

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

UZAKTAN BİLGİSAYAR KONTROLÜ

BİTİRME ÖDEVİ

HAZIRLAYANLAR

Erden SAÇAN – Tolunay ÖZBAY

Bitirme Yöneticisi :

Yrd.Doç.Dr Hasan Hüseyin BALIK

ELAZIĞ 2005

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

UZAKTAN BİLGİSAYAR KONTROLÜ

BİTİRME ÖDEVİ

HAZIRLAYANLAR

Erden SAÇAN – Tolunay ÖZBAY

Bu tez ,.....tarihinde aşağıda belirtilen jüri tarafından oybirliği / oyçokluğu ile başarılı / başarısız olarak değerlendirilmiştir.

Danışman:

Üye:

Üye:

TEŐEKKÜR

Bu alıőmanın gerek hazırlanmasında gerekse yürütülmesinde her türlü yardım ve desteęini esirgemeyen hocamız Yrd.Do.Dr Hasan Hüseyin BALIK'a, PDI Erkom (Ankara) Yazılım Müdürü Sn. Tamay ÖZBAY'a ve Siemens (İstanbul) Yazılım Bölümünden Sn. Hakan DİLEK'e teşekkür ederiz.

Erden SAAN - Tolunay ÖZBAY

ELAZIĞ 2005

İÇİNDEKİLER

1.BİLGİSAYAR PROGRAMLAMA.....	1
1.1 Bilgisayar Programlama Nedir ?.....	1
1.2 Program Nedir ?.....	1
1.3 Programlamanın Tarihi.....	2
1.4 Programlama Dilleri.....	3
1.5 Programlama Dillerinin Ortak Bileşenleri.....	3
1.5.1 Açıklamalar(Comments).....	4
1.5.2 Whitespace.....	4
1.5.3 Satır devam Karakteri.....	4
1.5.4 Literal.....	4
1.5.5 Rezerve sözcük.....	4
1.5.6 Hata.....	5
1.5.7 Sözdizimi Hatası.....	5
1.5.8 Çalışma Zamamı Hatası.....	5
1.5.9 Operatör.....	5
1.5.9.1 Aritmetik Operatör.....	5
1.5.9.2 İlişkisel Operatörler (Relational Operators).....	6
1.5.9.3 Mantıksal Operatörler (Logical Operators).....	6
1.5.10 Değişkenler (Variables).....	7
1.5.10.1 Değişken Adları.....	7
1.5.10.2 Veri Türleri (Data Types).....	7
1.5.10.3 Değişkenleri Tanımlamak.....	7
1.5.10.4 Değişkenlere İlk Değer Atamak (Initializing).....	8
1.5.11 Sabitler (Constants).....	8
1.5.12 Prosedürler (Procedures).....	8
1.5.13 Fonksiyonlar.....	8
1.6 Program Kontrolü.....	9
1.6.1 Karar Yapıları.....	10
1.6.2 Döngü Yapıları.....	10
1.6.3. Loop(Döngü).....	10
1.6.4 Seçim(Sapma).....	10
1.6.5 If...Then...Else Deyimi.....	11

1.6.6 Select Case Deyimi.....	11
1.7 Program Geliştirme.....	11
1.7.1 Kodlama.....	12
1.7.2 Test.....	12
1.7.3 Bakım.....	12
1.7.4 Akış Şeması.....	13
1.7.5 Karar Tabloları.....	13
2.INTERNET.....	14
3. WEB PROGRAMLAMA DİLLERİ.....	15
3.1 HTML (Hyper Text Markup Language) yada Web Sayfası Tasarımı.....	15
3.2 ASP (Active Server Pages) Web Programlama Dili.....	17
3.2.1 ASP'nin Mantığı.....	19
3.2.2 ASPUyumluluğu(Compatibility).....	19
3.2.3 PWS Nedir?.....	19
3.2.4 ASP Nasıl Çalışır?.....	20
3.2.5 ASP'nin Dili.....	21
3.2.6 ASP'nin Unsurları.....	21
4. VBScript.....	22
4.1 VBScript'te Yazım Kuralları.....	24
4.2 VbScript'te Değişkenler.....	25
4.3 VBScript'te Array Fonksiyonu.....	25
4.4 VBScript'te İşlemciler (Operatörler).....	27
4.5 VBScript'te Program Kontrolü.....	28
4.6 VBScript'te Mantıksal Sınamalar.....	29
4.6.1 If.. Else.....	29
4.6.2 Select Case.....	30
4.6.3 For..Next döngüsü.....	31
4.6.4 While...Wend.....	32
4.7 VBScript'te Sık Kullanılan Hazır Fonksiyonlara Örnekler.....	33
4.7.1 Tarih ve Saat.....	33
4.7.2 Karakter-dizisi Düzenleme.....	34
4.7.3 Test Fonksiyonları.....	35

TABLO VE RESİMLERİN LİSTESİ

TABLolar:

Tablo 1.1... Genel Aritmetik Operatörler.....	6
Tablo 1.2... Genel İlişkisel Operatörler.....	6
Tablo 4.1... VBScript Değişken Türleri.....	27
Tablo 4.2... VBScript'te Operatörler.....	27
Tablo 4.3... VBScript Operatör Hiyerarşisi.....	28
Tablo 4.4... VBScript String Fonksiyonları.....	34
Tablo 4.5... VBScript Test Fonksiyonları.....	35

RESİMLER:

Resim 1.... Sınırlayıcıların Test Edildiği ASP Programının Çıkışı.....	24
---	----

ÖZET

Internet sürekli aratan bir ivmeyle büyüyerek, içerik, hız ve kullanılabilirlik bakımından çok yol kat etmiş, iş, eğitim, kamu ve akla gelebilecek her tür alanın vazgeçilmezi olmuştur. Artan günlük ihtiyaçlara ve üzerine yüklenen yeni rollere cevap verebilmesi açısından Internet, yeni web programlama teknolojilerine bağımlıdır.

Kullanıcı artık durağan, kendini tekrarlayan web sayfaları yerine dinamik ve kişisel ihtiyaçlarına göre düzenleyebileceği web siteleri talep etmektedir. Bu durum ASP kullanımını zaruri bir ihtiyaç haline getirmiştir.

ASP “kişiselleştirilebilir web sayfaları” yeniliğinin yanında kullanıcının uzaktaki bir bilgisayara erişimini de çok daha kolay hale getirmiştir. Bu konuda halen her tür ihtiyaca cevap verebilecek (uzaktan eğitim, uzaktan hasta takibi, uzaktan sistem yönetimi, *uzaktan kaynak paylaşımı*, ... vb) amatör, profesyonel ve kurumsal pek çok uygulama geliştirilmektedir.

GİRİŞ:

Uygulama ödevimiz için “Uzaktan Bilgisayar Kontrolü” konusu belirlendiğinde ana hatları aşağı yukarı belliydi.Kodlamamız gereken program server olarak belirlenen bilgisayara ait paylaşımına açık belli başlı kaynakların, ağ üzerindeki başka bilgisayarlardan ve IP üzerinden Server’a ulaşan kullanıcılar tarafından görülebilmesi ve kullanabilmesini sağlamalıydı.

Biz bu noktada iki maddenin üzerinde durmaya karar verdik.Dosya yükleme ve karşı yüklemesi(upload/download) ve yazıcı(printer) paylaşımı.Her ne kadar artık bu özellikler işletim sistemlerinde built-in olarak gelse de ve haricen bu konuda pek çok program bulunsa da böyle bir çalışmanın kişinin kendi bilgisayarından uzakta bulunduğu oldukça faydalı olabileceğini (Örnek:kişisel bilgisayardan uzaktayken dijital fotoğraf aktarımı ve basımı gibi...) düşündük.

Web programlama üzerine ihtiyaç duyduğumuz temel bilgiler için ödevimizde de karşılaşacağımız türden kaynaklar toplayarak başladığımız uygulamamızı profesyonel olarak web programlama işiyle uğraşanlardan fikir,bilgi ve destek alarak tamamladık.

1.BİLGİSAYAR PROGRAMLAMA

1.1. Bilgisayar Programlama Nedir

Bilgisayar programlama,programlama dilleri aracılığıyla programların yazılmasını ve uygun bilgisayar ortamında çalıştırılarak belli işlemlerin yapılmasını sağlayan bir çalışma alanıdır.Genellikle kurumsal amaçlı ve ticari süreçleri otomatikleştirmek için programlar hazırlanır.

Programlama işi bir yada daha çok programcı tarafından ve belli yazılım araçları kullanılarak(derleyici,linker,vb..) yapılır.ancak programlama işleminin en önemli yanı bazı ön çalışmaları gerektirmesidir.bu anlamda programlama analiz ve tasarım gibi geniş çalışmaları gerektirir.

Programlama bilgisayar sektöründe yazılım alanı içinde bir alt alandır.İlk bilgisayarlarda donanım ön plandadır ve donanıma özel yazılımlar geliştirilmektedir.Bilgisayar donanımlarının standartlaşması sonucu işletim sistemleri,derleyiciler(compiler) ve diğer yazılım gelişmeleri programlama dillerinin gelişmesine ortam hazırladı.Günümüzde değişik amaçlarla yüzlerce programlama dili ile programlama yapılabilmektedir.

1.2 Program Nedir

Program bilgisayara ne yapacağını söyleyen komutlardır.Girişleri ve çıkışları arasında yapacağı düzenlemelerle bir kişinin kilosunun normal olup olmadığını söylemekten bir şirketin tüm üretim ve satış yönetimini düzenlemeye kadar pek çok iş görebilir.Programlar bir programlama diliyle yazılır.ardından bir alt düzey dile dönüştürülür.(derleme)

1.3 Programlamanın Tarihi

1822 yılında Charles Babbage'ın fark makinesi(difference engine)'nden beri insanoğlu makineleri programlamaya başlamıştır.ardından 1942 yılında ENIAC geliştirilmiştir.Elektronik olan bu makine anahtar ve kablolar aracılığıyla bir takım işleri otomatikleştirmek için programlanabilir nitelikteydi.

