi b yapiyi kullanarak yazarsak: (welcome2.asp)

```
<HTML>
<HEAD>
<TITLE>ASP ILE SAATE GÖRE SELAM</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>
<H2>
<CENTER>
<%
Select Case Hour(Now)
                Case 0,1,2,3,4,5,6,7,8,9,10,11
                Response.Write "Günaydin!"
                Case 12,13,14,15,16,17
                Response.Write "Tünaydin"
                Case Else
                Response.Write "Iyi Aksamlar!"
End Select
Response.Write "<BR>"
Response.Write "Site Onarim Sitesine Hosgeldiniz"
%>
</CENTER>
</H2>
</BODY>
</HTML>
```

Select Case komutuna, içindeki degerleri "durum" sayacagi dizi veya degisken olarak VBScript'in kullanilmaya hazir fonksiyonlarindan Hour(Now)'i verilir. Bu fonksiyondan, 0 ile 24 arasinda bir deger döner. Bu deger Select Case için bir durum demektir. Select Case, bu deger ile altta siralanan Case'leri karsilastirarak ve elindeki deger hangi Case'i tutuyorsa ona ait komutlari icra ettirir.Sonuncu Case'e dikkat edilmelidir. Burada Case olarak Else (baska) verilmistir. Bu sayede 17'den 23'e kadar olan saatler siralanmamis olur. 0'dan 11'e kadar olan saatlerle,12'den 17'ye kadar olan saatler siralandigindan baska hangi saat olursa olsun, ziyaretçiye "Iyi aksamlar!" söylenebilir.

3.8.3. Döngüler

Döngü ,sinamadan sonra, bir programin akisini kontrol için kullanilan en önemli unsur sayilir. Döngü (Loop), programin bir isi bitirmesini saglar. Ancak bu is sonsuza kadar sürecek olursa, buna Endless Loop (Sonsuz Döngü) denir. VBScript'te kullanilabilecek döngü yöntemleri sunlardir

3.8.4. For..Next döngüsü

Programin bir isi bazi kereler yapmasi istendiginde, bu bir sayaç degiskeniyle birlikte, For döngüsü kullanilarak yapilabilir.

For sayaç = baslangiç To son Step adim
yapilacak isler
Next

Burada, “sayaç” yerine istenilen bir degisken adi, “baslangiç” yerine sayacin baslamasi istenen sayi, “son” yerine sayacin durmasi istenen sayi, ve “adim” yerine de sayacin kaçar-kaçar artmasinin istendigi yazilir. En sondaki Next deyimi döngünün bir sonraki adima geçmesini saglar. Bu adimda sayaç, Step kelimesi varsa, karsisindaki deger kadar arttirilir ve yapilacak isler yeniden yapilir. Örnek: (days.asp)

```
<HTML>
<HEAD>
<TITLE>ASP ILE GÜNLERI SAYMA</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>
<H2>
<CENTER>
<%
Dim Gunler
Gunler = Array("Pazartesi", "Sali", "Çarsamba", "Persembe", "Cuma", "Cumartesi", "Pazar")
For sayac = 0 to 6
        Response.Write Gunler(sayac)
        Response.Write "<BR>"
Next
%>
</CENTER>
</H2>
</BODY>
</HTML>
```

Bu ASP kodunda, Gunler adiyla bir dizi-degisken olusturulmus ve bu degiskenin yedi hanesine, günlerin adlari atanmistir. Sonra, sayac adli sayaci 0’dan 6’ya kadar arttirilmistir (Bir sayaç birer birer artsin istenirse, Step bölümüne adim sayisi yazilmalidir). Ilk adimda sayaç 0 yapilir.Sonra, Gunler dizi-degiskeninden sayaç degeri ile ayni sayiyi tasiyan degisken alinip, bu ziyaretçinin Browser’ina yazilir(Gunler(0) Pazartesi olarak tanimlandigindan ziyaretçinin ekranina Pazartesi kelimesi yazilir). Daha sonra
 kodu

yazilir.Siradaki Next komutu ile bir sonraki adima devam edilir. Step degeri olmadigindan sayaç bir arttirilir.

VBScript, sayacin son degeri olan 6'ya ulasincaya kadar, biteviye Gunler dizi-degiskeninden sayacin degerine göre deger seçer ve bunu ekrana yazdirir.

3.8.5. While...Wend

Program mantigi her zaman yukarda verilen örnekteki gibi açik bir sayaç kurma imkani vermeyebilir. Sayaç olarak kullanilan deger, programin baska bir bölümü tarafindan üretiliyor veya ziyaretçi tarafindan belirlenmis olabilir. Özetle yapilmasi arzu edilen isin, ancak sayaç bir degerden azsa, çoksa veya esitse yapilmasi, bu durum degisirse durmasi istenebilir. Bu While (..iken) komutuyla yapilabilir. While döngüsü kullanildiginda sayaci programci arttirmalidir.örnegin, yukaridaki programda 7 günün tümünün degil ,gün sayisi 5'den küçük olmasi durumunda ekrana yazmasini istendiginde For.. Next arasinda kalan bölümde su degisik yapilmalidir:

```
While sayac <= 5
                Response.Write Gunler(sayac)
                Response.Write "<BR>"
sayac = sayac + 1
Wend
```

While döngüsünün Wend kelimesiyle sonlandirildigina dikkat edilmelidir. While satirindaki sayaç degistirilerek, programin sayaç 5'den küçük veya 5'e olmasi durumunda islemesi saglanmistir. For'dan farkli bir diger ifade ise sayaci arttiran "sayac = sayac + 1" ifadesidir.. VBScript ,sayaci bir arttirdiktan sonra ,önce While satirindaki sartin gerçeklesip gerçeklesmedigine bakar; gerçeklesmis ise Wend'i izleyen ilk satira gider; gerçeklesmemisse While döngüsünün içindeki isi yapmaya devam eder.

3.8.6. Do..Loop

Do (Yap) komutu ile kurulan döngüler iki ayri türde olabilir Bu döngü ile bir dizi komut, bir kosul dogru iken veya dogru oluncaya kadar yaptirilir. Bu yöntemlerden her biri iki ayri sekilde yazilabilir. Bir kosul dogru iken bazi islerin biteviye yapilmasi isteniyorsa, Do While yöntemi kullanilir:

Do While
kosul dogru iken yapilacak isler;
Loop

Bu ifade ile VBScript kosul dogru oldugu sürece istenilen islem yapilabilir. Buradaki Loop kelimesi, döngünün basa dönmesini saglar. Bu yöntemden su sekilde de yararlanabiliriz:

Do
kosul dogru iken yapilacak isler;
Loop While kosul

Burada, Loop komutu sartin hâlâ dogru olup olmadigini sinar ve dogru ise verilen isleri yapar; artik degilse bir sonraki satira geçer.

Döngünün bir sart gerçeklesinceye kadar bir isi yapmasi ise Do Until yöntemiyle saglanir. Bu durumda döngü söyle yazilir:

Do Until kosul
kosul gerçeklesinceye kadar yapilacak isler;
Loop

Bu ifade ile VBScript kosul dogru oluncaya kadar istenilen islemi yapar. Buradaki Loop kelimesi, döngünün basa dönmesini saglar. Bu yöntemden su sekilde de yararlanilabilir:

Do
kosul gerçeklesinceye kadar yapilacak isler;
Loop Until kosul

Burada, Loop komutu sartin henüz gerçeklesip gerçeklesmedigini sinar ve henüz gerçeklesmemisse verilen isleri yapar; gerçeklesmisse bir sonraki satira geçer.

Visual Basic metinlerinde bu döngüye verilen klasik örnek, bilgisayara yazi-tura attirmaktir! ASP sayfasinda yazi-tura attirabilmek için su kodlar yazilip ve yazi-tura.asp adiyla kaydedilmelidir;

```
<% Option Explicit %>
<HTML>
<HEAD>
<TITLE>ASP ILE YAZI-TURA</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>
<H2>
<CENTER>
<%
Dim ParaAt, Yazi, Tura, Atis
Randomize
Yazi = 0
Tura = 0
Atis = 0
Do While Tura < 3
                atis = Atis + 1
                ParaAt = Int(Rnd * 2) + 1
                If ParaAt = 1 Then
```

```
%>
Yazi!<P>
<%
                Yazi = Yazi + 1
                Else
%>
Tura!<P>
<%
                Tura = Tura + 1
                End If
Loop
%>
3 Tura getirebilmek için parayi <%=Atis%> kere atmak gerekti!
</HTML>
```

Randomize (Tesadüfî sayi bulma) fonksiyonudur.Önceleri bilgisayarin matematik islemlerde, özellikle istatistik hesaplamalarla kullanilmasi tesadüfî (rastlantisal) sayi üretmeyi gerekli kilmistir. Fakat daha sonra bilgisayar oyunlari bu islemi adeta zorunla hale getirmistir. Rastlantisal sayi, bir dizide tekrar etmesi belirli bir düzene tabi olmayan sayi demektir. Bilgisayar yokken, tesadüfî sayi tablolari matematikçiler tarafindan uzun ugraslarla üretilirdi.

VBScript, bu amaçla Visual Basic'in Randomize ve Rnd komutlarini almistir. Randomize, tesadüfî sayi üretme sürecini baslatir; Rnd da bu sayiyi kullaniciya verir. Kodun bir yerinde Rnd kullanilacaksa, ondan önce bir yerlerde mutlaka Randomize komutunun yer almasi gerekir. Örnek:

```
<% OPTION EXPLICIT %>
<HTML>
<%
Dim TesadufiSayi
Randomize
TesadufiSayi = Rnd
%>
<%=TesadufiSayi%>
</HTML>
```

Bu dosya sayilar.asp adiyla kaydedip çalistirildiginda; Browser'in Yenile dügmesinin her tiklanisinda ekranda yeni bir sayi görülür Bu sayilarin rastlantisal olmasi, bir kere daha gelmeleri için hiç bir kural (örnegin her rakamin 123 kerede bir gelmesi veya 1 milyon 245 bin kerede bir gelmesi gibi) bulunmamasidir.VBScript, her Rnd komutu icra edildiginde bilgisayarin saatini ögrenir; içinden seçecegi bir rakami son derece karmasik bir formülden geçirerek istemciye bir rakam verir. bu rakam daima 0 ile 1 arasinda olur. "(Rnd*6)+1" formülü kullaniciya 1 ile 6 arasinda , "(Rnd*13)+1" formülü ise 1 ile 13 arasinda bir deger verir. Fakat bu degerler tam sayi degildir!

Programin bütün islemi Do döngüsü bölümünde yapilir ve bilgisayarin bir tesadüfî sayi üretmesi esasina dayanir. Bu Randomize ve Rnd fonksiyonlari ile yapilir. Rnd'un verdigi tesadüfî rakam, iki ile çarpilip çikan sayiyi 1 ile toplanir, böylece ortaya 1'den büyük 3'den küçük bir kesirli rakam çikmis olur. Bu rakam Int() fonksiyonundan geçirilerek kesirden kurtarilir.

Rnd fonksiyonu ile ilgili yukaridaki örnekte, dönen sayi 0 ile 1 arasinda, yani daima kesirli bir degerdir. Bazen kullanici sayfalarinda hesaplamalar veya veritabanindan alinan degerler de kesirli olabilir. Örnegin ögrencilerin not ortalamalarinin hesabinda VBScript!te sonu gelmez kesirler elde edilir. Oysa çogu zaman bu rakamlarin ya yukari "yuvarlanmasi", ya da sadece tam sayi bölümü gereklidir.

VBScript'te Int() fonksiyonu,bir sayinin tam sayi bölümünü verir.örnegin bir KesirliSayi degiskeninin degeri 123,234567 olsun.

```
Tamsayi = Int(KesirliSayi)
```

Isleminden sonra Tamsayi degiskenin degeri 123 olur.

Round() fonksiyonu ise, kesirli bir sayiyi yukari veya asagi "yuvarlayarak" tam sayi haline getirir. örnegin KesirliSayi degiskeninin degeri bu defa 5,6 olsun.

```
Tamsayi = Int(KesirliSayi)
```

isleminden sonra Tamsayi degiskenin degeri 6 olur. Kesirli sayi 56,2 ise, Round() fonksiyonu ile sonuç 56 olur.

Programin, elde ettigi ve Paraat degiskenine kaydettigi sayi 1 ise, Yazi gelmis sayilir; ve Browser Penceresine "Yazi!" yazilir. Bu arada yapilan atis sayisinin kaydedildigi Atis ve gelen tura sayisinin tutuldugu Tura degiskenlerinin degeri bir arttirilir. ParaAt degiskeninin degeri baska bir seyse,program bu kez tura geldigine hükmedip Browser penceresine "Tura!" yazar. Do döngüsü, Tura gelen atislarin sayisi 3 oluncaya kadar devam eder. Çünkü Do döngüsünü While Tura < 3 (Tura 3'den az iken) deyimi ile çalistirilir.Sonuç olarak program 3 tura gelinceye kadar kaç atis yapildigini yazar.

Bu ASP sayfasi görüntülenirken, Browser'in Yenile dügmesi tiklandiginda, her seferinde Tura getirmek için farkli sayida atis yapmak gerekir ve ayni sayida atis yapilsa bile turalarla yazilarin yeri degisir.

3.8.7. Dizi degiskenler için döngü: For Each..Next

For..Next gibi çalisan bu özel döngü, sayaç degeri kullanmaz, fakat bir dizi degiskenin bütün degerleri için bir kere icra edilir. Dizi-degiskenler, VBScript ile yapilan islemlerde önemli bir yer tutar. Örnegin bir siniftaki ögrencilerin veya müsterilerimizin listesi bir dizi degiskenin elemanlari olabilirler. Yapilmasi istenen islem, dizi-degiskenin bütün elemanlari için tekrar edilecekse, For Each..Next döngüsü kullanmak uygun olur. Bir dizi-degiskenin eleman sayisi ilerde degisirse ve programci döngüyü For..Next ile kurmussa, döngünün sayaci için verilen alt ve üst siniri degistirmek zorunda kalir. Oysa For Each, kaç kere tekrar edecegine iliskin degeri her zaman dizi-degiskenin elemanlarin sayisindan alir.

Örnek: bu program bütün ögrencilerin listesini tutarak Ogrenciler dizi-degiskeninin bütün elemanlarinin degerini ekrana yazdirir.

```
For Each Ogrenci In Ogrenciler
Response.Write Ogrenci
Next
```

Burada "Ogrenci" Ogrenciler dizi-degiskeninde döngünün her adiminda okunan bir elemanin degerini tutar. For Each döngüsü tarafindan "okunmakta olan" dizi-degiskenin her bir degeri sirayla bu degiskene yazilir.

3.9. Döngünün Durdurulmasi

Program çalisirken belirlenen kosul gerçeklessin yada gerçeklesmesin döngüden çikilmasi gereken durumlarla karsilasilabilir. Bunu bir baska degiskendeki degisiklik zorunlu kilabilir. Bir döngüden çikmak için Exit (çik) ifadesi kullanilir. Bu ifade, döngünün yaptigi isler arasinda, genellikle bir If deyimi ile birlikte yer alir. Örnek:

```
For sayac = 1 to 10
        [..bir takim isler yap..]
        If Degisken1 &gt Degisken 2 Then Exit For
        [..bir takim islere devam et..]
Next
```

Bu durumda For..Next döngüsü, Degisken1'in degerinin Degisken2'den yüksek oldugunu belirlerse, derhal döngüyü durdurarak, Next'ten sonraki satira gider.
Do döngüsünden ise Exit Do ile çikilabilir. Bu ifadenin kullanimi da Exit For gibi olur.