İlk zamanlar programlar makine diliyle hazırlanıyordu.bir makine dili programı bilgisayar içinde yer alan ikilik anahtarların 1 ve 0 olarak bilinen iki durumunu temsil eder ve bu komutlarla çalışır.

Makine dili programları anlaşılması(okunabilirliği) zor olan ve tamamıyla donanıma (mikroişlemciye) dayalı olarak geliştirilen programlardır.Bu nedenle yazılması zordur ve standart değildir.Oysa günümüzde kullanılan program geliştirme araçlarıyla programcılar Pentium yada RISC işlemci gibi donanım bileşenlerin mimarilerini bilmelerine gerek kalmadan işlemlerin ve uygulamaların işlevleriyle uğraşarak programlarını yazarlar.

Makine dillerine yakın,ancak yine belli bir mikroişlemci için geliştirilmiş dillerden birisi de Assembly programlama dilidir.Assembly'de makine komutları yerine belli bir işlemi ifade eden Assembly kodları kullanılır.MOV,ADD,PUSH gibi komutlarla temel işlemler gerçekleştirilir.Bu komutlara mnemonic kod adı da verilir.

Makine dilinin ve Assembly dilinin zorluğu üst düzey programlama dillerinin geliştirilmesine neden olmuştur.Üst düzey programlama dilleri programın kullanıcıyadaha yakın ifadelerle geliştirilmesini sağlar.Ardından bu program(kaynak program) derleyici programlar aracılığıyla makine diline çevrilir.

Bu üç gelişim basamağına birer çok basit örnek verelim:

Bir makine dili programı:

01000111

10100011

01110110

10010010

10011001

Bir Assembly programı:

ADD AX,AX

PUSH AX

MOVE AX,DATA

MOVE DS,AX

Bir BASIC Programı:

10 PRINT "Dairenin alanini hesaplayan program"

20 PRINT

30 INPUT "Yaricapi girin: ";R

40 P=3.14159

50 A=P*R*R

60 PRINT "Dairenin alani: ";A

70 END

1.4 Programlama Dilleri

Bir programlama dili belli bir konuya ilişkin program yazmak için kullanılan bir araçtır. Günümüzde derleyici(compiler) programlama dillerinin yanı sıra daha çok geliştirme ortamı(development environment) olarak kullanılan programlama dilleri de vardır. Örnek olarak: "BASIC(Beginners All-purpose Symbolic Instruction Code), COBOL(Common Business Oriented Language), FORTRAN(FORmula TRANslation), Pascal, Visual Basic, Java, Delphi, Perl, ADA, PL/I, C, C++, ...

Üst düzey programlama dilleri sözcüksel bir dil yapısı ile işlemlerin kodlanmasını sağlarlar. Ardından derleme(compiler) programları aracılığıyla programın hataları kontrolö edilir. Daha sonra programın işletilebilir(Executable) dosyası(com veya exe) oluşturulur.

1.5 Programlama Dillerinin Ortak Bileşenleri

Programlama dilleri belli ortak özelliklere sahiptir.

- Literal'ler
- Aritmetik operatörler
- Built-in fonksiyonlar
- Değişkenler (Variables)
- Arreys
- Sabitler(Constants)
- İlişki Operatörleri
- If...Then Deyimleri
- Döngüler(Loops)
- Prosedürler
- Argümanlar
- Tanımlamalar
- Açıklamalar(Comments)...

1.5.1 Açıklamalar(Comments)

Programlar uzadıkça satırların ne işe yaradığı unutulabilir.Programcı kendisi için, diğer programcılar için yada belli bir kalite ve dokümantasyon anlayışı için açıklamalara yer verebilir.

Bütün programlama dillerinin yada ortamların kendine özgü bir açıklama sistemi vardır.Örneğin C dilinde /* ... */ aralığı kullanılırken Visual Basic'de ' işaretleri C++'da // kullanılır.

1.5.2 Whitespace

Whitespace programcılıkta tab,space veya yeni satır yerine kullanılan bir terimdir.Aslında çoğu programlama dili için kod satırları üzerinde yada aralarındaki boşluğun bir anlamı yoktur,yani kullanılabilir.bunların çoğu derleyici tarafından yok edilir.Ancak işin içine veri tanımları girdiğinde önemlidir.

1.5.3 Satır devam Karakteri(Line Continuation Character)

Program satırları uzun olduğunda yazmak ve okumak zorlaşır.bu nedenle özel bir satır devam karakteriyle yeni satırdan devam edilir.Örneği Visual Basic dilinde _ (alt tire) karakteri satırın sonuna eklenerek ilgili satıra bir alt satırdan devam edilebilir.

1.5.4 Literal

Literal bir değerdir.Örneğin 3 sayısı bir literaldir.bu anlamda litelaller, nümerik literaller ve karakter literaller gibi farklı türlere sahiptirler.

Örneğin "ALİ" verisi karakter bir literalken A1012C ise hegzadesimal bir literaldir.

1.5.5 Rezerve sözcük

Her programlama dilinin kendine ait bazı elemanları vardır.bu tanımlama yada komut sözcüklerine rezerve sözcük(reserved word) denir.

Örneğin If,Then deyimleri gibi sözcükler(komutlar) bir değişken adı olarak kullanılamazlar.

1.5.6 Hata

Hemen her zaman bir programın derlenmesi ve çalışması esnasında hatalarla karşılaşılır.bunları giderilmesi için programlama ortamlarının çeşitli araçları vardır.Başlıca hata çeşitleri:

- Syntax errors (sözdizim hataları)
- Runtime errors (çalışma zamanı hataları)
- Logical errors (mantık hataları)

1.5.7 Sözdizimi Hatası

Syntax error denilen sözdizimi (gramer) hataları programcının yazarken uygulamak zorunda olduğu kurallardan dolayı oluşur.Örneğin noktalama işaretlerine uymamak gibi.Bunun dışında yanlış yazılmış sözcükler, eksik yada yanlış yerde kullanılmış parantezler,virgüller vb..

Sözdizimi hataları derleme zamanında (compile time) bulunur.Yazılım geliştirme ortamları genellikle bu hatayı programcıya gösterir.Bunun dışında bu ortamlar hataların bulunması ve düzeltilmesi için çok sayıda tekniğe sahiptir.

1.5.8 Çalışma Zamanı Hatası

Çalışma zamanı hataları(Runtime Error) program çalışırken oluşur.Bu nedenle çok önemlidir.Programın yazılması sırasında oluşmayan bu hatalar eğer daha önceden kestirilememişse programın çalışırken hata vermesine ve kesilmesine neden olur.Örneğin bir donanım sorunu yada beklenmedik bir veri girişi gibi.Bu nedenle Visual Studio .NET ortamında ve diğer modern yazılım geliştirme ortamlarında TRY-CATCH yapıları geliştirilmiştir.

1.5.9 Operatör

Sabit ve değişken veriler üzerinde işlemler ve karşılaştırmalar yapmak için operatörler kullanılır.

1.5.9.1 Aritmetik Operatör

Bilgisayar ortamında aritmetik işlemlerle oldukça sık karşılaşılır.sabit ve değişken veriler üzerinde işlemler yapmak için aritmetik operatörler kullanılır.

Operatörler aslında iki yada daha çok değeri(operand) kullanarak yeni bir değer bulan fonksiyonlardır.

İşlem	Operatör
Toplama	+
Çıkarma	-
Çarpma	*
Üs alma	^
Bölme(Katsayı)	DIV
Bölme(Kalan)	MOD

Tablo1.1 Genel Aritmetik Operatörler

1.5.9.2 İlişkisel Operatörler (Relational Operators)

İlişkisel operatörler iki yada daha çok verinin karşılaştırmasını yaparlar ve bir Boolean(ikili) değer döndürürler: True yada False

Operatör	Adı
=	eşit
<>	eşit değil
<	küçük
<=	küçük yada eşit
>	büyük
>=	büyük yada eşit

Tablo 1.2 Genel İlişkisel Operatörler

İlişkisel operatörler kullanılarak yalnız sayısal (nümerik) değerler değil,karakter dizileri de karşılaştırılabilir.Karakter karşılaştırılmasında ASCII kodları kullanılır.

1.5.9.3 Mantıksal Operatörler (Logical Operators)

OR: Yalnızca biri doğru ise sonuç doğru

AND: Her ikisi de doğru ise sonuç doğru

XOR: Biri doğru biri yanlış ise sonuç doğru

NOT: Yanlış ise sonuç doğru

1.5.10 Değişkenler (Variables)

Değişkenler verilerin saklandığı alanlardır. Belli türdeki verileri saklamak yada temsil etmek amacıyla değişkenlere ihtiyaç duyulur.

Değişkenler bellekte bir kutu olarak düşünülebilir. Program çalışmaya başladığında bu kutu açılır, içine değişik değerler konur ve böylece program içinde verilerin temsil edilemesi sağlanır.

1.5.10.1 Değişken Adları

Bir değişken yaratıldığında ona unique bir ad vermek gerekir. Bunun yanı sıra değişken adları programlama dilinin kullandığı anahtar sözcüklerden de farklı olmalıdır. Örneğin bazı dillerde değişken adları %,!,#,\$,& içeremezler. Bunun dışında başka bir kısıtlama da Case-Sensitive'liktir. Visual Basic Case Sensitive değildir yani "toplam" ile "Toplam" aynı değişkeni temsil ederler. Ancak C++ gibi programlama dilleri Case-Sensitive'dir. Bu durumda değişken adlandırmaya daha çok dikkat etmek gerekir.

Birçok programlama dilinde değişkenler bir alfabetik karakterle başlar. Değişken adlarına genelde bir uzunluk sınırı getirilir. Örneğin Visual Basic'de 255 karakter uzunluğunda olabilir.

1.5.10.2 Veri Türleri (Data Types)

Veri türleri değişkenin içinde saklanacak verilerin türünü belirtir. Örneğin sayısal veriler, karakter veriler gibi. Programlama türüne göre farklılık gösterse de veri türleri genellikle numeric(sayısal), character, date(tarih), boolean(true-false) gibidir.

1.5.10.3 Değişkenleri Tanımlamak

Bazı programlama dillerinde tanımlamaya gerek yoktur. Değişkene bir değer verilerek yani doğrudan atama yaparak kullanılabilir.

Toplam=100

Alttoplam=20

Toplam=Toplam+Alttoplam

Ancak bu tür kullanımın sakıncaları vardır. Çünkü tanımlanmamış değişkenlerin yazılımında sık sık sorunlar çıkar. Örneğin yukarıdaki program parçacığında "Toplam" yerine "Toplan" yazan bir programcı hiçbir hata almaz ancak programın çalışması esnasında mantıksal hatalarla karşılaşacaktır.

1.5.10.4 Değişkenlere İlk Değer Atamak (Initializing)

Bir değişken tanımlandıktan sonra onun için bellekte bir kutu ayrılmış demektir. Ancak bu kutunun boş ve temiz olduğundan emin olmak için daha çok eski programlama dillerinden gelen bir alışkanlıkla ilk değer atanır. Bu temizleme amacıyla yada verilmesi gereken bir ilk değer şeklinde olabilir.

Toplam=0

Adi= " "

1.5.11 Sabitler (Constants)

Değişkenler bellekte açılan kutulardır. Değişkenlere sürekli farklı değerler atanabilir. Ancak değişkenlerden farklı olarak sabit değerlere de ihtiyaç vardır. Bu değerler program içinde bir kez tanımlandıktan sonra değişmeyecek değerlerdir.

Const KDV=18

1.5.12 Prosedürler (Procedures)

Kod bloklarına prosedür denir. Prosedürler sayesinde kodların birimler halinde yazılması ve paylaşımı sağlanır. Prosedürler modüler olarak program parçasını oluştururlar ve aynı program içinde yada farklı programlar içinde istenilen yerden çağrılabilirler.

Sub ToplamaIslemi

--kodlar

End Sub

Prosedürleri Çağırma

Call deyimi kullanılarak yada yalnızca adının yazılmasıyla prosedür çağrılabilir.

Call ToplamaIslemi

1.5.13 Fonksiyonlar

Bir prosedür türü olarak düzenlenen fonksiyonlar belli bir işlemi kod olarak içeren ve bir değeri döndüren program parçalarıdır. Hesaplama işlerinde yoğun olarak kullanılırlar.

Function KDVHesapla(deger1)

*deger1=deger1*1.18*

End Function

Built-in Fonksiyonlar

Built-in fonksiyonlar programlama dillerinin zenginlikleridir. Veri çevirme(dönüştürme) işlemleri yapan,karakterleri arayan yada bölen,karekök alan yada rastgele sayı üreten(random) fonksiyonlar programcıların çok işine yarar.

Built-in fonksiyonlar değişik kategorilerde incelenebilirler.

Matematiksel fonksiyonlara birkaç örnek verelim:

Round=Sayıları yuvarlar

Sonuc=Round (degisken,2)

Val=String değişkeninin sayısal değerini verir.

*Sonuc=Val (*421.3*)*

Int=Tamsayı değer döndürür

Sonuc i=int (degisken)

String (karakter) fonksiyonlara birkaç örnek verelim

Left=Bir stringin soldan belirtilen kadar karakterini döndürür.

Sonuc\$=left\$ (degisken\$,5)

Ucase=Bütün karakterleri büyük harfe çevirir.

Sonuc\$=ucase\$ (degisken\$)

1.6 Program Kontrolü

Program içinde tanımlama deyimleri,atama deyimleri yer alır ve sırasıyla işletilir.Ancak programın işleyişinde bu deyimler her zaman sırasıyla değil de belli bir mantıkla işletilmesi gerekir.İşte program deyimlerinin işleyiş sırasının kontrolüne program kontrolü denir.Program kontrolü belli yapılarla sağlanır.

1.6.1 Karar Yapıları

-If...Then...Else

-Select Case

Karar yapılarının yanı sıra yaygın olarak kullanılan bir diğer yapı da döngü(loop)dür.Döngüler programın etkinliği bakımından önemli bir tekniktir.

1.6.2 Döngü Yapıları

-Do...Loop

-For...Next

Döngüler işlemlerin belli bir sayıda yada belli bir koşula göre yinelenmesini(tekrarını) sağlar.Programcı gereksinimine göre karar yapılarından birini veya birkaçını birlikte kullanabilir.

1.6.3. Loop(Döngü)

Bir kod bloğunu yinelenerek çalıştırmaya döngü denir.Döngü sayesinde bir blok deyim verilen koşul doğru(True) olduğu sürece işletir.

Do döngüleri:

Do While ucret<100

'ücret 100den küçük olduğun da

Toplam=Toplam + ucret

Loop

1.6.4 Seçim(Sapma)

Seçim ve sapma(branch) yapıları If...Then...Else ve Select...Case deyimleri ile gerçekleştirilir.If...Then...Else deyimi belli bir deyimi yada bir blok deyimi bir koşula bağlı olarak işletmeyi sağlar.

Select Case deyiminde ise özellikle bir karar değişkeninin aldığı değerlere göre değişik işlemlerin yapılması sağlanır.Karar yapılarının temelinde değerler arasındaki ilişkiler yatar.İlişkiler değerler arasındaki operatörlerle kurulur.

1.6.5 If...Then...Else Deyimi

Klasik olarak kullanılan bir karar mekanizmasıdır.eğer koşul sağlanıyorsa (if) bunu yap,yoksa şunu yap (else) gibi.

Yapısı:
If *kosul* *Then*
[islem 1]
Else
[ilsem 2]
End If

1.6.6 Select Case Deyimi

Bir değere göre seçim yapılmasını sağlar.Select Case yapısı bir koşul yada değişken için olası değerlerin değerlendirilmesini sağlar.

Yapısı:
Case deger1
deyim blok 1
Case deger2
deyim blok 2
end Select

1.7 Program Geliştirme

Programlama yada uygulama geliştirme aslında görüldüğünden daha uzun bir süreçtir.Analizle başlar,ardından geliştirilen programın çalıştırılması,bakımı ve yükseltilmesiyle bu süreç devam eder.

Program geliştirme aşamaları:

- 1- Analiz/Tasarım
- 2- Kodlama
- 3- Test etme
- 4- Çalıştırma
- 5- Bakım

Analiz (sistem çözümlenme) sürecinde özellikle kullanıcıların istedikleri bilgileri üretecek programın tanımı yapılır.

Analiz sürecinde gereksinimler ve bu gereksinimlerin çözümü tanımlanır. Bütün bunların ifadesinde çeşitli teorik ve şematik yöntemler kullanılır. Analiz çalışmasının ilk aşamalarında gözlemlere ve görüşmelere yer verilir.

Program tasarımında yapılacak işleri açıklamak ve programcıya yol göstermek için kullanılan yaygın yöntemlerden birisi akış şemalarıdır (flow chart).

Özellikle If'li deyimleri açıklamak için karar tabloları (decision tables) kullanılır. Karar tabloları karmaşık işlemlerin açıklanması için bir kılavuz olacak biçimde hazırlanır.

Karar tablolarında durumlar ve işlemler yer alır. Bunlara göre de kurallar sıralanır. Örneğin bir fatura kesiminde iskonto uygulaması için bir tablo geliştirilebilir.

Program tasarımında programın birimlere (modüllere) ayrılması ve adım adım açıklanması çok önemli bir işlemdir. Bu birimler programın işlevlerini ve işlevlerin hiyerarşisini ortaya koyar.

Analiz aşamasında genel bir proje oluşturulurken tasarım aşamasında neyin nasıl yapılacağı ve hangi dilin kullanılacağı gibi somut kararlar verilir.

Programın kapsamı kullanıcı gereksinimlerini, fizibilite raporlarını ve gerekli zamanı içerir. Bu aşamalarda birçok araç kullanılır. Planlama araçları, şema çizmek için araçlar..vb.

Tasarımın kodlama sürecinde ayrıca flow chart, data modelleme teknikleri kullanılır.

1.7.1 Kodlama

Planlama aşaması yerine getirildikten sonra kodlama şaması başlar. Programcı bu aşamada tasarım spesifikasyonlarına uygun olarak kodlamayı yapar.

1.7.2 Test

Kod yazıldıktan sonra test edilmesi gerekir. Test sürecinde belli metotlar içinde yada daha serbest olarak programın fonksiyonları test edilir. Kullanıcı arabirimi, veri girişi, alanlara girilen bilgilerin boyutları...vb.

Programın işletilmesi, programın çalıştırılması anlamına gelir. Bu aşamada programın client (istemci) yada server (sunucu) olarak kurulması, düzenlenmesi ve çalıştırılması sağlanır.

1.7.3 Bakım

Programın bakımı programın güncel koşullara göre yeniden düzenlenmesini içeren bir konudur. Oluşan hataların giderilmesi, yeni eklemeler yapılması yada programın teknolojisinin yenilenmesi gibi işlemler programın bakımı konusuna girer.

1.7.4 Akış şeması

Programları yada program mantığını oluşturan algoritmaları tanımlamak için kullanılan çizim tekniklerinden en klasik olanı akış şemasıdır(flow chart).akış şemaları program yada program parçalarındaki akışı sergiler.böylece dokümanite edilen program mantığı programcılar tarafından kolayca kodlanır.

Akış şemaları değişik kapsamlarda hazırlanabilir:

- Sistem akış şemaları
- Program akış şemaları

Sistem akış şemaları program yada yazılımla ilgili genel bir görünüm sunar.

Verileri,kaynakları,işlemleri genel olarak gösterir.

Program akış şemaları ise program işleyişindeki işlemleri,karar yapılarını (IF-ELSE),döngüleri,veri alışverişini ve özel simgelerle daha birçok işlemi açılar.

Bir yazılımın sistem analiz ve tasarım aşamasında bütün ilişkileri ve bileşenleri çözüldükten sonra bilgisayar tarafından işletilecek adımlar oluşturulur.İşte bu aşamada akış şemaları kullanılır.akış şeması program mantığının kodlanmasında ve sorun giderme aşamalarında yardımcı olur.

1.7.5 Karar Tabloları

Programda çok sayıda IF'li deyim yer aldığında koşullara göre çok sayıda yapılacak işlem var demektir.Bu durumda program mantığını akış şemasıyla dokümanite etmek durumu iyice karmaşıktırabilir.Bu nedenle program içinde oluşan koşulları ve bu koşullara göre yapılacak işlemleri en iyi şekilde açıklamak için karar tabloları kullanılır.