3.10. Süreçler (Prosedürler)

VBScript'te programin akis kontrolünde kullanilan diger bir grup araç ise örnegin Javascript veya Perl'de fonksiyon denilen gruplandirilmis ve isimlendirilmis islem kümeleridir. Bu kümeler programin bir yerinde topluca dururlar ve programin baska bir yerinden isimleriyle çagrilirlar; veya bu kümelere isimleriyle referans yapilirlar.

VBScript'te bu kümelenmis kod gruplarina Prosedür (Süreç) denir. iki türlü olur: fonksiyon **(Function)** ve **Subroutine**. Bu iki süreç arasindaki baslica fark; fonksiyondan kendisini çagiran komuta daima bir deger döner, ancak Sub'dan dönmeyebilir. Sub, yapacagi isi yapar ve programin kontrolünü kendine atif yapilan noktaya devreder. VBScript'de bir programa farkli yerlerde sik sik ayni isi yaptiriliyorsa, bu Sub ile yaptirilir.Fakat programa bir deger gerekiyorsa, bu deger bir fonksiyona hesaplattirilir. Her ikisi de kendilerine atifta bulunan veya kendilerini göreve çagiran satirdan (komuttan, deyimden) verilebilecek degerleri kabul edebilirler.

Bir fonksiyonun adi, tipki bir degisken adi gibi, fonksiyonun ürettigi degeri tutar; ve bu deger kendisini çagiran komuta verilir.örnegin, programin çesitli noktalarinda yazi-tura atip, elde edilecek sonuca göre bir is yapilacaksa, yazi-tura komutlari yazilabilir.Ancak ortaya çok uzun bir program çikar. Oysa yazi-tura islemleri bir fonksiyonda toplanip , ihtiyaç halinde sadece bu fonksiyon çagirilirsa ve fonksiyon o anda yazi mi yoksa tura mi geldigini bildirirse,yapilacak is çok kolaylasmis olur.

Prosedür tekrarlanan islemleri barindirir ve islerimizi kolaylastirir.

Böyle bir fonksiyon ,yukarda verilen örnekten hareketle söyle olabilir;

```
<%
Function YaziTura
Dim ParaAt
Randomize
ParaAt = Int(Rnd * 2) + 1
If ParaAt = 1 Then
YaziTura = "Yazi"
Else
YaziTura = "Tura"
End If
End Function
%>
```

Bu fonksiyon, ASP programinin herhangi bir yerinden,asagidaki yöntemle çagirilip;verecegi sonuç programin akisina uygun sekilde kullanilabilir.

```
<%
NeGeldi = YaziTura
Response.Write NeGeldi
%>
```

Fonksiyonun sonunda End Function ifadesi bulunduguna ve fonksiyonun elde ettigi sonucun kendi adina atandigina dikkat edilmelidir. DIM ifadesiyle böyle bir degisken tanimlanmadigi halde VBScript, fonksiyon çagirildigi anda bunu kendiliginden yapar.

Ayni islem Subroutine (Sub) olarak yazilabilir. Fakat bu kez Sub, elde ettigi degeri kendisi kullanir ve bittigi anda kontrol programina geri döner:

```
<%
Sub YaziTura()
Dim ParaAt
Randomize
ParaAt = Int(Rnd * 2) + 1
If ParaAt = 1 Then
Response.Write “Yazi”
Else
Response.Write “Tura”
End If
End Sub
%>
```

Fonksiyon adlarinin sonuna, programcidan beklenen bir deger varsa onlari belirleyen degisken adlari parantez içinde yazilir. Fonksiyon böyle bir deger beklemiyorsa açilan kapanan (bos) parantezlere ihtiyaç yoktur.Sub'larin çagrilmasi, fonksiyondan farklidir. Sub'in icra edilmesinin istendigi noktaya sadece adi yazilir. Sub'lar isleyebilmek için programcidan deger bekliyorsa, bu degerler Sub adinin yanina, parantez içine alinmadan ve virgülle ayrilarak, yazilir. Örnegin, Hesapla isimli ve çalismak için iki deger bekleyen bir Sub söyle çagrilir:

```
Hesapla 10, 20
```

Sub isleyisi bittigi anda programin akisi, Sub'a atif yapilan noktadan devam eder.

3.11. Sik Kullanilan Hazir Fonksiyonlar

VBScript'te kullanilan tesadüfi sayi üreten Rnd() fonksiyonu , kesirli bir sayinin tam bölümünün alinabildigi int()fonksiyonu gibi iki hazir fonksiyon disinda VBScript'in kullanilmaya hazir daha bir çok fonksiyonu vardir.Ancak ASP uygulamalarinda sik kullanilan ve özellikle metin düzenlemeye ait olan bir kaçi söyle siralanabilir.

3.11.1. Tarih ve saat

Belki de Web'in zamana çok bagli olusu dolayisiyla, Visual Basic'in hemen hemen bütün zaman-tarih fonksiyonlari VBScript'te de kullanilir.

Date: Bugün tarihini verir. (15.04.2001 gibi)
Time: O andaki saati verir. (15:10:40 gibi)
Now: O andaki tarih ve saati birlikte verir. (15.04.2001 15:10:40 gibi)
VBScript'in buna ek olarak Weekday (haftanin günü), WeekdayName (günün adi) ve Monthname (ayin adi) fonksiyonlari da vardir. Bu fonksiyonlar degerlerini Date fonksiyonuna göre alirlar. Örnegin,

```
<%= WeekdayName(Weekday(Date))%>
```

komutu bugün Pazar ise "Pazar" degerini verir.

```
<%= MonthName(Month(Date))%>
```

komutu bu ay Nisan ise "Nisan" degerini verir. VBScript'in bunlara ek olarak Day (gün), Month (ay) ve Year (yil) fonksiyonlari da degerlerini Date fonksiyonundan alarak,bir rakam verirler. Eger tarih 15 Nisan 2001 ise:

<%= Day(Date)%>... 15
<%= Month(Date)%>... 4
<%= Year(Date)%>...2001

degerini verir. VBScript, bu degerleri dogruca isletim sisteminden alir. Dolayisiyla isletim sisteminin bölgesel ayarlari Türkiye için yapilmissa, gün adlari Türkçe olarak döner. Ayrica, tarih ve saat biçimleri de bölgesel ayarlara bagli olarak, ay önde, gün arkada veya tersi, saat de 12 saat veya 24 saat esasina göre döner. ASP programlari kisisel Web Server'da denenirken bilgisayarin tarih ve saati; gerçek Internet'te çalistirirken Server'in tarih ve saatini alir. Sayfalarda ay ve gün adlarinin Türkçe görüntülenmesi için, önce Server'in bölgesel ayarlari sinanmali ve eger isimler Türkçe gelmiyorsa, bunlari çeviren Sub'lar veya fonksiyonlar yazilmalidir.

3.11.2. Dizi-Degisken (Array) Fonksiyonu

VBScript'in dizi-degisken olusturmada Array() fonksiyonu ile sagladigi kolayliklara kisaca deginildi. Fakat Array ile daha bir çok is yapilabilir; ve dizi degisken olusturmakta VBScript'in diger bazi kolayliklarindan yararlanilabilir. Dizi-degiskenler, özellikle Web ziyaretçilerinden gelecek bilgilerin kaydedilmesinde; veritabanindan çekecegimiz verilerin kullanilir hale getirilmesinde yararli bir araçtir. Dolayisiyla ASP sayfalarinda sik sik çok-boyutlu dizi degiskenlerden yararlanilabilir. Bunun için gerekli araçlari kisaca ve topluca ele almamiz yerinde olur.

Bir dizi degisken olustururken, degiskenin eleman sayisi belirtilmezse, VBScript, elemanlarinin degerleri sonradan belirtilebilecek ve eleman sayisi sonradan arttirilabilecek bir dinamik dizi-degisken olusturur.

Örnek:

```
Dim Ogrenciler()
```

Bu komutla, Ogrenciler dizi-degiskeni olusturulur; ancak eleman sayisi belirtilmedigi için dizi dinamiktir; yani daha sonra bu dizinin eleman sayi belirlenebilir. Bu:

```
ReDim Ogrenciler(15)
```

gibi bir komutla yapilabilir. Dizi-degiskenimizin eleman sayisini önceden belirlememizin sebebi; programin akisi içinde bu sayi, baska bir fonksiyonun, Sub'in veya kullanici girdisinin sonucu olarak belirlenebilir olmasindan dolayidir. Fakat ReDim komutu, mevcut bir dizi-degiskenin içindeki her seyi siler. Mevcut dizinin elemanlari ve onlarin degerleri korunmak istenirse:

```
ReDim Preserve Ogrenciler(20)
```

yazilmasi gerekir. Buradaki Preserve (koru) komutu, VBScript'e mevcut dizi içindeki elemanlari korumasini, ve eleman sayisini 20'ye çikartmasini bildirir. Çünkü ziyaretçinin tercihleri degisebilir; örnegin bir elektronik alisveris sitesinde ziyaretçi yeni seyler alabilir; daha önceki alisverislerine iliskin verilerin tutuldugu dizi-degiskenin eleman sayisini, daha önceki bilgilerin silinmeden arttirilmasi gerekir.
VBScript'in dizi-degiskenleri tümü ayni adi tasiyan bir liste gibidir; sadece degisken adinin yaninda dizinin kaçinci elemani oldugunu belirten sayi bulunur:

```
Ogrenciler(1): Necip
Ogrenciler(2): Serap
Ogrenciler(3): Neslihan
```

Fakat VBScript çok boyutlu dizi degisken de olusturabilir. Iki boyutlu dizi-degisken tablo gibi düsünülürse; dizinin elemanlari ayni adi tasiyan degiskenler fakat bu kez sadece tek sayi degil sira ve sütun numaralari ile belirlenirler:

```
Ogrenciler(1,1): Necip
Ogrenciler(1,2): Serap
Ogrenciler(1,3): Neslihan
Ogrenciler(2,1): Selim
Ogrenciler(2,2): Murat
Ogrenciler(2,3): Merve
Ogrenciler(3,1): Elif
Ogrenciler(3,2); Hande
Ogrenciler(3,3): Leyla
```

Simdi, burada üç sirali, üç sütunlu bir tablo getirebilirsiniz gözünüzün önüne. Bu dizi-degisken su komutla olusturulabilir:

```
Dim Ogrenciler(3,3)
```

Böyle bir degiskende sözgelimi birinci sira (numarasi 1,x olanlar) çaliskanlari, ikinci siradakiler (2,x'ler) daha az çaliskanlari vs., belirtebilir. VBScript, üç, dört ve hatta

bes boyutlu dizi-degisken olusturur. Ama bunun nerede kullanilacagini tasarimci kararlastirir.

Bir dizi-degiskenin herhangi bir elemanin degeri, programin herhangi bir asamasinda degistirilebilinir:

```
Ogrenciler(3,2) = "Caner"
```

komutu, Hande'nin adini siler ve yerine Caner'in adini yazar.

Dizi-degiskenlerinin eleman sayisini bilmek isteyebiliriz. Bazen dizi-degiskenlerinin eleman sayisi bizim tarafimizdan belirlenmez; bu bilgi bir formdan gelebilir; bir veritabanindan alinabilir; fakat mesela bir döngü için bu degiskenin kaç elemani oldugunu bilmek gerekir. Örnegin elimizde 35 elemani olan Ogrenciler dizi-degiskeni varsa, bu sayiyi

```
ElemanSayisi = UBound(Ogrenciler)
```

komutu ile ElemanSayisi degiskenine yazdirabiliriz. ElemanSayisi'nin degeri bu durumda 35 olacaktir.

3.11.3. Test Fonksiyonlari

VBScript'te kullanilan bazi degiskenlerin o andaki durumu, programin akisini kontrolde kullanilan bilgiyi saglayabilir. Sözgelimi bir degiskenin degeri bos ise, ziyaretçinin formu tam olarak doldurmadigi düsünülür. VBScript de degiskenlerin durumunun sinanmasi için bazi özel fonksiyonlar mevcuttur. Bu özel fonksiyonlardan dönen deger True (dogru) veya False (yanlis) olur; dogru sonucun degeri –1, yanlis sonucun degeri ise 0'dir:

IsArray	Bir degiskenin dizi-degisken (Array) olup olmadigini sinar.
IsDate	Bir degiskenin degerinin tarihe (Date) çevrilip çevrilemeyecegini sinar.
IsEmpty	Bir degiskenin tanimlanip deger atanmis olup olmadigini sinar.
IsNull	Bir degiskenin geçerli bir deger tutup tutmadigini sinar.
IsNumeric	Bir degiskenin sayi olarak isleme tabi tutup tutulamayacagini sinar
IsObject	Bir ifadenin geçerli bir ActiveX veya OLE nesnesine referansta bulunup bulunmadigini sinar.
TypeName	Bir degiskenin türünü belirtir.
VarType	Bir degiskenin türünü belirten sayiyi verir.

4. ASP NESNELERI

ASP tekniginde de amaç nesnelerin özelliklerini kullanarak, ya bu özellikleri belirlemek, ya da degistirmektir. Nesne Yönelimli Programlama (Object Oriented Programming, OOP) ile nesneye yönelik programlama yapilir. Örnegin her ögrencinin notunu veritabanina isleyen, veritabanindan notlari alarak geçeni-kalani belirleyen veya ögrencilerle ilgili daha yapilmasi gereken bir çok isi yapan fonksiyonlari ve Sub'lari olan bir ASP programinda; bir çok degisken vardir; ve burada nesne"ögrenci" dir. Böylelikle bu nesneye yönelik program yapilmistir.

Her “program nesnesi” iki unsura sahiptir:

Özellik (Property, Attribute): Bir nesnenin özellikleri, onun degiskenleridir. “Ögrenci” nesnesinin “Ögrencinin Adi,” “Notlari,” “Adresi” gibi degiskenleri, yani özellikleri vardir.

Metod (Method): Bir nesnenin islemesi, çalismasi için, kisaca kendisinden bekleneni yerine getirebilmesi için çalisma yöntemlerine ihtiyaci vardir. Dolayisiyla bir ASP nesnesinin fonksiyonlari, onun metodlaridir.

Fakat ASP’de nesneler sadece öbekler halinde toplanan fonksiyonlar ve degiskenlerden ibaret degildir. ASP programinda kullanilan Script dilinin getirdigi nesneler vardir. ASP sayfasi Javascript ile yazilirsa baska, VBScript ile yazilirsa baska dil nesnelerine sahip olunur; ancak her ikisinde de ortak olan “Scripting” nesneleri vardir.Bir de Web Server’in hazir sundugu nesneler vardir. Bunlar daha sonraki bölümde açiklanacaktir. Ayrica, Browser’in bir HTML sayfasinin bölümlerini nesne sayarak olusturdugu nesneler vardir.

Nesneler nasil olusmus olursa olsunlar, daima kullaniciya bir deger verirler:

Nesne.Özellik = Deger

Bir nesnenin bir özelliginin degeri,bir degisken degeri gibi önem tasir:

If Nesne.Özellik > Deger Then ...

Nesnelerin özelliklerinin degerleri degiskenlere atanabilir; ancak bunu yaparken Nesne’nin bir metoduna (fonksiyonu) göndermede bulunulmali ve gerekiyorsa bu fonksiyona kullanmasi için veri göndermelidir (bir fonksiyona kullanmasi için gönderilen degere argüman/argument denir):

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

Daha sonra bu degisken istenilen yerde kullanilabilir.