Karar Tablosu:

- Bir kararın matris gösterimidir
- Olası koşulları ve sonuçlarını gösterir
- Özellikle karmaşık kararların çözümünde kullanılır

Karar tablosu üç kısımdan oluşur:

Koşullar:Karara ilişkin koşulları ortaya koyar.

İşlemler:Koşullara göre oluşacak aktiviteler.

Kurallar:Hangi koşul oluştuğunda hangi işlemlerin yapılacağını gösterir.

İyi Bir Programın Nitelikleri

- Görsellik
- Kolaylık
- Doğruluk
- Hız
- İyi bir belgeleme(Programa ait dokümantasyon)
- Kolayca değiştirilebilme,güncellenebilme
- Etkin bir kodlama
- Etkili bir işletim(daha az sistem kaynağı ihtiyacı)

2.İNTERNET

İnternet dünya genelinde bilgisayar ağlarını birbirine bağlayan ve .ağların ağı. olarak da anılan adeta .sınırsız. bir iletişim ve bilişim ortamıdır. 1969 yılında Amerika Birleşik Devletleri İleri Savunma Araştırma Projeleri Teşkilatı tarafından olası bir savaş çıkmasına karşı iletişim ve verileri korumak amacı ile geliştirilen İnternet, birçok bilgisayar ağıyla birleşerek 1983 yılından sonra TCP/IP destekli, uluslararası bir bilgisayar ağı olarak ortaya çıkmıştır. Köklerinin bu kadar eskiye dayanması ile birlikte, İnternet kullanımındaki inanılmaz artış son birkaç yılda olmuştur [2]. Bunda bilgisayar donanımı ve iletişim maliyetlerinde süregelen düşüşler ve iletişim hızlarının artması yanında, İnternette bilgi dağılımı ve kullanımında yeni ufuklar açan yazılım ve araçların (İnternet tarayıcıları, HTML, Java ve VB script, ASP, CGI, Perl, Flash vs.) geliştirilmesi önemli rol oynamıştır. Günümüzde bir milyanın üstünde kullanıcıya ulaştığı tahmin edilen bir siberuzay ortamı haline gelen İnternet, kabaca ayda %10 oranında büyümektedir [1].

Sunduğu sınırsız imkanlarla, başlangıçta yalnızca teknik amaçlara yönelik olarak geliştirilmiş olan İnternet, bilgi teknolojilerindeki gelişmelerin de desteğiyle, artık sosyoekonomik hayatın vazgeçilmez bir aracı olmuştur. İnternet adeta kültürler değiştirmektedir. Elektronik posta, veri transferi, bilgi dağıtımını, bilgi arama/tarama, video konferans gibi teknik işlevler yanında rezervasyon, alışveriş vs. için de İnternet yoğun olarak kullanılmaktadır [2]. Görüntü ve sesin sayısallaştırılması, gün geçtikçe daha az yer kaplayacak şekilde saklanabilmesi ve hızlı transferini mümkün kılan teknolojiler sayesinde günümüzde artık İnternet üzerinden TV izlemek, müzik dinlemek, görüntülü iletişim vs. mümkün olmakta ve hayal sınırlarını zorlamaktadır. İnternet üzerinden yapılan ürün ve hizmet gelirleri 2000 yılı itibarı ile 10 milyar doların üzerindedir.

Büyüme hızı ve boyutlarını kavramak açısından, İnternet dijital evreninde iki dakika içinde gerçekleşen olaylar için, Intel.in yöneticisi Craig Barrett tarafından dikkat çekilen örnekler şöyledir [7]:

- 400 yeni abone katılıyor.
- Açık arttırma sitelerine 1400 yeni ürün katılıyor.
- Amazon.com.dan 11 bin dolarlık alışveriş yapılıyor.
- Google arama motorundan 83 bin arama yapılıyor.
- Intel Internet üzerinden 103 bin dolarlık yeni iş teklifi alıyor.
- AOL.den 900 bin e-posta gönderiliyor.,
- 50 trilyon Bit.lik ses dosyası transfer ediliyor.
- 100 trilyon Bit.lik data dosyası gönderiliyor.
- Yahoo.dan 1,5 milyon sayfa indiriliyor.
- 500 milyarın üstünde olan toplam Internet sayfalarına 50 yeni sayfa ekleniyor.

WWW (World Wide Web) olarak adlandırılan web siteleri Internette bilgi transferi için kullanılan temel araçlardır. WWW, http (Hyper Text Transfer Protocol) adı verilen protokolü kullanarak web sayfalarını görüntüler. Http bir anlamda Internet ağı üzerindeki iletişim dilidir. WWW, hypertext adı verilen bir sistemin üzerine dizayn edilmiştir. Hypertext mantığında bir dokümanın içindeki bir kelime bir başka dokümana ya da referansa (link) bağlanmaktadır [3]. Kullanıcının bulunduğu dökümandan başka bir dökümana geçiş yapabilmesi için özel bir komut bilmesine gerek yoktur; o doküman ile ilgili sözcüğe, sözcük grubuna yada simgeye tıklaması yeterlidir. Doküman kelimesi ile kastedilen sayfa, resim, video, ses, dosya vs. dir. Bu özelliği sayesinde WWW, kullanıcı dostu (user friendly) ve ek özellikleri sebebiyle de çok fonksiyoneldir.

3. WEB PROGRAMLAMA DİLLERİ

3.1 HTML (Hyper Text Markup Language) yada Web Sayfası Tasarımı

Web sayfalarının oluşturulması ve http tarafından anlaşılabilir dökümanlar oluşturulabilmesi için kullanılan yazılım dili ise html.dir (Hyper Text Markup Language). Bu dille yaratılan dosyalar düz text (ASCII Plain text) olup uzantı isimleri .htm yada .html.dir. Anlaşıldığı üzere web sayfası tasarımı dediğimiz şey esasen html dilinde yazılmış bir programdır. Html, tüm programlama dillerinin genel mantığında olduğu gibi, yapılacak işleri bilgisayara tarif eden, çalıştırılacak prosedürü içeren, text formatında bir kaynak kodudur. Html.nin diğer programlama dillerinden farklı yönü sadece görsel düzenleme için kullanılmasıdır. Html kodu basitçe, sayfanın şurasına şu resmi yerleştir, şu yazı kalın olsun, tablonun şekli şöyle olsun, zemin rengi bu olsun, bu kelimedden şu sayfaya link verilsin gibi yönergeleri içerir.

Her programlama dilinin yazılan yönergeleri uygulayan bir yorumlayıcısı/derleyicisi vardır. Html dilinin yorumlayıcıları ise Internet browser dediğimiz tarayıcılardır. Değişik firmalarca üretilmiş bir çok html yorumlayıcısı mevcuttur. Bunlardan en yaygın olarak kullanılanları Microsoft Internet Explorer ve Netscape Navigator.dur. Internet Browserlar html dilini yorumlayarak, web sayfalarını ekranda gördüğümüz haliyle kullanıcıya gösterirler. Kullanıcının bir web sitesini görebilmesi için kendi bilgisayarında çalışan bir html yorumlayıcıya (browser) ihtiyacı vardır. Web siteleri sunucu (server) adı verilen bilgisayarlarda saklanırlar. Herhangi bir kullanıcı (client) web browser.ının komut satırına bir web sayfasının adrsini yazdığında, browser Internet ağı üzerinden o sitenin kayıtlı olduğu sunucuya ulaşır ve sözkonusu sayfaya ait kaynak kodunu ve ilgili dökümanları (resim vs.) talep eder. Bu dosyalar önce kullanıcının bilgisayarına indirilir (download), ardından browser tarafından çalıştırılır. Dolayısıyla Html, kullanıcı tarafında çalışan bir dildir. Internet Explorer.da bu şekilde ekranımıza gelmiş bir web sayfasının, html dilindeki kaynak kodunu Görünüm/Kaynak menüsünden görmek mümkündür.

Bir text editöründe (mesela Notepad) yazdığımız html kodunu .htm uzantılı olarak kaydedip, herhangi bir browser ile çalıştırdığımızda hazırladığımız sayfanın görüntüsü karşımıza gelecektir. Fakat bir web sayfası hazırlamak için ille de html bilmek gerekli değildir. Html dilini bilmeyenlerin de kolayca kullanabileceği, web sayfası tasarlamak üzere üretilmiş bir takım programlar mevcuttur. Basit ve sade sayfalar hazırlamak için çoğu kez Frontpage Express ve Netscape Composer yeterlidir. Daha profesyonel ve komplike web siteleri üretmek için ise Macromedia Dreamweaver yada MS Frontpage yaygın olarak kullanılan yazılımlardandır. Bunun yanında office uygulamaları ve resim işleme yazılımları gibi görsel olan bir çok program .htm formatında kayda izin vermektedir. Örneğin Word ile hazırladığımız bir sayfayı .Web sayfası olarak. (.htm) kaydetmemiz mümkündür. Şurası da bir gerçektir ki bu türden kolaylaştırıcı programlar ne kadar faydalı olurlarsa olsunlar html dilini bilmek her zaman bir artıdır. Bu sayede, bu programlarda karşılaşılan bazı güçlükleri bizatihi kodu inceleyerek gidermek, gerektiğinde manual olarak müdahale etmek ayrı bir avantaj sağlayacaktır. Hızlı bir şekilde sayfa tasarlamak açısından bu yazılımları etkin bir şekilde kullanmak, ama gerektiğinde koda müdahale edebilmek iyi bir web tasarımcısının özelliklerindedir.