Nesneler, diger yararlarinin yani sira, birbiri ile ilgili Sub’larin, fonksiyonlarin ve degiskenlerin bir arada tutulmasini saglar. VBScript ile olusturulan bir nesne örnegi verirsek:

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

```
End Class
%>
```

Burada Nesne(object) adi verildi fakat Class(sinif) olusturuldu.“Sinif” ancak Vbscript tarafindan kullanilmaya baslanirsa Nesne olur. Dolayisiyla “sinif” yazilir ve VBScript onu Nesne haline getirir. Nesne olusturulduktan sonra bu nesneden yeni bir olgu (instance) olusturulur :

```
<%
Dim Ogr1
Set Ogr1 = New Ogrenci
Ogr1.Adi = “Mehmet”
Ogr1.Soyadi = “Can”
Ogr1.No = “181”

Response.Write Ogr1.AdiSoyadi
%>
```

Her nesne, New (yeni) komutu ile yeni bir degiskene bütün özelliklerini verir. Burada Ogr1 degiskeni, yukarida olusturulan Ogrenci nesnesinin bütün özelliklerini kazanmis olur. Ogrenci nesnesinin “.Adi”, “.Soyadi” ve “.No” özellikleri olmasi gerekir; nitekim Ogr1’e bu özellikleri burada verilir. Ogrenci nesnesinin bir de metodu (fonksiyonu) vardir; Ogr1 bunu da kazanir,ve hem de bunu ziyaretçinin Browser penceresine yazdirabilir.ASP sayfalarin nesne olusturarak çalismak büyük kolayliklar saglar.

4.1. Hata (Err) Nesnesi

Hangi dille olursa olsun program yazarken hata yapmak kaçinilmaz bir kuraldir. Dolayisiyla kullanilan programlama dili hatalarin kolayca yakalanmasina imkan vermelidir.
ASP programlarinda yazim yanlisligi, olmayan degiskene gönderme gibi Script hatasi olmamasi gerekir. Bu tür hatalar, program Web’e gönderilmeden mutlaka ayiklanmalidir.

Fakat programcinin öngöremeyecegi, ve çogu Web ziyaretçisinden veya ziyaretçinin bilgisayarindan kaynaklanan hata durumlari olabilir. VBScript, su standart komutla beklenmedik hata durumlarinda programin yoluna devam etmesini saglayabilir:

```
<%Hata Error Resume Next %>
```

Bu komutla VBScript'e, hata halinde bir sonraki satirdan yoluna devam edecektir. Fakat olusan hata, programin daha sonra vermesi beklenen sonucu vermesini önleyebilir.VBScript, Err (Hata) Nesnesi'nin bir çok özelliginden özellikle hata sayisi (Number), tanimi (Description) ve kaynak (Source) özellikleri ile hatanin ne oldugunu ve nereden kaynaklandigini gösterebilir. Örnegin,

```
If Err:Number = xx Then
```

seklinde bir ifade ile hatanin türüne göre programin kazasiz yürümesi saglanabilir. Burada xx yerine 108 ayri hata numarasi yapabilirsiniz. Hata numaralari, Microsoft'un VBScript sitesinden edinilebilir.

4.2. Dosya Sistemi Nesnesi

Dosya Sistemi Nesnesi (FileSystemObject), ASP programinin, Web Sunucusunun sabit disk sisteminde, sürücüleri, klasörleri ve dosyalari yönetmekte kullanilan temel araçtir.Bunu açiklamak için su kod yazilip dosya_yaz.asp adiyla kaydedilir:

```
<% Option Explicit %>
<HTML>
<HEAD>
<TITLE>ASP ILE DOSYA YAZMA</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>
<Dim YaziFSO, yaz
Set YaziFSO = CreateObject("Scripting.FileSystemObject")
Set yaz = YaziFSO.CreateTextFile("c:\yazi_deneme.txt",True)
yaz.WriteLine("Bu bir denemedir.")
yaz.Close
%>
<H2><CENTER>Bu Web sayfasi sabit diske yazi yazdirir!!
<BR>simdi C: sürücüsünde yazi_deneme.txt adli bir dosya olmasi gerekir!
<BR>Lütfen bakar misiniz?</H2></CENTER>
</BODY>
</HTML>
```

Kodun Dim satirinda iki degisken belirlenmistir.Fakat bu iki degiskeni sistem nesnesi olan Scripting'in yeni bir olgusu olarak kullanilacagindan daha önce standart degiskenlere deger atandigi gibi degil, fakat Set komutundan yararlaniriz, ve YaziFSO degiskeninde bir "Scripting.FileSystemObject" nesnesi olusturulmasini saglariz. (ASP uzmanlari arasinda gelenek, nesne degeri tutan degiskenlere, ilgili nesnenin bas harflerini eklemektir. Böylece bir degiskenin adina bakarak, islevini anlamak mümkün olur.)

"yaz" degiskeni YaziFSO'da yeni bir olgusunun olusturuldugu FileSystem Object'in CreateTextFile (Düzyazi dosyasi olustur) metodunu kullanir; bu metod olusturulacak dosyanin adini ve eger bu dosya varsa üzerine yazilmasina izin veren True (dogru) veya buna izin vermeyen False (yanlis) kelimesini argüman olarak alir. "yaz" degiskeni simdi kendisi bir metod kullanabilecek sekilde, FileSystemObject'in bir örnegidir; nitekim WriteLine metodu ile biraz önce olusturulan dosyaya, argüman olarak verilen metni yazdirmaktadir. Bu kod çalistirildiginda sabit diskte düzyazi dosyasi görülür.

Böylece sistem nesneleri kullanarak çok daha farkli seyler yapilabilir .File System Object nesnesi bize sabit diske erisme ve onun kaynaklarini kullanma imkani verir. Bütün nesneler gibi kullanilabilmesi için önce bir degiskenin bünyesinde olusturulmasi gerekir:

```
<%
Dim DosyaSistemi
Set DosyaSistemi = CreateObject("Scripting.FileSystemObject")
%>
```

Dosya Sistemi Nesnesi'nin 20'den fazla metodu vardir; fakat bunlardan önemlileri söyle siralanabilir:

CopyFile (dosya kopyala), MoveFile (Dosya tasi), CopyFolder (klasör kopyala), MoveFolder (klasör tasi), Create Folder (klasör olustur), DeleteFile (dosya sil), DeleteFolder (klasör sil).
Bunlardan birinin nasil kullanilabilecegine dair örnek verilirse:

```
<%
Dim DosyaSistemi
Set DosyaSistemi = CreateObject("Scripting.FileSystemObject")
DosyaSistemi.DeleteFile "c:\belgelerim\test.*"
%>
```

Bu program ile "Belgelerim" klasöründeki "test" isimli bütün dosyalari silmis olunur.Bu program dikkatli çalistirilmalidir ; çünkü ASP yoluyla silinen dosyalar , Geri Dönüsüm Kutusu'na gitmez ,daha az zararli bir diger örnek ise söyle olabilir:

```
<%
```

```
Dim DosyaSistemi
Set DosyaSistemi = CreateObject(“Scripting.FileSystemObject”)
DosyaSistemi.CopyFile “c:\belgelerim\*.*”, “c:\yedekler\”
%>
```

Bu program “Belgelerim” dizinindeki bütün dosyalari “Yedekler” dizinine kopyalar.

File System Object’in sadece bir özelligi (Property) vardir: Drives (sürücüler). Fakat bu özellik, bir degil bir çok elemandan olusan bir dizi-degisken gibi Kolleksiyon (Collection) sayilir. Çünkü bir Web Server’da birden çok sürücü bulunur. Her sürücü, bu kolleksiyonun üyesidir (FileSystem.Drives) ve her birinin sürücü harfi (.DriveLetter), disk adi (.VolumeName), byte olarak bos alani (.FreeSpace) özellikleri vardir. suruculer.asp adiyla kaydedilecek su program, denenmek istenen sistemin disk-disket-CD-ROM durumunu listeler.

```
<% Option Explicit %>
<HTML>
<HEAD>
<TITLE>ASP ILE SÜRÜCÜ KOLLEKSIYONU</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>
<%
Dim DosyaSistemi, Surucu, Suruculer
Set DosyaSistemi = CreateObject("Scripting.FileSystemObject")
Set Suruculer = DosyaSistemi.Drives
For Each Surucu In Suruculer
%>
<b>Sürücü:</b> <%=Surucu.DriveLetter%><br>

<% If Surucu.IsReady = True Then%>
<b>Disk Adi:</b> <%=Surucu.VolumeName%><br>
<b>Bos alan:</b> <%=Surucu.FreeSpace%><br>
<% Else %><i>Sürücü hazir degil!</i><br>
<% End If
Next %>
</BODY>
```

```
</HTML>
```

Burada özelliklerinin ve metodlarinin Dosya Sistemi adli degiskene atandigi Dosya Sistemi Nesnesi'nin sürücüler kolleksiyonu dizi-degisken gibidir. For..Next akis kontrolü ile bu kolleksiyonun bütün üyelerinin sirayla sürücü harfi, ve hazirsa disk adi ve bos alani bilgileri alinir. Drives kolleksiyonunun diger özellikleri arasinda toplam yüzey genisligi (TotalSize), sürücü türü (DriveType; 0=bilinmiyor; 1=çikartilabilir; 2=sabit; 3=ag; 4=CD-ROM; 5= RAM-Drive), ve dosya sistemi (FileSystem; FAT, NTFS, CDFS), kök dizin (RootFolder) vardir. Bu program bir PWS'da çalistiginda, su sonucu alinir:

Script açisindan, her sürücüde klasörler (Folders) ve onlarin içinde alt-klasör (Subfolders) ve dosya (Files) kolleksiyonlari bulunur. (Her klasörün içinde bir alt-klasör nesnesi bulundugu için ASP ile sonsuza kadar bütün klasörlere gönderme yapilabilir) Klasör nesnesinin bazi özellikleri sunlardir:

Adi (Name), olusturulma (DateCreated), erisim (DateLastAccessed), degistirme (DateLastModified) tarihleri, içindeki dosyalar ve alt-klasörlerdeki dosyalarla birlikte boyutu (Size), bulundugu sürücü (Drive), içinde bulundugu klasör (ParentFolder), alt-klasörler (SubFolders), kök dizin olup olmadigi (IsRoot).

Klasör nesnesinin kopyala (Copy), sil (Delete) ve Tasi (Move) metotlari vardir.
Dosya (File) nesnesinin de ad, olusturma, erisim, degistirme, boyut, sürücü ve içinde bulundugu sürücü özellikleri, ve kopyala, sil, tasi metodlari vardir.

PWS'in bulundugu sistemde, söz gelisi C: sürücüsünün kök dizinindeki bütün dosyalarin listesini veren bir kod söyle yazilir (dosyalar.asp):

```
<% Option Explicit %>
<HTML>
<HEAD>
<TITLE>ASP ILE KLASOR - DOSYA KOLLEKSIYONU</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>
<%
Dim DosyaSistemi, Surucu, Dosya, KokDizin, KokDosyalar, DosyaNesnesi
Dim SurucuHarfi
SurucuHarfi = "C:"
Set DosyaSistemi = CreateObject("Scripting.FileSystemObject")
Set Surucu = DosyaSistemi.GetDrive(SurucuHarfi)
Set KokDizin = Surucu.RootFolder
```

```
Set KokDosyalar = KokDizin.Files
For Each DosyaNesnesi In KokDosyalar
%>
<%=DosyaNesnesi.Name%><br>
<% Next %>
</BODY>
</HTML>
```

SurucuHarfi degiskenin degerini degistirerek, arzu edilen disk/disket veya CD-ROMa ulasilabilir. GetDrive metodu ile; VBScript, fiilen disk/disket sistemine erisir.

4.3. Metin(TextStream) Nesnesi

Dosya sistemi nesnesi disk sistemine, klasörlere ve dosyalara erisme imkani verir ama yeni dosyalari olusturmak veya mevcutlara ek yapmak için yeterli özellik ve metoddan yoksundur. Bunu TextSream nesnesi saglar.

Bir isletim sistemi, metin dosyalarini okurken, yazarken bir metin akisi olur; TextStream nesnesinin adi da bunu anlatir: Metin Akimi. Web Server ve dolayisiyla ASP açisindan sabit diske bir metin yazarken, veya sabit diskten bir metin okurken, bir metin akisi nesnesi olusur. Bu nesnenin özellikleri ve metodlarini kullanarak, örnegin ziyaretçilerin siteye birakacaklari form bilgileri Web Server'in sabit diskine yazdirilabilir. Veya mevcut metinler okunabilir ve bunlarin içerigi ziyaretçiye gönderilecek HTML sayfasinin etiketlerinin içerigi olarak kullanilabilir. Metin dosyasi okumak ve yazmak disk sistemini ilgilendiren bir eylem oldugu için yine Scripting nesnelerinden FileSystemObject nesnesinden yararlanilir fakat bitirme ödevimizde bunun yaninda degisik metodlar da kullanildi.

4.3.1. Metin Dosyasi Olusturma (CreateTextFile)

ASP nesnelerinin neler yapabilecegine örnek olarak yazilip,dosya_yaz.asp adiyla kaydedilen program, bir metin dosyasini yazdirma islemidir.Buradaki kodlar CreateTextFile (metin dosyasi olustur) metoduna argüman olarak yeni metin dosyasinin yolunu ve adini veriyor. Bu metod TextStream nesnesinindir; ve otomatik olarak bu nesnenin diger metodlarinin kullanilmasini saglar. Kullanilan metodlar ise WriteLine (satir yaz: bir String'i sonuna yeni satir karakteri koyup dosyaya yazar) ve Close (kapat: açilan metin dosyasini kapatir).

TextStream'in burada kullanilan ikisinin disinda iki metodu daha vardir:

Write (yaz): Bir String dosyaya yazdirilir; satir sonuna yeni satir karakteri (Return kodu) konulmaz.
WriteBlankLines (bos satir yaz): Bir metin dosyasina argüman olarak verilen sayida bos satir yazdirilir.

4.3.2. Varolan Metin Dosyasina Ek Yapma (OpenTextFile)

Mevcut bir metin dosyasina ek yapmak için OpenTextFile metodu kullanilir.Site konuk defterine yazilanlar, bir metin dosyasinda toplanmak istenirse ,bu metod ile, açilan dosyanin yolu ve adi istenir.

dosya_yaz.asp'nin ilgili satirina su satir eklenir:

```
Set yaz = YaziFSO.OpenTextFile("c:\yazi_deneme.txt",8,0)
```

Burada dosya yolunu ve adini veren birinci argümana ek olarak "8,0" seklinde iki yeni argüman görülür.Bunlardan birinicisi girdi/çikti durumu (I/O Mode), ikincisi ise biçim (Format) ile ilgilidir. I/O Mode parametreleri sunlardir:

1: okumak için aç
8: eklemek için aç

Açilacak dosyanin biçimini belirttigimiz son argüman ise su degerlerden birini alabilir:

 0: ASCII dosyasi olarak aç
-1: Unicode dosyasi olarak aç (Örnegin içinde Türkçe karakterler varsa)
-2: Sistemin varsayilan dosya türü olarak aç

Buna göre, bir dosyayi salt okumak amaciyla açmak için "1,0" argümanlari kullanilir.Bir dosya açilip, içindekiler okunmak istenirse asagidaki örnek yazilir:

```
<% Option Explicit %>
<HTML>
<HEAD>
<TITLE>ASP ILE DOSYADAN METIN OKUMA</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>
<%
Dim DosyaSistemi, MetinDosyasi, Satir
Set DosyaSistemi = CreateObject("Scripting.FileSystemObject")
Set MetinDosyasi = DosyaSistemi.OpenTextFile("c:\yazi_deneme.txt",1, 0)
Do
```

```
Satir = MetinDosyasi.ReadLine
%>
<%=Satir%>
<%
Loop Until MetinDosyasi.AtEndOfStream
MetinDosyasi.Close
%>
</BODY>
</HTML>
```

Burada kullanilan metod ReadLine metodudur. Bu, açilan metin dosyasindan bir satir okunmasini saglar. Ikinci ve son satirlarin okutulmasi ise Do..Loop kontrolü ile saglanir.