Html dilinin yapısından kısaca bahsedecek olursak; Komutlar .<. ve .>. işaretleri arasında yazılır ve .tag. adını alırlar. Bir tag <tag_adi> şeklinde başlar ve bazı taglar dışında </tag_adi> şeklinde biter. Mesela <table> ile bir tablonun çizilmeye başlandığı belirtilir. Tablo içindeki tüm işlemler bittikten sonra </table> yazılarak tablo kapatılır. Bu işaretler arasına yazılmayan metinler sayfa üzerinde kullanıcıya sunulan metinlerdir. Dolayısı ile bu işaretlerin düzgün kullanılmasına dikkat edilmeli, eksik yazılması halinde, sayfanın biçimsiz görüntüler alacağı, istenmeyen metinlerin kullanıcıya gösterileceği bilinmelidir.


```
<html>
<head>
<title> hizalama ve borderlar </title>
</head>
<body>
<table align="center" border="2">
<tr>
<td>
<center><b><font color="#ff0000"><font size=+3>Satır1</font>
<br></font></b></center>
</td>
</tr>
</table>
</body>
</html>
```

3.2 ASP (Active Server Pages) Web Programlama Dili

Yukarıda da bahsedildiği gibi HTML dosyalarının görüntülenmesinde tek yönlü bir iletişim sözkonusudur. HTML dosyalar etkileşimli değildirler. Yalnızca sunucudan kullanıcıya doğru bir akış sözkonusudur. Dolayısı ile HTML ile yapılabilecekler sınırlıdır. Sadece resimler, metinler ve linkler üzerine kurulmuş bu sistem web sayfalarında monoton bir ortam yaratmaktadır. HTML sayfalarının tarayıcıda görüntülenmesi adeta bir mağazanın vitrininden bakmakla eşdeğerdir. Oysa artık günümüzde Internet çift yönlü bir iletişimi gerektirmektedir. Her türlü işlem web üzerinden yapılmaktadır. Dolayısı ile kullanıcının sunucuya ve sunucunun da kullanıcıya veri gönderdiği etkileşimli sayfalara ihtiyaç duyulmaktadır. Bu amaçla değişik Internet programlama ve interaktif sayfa hazırlama teknikleri geliştirilmiştir. CGI/PERL, PHP ve ASP gibi. Kolay kullanımı, dil alternatiflerinin genişliği, anlaşılır programlama mantığı, güvenli oluşu ve Microsoft tarafından geliştirildiği için Microsoft tabanlı tüm uygulamalarla problemsiz çalışması gibi özelliklerinden dolayı ASP, bugün en yaygın olarak kullanılan tekniği haline gelmiştir. Açılımı Active Server Pages (Aktif Sunucu Sayfaları) olan ASP kullanıcı tarafına etkileşimli, dinamik Web sayfaları göndermek için kullanılır. Dinamik sayfalar sayesinde her kullanıcıya farklı bir sayfa gösterilmesini sağlar. Html ile bunu yapmak imkansızdır, ancak farklı sayfalar yaparak bu sorun çözülebilir. Bu da zamandan ve yerden büyük kayıp demektir. Asp bir Internet programcılığı tekniğidir. Fakat bizatihi bir dilin ismi değil, bir teknolojiye

verilen addır. Tercihe bađlı olarak ASP proramlamada JScript, JavaScript yada VBScript dillerinden biri kullanılabilir. En yaygın olarak VB script kullanılmaktadır. Visual Basic Scripting de adından da anlaşılacağı gibi VB programlama dilinin bir web için uyarlanmış bir versiyonudur. Tam anlamı ile bir VB bilgisi gerektirmez. Ne var ki, VB' ne kadar iyi bilinir ve hakim olunursa o kadar çok karmaşık işler yapılabilir.

Kullanıcı taraflı bir dil olan html.ve karşın, ASP sunucu taraflı bir teknolojidir (server side application). Şöyleki; kullanıcı bir html dosyası vasıtasıyla sunucudaki asp komut dosyasına veri gönderir ve bu veriye bađlı olarak bir talepte bulunur. ASP dosyası içerisindeki kod, sunucu üzerinde çalıştırılır, kullanıcının talep ettiği bilgi hazırlanır ve kullanıcıya gönderilir. Dolayısı ile kullanıcı sadece kendisine gönderilen sonuçları görebilir, ASP dosyasının içeriğini göremez. Kullanıcı sunucu tarafında neler olup bittiğini bilmezken kendisine gelen sayfanın kaynağına (source) baktığında ASP kodlarını deđil, Internet sunucusunun onun için hazırladığı HTML kodlarını görür.

Kullanıcı, sunucu bilgisayardan herhangi birşey talep ettiğinde sunucudaki Internet Information Server (IIS) devreye girer. IIS çağrılan dosyanın uzantısına bakarak farklı şekilde davranır. Eđer çağrılan bir html dosyası ise IIS bu dosyayı ve bileşenlerini kullanıcıya paket halinde gönderir ve dosya kullanıcının tarayıcısında yorumlanır. İstemcinin talebi bir ASP sayfasıysa IIS bu sayfanın içeriğine bakar, sayfada kullanılan scripting dilinin yorumlayıcısını çağırır. Yorumlayıcı ASP dosyasındaki komutları uygulayarak kullanıcıya gönderilecek veriyi hazırlar ve sonuç veri IIS tarafından HTML formatında kullanıcıya gönderilir. ASP sisteminin dezavantajı Web sunucunuza daha fazla yük getirmesidir. Normal bir HTML sayfasının sabit diskten alınarak sunulması sunucuya fazla yük getirmezken ASP sayfalarının işlenmesi sunucunuzun deđerli bellek ve işlemci kaynaklarından daha fazla yararlanacaktır. (HTML sayfalarının kullanıcıya gösterilmesi direk olarak dosyaların transferi işlemidir Buna karşılık, ASP talebiyle sunucuda bir komut dosyası çalıştırılmakta, bir takım deđişkenlerin hafızaya alınmakta, dolayısı ile bellek işgal edilmektedir.)

Sunucu taraflı bir teknoloji oluşu, özellikle karmaşık ve güvenlik gerektiren Internet uygulamalarında ASP.nin başlıca alternatif olarak tercih edilmesine sebep olmuştur. Ziyaretçi defteri, mesaj panosu, alışveriş sitesi gibi özellikle web üzerindeki veri tabanı uygulamaları için ASP çok elverişlidir. Access veya Oracle gibi veri tabanlarına rahatlıkla erişebilmekte, SQL dili kullanabilmektedir. ODBC bađlantısı yapılabilen tüm veritabanlarına ASP kullanarak ulaşmak mümkündür. ASP.nin güvenliği, kullanıcının ASP dosyasının içeriğine erişememesi, sadece verilen kadarı ile sonuçları görebilmesinden kaynaklanmaktadır.

Sunucu taraflı olmasının bir başka avantajı da kullanıcının bilgisayarının yada tarayıcısının (browser) ASP yi desteklemesi gibi bir şeye ihtiyaç duymamasıdır. Kullanıcının bilgisayarından bađımsız olarak tüm işlemler sunucu tarafında yapıldığı için farklı tarayıcılarda farklı farklı sonuçlar almak gibi bir ihtimal söz konusu deđerdir.

HTML zaten kullanıcı tarafında çalıştığı için kullanıcı kendi bilgisayarındaki herhangi bir tarayıcıda, hazırladığı sayfaları test edebilir. Fakat Win 95 ve 98 işletim sistemlerinde ASP kodlarının düzgün çalışıp çalışmadığını direk olarak test etmesi mümkün değildir.

3.2.1 ASP'nin Mantığı

ASP bir Server Side Script olduğu için Web sayfası kullanıcının ekranında görüntülenmeden işlenir, gerekli işlemler yapıldıktan sonra html kodları oluşturulur ve kullanıcının bilgisayarına gönderilir. Yani eğer sayfada hepsi varsa önce server side scriptler(ASP,cgi gibi) sonra client side scriptler ve en son bunların işlenmesinden oluşan HTML kodları üretildikten sonra yazılan diğer HTML kodları ile birlikte işlenir.

ASP komutları normal HTML komutlarından <%...%> tagları ile ayrılır. ASP'nin default dili VBScript'tir. Eğer JavaScript ile yazmak istenirse sayfanın başına <% LANGUAGE=JSCRIPT %> yazılması gereklidir. Bundan sonra sayfada ASP kullanılacak yerlere <%...%> yazılması yeterlidir.

3.2.2 ASP Uyumluluğu(Compatibility)

1. ASP bir Microsoft teknolojisidir.
2. Bilgisayara IIS (Internet Information Server) kurabilmek için bilgisayarda Windows NT 4.0 veya sonraki versiyonların kurulu olması gereklidir.
3. Bilgisayara PWS (Personal Web Server) kurabilmek için bilgisayarda Windows 95 veya sonraki versiyonların kurulu olması gereklidir.
4. ChiliASP Windows İşletim sistemi olmadan ASP'nin çalıştırılmasını sağlayan bir teknolojidir.
5. InstantASP de Windows İşletim sistemi olmadan ASP'nin çalıştırılmasını sağlayan bir diğer teknolojidir

3.2.3 PWS Nedir?

ASP ile programlamaya başlamadan önce bilgisayara Windows CD-ROM'u içinde yer alan PWS (Personal Web Server-Kişisel Sunucusu) programı kurulmalıdır.ASP, PHP, CGI ve Perl gibi sunucu taraflı yazılım geliştirme ortamlarında dosyalar Internet'e yerleştirmeden önce kontrol edilmelidir. Bu sebepten dolayı bilgisayara Kişisel Web Server Programı (PWS'yi) kurulmalıdır. Sonuç olarak PWS, Web sayfalarını Internet'e koymadan önce kontrol etmek ve düzenlemek amacı ile kullanılabilen bir masaüstü server (sunucu) yazılımıdır

PWS kurulumu ile ilgili olarak:

1. Sistem Windows2000 Professional veya Windows2000 Server ise:
Kişisel Web Server programını kendiliğinden kurar.
2. Sistem Windows95/98, NT4 Workstation veya NT4 Server ise:
Kişisel Web Server elle kurulmak zorundadır. PWS kurulmadan önce kontrol edilmesi ve uygulanması gerekenler:
Windows 95/98
 - Bilgisayarım>Denetim Masası>Program Ekle/Kaldır>Internet Araçları>Personel Web Server Seçeneğinin işaretlenmiş olması gerekir. Eğer Windows normal seçeneği ile kurulmuşsa bu seçenek işaretli demektir.
 - Bilgisayara bir kimlik verilmesi gerekir. Bunun için Bilgisayarım>Denetim Masası>Ağ' ı tıklayarak, diyalog kutusunun ikinci sekmesi olan Tanımlama açılır ve "Bilgisayar Adı" kutusuna istenilen isim verilir.
 - Bilgisayarın IP yapılandırmasının doğru çalışıp çalışmadığını kontrol etmek içinde Başlat>Çalıştır' ı açılarak, "winipcfg" çalıştırılarak kontrol edebilir.