Bu döngü,Metin Dosyasi nesnesi, AtEndOfStream (akimin sonunda) oluncaya kadar sürer.Böylece Dosyanin sonuna gelindigi anda Do..Loop yaptigi isi durdurur, bir sonraki komuta geçer. Burada kullanilan

ReadLine metoduna ek olarak yararlanabilecek diger metodlar ise sunlardir:

Read (oku): Bir sayi argümani ile çalisir ve verdiginiz sayi kadar karakter okur.
ReadLine (satir oku): Bir satir okur ve String olarak verir.
ReadAll (tümünü oku): Bütün satirlari okur ve tek String olarak verir.
Skip (atla): Bir sayi argümani ile çalisir ve verdiginiz sayi kadar karakteri atlar.
SkipLine (satir atla): Bir sonraki satira atlar.

Bu metodlarla sagladigimiz okuma isinin kontrolü amaciyla su özellikleri de kullanabiliriz:

AtEndOfStream (akimin sonunda): Okutulan dosyanin sonuna gelinmesi halinde True (dogru) olur.

AtEndOfLine (satirin sonunda): Okutulan satirin sonuna gelinmesi halinde True (dogru) olur.

4.4. Sunucu (Server) ve Talep (Request) Nesneleri

Buraya kadar ele aldigimiz nesneler bir anlamda bizim sadece tek tek sayfalarda yararlanacagimiz araçlari sagliyor. ASP'yi diger CGI teknolojilerinden ayiran baslica özelliklerden biri, tek tek Web sayfalarini diyalog kutulari, mesaj kutulari, girdi kutulari gibi, birarada bir "uygulama programi" olarak baglayabilmesidir. Kisaca Web Server, ziyaretçinin siteye baglandigi ve ana sayfayi açtigi andan itibaren,sitenin bir program bütünlügünde çalismasini saglar.

ASP sayfalarinda kullanilan ikinci grup nesne, Sunucu Nesneleri'dir. Bu grupta önce Sunucu'nun kendisi yer alir; sonra ziyaretçi ile kurulan iliski gelir. Ziyaretçi ile olan

iliski iki yönlü trafige benzetilirse:ondan sunucuya gelen talepler, sunucunun ona karsiliklari.

Ziyaretçiden sunucuya gelen trafige Talep denir.Ziyaretçi, Browser'inin URL hanesine yazdigi her adresle, veya formlardaki bir dügmeyi veya sayfalarimizdaki herhangi bir köprüyü tiklamakla, Server'dan istenileni göndermesini ister. Bu taleptir. Ziyaretçi taleplerinin tümü Talep Nesnesi (Request Object) olarak bir arada ele alinabilir. Server'in bu taleplere verdigi karsiliklar, yani ziyaretçinin Browser'ina gönderdigi sayfalar, resimler, sesler, videolar ise karsiliktir ve ASP açisindan Karsilik Nesnesi'ni (Response Object) olusturur.

4.4.1. Server Nesnesi

Web Server site adina yönetimden sorumludur. Az özelligi ve metodu olmasina ragmen bu nesneden çok yararlanilir. ActiveX ve COM bilesenlerini çalistirmak Server'in görevidir. Web Server ,ASP için bir nesnedir ,ASP'nin bir çok isini bu nesnenin özellikleri ve metodlari halleder.Server nesnesinin bir özelligi (ScriptTimeout) ve dört metodu (CreateObject, HTMLEncode, URLEncode, MapPath) vardir. Web Server çalistigi bilgisayarin, site adina yönetiminden sorumludur.

ScriptTimeout Özelligi: ASP Script'i, programcinin, ziyaretçinin, veya Server'in bir hatasi sebebiyle sonsuz döngüye girerse,programin bir sekilde durdurulmasi için Web server programinin Script Timeout (Script süre siniri) diyalog kutusuna bir deger girilir. Öregin MS-Internet Information Server için varsayilan Script Timeout süresi 90 saniyedir. Yani ISS, herhangi bir Script'in çalisip-durmasini 90 saniye bekler; bu sürenin sonunda Script'in çalismasi tamamlanmazsa ziyaretçiye arzu ettigi sayfanin veya unsurun bulunamadigini bildirir. Bu süreyi (Server'in varsayilan degerinin altinda) kisaltmak degilse bile uzatmak mümkündür. Bu ScriptTimeout özelligi kullanilarak yapilabilir. Bunun için sayfanin herhangi bir yerine su kodu koymak gerekir:

```
<% Server.ScriptTimeout = 100 %>
```

Bu scipt ile serverin varsayilan Script Timeout süresi 90 saniye ise 100 saniyeye çikarilmis olur.Script çok karmasik ise veya baska bir Server'daki veritabanindan veri çekiyor ise bu yöntem kullanilmalidir.ASP sayfalari çok karmasiksa ve sürekli Timeout hatasi veriyorsa, hata aramadan önce bu süre uzatilmalidir.

CreateObject Metodu: Buraya kadar kullanilan CreateObject Scripting nesnesinin bir metodu idi; bu ise Server nesnesine aittir. Eger sayfada reklam amaçli banner grafiklerini belirli zaman araligi ile veya ziyaretçiye gönderilen Cookie (çerez) bilgilerine göre degistirmek istenirse,örneginin MS-Web Server Programinin AdRotator bilesininden yararlanilirsa söyle bir kod yazilmalidir:

```
<% Set Reklam = Server.CreateObject ("MSWS.AdRotator")%>
<%= Reklam.GetAdvertisement("/reklamlar/buyukbanka.txt")%>
```

Burada GetAdvertisement, Server'in AdRotator bileseninin bir metodudur. Server'in CreateObject metodundan, veritabanina ulasirken de yararlanilir.

MapPath (Yolu belirle) Metodu: Web Server açisindan "kök dizin" (root directory) Server'in bulundugu bilgisayarin sabit diskinde, herhangi bir klasör olabilir. Örnegin IIS için bu varsayilan deger olarak "C:\inetbup\wwwroot" klasörüdür. Özellikle ASP ile "program niteliginde siteler" yapilirsa, sitenin ilgili bütün dosyalarinin bulundugu bir dizin için yol belirlemek gerekir. Bu Server nesnesinin MapPath (Yolu belirle) metodu ile yapilabilir:

```
WebDizini = Server.MapPath("/benim_site")
```

Bu komutla WebDizini degiskenin degeri söyle olur:
"C:\inetbup\wwwroot\benim_site\"

Bazen sayfalarda ziyaretçi ile etkilesmenin sonucu olarak varsayilan Web dizini degistirmek istenebilir. örnegin biri Türkçe, digeri Ingilizce iki site varsa, ve ana sayfada ziyaretçi Türkçe'yi seçtiyse,bu seçimden itibaren Web uygulamasi için Web kök-dizini, "/turkish/" olacak ve sözgelimi resimler için verilen "/resimler/" dizini kök dizinde degil, "/turkish/resimler/" klasöründe aranacaktir.

HTMLEncode, URLEncode

Örnegin içinde html kodu olan bir metin olsun:

```
Server.HTMLEncode("Degisken1 < Degisken2")
```

yazilirsa, bu metin ASP tarafindan HTML kodu olarak yorumlanmaz, metin olarak algilanir.Internet'te bazen özellikle sayfa adresleri belirtilirken bazi degerlerin "URL Kodu" denilen sekilde kodlanmis olarak gönderilmesi gerekir. Bu kodlama türünde bosluklarin yerine + isareti konmus olmasi gerekir. Bu tür bilgiler gönderilecegi zaman:

```
Server.URLEncode("kelime 1 kelime2 kelimeme2")
```

seklindeki bir kod Bunu hemen su sekle sokar:
kelime1+kelime2+kelime3

4.4.2. Talep (Request) Nesnesi

Web Server , bir Web ziyaretçisi herhangi bir talepte bulundugu, yani bir sayfanin gönderilmesini istedigi anda, bu talebi, bir nesne halinde ele alir; kolleksiyonlar olusturur. Bu kolleksiyonlar, HTTP protokolü ile iletisimin sonucu olarak ziyaretçinin Browser'indan ve Internet'e giris noktasi olan ISS'in bilgisayarindan baslayan ve Web Server'dan derlenen bir dizi bilgidir. Bir anlamda, Request nesnesi, Web programinin Girdi (Input) bölümünü olusturur.

Request nesnesi kendi içinde dört ana nesne barindirir:

4.5. QueryString ve Form

Web ziyaretçisinin bilgisayarindan Server'a gelen her sey, QueryString Web kolleksiyonunu olusturur. Bu, ziyaretçinin Browser'in URL adresi hanesine yazdigi bir basit HTML sayfasi yolu ve adi veya bir Form'un Gönder dügmesini tikladiginda gelen bilgiler olabilir. Bu bilgilerin su özellikleri kullanilabilir:

Content_Length: Bir Form'dan gelen bilgilerin tümümün byte olarak boyutudur.
Remote_Host: Ziyaretçinin IP adresini verir; ancak Internet'e çevirmeli ag ile baglanan ziyaretçiler her seferinde farkli bir IP bildirebilirler. Bu yüzden bu bilgi ziyaretçinin kimligi sayilamaz.
Request_Method: Form'da kullanilan GET veya POST metodunu bildirir. Iki yöntemle gelen bilgi farklidir. Form'un olusturdugu bilgiler GET yöntemi ile alinirsa bu, çevre degiskenlerinden QUERY_STRING degiskeninin içine yazilir. Baska bir ifade ile Form'daki bütün degiskenlerin adlari ve bu degiskenin içerdigi deger birlestirilir.ve Server'da QUERY_STRING degiskeninin degeri olarak yazilir. Form'un bilgileri POST yoluyla aliniyorsa bunlar Request nesnesinin Form kolleksiyonunun içinde Form'un degisken adlari ve ziyaretçinin bu degiskenler için sagladigi degerler olarak ayri ayri yazilir. GET ile sinirli, POST ile sinirsiz bilgi alinabilir.
Script_Name: O anda çalistirilmakta olan ASP sayfasinin adini verir.

4.6. ServerVariables (Server Degiskenleri)

Request nesnesinin bir diger kolleksiyonu, Web Server'in o anda çalismakta olan ASP sayfasi için olusturdugu ortamin degiskenleridir. Bunlarin arasinda ziyaretçinin Browser'ina iliskin bilgiler de vardir.Su kisa ASP sayfasi çalistirilirsa Server'in su andaki degiskenleri görülebilir.(SerDeg.asp):

```
<HTML>
<HEAD>
<TITLE>HTTP ServerDegiskenleri Kolleksiyonu</TITLE>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<EAD>
<BODY BGCOLOR=white>
<CENTER>
<H2>HTTP Server Degiskenleri Kolleksiyonu</H2>
</CENTER>
<TABLE BORDER=1>
<TR><TD><B>Degiskenin adi</B></TD> <TD><B>Degeri</B></TD></TR>
<% For Each key in Request.ServerVariables %>
                <TR>
```

```
                <TD><% = key %></TD>
                <TD>
                <%If Request.ServerVariables(key) = "" Then
                        Response.Write " "
                Else
                        Response.Write Request.ServerVariables(key)
                End If
                Response.Write "</TD>"%>
                </TR>
<% Next %>
</TABLE>
<p>
Hostadi:<B><%=Request.ServerVariables("HTTP_HOST")%></B>
</BODY>
</HTML>
```

4.7. Cookie (Çerez)

HTTP ile yapilan iletisim, belirgin olmayan durum baglantisina dayanir.Yani ne istemci sunucunun, ne de sunucu istemcinin o anda hatta (on-line) oldugunu bilmek zorunda degildir; birbirlerinden istenilen ve gönderilen seyleri karsi tarafindan almaya hazir oldugu bilinmeden gönderilir. Örnegin elektronik alisveris gibi ziyaretçinin bir yerlere bir seyler kaydettigi, geçici degiskenler olusturdugu durumlarda sitede kimin ne yaptigi bilinmek zorundadir. Ziyaretçinin bir sayfada yaptigi tercihler diger sayfalarda ona sunulan içerigi etkileyebilir, belirleyebilir. Oysa ayni ziyaretçi bir sayfadan digerine geçerken Server ile iliskisini kaybedebilir. Bunun için ziyaretçinin Internet'ten kopmasi gerekmez; sadece TCP/IP protokolü geregi baglantisi kesilebilir. Bunu ziyaretçi fark etmeyebilir; ama Server etmelidir. Her yeni iliskiye yeni bir "application" (uygulama programi) baslatilamaz; ziyaretçinin bir önceki sayfada yaptigi tercihlerin devam etmesi gerekir. Bu devamlilik ziyaretçi isaretlenerek yapilir; bu isareti de Cookie saglar.

ASP teknigiyle tasarlanan sitede, ziyaretçilere Cookie göndermek zorunludur.ASP'de iki ayri grup Cookie nesnesi olusturulur.Bunlar verilenler ve hakkinda bilgi alinan varolan Cookie'ler. Birinci grup Request (talep) nesneleri, ikinci grup ise Response (karsilik) nesnelerini içinde ele alinabilir.

4.8. Sertifika Nesnesi

Sertifika, HTTP baglantisinda istemcinin taninmasini saglar. Bir yazilimdan ibaret olan sertifikalari yetkili bir kurum veya sirket verir; bir seri numarasi olur. Sifreleme teknikleri gelismis oldugu için taklit edilmesi zordur. Sertifika uygulamasi için Web

Server'in Secure Socket Layers denen güvenli HTTP protokolünü kullanmasi gerekir. Bu durumda Server'in URL'i," http:// "olarak degil "https:// "olarak yazilir.

ASP için sertifika ile ilgili her türlü bilgi Client Certificate kolleksiyonunda durur. Örnegin bir ziyaretçinin siteye girmeye yetkili olup olmadigini anlamak için:

```
<%SertifikaNo = Request.ClientCertificate(SerialNumber) %>
```

yazilan kodla istemcinin Sertifika seri numarasi SertifikaNo degiskenine atanir ve daha sonra bu degiskenin degeri bir liste ile karsilastirilabilir.

4.9. Karsilik (Response) Nesnesi

Web Server'in Çikti (Output) sagladigi,istemciye giden karsiliklarin olusturuldugu nesnedir. Server'dan Browser'a giden her sey karsiliktir. Bu bir ASP veya HTML sayfasi olabilir, sayfanin içindeki GIF, JPG veya PNG grafigi, bir Flash, video veya ses dosyasi olabilir. Zengin bir içerigi olmakla birlikte Response nesnesinde koleksiyon olarak sadece Cookie'ler vardir.Fakat Response nesnesinin bir çok özelligi ve metodu vardir.

4.9.1. Response Nesnesinde Cookie'ler

Örnegin ziyaretçinin bir Form'a yazdigi adi,soyadi ve elektronik posta adresini Cookie olarak onun bilgisayarina kaydettigimizde ziyaretçi isaretlenmis olur.Bir ziyaretçi sayfayi talep ettigi anda, Cookie'den bu kisinin adi ögrenilir ve sayfada ismi görüntülenebilir.

Ziyaretçinin daha önce sitede ziyaret ettigi sayfalar veya elektronik ticaret sitesinden satin aldigi kitap türleri Cookie'ye kaydedilip,ikinci ziyaretinde ona önce bu sayfalarin veya kitaplarin köprüleri sunulabilir.

Bir Cookie'nin adi ve anahtarlari (key) ile bu anahtarlara karsilik degerler olur. Örnegin:

```
<% Response.Cookie("Bizim_Cerez")("Adi_Soyadi")= "Mehmet Can" %>
```

Bu, ziyaretçinin Browser'ina (yani sabit diskine) "Bizim_Cerez" isimli bir Cookie gönderir; bu Cookie'nin "Adi_Soyadi" adli bir anahtari vardir; bu anahtarin degeri ise "Mehmet Can" olur.

Cookie kolleksiyonunun bir özelligi ise zaman asimi süresidir .

Expires (zaman asimi süresi): Bir cookie'nin artik geçersiz oldugu tarihi gösterir.:

```
<% Response.Cookie("Bizim_Cerez").CookieExpires = "August 7, 2000" %>
```

Bunun anlami Cookie'nin 7 Agustos 2000 tarihinden sonra kullanilamayacagidir.

4.9.2. Metodlar

Response nesnesinin bir çok metodu vardir; bunlardan .Write en sik kullanilanidir:

```
<%
DIM Adi_Soyadi
Adi_Soyadi = “Gülden ve Ülkü”
Response.Write(“Merhaba, Bizim Isimlerimiz, “ & Adi_Soyadi)
%>
```

örnegi ile ziyaretçinin Browser penceresine “Merhaba, Bizim Isimlerimiz, Gülden ve Ülkü” yazdirilmis olunur. Fakat VBScript’in “esittir metodu” ile bu islem daha da kolaylasir:

```
<%
DIM Adi_Soyadi
Adi_Soyadi = “Gülden ve Ülkü”
%>
<%= “Merhaba, Bizim Isimlerimiz , “ & Adi_Soyadi %>
```

4.9.3. Özellikler

Response nesnesinin bir çok özelligini kullanarak ziyaretçiye gönderilecek sayfalar ve diger unsurlar yönetilir:

Buffer (Tampon) : True (dogru) olarak ayarlandiginda ziyaretçiye gönderilecek sayfanin bütün unsurlari bir tampon bölgede toplanir, Script’in çalismasi bitinceye kadar beklenir ve HTML sayfa toptan gönderilir. Kimi zaman ASP kodumuz sonuna kadar çalistiginda ziyaretçi baska bir sayfaya ve siteye yönlendirilebilir. Bu gibi sebeplerle, özellikle çok islem gerektiren ASP sayfalarinin bas tarafina bunu saglayan kodu koymak gerekir:

```
<%
Option Explicit
Response.Buffer = TRUE
%>
```

Flush (hemen gönder): Buffer metodu sayfanin tümünü, Script’in icrasi bitmeden gönderilmesine izin vermez. Flush bunun tam tersini yaparak o ana kadar icra edilmis kodun sonucu olan HTML’i Browser’a gönderir:

```
<%
Option Explicit
Response.Flush
```

```
%>
```

Clear (Bosalt): Buffer metodu ile Script'in sonunu beklerken geçici bir alanda tutulmakta olan HTML, Clear metodu ile temizlenir, yok edilir. Flush metodunda tampondaki HTML Browser'a gönderilir; ancak Clear metodu tampon bölgedeki her seyi yok eder.Kullanilma sebebini bir örnek ile açiklarsak:ziyaretçinin elektronik alisveris sitesinde alisveristen vazgeçtigini belirttigi zaman tampon bölgede tutulmakta olan ve alinan mallar listesini içeren HTML bu yöntemle temizleriz:

```
<%
Option Explicit
Response.Clear
%>
```

Expires (Süresi dolar): Kullanici tarafindan tersine bir ayar yapilmadiysa, Browser genellikle görüntüledigi sayfalari Geçici Internet Dosyalari dizinine (cache) kaydeder ve tekrar ayni sayfa görüntülenmek istendiginde sayfayi Internet'ten edinmek yerine sabit diskinden alir. Oysa bu,özellikle haber gibi süreli bilgilerin sunuldugu Web sitelerinin itibarini sarsar. ASP teknigiyle bu önlenebilir. Bu sayfanin sözgelimi 60 dakikadan fazla cach dizinde tutulmasini önlemek için Expires metodu kullanilabilir:

```
<%
Option Explicit
Response.Expires = 60
%>
```

"Expires = 0" sayfanin hiç saklanmamasini saglar.Yani buradaki rakam degistirilerek süre ile oynanabilir.

End (Son): End metodu ile, Response nesnesinin o anda icra edilmekte olan Script'i durdurarak, o ana kadar ne elde edilmisse hepsini Browser'a göndermesini saglar.Ayni zamanda Buffer metoduyla tutulan HTML'in de gönderilmesine yol açar. Bu metoddan sonraki HTML veya ASP kodlari icra edilmez:

```
<%
Option Explicit
Response.End
%>
```

4.10 Uygulama(Application) ve Oturum (Session) Nesneleri

ASP ile Web Server her bir Web ziyaretçiyi oturumun basindan sonuna izleyebilir.CGI ise bu konuda yetersiz kalmistir. ASP açisindan, bir site "uygulama programi" (Application) sayilir. Her ziyaretçi de bir "oturum" (Session) sayilir.

Application nesnesi, sitenin tümüyle ilgili bilgileri (degiskenleri, nesneleri ve metodlari) tutar; Session nesnesi ziyaretçinin siteye girmesinden itibaren izini sürer. Örnegin bir borsa sitesinde ziyaretçiler tarafindan satisa sunulan hisse senetlerinin degerlendirmeleri okunur ve çesitli talimatlar verilir. Bütün ziyaretçilerin erisecegi bir tane veritabani vardir fakat her bir ziyaretçi farkli tercihler yapmaktadir. Application nesnesi ile, site ile veritabanina erisme, alisverisler, sitede yapilan bütün islerin kurallari bilinir ve uygulanir; Session nesnesi ise sahislarin alisverisleri, tercihleri bilinir.

Bir ASP sayfasinda, herhangi bir degisken fonksiyon disinda tanimlanip deger atanirsa bütün fonksiyonlar için geçerli hale getirilir. Bir fonksiyonun degerinin bütün sayfalarda ayni olmasi istenirse ,degiskenler Session nesnesi için olusturulur ve bu deger ziyaretçinin oturumu boyunca devam eder; bütün ASP sayfalarindaki bütün Fonksiyonlar tarafindan bilinebilir. Örnegin:

```
Session ("Tupras") = 44500
```

bütün Session için geçerli bir Tupras degiskeni olusturulur ve ona "44500" degeri atanir. Kimi zaman, degiskenin çok daha genis kapsamli olmasi, yani ömrünün Session ile degil bütün Application boyunca belirli olmasi istenir. O zaman bu degisken Application düzeyinde tanimlanabilir:

```
Application ("Tupras") = 44500
```

Bu durumda Tupras degiskeni bütün ziyaretçiler için ayni degere sahip olur.Session nesnesinin olusabilmesi için, ziyaretçiye Cookie gönderilerek bir isaret verilmesi gerekir. HTTP ile kurulan baglanti, belirsiz durum baglantisidir.Yani Server bir ziyaretçiye arzu ettigi sayfayi gönderdikten sonra, onu alip almadigini, o sayfada ne tercihler yaptigini bilemez. Fakat ziyaretçiye siteye baglandigi anda bir Session kimligi verilirse ve her yeni sayfa talebinde bu kimlik kontrol edilirse, kimin hangi oturumu sürdürdügü bilinir. ASP uyumlu bir Web Server varsa, yeni bir tercih yapilmadigi taktirde her Session nesnesi 20 dakika açik tutulur sonra silinir. Bu süre Session nesnesinin Timeout özelligi ile degistirilebilir. Session belirleyen Cookie, ASP-uyumlu Web Server tarafindan otomatik olarak gönderilir ve takip edilir tasarimcinin bu konuda bir sey yapmasina gerek yoktur.

Bir Web programina ayni anda kaç kisi ulasirsa o kadar. Global.asa dosyasi Appcation nesnenin bütün Session'lar için sitenin ihtiyaçlarina uygun ve ayni uygulama kurallarina sahip olmasini saglar.Bu dosya PWS veya IIS kurulurken olusturulur. ASP ile Web programlari, örnegin MS Visual Studio ile olusturuluyorsa, program seçilen dizinde bir Global.asa dosyasi olusturulur. Bu dosyanin içeriginde, siteye ilk ziyaretçinin gelmesiyle olusan Application_OnStart ve son ziyaretçinin çikmasiyla olusan Application_OnEnd ile herhangi bir ziyaretçinin bir sayfaya erismesiyle olusan Session_OnStart ve ziyaretçinin sitemizden çikmasi ile olusan Session_OnEnd olaylari halinde ne yapilacagi yazilidir.Dosya uzantisinin .asp degil de .asa olmasinin sebebi, dosyanin Active Server Application dosyasi olmasidir. ASP-uyumlu bir Web Server programi tarafindan ilk ziyaretçiyi ilk ziyaretçi siteye ulastigi anda Global.asa dosyasi çalistirilir.

Application ve Session nesneleri en çok siteye gelen ziyaretçilerin sayisinin tutulmamasinda kullanilir. Bu genellikle Global.asa programina bir sayaç yerlestirilerek yapilir.

4.11. ActiveX Veri Erisim (ADO) Nesneleri

ASP ile veri erisimi çok kolaylasmistir.ADO Server Component'i (sunucu bileseni) dir. Bu bilesene ASP içinden bir ActiveX nesnesi ile ulasilir.

Veritabaninin site güncellestirme islerini kolaylastirmasi sebebiyle Web Programlarinin temelini olusturmaktadir. Sayfalarin unsurlari veritabani dosyasindan alinir; ziyaretçilerin verdikleri bilgiler veritabanina yazilir.VBScript kodu hiç degismeden kalir ve veritabani dosyasinda ilgili verinin alindigi alana yeni degerler girilerek sayfanin sürekli güncel tutulmasi saglanir.Veritabani dosyalarinin idaresi kolaydir .ASP sayfalari ile Access, Excel, Paradox, FilePro, SQL Server ve Oracle veritabanlarina ve spreadsheet dosyalarina erisilebilir; bu dosyalardan veri okunur ve bu dosyalara veri yazilabilir. ASP programlari ile, SQL-uyumlu veya Windows ve diger sistemler için yazilmis ODBC (Open Database Connectivity/Açik Veritabani Baglantisi) ile uyumlu her türlü dosyaya, ADO nesnesi araciligiyla ulasilabilir.

4.11.1. ODBC ve OLE-DB

ODBC ve OLE-DB ADO'nun kullandigi sistemlerdir. ODBC, iliskilendirilmis (relational) veritabanlarina erismek üzere tasarlandigi halde OLE-DB her türlü veritabanina erisebilir. OLE-DB, ASP programlarina yeni nesneler kazandirabilir; kullanilmaya hazir elektronik ticaret bilesenlerini kullanmaya imkan verir.

Bitirme ödevinde ODBC kullanildi. OLE-DB, ODBC'nin yerini almakla birlikte içinde ODBC'yi bulundurur.ODBC'li ASP uygulamalari OLE-DB teknigi ile çalisan sunucularda da isleyebilir.

Bir örnekle açiklanirsa:Bilgisayarda kurulu bir veritabani programini kullanarak(burada access kullanildi) bir veritabani dosyasinda kayitlar.mdb adiyla bir dosya olusturulur.Bu dosya, kisisel Web Server'in kök dizinine kopyalanir. Denetim Masasi açilir ve adi ODBC, ODBC 32 Bit, ya da ODBC Data Source olan simge çalistirilir; ikinci sekme olan System DSN tiklanir.

Açilan kutuda Add/Ekle dügmesi tiklandiginda, veriyi okumakta kullanilan sürücünün seçildigi kutu açilir ve olusturulan veri dosyasinin sürücüsü seçilir kayitlar.mdb dosyasi için birinci seçenek olan Microsoft Access Driver seçilir. Son dügmesi tiklanir. Buradaki Data Source Name (DSN, Veri Kaynak Adi), daha sonra ADO nesnesiyle ilgili metodlar ve deyimler yazilirken kullanilan veri adidir; buraya "kayitlar" yazilir; çünkü örneklerde bu veriye "kayitlar" adiyla gönderme yapilacaktir. Istenirse, Description/Açiklama bölümüne veritabaninin niteligini belirten bir kaç kelime yazilabilir. Select/Seç dügmesini tiklanarak açilan diyalog kutusu yardimiyla veritabani dosyasi kopyalandigini yerde bulunur; OK/Tamam'i tiklanarak, veritabani seçme islemi tamamlanir.

DSN olusturma kutulari sirasiyla OK/Tamam dügmeleri tiklanarak kapatilir; böylece "kayitlar" verisi, su andan itibaren bütün Web uygulamalarida hizmete girmis olur.
Veritabani dosyasi Internet sitesinde kök dizinine veya bir diger dizine kopyalandiktan sonra sistem yöneticisine ya elektronik mektupla, ya da evsahibi firmanin yönetim ve teknik destek yardimi saglayan sayfasinda veritabaninin dosya adini, yolunu, ve DSN olarak kullanmak istenilen ismi bildirerek,ayni islemleri Server yöneticisinin yapmasi saglanir.

ADO'nun nesneleri ve metodlari açiklanirsa:

4.11.2. Connection (Veritabanina baglanti)

ADO'nun ilk nesnesi Connection'dir. Bu nesne veritabani ile baglanti saglar:

```
<%
Dim Veriyolu
Set Veriyolu = Server.CreateObject("ADODB.Connection")
Veriyolu.Open "Veri_adi"
%>
```

Server'in CreateObject metodu ile ADOBD.Connection nesnesi olusturulur. Olusturulan baglantiya istenilen degisken adi verilebilir.Veriye kurulan bu baglanti Veriyolu adini almistir. Bu yolla saglanan veriler, ASP programi boyunca bir isimle bilinir.Veriyolunun açacagi veri kümesinin ismi "Veri_adi" kelimelerinin yerine yazilir. Bu isim, baglantinin .Open metodu ile açacagi verinin adidir. Bu, kullanilan veritabani dosyasinin adi degildir. Bu isim ile veritabani dosyasini isletim sisteminin ODBC aracina tanitirken kullanilan isim ayni olmalidir. ASP programi, Server'dan ADO araciligiyla, sistemin "Veri_adi"yerine yazilan isimli veriye yol açar.

4.11.3. Recordset (Kayit dizisi)

Veritabanina baglanti olusturulduktan sonra verilerin kullanilir kayitlar haline getirilmesi ADO'nun Recordset nesnesi ile saglanir. Kurulan veri yolundan programa bilgi gelmesi için .Execute (icra et) metodu kullanilir; fakat bu komuta icra edecegi bir komut verilmelidir.

ADO ile kullanilabilecek veritabaninin SQL (Structured Query Language/Yapisal Sorgu Dili) uyumlu olmasi gerekir. Bu dil, verilerin sabit diske yazilmasi ve okunmasini düzenleyen bir çok veritabani dilinden sadece biri, fakat en yayginidir. Bir veritabanindan veri okumak, veri degistirmek veya eklemek için komutlarin bu dille verilmesi gerekir.

ASP amaciyla SQL komutlarindan çok az kismini kullanilir; bu bakimdan ASP sayfasi olusturmak için sinirli da olsa SQL bilmek gereklidir.