ASP veya Active Server Pages (Aktif Sunucu Sayfaları) sayfalara dinamiklik kazandıracak bir web programlama tekniğidir.

3.2.4 ASP Nasıl Çalışır?

ASP' nin anlamının Aktif Sunucu Sayfaları olduğunu biliyoruz. O halde ASP'nin sunucu tarafında çalışan bir teknik olduğunu aşıkardır.Şimdiye kadar hazıladığımız sayfalarda yazdığımız kodlar istemci tarafında çalışırdı. Yani Internet kullanıcısı www.programcitr.com/index.asp diye yazdığında index.htm sayfası web server'ınızdan alınıp, olduğu gibi kullanıcının bilgisayarına gönderilirdi. Fakat ASP tekniği ile yapılmış sayfalarda HTML tagları içerisine yazdığımız kodlar önce web serverda icra ettirilir, sonra bu kodları ayıklanır ve geriye sadece işlenmiş salt HTML kodu gönderir. Şimdi buraya kadar olan bölümde ASP ile ilgili çıkan sonuçlar:

- ASP sayfasının içerisinde kodlar vardır.
- ASP sayfasının içerisindeki kodlar web serverda icra ettirilir.
- ASP sayfası,kullanıcının browser'ına salt HTML olarak gönderilir.
- ASP içerisinde yazılan kodun web server'da çalıştığı için kullanılan browser'a göre görüntülenecek olmamasıdır.

Bu arada devamlı olarak ASP kodlarından bahsettik. Nedir bu ASP kodları? ASP bir çok script dili ile yazılıp, HTML taglarımızın içerisine yazılan kodlardır. ASP'nin temel olarak yazılabildiği script dilleri;

- VBScript
- JavaScript
- JScript

- PerlScript
- PHPScrip

3.2.5 ASP'nin Dili

ASP, bir teknolojidir. Kendi başına bir yazım kuralı yoktur. ASP tekniğini kullanabilmek için, ASP sayfasının talep edilmesi halinde ziyaretçiye gönderilmeden önce ASP.DLL'ye teslim edilmesi bu teknolojinin kullanılabilmesi için hemen hemen tek şarttır. Bunu, dosya uzantısını .asp yaparak sağlarız.ASP.DLL ise, dünyada mevcut bütün Script dilleri ile verilecek komutları kabul edebilir. Sadece ASP.DLL'e sayfadaki kodların hangi dilde olduğunu söylemeniz gerekir. Bunu, ASP sayfasının birinci satırında yaparız. Örneğin ASP'ye VBScript dilini kullanmasını belirtmek için bu satırı şöyle yazarız:

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

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

3.2.6 ASP'nin Unsurları

ASP kodları nesnelere yöneliktir, onları kullanma ve onlardan bir sonuç alma veya onlara bir sonuç aktarma amacına yöneliktir. ASP'nin Nesneleri altı grupta toplanır:

Application/Uygulama: Bir ASP sitesi, gerçekte bir Uygulama Programı olarak görülür. Bu, HTML/CGI geleneğine aşina tasarımcı için yeni bir kavram. ASP kullanan tasarımcılar; bir ziyaretçi bir ASP sayfasından girerek, bir sitede surfing'e başladığında, onu bir programı işleten bilgisayar kullanıcısı olarak görürler. Böylece site her ziyaretçinin karşısına çıktığında "bir program çalışmış" gibi sayılır.

Session/Oturum: Bir ziyaretçi siteye geldiğinde, hangi sayfayı talep ederse etsin, bu bağlantı ASP açısından bir oturum sayılır. Her oturumun belirli bir süre devam eden özellikleri, değişkenleri ve değerleri vardır. Site tasarımında oturum özelliklerinden geniş ölçüde yararlanır.

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

Response/Karşılık: Server'dan ziyaretçinin bilgisayarına gönderilen bütün bilgiler, çerezler (cookie) ve başlıklar (Header) Response (Karşılık) nesnesinin ögeleridir. Bu nesneyi kullanarak ziyaretçiye göndermek istenilenler gönderilir.

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

ObjectContext/Nesne Bağlamı: Microsoft'un Transaction Server (MTS) programının sunduğu hizmetlere erişimi sağlar. MTS, ASP sayfaları içinden, uygulama programlarından yararlanılmasını sağlar.

HTML.de olduğu gibi ASP kodları da herhangi bir text editöründe yazmak mümkündür. Aslında ASP dosyası HTML kodlarını da içeren bir yapıya sahiptir. Bir dosyanın uzantısının .ASP olması o dosyanın içeriğinin tamamında yada bir kısmında ASP kodlarının yazılı olduğunu gösterir. Bunun dışında HTML kodlarını da içerebilir. Böyle bir dosyada <% tagıyla ile başlayıp %> ile biten satırlar VBScriptle yazılmış ASP kodlarını gösterir ve sunucuya bu aralıktaki komutların sunucuda çalıştırılmasını söyler. Bu aralığın dışındaki HTML kodları ise kendi browserında yorumlanmak üzere kullanıcıya gönderilir.

4. VBScript

Visual Basic dili ile VBScript oldukça benzeşen iki dildir. VBScript, güçlü bir dildir; ancak Netscape firmasının hiç bir zaman Browser'ında istemci tarafında çalıştırılabilecek diller arasında kabul etmemesi sebebiyle VBScript, Web'in istemci tarafında etkili olamamıştır. MS'un Browser'ı Internet Explorer ise VBScript ile yazılan İstemci-Tarafı kodları okuyabilir ve icra edebilir.

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

VBScript'in hemen hemen bütün komutların ve yöntemleri ASP'de kullanılabilir. Ancak bunun bir kaç kısıtlaması vardır. VB veya VBScript'te mesaj kutusu (MsgBox) ve girdi kutusu (InputBox) aracılığı ile programlara kullanıcının bilgi girmesi sağlanabilirken, bu iki komut ASP içindeki VBScript kodunda kullanılamaz. Ayrıca ASP teknolojisi zaten VBScript'in bütün komutlarını ve deyimlerini kullanılmasına gerek bırakmaz.

ASP sayfalarındaki HTML kodları ile VBScript (veya diğer Script dillerinin) kodlarını birbirinden ayrılmalıdır. Bu ASP.DLL'ye, HTML'in nerede bittiğini, Script diliyle yazılmış kodun nerede başladığını gösterilebilmesi için gerekir. Bunu sağlamak için Script diliyle yazılmış her şey "<%>" ve "%>" işaretleri arasına alınır. ASP.DLL bu işaretleri görünce, içindekileri "yazmak" yerine "yapar." Bir ASP sayfasında HTML'in klasik "<" ve ">" işaretleri arasındaki unsurlar, ASP.DLL

tarafından ziyaretçiye gönderilecek olan sayfaya aynen aktarılır; ancak "<%>" ve "%>" arasındaki her şey, başta belirtilen LANGUAGE etiketinde yazılı Script dilinin yorumlayıcısına verilir; yorumlatılarak, gereği yerine getirilir.

"<%>" ve "%>" işaretlerine "sınırlayıcı" denir. Sınırlayıcının içinde bir veya daha çok satır kod bulunabilir. Sınırlayıcılar ve içindeki Script, HTML etiketlerinin içinde veya dışında yer alabilir. Sınırlayıcının içindeki kodları açıklamak için eklenen yorum satırlarının başına tek tırnak işareti (') konulur. Bu kuralların uygulandığı bir ASP sayfası örneği:

```
<% @LANGUAGE=VBScript %>
<html>
<head>
<title>Hoşgeldiniz!</title>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
</head>
<body>
<center>
<%
' Yazı tipi boyutunu tutacağımız bir değişken
tanımlayalım
Dim fontBoyut
%>
<%
' yazı tipi boyutunu 1'den 7'ye kadar değiştirelim
For fontBoyut = 1 To 7
%>
<font size = <%=fontBoyut%>>
Hoşgeldiniz!<br>
<% Next %>
</center>
<h3>Bugün <%=WeekdayName(Weekday(Date)) %>, <%= Date %>.
Şu anda Server'da saat: <%= Time %>.<p>
</h3>
</body>
</html>
```

Burada sınırlayıcı arasında tek veya çok satırlı VBScript kodları ile başında tek tırnak olan icra edilmeyen, yorum satırları görülmektedir. HTML etiketinin içine gömülmüş VBScript kodu ise HTML'in etiketinde yer alır: <font size = <%=fontBoyut%>>. Burada karşılaşılan "<%=>"

ifadesi, ASP'ye, "Bu deęişkenin deęerini bul ve tam buraya yaz!" denmesini saęlar. Bu ifade Response.Write metodunun kısaltılmıř halidir.



Resim 1 Sınırlayıcıların Test Edildięi ASP Programının ıkıřı

4.1 VBScript'te Yazım Kuralları

VBScript komutları, anahtar kelimeleri ve deęişken adlarının büyük harf-küçük harf olması önemli deęildir. Yani yukarıdaki ifadelerden biri řu biimlerden birinde de yazılabilir; kod yine de alıřırdı:

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

Eęer bir kod satırı ok uzun geliyor ve daha sonra anlaşılması imkansız derecede uzuyorsa, bu satırı altizgi (_) ile ařaęı satırda devam edilebilir. Örnek:

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


4.2 VbScript'te Değişkenler

VbScript'te, bir çok başka bilgisayar programlama dilinden farklı olarak değişkenlerin tanımlanması veya "beyan edilmesi," "boyutlandırılması" gerekmez. Belirtilmemiş, önceden tanımlanmamış bir değişkene değer atamaya kalkılırsa, VbScript bunu mükemmel şekilde kabul eder. Fakat bu kötü bir programcılıktır. İyi programcılık değişkenlerin önceden beyan edilmesini gerektirir. Bu DIM komutuyla yapılır. DIM, Dimension (boyutlandır) kelimesinden kısaltılmıştır. Bu komut bilgisayarın değişken yeri olarak bir bellek alanının boyutunu belirtmesini sağlar. Örnekler:

```
<%  
DIM Gun, Ay, Ogrenci, Not  
Gun = "Pazartesi"  
Ay = "Ocak"  
Ogrenci = "Necip"  
Not = 5  
%>
```

Değişken isimleri, mutlaka harfle başlamalıdır; içinde noktalama işaretleri bulunamaz ve uzunluğu 255 karakteri geçemez.