4.11.4. SQL

ASP amaçli olarak kullanilacak komut sadece SELECT'tir. Fakat veritabani ilkeleri bilinmelidir. Bir veritabani kabaca alanlar (sütunlar) ve bunlarin içinde yazili degerler (satirlar) den olusur; her satir bir elemanin degerleridir; ve Kayit adini alir.

SELECT:

Bir veritabanindan veri seçmeye yarar. SQL Sorgusu da denir. Dört bölümü vardir. Tipik bir
SELECT komutu söyle yazilir:

```
SELECT alan1, alan2.. FROM ORDER BY alan1 tablo WHERE kosul = deger
```

Seçilen alanlarin adi SELECT komutunun ilk bölümünü olusturur. Bir veritabaninda birden fazla tablo bulunabilir; seçimin hangi tablodan yapilacagi FROM bölümünde gösterilir.Bir alandaki degerlerin verilen bir kosula uymasi istenirse "kosul = deger" testi WHERE bölümünde yapilir. Seçilen degerlerin hangi alandaki kayitlara göre siralanmasini isteniyorsa, ORDER BY bölümünde belirtilir.

Bir tablodaki bütün alanlarin bütün degerlerinin seçilmesi için SELECT komutu asagidaki gibi yazilir:

```
SELECT * FROM Veri_adi
```

Buradaki "Veri_adi" yerine DSN'e verilen ad(örnegin yukaridaki örnekte oldugu gibi, "kayitlar" kelimesi)
yazilir.

Connection metodu kullanilirken yazilan kod gelistirilirse:

```
<%
Dim Veriyolu, Kayitdizisi
Set Veriyolu = Server.CreateObject("ADODB.Connection")
Veriyolu.Open "Veri_adi"
Set Kayitdizisi = Veriyolu.Execute("SELECT * FROM Veri_adi")
%>
```

.Execute metodu ile, DSN'ini verilen kaynaktaki veritabanindan veriler alinir ve bir Recordset (Kayit dizisi) olusturulur. .MoveNext (bir sonrakine git) metodu ile kayitlar tek tek okunur ve kayitdizisi bir sonraki kayda gider. Okunan her kayit Kayitdizi adli degiskenin içindedir..Movenext metodu ile bir sonraki kaydin okunmasi saglanir.

4.11.5. Recordset.Open

Veritabanina dayanan Web uygulamalarinda sorun buradaki gibi sadece veriyi okumakla bitmeyebilir; veriyi güncellestirmek veya silmek istenebilir. Bunun için dogruca ADO'nun .Recordset metodundan yararlanilmalidir.Recordset metodu tipki ekrandaki bir yazinin içinde duran imleç (cursor) gibi hayalî bir imleci götürür verilerin en basina koyar. Bu hayali imleci veritabani üzerinde dolastirmak ve gittigi yerdeki degeri okutmak programcinin isidir.

.Recordset metodu, ile bir veritabanini okuyacak imleç üç sekilde ayarlanabilir:

Static	(Duragan) SELECT komutu icra edilir ve okunan kayit arzu edilen degiskene yazilir. (ADO Sabit Degerleri dosyasindan yararlaniliyorsa, adOpenStatic)
Forward only	(Sadece ilerle) Imleç veritabani içinde sadece ileri dogru gider ve her seferinde bir kayit okunur. (Varsayilan imleç türü budur.) (ADO Sabit Degerleri dosyasindan yararlaniliyorsa, adOpenForwardonly)
Dynamic	(Dinamik) Veritabanina ulasan ve degisiklik yapan baska bir kullanici varsa, bu degisiklik programciya aninda yansitilir. (ADO Sabit Degerleri dosyasindan yararlaniliyorsa, adOpenDynamic)

Bu yöntemlerden birinin seçilmesiyle veri belirli bir okuma tarzinda açilmis olur. Bu yöntemlerden hangisinin seçildigi .Recordset metodunu kullanacak olan .Open komutunun argümani olarak açikça belirtilmelidir. ADO, bunun için sayilar halinde argümanlar ister.

4.11.6. ADO Sabit Degerleri

ADO+ODBC yoluyla kurulan veri baglantilari, çogu zaman adeta sifreli ifadeler içerebilir ve bir çok komutun argümani ögrenmesi zor sayilar halinde verilir. Microsoft ve kullanilmaya hazir ASP Uygulamalari üreten firmalar, bu karmasik ifadeleri düz metinler olarak ifade etmeye yarayan haricî dosyalar (include files) hazirlar ve sunarlar. Bunlar arasinda en yaygin olani Microsoft'un ADOVBS (adovbs.inc) dosyasidir. Bu dosyadan yararlanabilmek için, siteye kopyalanmasi ve daha sonra sayfalara su kodlarin eklenmesi gerekir:

```
<!- - #include file="adovbs.inc" - - >
```

Bu dosya, Server tarafindan icra edilir ve ADO nesnesinin sayi halindeki bütün argümanlari anlasilabilir Ingilizce kelimelere çevirir.

Bir veriye baglanti kurulduktan sonra kayit dizisi .Recordset metodu ile saglanacaksa, yukaridaki örnek kodu söyle yazilmalidir:

```
<!- - #include file="adovbs.inc" - - >
<%
Dim Veriyolu, Kayitdizisi, Sorgu
Set Veriyolu = Server.CreateObject("ADODB.Connection")
Veriyolu.Open "Veri_adi"
```

```
Set Kayitdizisi = Server.CreateObject("ADODB.Recordset")Sorgu = "SELECT * FROM
Veri_adi"Kayitdizisi.Open Sorgu, Veriyolu, aOpenStatic
%>
```

Bu kod ile, .Recordset metodu, son .Open komutu ile veri baglantisini saglar; verilen SQL Sorgusu icra edilir ve kayit dizisi Kayitdizisi'ne kaydedilmeye hazir hale gelir.Su durumda imleci ilerleterek, veriyi fiilen okutmak gerekir;ki bu yukarda kolayca .Execute metoduyla olusturulan kayit dizisinde kullanilan basit .MoveNext'ten daha çok imkan saglar:

MoveFirst:	Kayit dizisinin (Recordset'in) birinci satirina gider.
MoveLast:	Kayit dizisinin (Recordset'in) son satirina gider.
MoveNext:	Kayit dizisinin (Recordset'in) bir sonraki satirina gider.
MovePrevious:	Kayit dizisinin (Recordset'in) bir önceki satirina gider.
Move:	Kayit dizisinin (Recordset'in) içinde verilen sayiya göre ilerler.

Bunun için iki sayi vermek gerekir(baslangiç noktasi ve ilerlenecek kayit sayisi)

4.11.7. Recordset.Update

Veritabanindan alinan degerlerin, kimi zaman ziyaretçinin verecegi degerlerle veya ziyaretçinin bir takim tercihleri sonucu güncellestirilmesi gerekebilir. Bu Recordset nesnesinin .Update metodu ile kolayca yapilir.

Recordset'in .Open metodunun imleçlerinden söz edilirken, okumanin yönünü veya imlecin hareket tarzini belirleyen argümanlar siralanmisti. Bu argüman dizisine bir yenisi eklenerek, veritabanina erisimin niteligi ve güncellestirmenin nasil yapilacagi ve yansitilacagi da belirlenebilir. Bu islemin temel ilkesi veritabani kayitlarinin kilitlenmesi esasidir. Bu kilitlemenin türünü belirleyerek, güncellestirmenin de nasil yapilacagi belirlenmis olur. Burada kullanilan argümanlar da ADO'nin sifreli sayilari olmasi gerekirken, adovbs.inc dosyasi sayesinde Ingilizce (ve dolayisiyla anlasilabilir) kelimeler olur. advbs.inc dosyasini devreye sokulduysa, su iki tür kilit kullanilabilir:

adLockReadOnly	Kayitlarin güncellestirilmesini önler; ziyaretçi veritabanina kayit yapmayacaksa, bu kilit türünü kullanmak gerekir.
adLockOptimistic	Veritabanina ek yapilacaksa, mevcut kayitlar düzeltilecekse ve bazilari silinecekse, bu kilit türü kullanilmalidir.

Yukaridaki kod örneginin sadece son satiri, bu metodu kullanmak amaciyla, söyle yazilabilir:

```
Kayitdizisi.Open Sorgu, Veriyolu, aOpenStatic, adLockOptimistic
```

Veritabanini güncellestirmek için imleç veritabaninda dogru kaydin üzerine götürülmek ve bu arada Recordset'in sagladigi mevcut verilerin yerine yeni degerlerin

atanmis olmasi gerekir. Bu saglandiktan sonra bütün yapilacak sey .Update metodunu kullanmaktir:

```
Kayitdizisi(“Adi”) = “Hasan Hüseyin”
Kayitdizisi(“Soyadi”) “Balik”
Kayitdizisi.Update
```

Bu komut, imleç o sirada hangi kaydin üzerinde ise o kaydin “Adi” ve “Soyadi” alanlarindaki veriyi “Hasan Hüseyin” ve “Balik” haline getirir. Bu metod kullanilirken bir kaydin bütün alanlarinin güncellestirilmesine veya güncellestirilmeyen alanlar eski degerlerinin ile tekrar edilmesine gerek yoktur.

4.11.8. Recordset.Delete

ADO ile bir veritabanindaki kaydi silmek de oldukça kolaydir. Imleç, silinecek kaydin üzerine götürüldükten sonra, Recordset’in , .Delete metodu çagirilarak o andaki kayit silinir. Bu metod, bir kaydin bütün alanlarindaki degerlerle birlikte (yani veritabaninin bir satirini tümüyle) siler:

```
Kayitdizisi.Update
```

4.11.9. Recordset.AddNew

Bir veritabanina yeni kayit eklemek istendiginde, Recordset’in .AddNew (yeni ekle) metodundan yararlanilir. Bu metodun özelligi kullanicinin imleci veritabani içinde bir yere götürme zorunlugunun olmamasidir. Bu metod kendiliginden imleci dosyanin en son satirinin altina götürür. .AddNew metodu bir veritabani dosyasina kayit eklerken, veritabaninda mevcut bütün alanlar için deger verilmesini ister. Örnegin

```
<%
Kayitdizisi.AddNew
Kayitdizisi(“Adi”) = “Hasan Hüseyin”
Kayitdizisi(“Soyadi”) “Balik”
Kayitdizisi(“TelNo”) = “0424-2128970“
Kayitdizisi.Update
%>
```

Burada veritabanina yeni kaydinin .Update metodu tarafindan yaptigina dikkat edilmelidir.

4.12. DSN’siz Veri Baglantisi

Aslinda DSN olmadan da veritabanlarina ulasilabilir,ancak isin dogrusu DSN yoludur. Fakat yine de DSN olusturmadan da veritabanina ulasabilecegi bilinmelidir.

DSN, genellikle Web Server'lari yavaslatir; Web Server, DSN'ini belirtilen veriye ulastirmak için önce ODBC'nin yardimini ister; ODBC, bir takim sürücüleri devreye sokarak veriye ulasir. Bir DSN-verisine 20-30'dan fazla kullanici ayni anda eristigi zaman bu yavaslama gözle görünür hale gelebilir. Bir süre öncesine kadar Microsoft firmasi, veriye dayanan Web sitelerinin veri-bagini DSN yoluyla kurmasini tavsiye ederken, simdi MS yayinlarinda sik sik DSN'siz veri baglantisinin da etkin sekilde kullanilacagi belirtmektedir.

Örnek:

```
<%
Dim Veriyolu, Kayitdizisi
Set Veriyolu = Server.CreateObject("ADODB.Connection")
Veriyolu.Open "Veri_adi"
Set Kayitdizisi = Veriyolu.Execute("SELECT * FROM Veri_adi")
%>
```

Böyle bir DSN baglantisinin kullanilabilmesi için, kullanici kisisel Web Server ortaminda Denetim Masasi'ndaki ODBC aracini kullanarak bir DSN olusturmali; Internet ortaminda ise bu adi verilen veritabani dosyasina DSN olusturulmasi için Web Server yöneticisinin yardimi istenmelidir. Oysa ayni islem DSN'siz veri baglantisi kurularak da yapilabilir. Bunun için, DSN'e yukaridaki gibi dogrudan göndermede bulunmak yerine; ya ODBC sürücüsüne ya da ODBC'nin kullandigi Microsoft Jet OLEDB sürücüsüne dogrudan atifta bulunulur. Örnek:

```
Veriyolu.Open "Veri="&Server.MapPath("..../veriler/kayitlar.mdb") & "; Driver = {Microsoft
Access Driver (*.mdb);"
```

Burada, DNS'siz baglanti için veritabani dosyasinin Server'daki göreli yeri, adi ve hangi sürücünün kullanilacagi belirtilir. Ayni baglanti, dogruca Jet sürücüsü için de yazilabilirdi.

```
Veriyolu.Open "Veri="&Server.MapPath("..../veriler/kayitlar.mdb") & ";
Provider=Microsoft.Jet.OLEDB4.0;"
```

Ancak buradaki sorun kullanilan veritabani dosya türüne uygun Microsoft Jet sürücüsünün seçilebilmesidir.

4.13. HTML Etiketlerinin Veri ile Doldurulmasi

Su ana kadar ADO nesnesi taninmis ve metodlari görülmüstür. ADO'nun veritabani ile DSN ile ve DSN'siz nasil baglanti kuracagi ele alinmistir. Burada "doldurmak" kelimesi bir etiketin deger (value) bölümünü yazmak anlamina gelmektedir.. Burada bazi örneklerde ODBC konusunu ele alinirken olusturulan kayitlar.mdb DSN

olarak “kayitlar” verisi seklinde kullanilmistir.Ancak bir sinirlama yoktur yani istenilen veri kullanilabilir.

4.13.1. Seçme Kutulari: SELECT

SELECT, ziyaretçilere önceden belirlenmis bir çok unsurdan birini veya daha fazlasini seçme imkani veren bir etikettir. Ziyaretçi, seçimini SELECT’in OPTION’lari arasindan yapar. Seçenekler (OPTION), sahip olduklari degeri Server’a gönderirler. Genel yazim kurali su sekildedir:

```
<FORM ACTION=”...” METHOD=POST|GET>
<SELECT NAME=”metin”>
<OPTION VALUE=”deger1”>Tercih 1
<OPTION VALUE=”deger2”>Tercih 2
<OPTION VALUE=”deger3”>Tercih 3
</SELECT>
```

Bu Form’un gönder (Submit) dügmesi ile saglanan hareket (ACTION), seçilen degeri veya degerleri, Form’u isleyecek ASP programina göndermektir.

Ziyaretçiye sunulan seçenekler, iki-üç adet ise, bu HTML dosyasi yazilirken, OPTION’lar halinde kodlamak kolay olabilir. Ancak seçenek sayisi arttikça, veya seçenekler sik sik degistikçe, bunlari bir veri tabaninda toplamak ve OPTION degerlerini veritabaninin bir alanindan alarak ziyaretçiye sunmak çok daha kolay olur. Böylece ASP sayfasi degismeden kalir; sadece veritabani güncellestirilmis olur. Çogu zaman bu güncellestirme ziyaretçilerin yapacaklari ekleri veritabanina yazmasiyla saglandigi için, ortaya gerçekten dinamik bir Web Uygulamasi çikmis olur.

Örnegin, grubumuzun üyelerini gösteren yukarida olusturulan kayitlar.mdb (DSN’i uyeler olan veritabani) dosyasinin ad-soyad alanlari birlestirilerek, sayfada bir SELECT etiketinin OPTION’larina yazilmak istendiginde,bunun için önce sayfada kullanilan degiskenler asagidaki gibi tanimlanmalidir:

```
<%
Dim connVeriyolu, rsVeri, SQL
%>
```

Daha sonra, bu degiskenlerden veri ile ilgili olanlara .Connection ve .Recordset için gerekli ifadeler yazilmalidir. Veri ile çalisirken tasarimcinin degisken adlarina bakarak hangisinin .Connection, hangisinin .Recordset degerlerini içerdigini anlamasi zorlasabilir. Bu bakimdan degisken adlarinin önüne .Connection için olaninda conn, .Recordset için olaninda rs harflerini kullanmak yararli olabilir. Veritabanindan fiilen hangi verilerin çekilecegini gösteren SQL deyimi belirgin bir sekilde SQL degiskenine yazilmalidir:

```
<%
Set connVeriyolu = Server.CreateObject("ADODB.Connection")
SQL ="SELECT uyeAdi, uyeSoyadi FROM uyeler"
%>
```

Bu degerlere dayanan ve adina kayitlar denilen veri kümesi asagidaki :

```
<%
connVeriyolu.open "uyeler"
Set rsVeri=connVeriyolu.execute(SQL)
%>
```

Bu islemler sonunda içinde bütün üyelerin ad ve soyadini tutan bir dizi-degisken elde edilir. Veritabanindan veri satir-satir okunur. Birinci satirin okunmasi sirasinda bu degiskenin degerleri:

rsVeri (0) = üye 1'in adi
rsVeri(1) = üye 1'in soyadi

olur. Veritabanindan ikinci satirin okunmasinda ikinci üyenin adi ve soyadi, üçüncü satirin okunmasinda üçüncü üyenin adi ve soyadi bu degiskenlerin degeri olacaktir.Bu degerler bir SELECT etiketinin OPTION degeri olarak kullanilacaksa, bu islem ikinci satir okunmadan yaptirilmalidir.Yani;

```
<SELECT NAME="AdSoyad">
<% Do While Not uyeler.eof %>
<OPTION VALUE = "<%= rsVeri(0) & " " & rsVeri(1)%>"><%= rsVeri(0) & " " &
rsVeri(1)%>
</Option>
<%rsVeri.movenext
loop%>
</select>
<% rsVeri.close %>
</SELECT>
```

Do döngüsünün içinde iken veritabanindan alinan deger, herhangi bir degiskenin degeri gibi kullanilabilir. Burada verilerin uyeler dizisinin dosya sonuna (eof, End Of File) okunduguna dikkat edilmelidir.Yukaridaki kodlar bir Form içinde asagidaki gibi birlestirilebilir:

```
<%@ LANGUAGE="VBSCRIPT" %>
<% Option Explicit %>
```

```
<HTML>
<HEAD>
<TITLE>ASP SELECT DOLDURMA</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>
<%
' Degiskenleri tanimlayalim
Dim connVeriyolu, rsVeri, SQL
Set connVeriyolu = Server.CreateObject("ADODB.Connection")
SQL ="SELECT uyeAdi, uyeSoyadi FROM uyeler"
connVeriyolu.open "uyeler"
Set rsVeri=connVeriyolu.execute(SQL)
%>
<BODY>
Bu listeden bir üyenin adini seçiniz:
<SELECT NAME="AdSoyad">
<% Do While Not rsVeri.eof %>
<OPTION VALUE = "<%= rsVeri(0) & " " & rsVeri(1)%>"><%= rsVeri(0) & " " &
rsVeri(1)%>
</Option>
<%rsVeri.movenext
loop%>
</select>
<% rsVeri.close %>
</SELECT>
</BODY>
</HTML>
```

Bu sayfa option.asp adiyla kaydedilip sinanabilir. Alinan sonuç suna benzemelidir:

Burada yapilan seçim sonucu elde edilen deger Server'a gönderilebilir; ve sözgelimi ziyaretçinin seçtigi kisiye ait bilgiler kendisine ulastirilabilir.

4.13.2. Isaretleme Alanlari:

4.13.2.1. Input-Radio

INPUT etiketi türleri ziyaretçilerin önceden belirlenmis bir çok unsurdan birini veya daha fazlasini seçmelerine veya kendilerinin girdi yapmalarina imkan veren bir etikettir. INPUT türlerinden Radio ve Checkbox (isaretleme kutusu) veritabanindan çekilen degerlerle doldurularak ziyaretçiye sunulabilir. Ziyaretçi, seçimini radyo dügmelerinden veya isaret kutularindan birini isaretleyerek yapar.

INPUT etiketinin radyo dügmesi türünün genel yazim kurali söyledir:

```
<FORM ACTION=”...” METHOD=POST|GET>
<INPUT TYPE=”Radio” NAME=metin1 VALUE=deger1>
<INPUT TYPE=”Radio” NAME=metin1 VALUE=deger2>
<INPUT TYPE=”Radio” NAME=metin1 VALUE=deger3>
</SELECT>
```

Bu Form’un gönder (Submit) dügmesi ile saglanan hareket (ACTION), seçilen degeri Form’u isleyecek ASP programina gönderir. Bir Form’daki bir grup olusturan bütün radyo dügmeleri ayni adi alirlar, ki böylece ASP programina bir degisken için deger gönderilmis olur.

Örnegin, ziyaretçiden begendigi rengi seçmesi istenen bir grup radyo dügmesini sunan bir Form yapmak için asagidaki kodlar yazilip radyo.asp adiyla kaydedilebilir:

```
<%@ LANGUAGE="VBSCRIPT" %>
<HTML>
<HEAD>
<TITLE>ASP OPTION-RADIO DOLDURMA</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>
<%
' Degiskenleri tanimlayalim
Dim connVeriyolu, rsVeri, SQL
Set connVeriyolu = Server.CreateObject("ADODB.Connection")
SQL ="SELECT renk FROM renkler"
connVeriyolu.open "uyeler"
Set rsVeri=connVeriyolu.execute(SQL)
```

```
%>
<BODY>
<FORM><DIV ALIGN="center"><center><TABLE BORDER="0">
<TR>
<TD colspan="2" align="center"><h3>Renk</h3></TD>
</TR>
<% Do While Not rsVeri.eof %>
<TR><TD><INPUT TYPE="radio" VALUE="<%=rsVeri(0)%>" NAME="Radyo"></TD>
<TD><%=rsVeri(0)%></TD></TR>
<%rsVeri.movenext
loop%></TABLE></CENTER></DIV></FORM>
</BODY>
</HTML>
```

Burada "Radyo" isimli radyo dügmesine verilen degerler, ODBC'nin "uyeler" adiyla tanidigi veritabanindan alinmaktadir.Bu örnekte bir önceki örnekten farkli olarak ayni veritabanindaki farkli tablodan, renkler tablosundan ve sadece bir alanin, renk alaninin degerleri çekilir. Bu örnek programda da ziyaretçinin seçtigi degerleri Server'a gönderecek bir Gönder dügmesi yoktur. Ama içerik veritabanindan alinan degerlerle doldurduktan sonra, ziyaretçinin radyo dügmeleriyle yapacagi tercihler, tipki klasik HTML'deki gibi kullanilabilir; Server'a degisken olarak gönderilebilir; veya ziyaretçinin bilgisayarinda (client-side) herhangi bir Script tarafindan kullanilabilir.

4.13.2.2. INPUT-CHECHBOX

INPUT etiketinin ziyaretçiye isaretleyerek tercih imkani veren diger araci Checkbox (isaretleme kutusu) türüdür. Tipki radyo dügmesinde oldugu gibi veritabanindan çekilen degerlerle doldurularak ziyaretçiye sunulabilir. Ziyaretçi, seçimini isaret kutularindan birini isaretleyerek yapar.

INPUT etiketinin Checkbox türünün genel yazim kurali söyledir:

```
<FORM ACTION="..." METHOD=POST|GET>
<INPUT TYPE="checkbox" NAME=metin1 VALUE=deger1>
<INPUT TYPE="checkbox" NAME=metin1 VALUE=deger2>
<INPUT TYPE="checkbox" NAME=metin1 VALUE=deger3>
</SELECT>
```

Bu Form'un gönder (Submit) dügmesi ile saglanan hareket (ACTION) seçilen degeri Forma gönderecektir. Radyo dügmesi ile Checkbox'in arasindaki fark, ziyaretçinin ayni ismi tasiyan radyo dügmelerinden birini isaretleyebilirken; istedigi kadar Checkbox'a

isaret koyabilmesidir. Birden fazla Checkbox isaretlendigi taktirde Server'a "metin1=deger1, deger2.." seklinde bilgi gönderirler. (ASP programlama açisindan, bu degisken Request.Form nesnesinde Checkbox'in adini tasiyan kolleksiyonun içinde dizi-degisken olarak yazilir.)

Yukarida radyo dügmesi örnegi sadece Do döngüsüne ait kismini degistirilerek,Checkbox'a uyarlanabilir.Bu örnek yazildiktan sonra isaret.asp adiyla kaydedilebilir:

```
<% Do While Not rsVeri.eof %>
<TR><TD><INPUT TYPE="checkbox" VALUE="<%=rsVeri(0)%>"
NAME="Isaret"></TD>
<TD><%=rsVeri(0)%></TD></TR>
<%rsVeri.movenext
loop%>
```

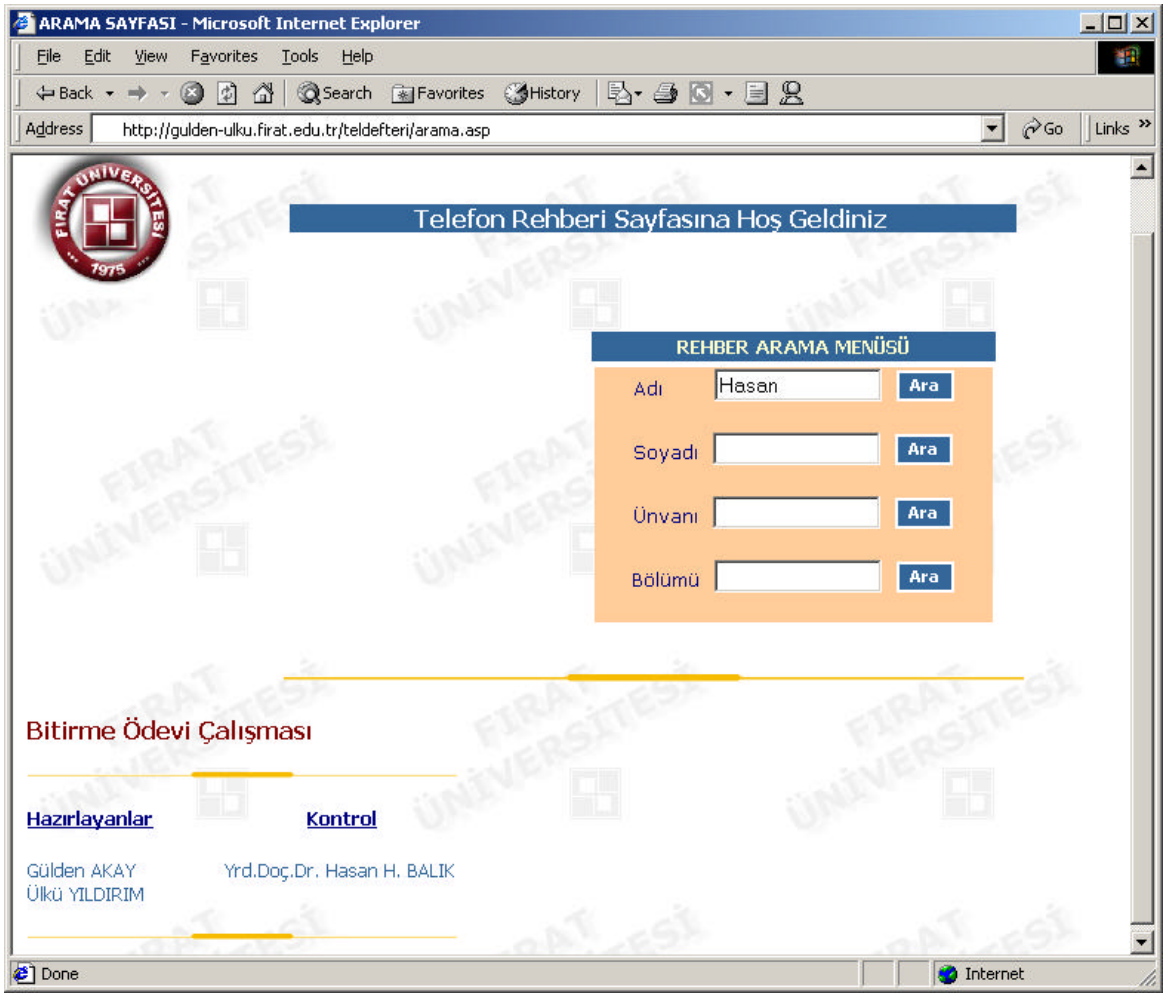
Ziyaretçinin bu form ile birden fazla kutuyu isaretleyerek Gönder dügmesine basmasi halinde, Server'a gelecek bilgi, örnegin, "Isaret=Kirmizi, Mavi" seklinde olur.

TELEFON REHBERI ÖRNEGI

ARAMA.ASP DOSYASI

Programin ana sayfasinda gelen menüde ilgili alanlardan sadece biri doldurularak arama islemi yapilabilir. Hangi kategoriye göre arama yapilmak isteniyorsa ilgili kutucuk doldurularak aratilir ve ilgili tüm kayitlar listelenir.

Ekrana gelen menülü ana sayfayi çalistiran dosya **arama.asp** dosyasidir. Bu dosyanin açik ve tam kodlari asagida verilmistir.