4.3 VbScript'te Array Fonksiyonu

VbScript'in kullanılmaya hazır bir çok fonksiyonu vardır; bunlardan biri olan Array ile, kolayca dizi değişken oluşturulabilir. Diyelim ki, Gunler(7) dizi-değişkenini gün adları ile doldurarak oluşturulmak isteniyor:

```
<%  
Dim Gunler = Array ("Pazartesi" , "Salı" , "Çarşamba" ,  
"Perşembe" , "Cuma" , "Cumartesi" , "Pazar")  
%>
```

ile hem dizi-değişkeni oluşturulabilir; hem de değerlerini atanabilir. Bu suretle oluşturulan dizi değişkenin üyelerine daha sonra sıra numaraları ile atıfta bulunulabilir. Örneğin:

```
<%=Gunler(6)%>
```

bize Pazar'ı verir. Çünkü dil tasarımcıları, sayı saymaya 1'den değil 0'dan başlarlar; dolayısıyla Gunler dizi-değişkeni Gunler(0)'dan başlar.

Dizi değişkenlerde bir değişkene numarasıyla atıfta bulunabilirsiniz. Numara yerine bir başka değişken kullanılabilir. Örneğin, OgrenciNo değişkeninin değeri 12 ise

```
<%  
....  
If Ogrenciler(OgrenciNo) = "Necip"
```

...
<%>

şeklinde bir döngü ile aranılan öğrencinin Necip olup olmadığını sınanabilir.

Başka programlama dillerinde bir değişkenin değeri harf ve rakamlardan oluşuyorsa, yani matematik işlem yapmaya elverişli değilse bunlara **String** (Alfanümerik, karakter değerler) denir. Programlama dillerinde bir de matematik işlem yapmaya elverişli değişken türü vardır: Sayı (**Number**). VBScript, bir değişkene alfanümerik (karakter, metin) olarak atadığınız değeri çift tırnak içine almanızı ister. Sözelimi **Ogrenci(12)** değişkeni için Necip değerini atamak isteniyorsa Necip kelimesi çift tırnak içine alınmak zorundadır. Sayı olarak kullanılacak değerler ise tırnak içine alınmaz. Fakat VBScript açısından değişken türü diye bir şey yoktur. Visual Basic, küçültülüp Script dili haline getirilirken, değişken türü özelliği alınmayınca ortaya böyle bir durum çıkmıştır. Karışıklık, çift tırnak içinde verilen bir değer matematik işlemde kullanmaya kalkılması ile ortaya çıkabilir. Rakam olmayan bir karakter-dizisini bir değişkene tırnaksız olarak atadığında VBScript "tanımsız değişken" verilmeye kalkıldığını söyleyerek, duracaktır.

VBScript'in bu eksikliğini gidermek için bazı ASP programcıları değişken adlarının önüne karakter-dizileri için "str" harflerini koyarlar: strAy, strOgrenciler, gibi.

```
<% Option Explicit %>  
<HTML>  
<%  
Dim Degisken(2), Toplam  
Degisken(1) = "Necip"  
Degisken(2) = "Dayanır"  
Toplam = degisken(1) + Degisken(2)  
>%>  
<% =Toplam %>  
</HTML>
```

Sonuç: "NecipDayanır" dır.

VBScript'in kullandığı tek tür değişkene variant denir. Variant, karakter-dizini (**String**) de olabilir, sayı (**Number**) da. Fakat bir variant'ın içindeki değer, veri olarak nitelendiğinde şu türlerden birine girer:

Boolean	Değişkenin değeri ya True (doğru) ya da False (yanlış) olur; True'nun değeri -1, False'un değeri ise 0'dır.
Byte	0 ile 255 arasında bir sayısal değer olabilir.
Double	Yüzen noktalı değer denilen veri türüdür. Pozitif sayılar için 4.9E-324 ile 1.8E308 arasında, negatif sayılarda -4.9E-324 ile -1.8E308 arasında bir değer alabilir.
Date/Time	Önceden belirlenmiş biçimlerde tarih ve zaman bilgisi içerir.
Empty	Tanımlanmış ancak henüz değer atanmamış (boş) değişken türüdür.
Error	Programın hata mesajlarını ve hata değerlerini tutar.
Integer	Noktalık bölüm içermeyen tamsayı verilerdir; değeri -32.768 ile +32.767 arasında olabilir.
Long	Noktalık bölüm içermeyen tamsayı verilerdir; değeri -2.147.483.648 ile 2.147.483.648 arasında olabilir.
Null	İçinde veri bulunmamak üzere tanımlanmış değişkenlerdir.
Object	Windows OLE Nesnelerini tutmak üzere tanımlanmış değişkenlerdir.
Single	Yüzen noktalı değer denilen veri türüdür. Pozitif sayılar için 1.4E-45 ile 3.4E38 arasında, negatif sayılarda -1.4E-45 ile -3.4E38 arasında bir değer alabilir.
String	Alfanumerik veri tutan değişkenlerin değeridir

Tablo 4.1 VBScript Değişken Türleri

4.4 VBScript'te İşlemciler (Operatörler)

Operatör	İşlev	Sınıfı
+	Toplama	Aritmetik
-	Çıkartma	
*	Çarpma	
/	Bölme	
^	Üssünü alma	
\	Tamsayı bölme	
Mod	Modüler aritmetik	
=	Bir değişkenin diğerine eşit olduğunu sınar	Karşılaştırma
<>	Bir değişkenin diğerine eşit olmadığını sınar	
>and<	Bir değişkenin diğerinden büyük veya küçük olduğunu sınar (and kelimesi var)	
>= and <=	Bir değişkenin diğerinden büyük veya eşit, veya küçük veya eşit olduğunu sınar (and kelimesi var)	

Is	Bir ifadedeki iki referansın aynı Nesne'ye yapılıp yapılmadığını sınar	
And	Bir veya daha fazla değişkeni test olarak karşılaştırır	Mantıksal
Or	Bir işlemin devamı için hangi koşulun oluşması gerektiğini sınar	
Not	Bir ifadeyi negatif hale getirir	
XoR	Sadece bir koşulun doğru olup olmadığını sınar	
Eqv	İki değişkenin eşitliğini sınar	
Imp	İki ifadede mantıksal implikasyon işlemi yapar.	

Tablo 4.2 VBScript'te Operatörler

VBScript ile yazılan ASP sayfalarında bu işlemcilerin beklenen sonucu verebilmesi için kullanım sıraları önemlidir. VBScript'te bu sıralama hesap makineleri ile aynı mantıkla işler. Yani aritmetik işlemlerin öncelik sırası şöyledir:

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

Tablo 4.3 VBScript Operatör Hiyerarşisi

VBScript ile hesap işlemi yaparken, aritmetik işlem sırasını karıştırarak hatalı sonuç almamak için sıklıkla parantez kullanmak gerekir.

4.5 VBScript'de Program Kontrolü

İster Script diliyle, ister gerçek programlama diliyle yazılsın, bir bilgisayar programının varlık sebebi, çeşitli durumları değerlendirerek, belirli durumlarda belirli kararlar verebilmesidir. Bunu programın kontrol öğeleri kullanılarak yapılır. Programlar, bu öğeler sayesinde karşılaştırma yaparlar; belirli durumların oluşup oluşmadığını sınarlar; veya belirli bir durumun oluşmasına veya sona ermesine bağlı olarak bir iş yaparlar veya yapmazlar. Bu sınamalarla (koşullu ifadelerle) veya döngülerle sağlanır. Kimi zaman da, programa (programın mantığı çerçevesinde) istenilen anda

yapmakta olduđu işi durdurarak, başka bir işi yapması bildirilebilir. Bunlara da **Süreçler** (veya Prosedürler) denir.

4.6 VBScript'te Mantıksal Sınamalar

VBScript'te programın karar vermesini sağlayan ilk kontrol ögesi "eğer ... ise... yap!" şeklinde özetlenebilir. VBScript bu işlemi iki ayrı ifadeyle yapar:

4.6.1 If.. Else

VBScript'in verilen bir durumun bulunup bulunmadığını sınamasını sağlar. Genel yazım kuralı şöyledir:

```
If şart Then
    [şart doğru ise yapılacak işler]
Else
    [şart doğru değilse yapılacak işler]
End If
```

Bir örnekle ifade edilirse: Eğer saat 12'den önce ise sayfaya "Günaydın" , saat 12'den sonra ise "Tünaydın" , saat 18'den sonra ise sayfaya "İyi akşamlar!" yazdırılması istenirse.

```
<HTML>
<HEAD>
<TITLE>ASP İLE SAATE GORE 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>
<%
If Hour(Now) <12 Then
    Response.Write "Günaydın! "
ElseIf Hour(Now) >= 18 Then
    Response.Write "İyi akşamlar! "
```

```
Else
    Response.Write "Tünaydın! "
End If
Response.Write "<BR>"
Response.Write "Site Onarım Sitesine Hoşgeldiniz"
%>
</CENTER>
</H2>
</BODY>
</HTML>
```

4.6.2 Select Case

VBScript'in bir diğer duruma bakarak karar verme ifadesi, **Select Case** (Durum Seç) yapısıdır. Bu kontrol ögesinin çalışması şöyle özetlenmiştir:

Durum Seç (Durumların listesi veya durumları belirten bir değişken)