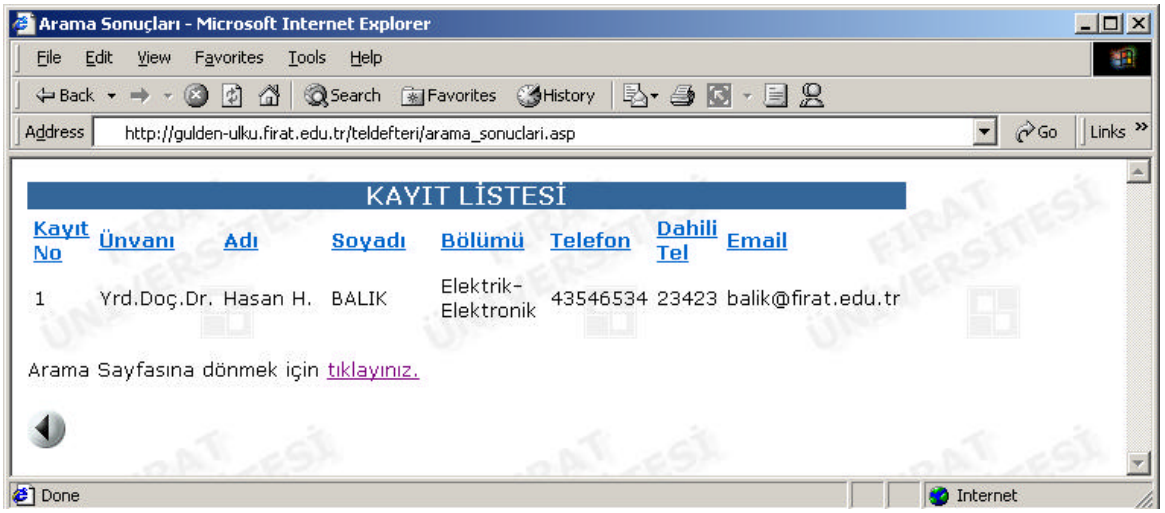

```
<html>
<head>
<title>ARAMA SAYFASI</title>
<meta http-equiv="Content-Type" content="text/html; charset=1254">
</head>

<body bgcolor="#FFFFFF" background="background.gif" bgproperties="fixed">

<div align="center">
  <center>
<br>
<img border="0" src="firat.jpg" width="105" height="104" align="left">
```

```
<p> </p>
<div align="center">
  <center>
<table border="0" cellspacing="2" cellpadding="0" width="495">
  <tr>
    <td bgcolor="#336699" align="center" width="489" valign="top">
      <div align="center">
        <p align="center"><font face="Tahoma" color="#FFFFFF" size="3"><b>Telefon
        Rehberi Sayfasina Hos Geldiniz</b></font></div>
    </td>
  </tr>
  <tr>
    <td width="489" align="center" valign="top">
      <div align="right">
      <table border="0" cellspacing="2" cellpadding="2" width="270">
        <tr>
    <td width="272">
    </td>
    <td width="272">
    </td>
    <td width="272">
    </td>
          <td valign="top" align="center" width="6"> 
            <p> </td>
        </tr>

        <tr>
    <td width="272">
    </td>
    <td width="272">
    </td>
    <td width="272" bgcolor="#336699">
      <div align="center">
        <p align="center"><b><font size="2" face="Tahoma" color="#FFFFCC">REHBER
        ARAMA
        MENÜSÜ</font></b></div>
    </td>
          <td valign="top" align="center" width="6"></td>
        </tr>
        <tr>
    <td width="272" align="left" valign="top" height="112">
    </td>
    <td width="272" align="left" valign="top" height="112">
    </td>
    <td width="272" align="left" valign="top" height="112">
      <table width="100%" border="0" cellspacing="0" cellpadding="0">
        <tr bgcolor="#FFFFCC">
          <td bgcolor="#FFCC99">
           <div align="center" style="width: 269; height: 117">
           <font size="2" face="Verdana">
            <form method="post" action="arama_sonuclari.asp" name="fname">
                <font color="#000080">
                Adi       <input type="text"
name="arama" size="15" maxlength="15">
                <input type="hidden" name="alan" value="Ad"> <input type="submit"
name="ara" value=" Ara " style="background-color: #336699; color: #FFFFFF; font-
size: 8pt; font-family: Tahoma; font-weight: bold; border-style: solid; border-
color: #FFFFFF">
                </font>
           </font>
            </form>
           <font size="2" face="Verdana">

            <form method="post" action="arama_sonuclari.asp" name="fname">
                <font color="#000080">
```

```
                Soyadi  <input type="text" name="arama" size="15"
maxlength="15">
                <input type="hidden" name="alan" value="Soyad">
                <input type="submit" name="ara" value=" Ara " style="background-
color: #336699; color: #FFFFFF; font-size: 8pt; font-family: Tahoma; font-weight:
bold; border-style: solid; border-color: #FFFFFF">
                </font>
            </form>

            <form method="post" action="arama_sonuclari.asp" name="fname">
                <font color="#000080">
                Ünvani  <input type="text" name="arama" size="15"
maxlength="15">
                <input type="hidden" name="alan" value="Unvan">
                <input type="submit" name="ara" value=" Ara " style="background-
color: #336699; color: #FFFFFF; font-family: Tahoma; font-size: 8pt; font-weight:
bold; border-style: double; border-color: #FFFFFF">
                </font>
            </form>

            <form method="post" action="arama_sonuclari.asp" name="fname">
                <font color="#000080">
                Bölümü  <input type="text" name="arama" size="15"
maxlength="15">
                <input type="hidden" name="alan" value="Bolum">
                <input type="submit" name="ara" value=" Ara " style="background-
color: #336699; color: #FFFFFF; font-family: Tahoma; font-size: 8pt; font-weight:
bold; border-style: double; border-color: #FFFFFF">
                </font>
            </form>
           </font>
           </div>
          </td>
        </tr>
      </table>
    </td>
          <td valign="top" align="center" width="6"></td>
        </tr>

      </table>
      </div>
       </td>
  </tr>
</table>
  </center>
</div>
  </center>
</div>
<p align="center” <img border="0" src="yellowline.gif"></p>
<p align="left"><font color="#800000" face="Tahoma" size="3"><b>Bitirme Ödevi
Çalismasi</b></font></p>
<p align="left"><img border="0" src="yellowline.gif" width="290" height="4"></p>
<p align="left"><b><font size="2" face="Tahoma"
color="#000080"><u>Hazirlayanlar</u>       &nb
sp;             
    
<u>Kontrol</u></font></b></p>
<p align="left"><font size="2" face="Tahoma" color="#336699">Gülden
AKAY             
; 
Yrd.Doç.Dr. Hasan H. BALIK<br>
Ülkü YILDIRIM</font></p>
<p align="left"><img border="0" src="yellowline.gif" width="290" height="4"></p>
</body>
</html>
```

ARAMA_SONUÇLARI.ASP DOSYASI

Aramaya baslarken **arama_sonuclari.asp** dosyasi çalismaya baslar ve aratilan kayit bulunduktan sonra asagidaki tablo seklinde ekrana gelir. Gelen sayfa Server üzerinde çalisan **arama_sonuclari.asp** dosyasidir. Bu dosyanin açik ve tam kodlari asagida verilmistir.

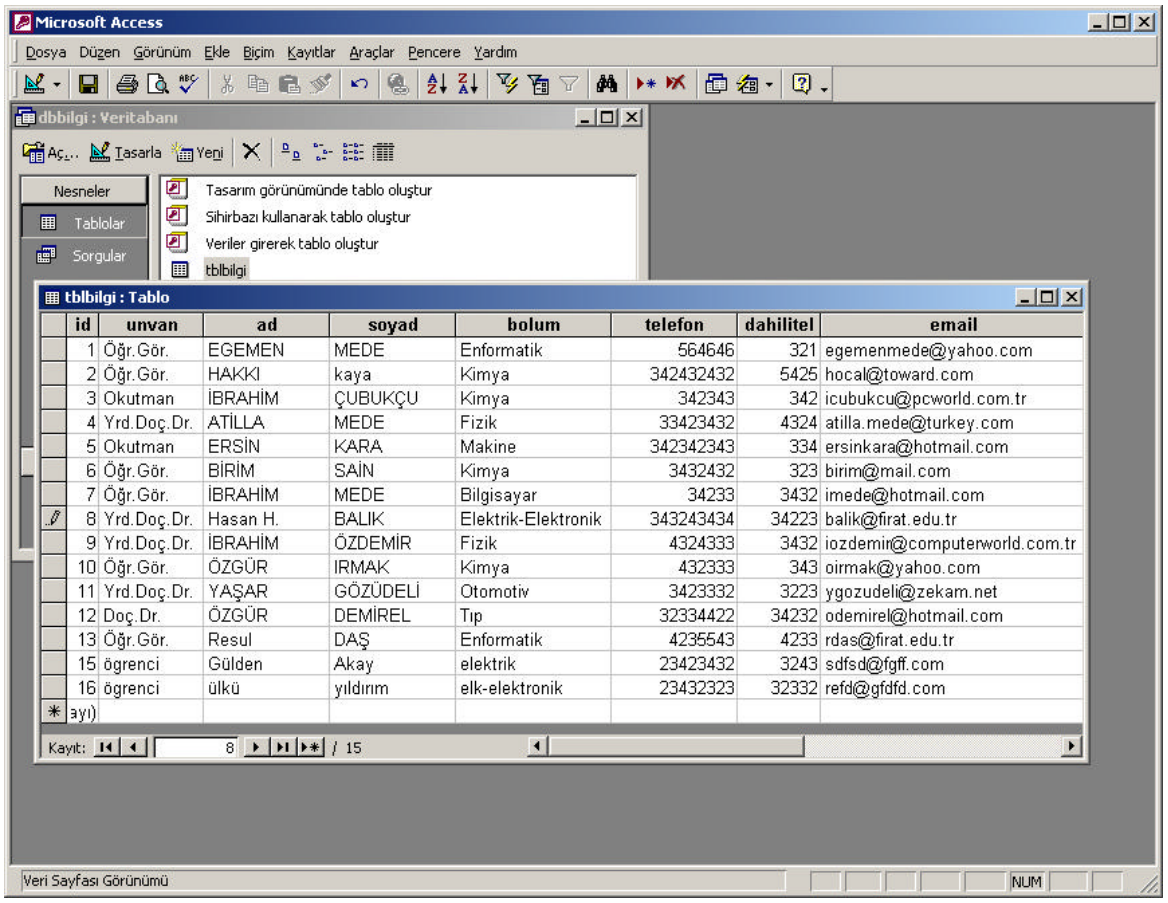



```
<html>
<head>
<title>Arama Sonuçlari</title>
<meta http-equiv="Content-Type" content="text/html; charset=1254">
</head>

<body bgcolor="#FFFFFF" background="background.gif" bgproperties="fixed">
<%
set conn=server.createobject("adodb.connection")
dsnpath="DRIVER={MICROSOFT ACCESS DRIVER (*.mdb)};"
    dsnpath=dsnpath & "DBQ=" & Server.mappath("db/dbbilgi.mdb")
conn.open dsnpath
if request("arama")="" then
   Response.Redirect("arama.asp")
end if
if request("alan")="Ad" then
   sql="SELECT * FROM tblbilgi where ad like '"&request("arama")&"%'"
end if
if request("alan")="Soyad" then
   sql="SELECT * FROM tblbilgi where soyad like '"&request("arama")&"%'"
end if
if request("alan")="Unvan" then
   sql="SELECT * FROM tblbilgi where unvan like '"&request("arama")&"%'"
end if
if request("alan")="Bolum" then
   sql="SELECT * FROM tblbilgi where bolum like '"&request("arama")&"%'"
end if
if request("arama")="*" then
   sql="SELECT * FROM tblbilgi"
end if

Set rs=conn.execute(sql)
tpl=0
say=1
%>
```

```
<table width="81%" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td colspan="4" bgcolor="#336699">
      <div align="center"><b><font face="Verdana" color="#FFFFFF" size="3">KAYIT
LISTESI</font></b></div>
    </td>
  </tr>
  <tr>
    <td width="85%" colspan="4" rowspan="2" align="left" valign="top">
      <table width="100%" border="0" cellspacing="2" cellpadding="2">
        <tr>
          <td width="10%"><b><font size="2" face="Verdana"
color="#0066CC"><u>Kayit No</u></font></b></td>
         <td width="10%"><b><font size="2" face="Verdana"
color="#0066CC"><u>Ünvani</u></font></b></td>
          <td width="15%"><b><font size="2" face="Verdana"
color="#0066CC"><u>Adi</u></font></b></td>
          <td width="15%"><b><font size="2" face="Verdana"
color="#0066CC"><u>Soyadi</u></font></b></td>
         <td width="15%"><b><font size="2" face="Verdana"
color="#0066CC"><u>Bölümü</u></font></b></td>
          <td width="12%"><b><font size="2" face="Verdana"
color="#0066CC"><u>Telefon</u></font></b></td>
         <td width="10%"><b><font size="2" face="Verdana"
color="#0066CC"><u>Dahili Tel</u></font></b></td>
          <td width="40%"><b><font size="2" face="Verdana"
color="#0066CC"><u>Email</u></font></b></td>
        </tr>
        <%Do while not rs.eof%>
        <tr>
        <td width="10%" align="left"><font face="Verdana"><font
size="2"><%=say%></font></font></td>
        <td width="10%" align="left"><font face="Verdana"><font
size="2"><%=rs("unvan")%></font></font></td>
         <td width="15%" align="left"><font face="Verdana"><font
size="2"><%=rs("ad")%></font></font></td>
         <td width="15%" align="left"><font face="Verdana"><font
size="2"><%=rs("soyad")%></font></font></td>
        <td width="15%" align="left"><font face="Verdana"><font
size="2"><%=rs("bolum")%></font></font></td>
        <td width="12%" align="left"><font face="Verdana"><font
size="2"><%=rs("telefon")%></font></font></td>
        <td width="10%" align="left"><font face="Verdana"><font
size="2"><%=rs("dahilitel")%></font></font></td>
         <td width="40%" align="left"><font face="Verdana"><font
size="2"><%=rs("email")%></font></font></td>
          </tr>
        <%
tpl=tpl+1
say=say+1
%>
<% rs.movenext
loop
%>
      </table>
      <%
rs.close
set rs=nothing
conn.close
set conn=nothing
%>
      <p><font size="2" face="Verdana">Arama Sayfasina dönmek için <a
href="arama.asp">tiklayiniz.</a></font></p>
      <p><font size="2" face="Verdana"><a href="arama.asp"><img border="0"
src="geri.gif"></a></font></p>
```

```
    </td>
  </tr>
</table>
</body>
</html>
```

VERI TABANI DOSYASI



id	unvan	ad	soyad	bolum	telefon	dahilitel	email
1	Öğr.Gör.	EGEMEN	MEDE	Enformatik	564646	321	egemenmede@yahoo.com
2	Öğr.Gör.	HAKKI	kaya	Kimya	342432432	5425	hocal@toward.com
3	Okutman	İBRAHİM	ÇUBUKÇU	Kimya	342343	342	icubukcu@pcworld.com.tr
4	Yrd.Doç.Dr.	ATİLLA	MEDE	Fizik	33423432	4324	atilla.mede@turkey.com
5	Okutman	ERSİN	KARA	Makine	342342343	334	ersinkara@hotmail.com
6	Öğr.Gör.	BİRİM	SAİN	Kimya	3432432	323	birim@mail.com
7	Öğr.Gör.	İBRAHİM	MEDE	Bilgisayar	34233	3432	imede@hotmail.com
8	Yrd.Doç.Dr.	Hasan H.	BALIK	Elektrik-Elektronik	343243434	34223	balik@firat.edu.tr
9	Yrd.Doç.Dr.	İBRAHİM	ÖZDEMİR	Fizik	4324333	3432	iozdemir@computerworld.com.tr
10	Öğr.Gör.	ÖZGÜR	IRMAK	Kimya	432333	343	oirmak@yahoo.com
11	Yrd.Doç.Dr.	YAŞAR	GÖZÜDELİ	Otomotiv	3423332	3223	ygozudeli@zekam.net
12	Doç.Dr.	ÖZGÜR	DEMİREL	Tıp	32334422	34232	odemirel@hotmail.com
13	Öğr.Gör.	Resul	DAŞ	Enformatik	4235543	4233	rdas@firat.edu.tr
15	ögrenci	Gülden	Akay	elektrik	23423432	3243	sdfsd@fgff.com
16	ögrenci	ülkü	yıldırım	elk-elektronik	23432323	32332	refd@gfdfd.com
ayı)							

KAYNAKLAR

- Asp ile Web Programciligina Giris (Yrd.Doç.Dr.Adem KARACA)
- Active Server Pages (PCLIFE), (Dr. Hakki ÖCAL)
- ASP, (Zafer DEMIRKOL)
- PCLIFE Bilgisayar Dergisi (http://www.pclife.com)
- PC Magazine Bilgisayar Dergisi (http://www.pcmagazine.com)
- CHIP Bilgisayar Dergisi (http://www.chip.com)
- http://www.itu\itu_ASPdersnotu
- http://www.aspsite.com
- http://www.aspTurk.com
- http://www.aspin.com
- http://www.maxi.asp.com
- http://programlama.com
- http://www.asplist.com
- http://www.asp101.com
- http://aspzone.com
- http://www.asdj.com