Durum 1 : Yapılacak işler

Durum 2: Yapılacak işler

Durum 3: Yapılacak işler

Durum n: Yapılacak işler

Seçmeyi Bitir

VBScript, verdiğiniz durum listesine veya içinde çeşitli değerler bulunan değişkene bakarak, bu değişkenin her bir değerini bir "durum" sayacak ve verdiğiniz durumlardan hangisini tutuyorsa, ona ait komut dizisini icra edecektir. Yukarıdaki örneği bu kez bu yapıyı kullanarak yazarsak.

```
<HTML>
<HEAD>
<TITLE>ASP İLE 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 "Günaydın!"
Case 12,13,14,15,16,17
Response.Write "Tünaydın"
Case Else
Response.Write "İyi Akşamlar!"
End Select
Response.Write "<BR>"
Response.Write "Site Onarım Sitesine Hoşgeldiniz"
%>
</CENTER>
</H2>
</BODY>
</HTML>
```

Select Case komutuna, içindeki değerleri "durum" sayacağı dizi veya değişken olarak VBScript'in kullanılmaya hazır fonksiyonlarından **Hour(Now)**'ı verilmiştir. Bu fonksiyondan, 0 ile 24 arasında bir değer dönecektir. Bu değer **Select Case** için bir durum demektir. **Select Case**, bu değer ile altta sıralanan **Case**'leri karşılaştıracak ve elindeki değer hangi **Case**'i tutuyorsa ona ait komutları icra edecektir. Sonuncu **Case**'de **Case** olarak **Else** (başka) verilmiştir. Bu 17'den 23'e kadar olan saatleri sıralamaya gerek bırakmaz. 0'dan 11'e kadar olan saatlerle 12'den 17'ye kadar olan saatleri sıraladığına göre başka hangi saat olursa olsun, ziyaretçi "İyi akşamlar!" mesajıyla karşılaşır.

4.6.3 For..Next döngüsü

Programın bir işi belirli kere yapması isteniyorsa, ona yapacağı iş bir sayaç değişkeniyle birlikte, **For** döngüsüyle bildirilir:

```
For sayaç = başlangıç To son Step adım
```

```
...yapılacak işler...
```

```
Next
```

Burada, "sayaç" yerine istenilen bir değişken adını, "başlangıç" yerine sayacın başlaması istenilen sayı, "son" yerine sayacın durmasını istenilen sayı, ve "adım" yerine, sayacın kaçır-kaçır artmasının istendiği yazılabilir. En sondaki **Next** deyimini ise döngünün bir sonraki adıma geçmesini sağlar. Bu adımda sayaç, **Step** kelimesi varsa, karşısındaki değer kadar arttırılır ve yapılacak işler yeniden yapılır.

```
<HTML>
<HEAD>
<TITLE>ASP İLE GÜNLERİ 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", "Salı", "Çarşamba", "Perşembe", "Cuma", "Cumartesi", "Pazar")
For sayac = 0 to 6
    Response.Write Gunler(sayac)
    Response.Write "<BR>"
Next
%>
</CENTER>
</H2>
</BODY>
</HTML>

```

Bu kodda sayaç 0 iken ekrana “Pazartesi” yazılacak, daha sonra sayaç bir artacak ve karşılık gelen “Salı” ekrana yazılacak, bu işlem 6 değerine ulaşılan kadar devam edecek ve son bulacaktır.

4.6.4 While...Wend

Program mantığı bazen bize böyle açık ve seçik bir sayaç kurma imkanı vermez. Sayaç olarak kullanılan değer, programın başka bir bölümü tarafından üretiliyor olabilir veya bu değer ziyaretçi tarafından belirlenmiş olabilir. Özetle yapılmasını arzu edilen işin ancak sayaç bir değerden azsa, çoksa veya eşitse yapılması, bu durum değişirse durması istenebilir. Bu **While** (..iken) komutuyla yapılabilir. **While** döngüsünü kullanıldığı zaman sayacın artırılması gerekir. Sözgelimi, yukarıdaki programın 7 günün tümünü ekrana yazmasını değil de, mesela gün sayısı 5'den küçük ise yazmasını isteniyor olabilir. Bu durumda kodda **For.. Next** arasında kalan bölümde şu değişiklik yapılmalıdır:

```

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

```


Wend

Burada **While** döngüsü **Wend** kelimesiyle sonlandırılmıştır. **While** satırındaki sayacı değiştirildi, programın sayaç 5'den küçük veya 5'e eşit iken işlemesi sağlandı.. For'dan farklı bir diğer ifade ise sayacı arttıran "sayac = sayac + 1" ifadesidir.Bu "sayac'ın o andaki değerini al, 1 ile topla ve bulduğun yeni değeri sayacın mevcut değerinin yerine yaz" demektir. VBScript sayacı bir arttırdıktan sonra önce **While** satırındaki şartın gerçekleşip gerçekleşmediğine bakar; gerçekleşmiş ise **Wend**'i izleyen ilk satıra gider; gerçekleşmemişse **While** döngüsünün içindeki işi yapmaya devam eder.

4.7 VBScript'te Sık Kullanılan Hazır Fonksiyonlara Örnekler

Hemen hemen tüm programlama dillerinde olduğu gibi VBScript'te de kullanılabilecek hazır-fonksiyonlar mevcuttur.

4.7.1 Tarih ve saat

Web'in zamana çok bağlı oluşu dolayısıyla, Visual Basic'in hemen hemen bütün zaman-tarih fonksiyonları VBScript'te de kullanılır.

Date: Bugün tarihini verir. (25.04.2005 gibi)

Time: O andaki saati verir. (22:24:40 gibi)

Now: O andaki tarih ve saati birlikte verir. (25.04.2005 22:24:40 gibi)

VBScript'in buna ek olarak **Weekday** (haftanın günü), **WeekdayName** (günün adı) ve **Monthname** (ayın adı) fonksiyonları da vardır. Bu fonksiyonlar değerlerini **Date** fonksiyonuna göre alırlar. Örneğin,

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

komutu bugün Cumartesi ise "Cumartesi" değerini verir.

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

komutu bu ay Nisan ise "Nisan" değerini verir. VBScript'in bunlara ek olarak **Day** (gün),

Month (ay) ve **Year** (yıl) fonksiyonları da değerlerini **Date** fonksiyonundan alarak,birer rakam olarak bildirirler. Eğer tarih 25 Mart 2000 ise:

```
<%= Day(Date)%>... 25
```

```
<%= Month(Date)%>... 4
```

```
<%= Year(Date)%>... 2005
```

değerini verir. VBScript, bu değerleri doğruca işletim sisteminden alır. Dolayısıyla işletim sisteminin bölgesel ayarları Türkiye için yapılmışsa, gün adları Türkçe olarak dönecektir. Ayrıca, tarih ve saat biçimleri de bölgesel ayarlara bağlı olarak, ay önde, gün arkada veya tersi, saat de 12 saat veya 24 saat esasına göre döner. ASP programları kişisel Web Server'da denenirken kullanılan bilgisayarın

tarih ve saati; gerçek Internet'te çalıştırırken Server'ın tarih ve saati alınır. Sayfada ay ve gün adlarını Türkçe görüntülemek için, önce Server'ın bölgesel ayarlarını sınanmalı ve eğer isimler Türkçe gelmiyorsa, bunları çeviren Sub'lar veya fonksiyonlar yazılması gerekir.

4.7.2 Karakter-dizisi Düzenleme

Karakter-dizisi veya **String**, VBScript için herşey olabilir. "Fırat Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü" bir **String**'dir. "Bugün 25 Nisan 2005" bir **String**'dir. "Doğum Günün Kutlu Olsun!" bir **String**'dir. Web sitesini ziyaret eden kişinin formlara yazacağı ve Gönder tuşunu tıklayarak Server'a göndereceği bilgiler **String**'dir. Fakat bunların hepsi olduğu şekliyle işimize yaramaz. Bunları yeniden düzenlemek, içinden seçmeler yapmak veya biçimlerini değiştirmek gerekebilir. VBScript bu amaçla kullanılmaya hazır bir dizi fonksiyon verir:

InStr Uzun bir String'in içinde vereceğiniz daha kısa bir String'in bulunup bulunmadığını arar; bulursa bu kısa String'in başlama noktasının değerini verir. Diyelim ki, "Fırat Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü" String'ini Elazig değişkenine, "Elektrik" kelimesini de Ara değişkenine atandı. InStr fonksiyonu ile Ara'nın değerinin yeri Elazig'in değerinin içinde bulunabilir:

Yer = InStr(Elazig, Ara)

Yer'in değeri 20 olacaktır; çünkü "Elektrik" kelimesi, uzun **String**'de 20'nci karakterden başlamaktadır.

Len Bir **String**'in uzunluğunu belirler. Yukarıdaki örnekte yer alan Elazig değişkeninin uzunluğu şöyle belirlenebilir:

Uzunluk = Len(Elazig)

Uzunluk değişkeninin değeri 58 olacaktır.

UCase Verilen bir **String**'in tüm karakterlerini büyük harfe çevirir.

YeniString = UCase(Elazig)

Yeni String'in değeri: "FIRAT ÜNİVERSİTESİ ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ" olacaktır.

LCase Verilen bir String'in tüm karakterlerini küçük harfe çevirir.

YeniString = LCase(Kayahan)

Yeni String'in değeri: " firat üniversitesi elektrik-elektronik mühendisliği bölümü " olacaktır.

LTrim, Verilen **String**'in (sırasıyla) solunda yani baş tarafında; sağında yani sonunda ve hem başında

RTrim, ve hem de sonundaki boşlukları temizler.

Trim

- Space** İçinde, verilen sayı kadar boşluk olan boş bir **String** oluşturur. Örneğin
 Bosluk = Space(20)
 Bosluk değişkeninin değeri " " (20 boşluk) olacaktır.
- String** İstenilen sayıda ve istenilen bir karakterle bir **String** oluşturur.
 YeniString = String(3, "*")
 YeniString değişkeninin değeri "***" olacaktır.
- Left, Right** Bir **String**'in içinde soldan (baştan) veya sağdan (sondan) verilen sayıya kadar olan karakterleri verir. Örneğin, yine yukarıdaki Elazig değişkeni kullanılırsa:
 Solda = Left(Elazig, 5)
 Solda değişkeninin değeri "Fırat" olacaktır; çünkü Elazig değişkeninin soldan itibaren beş harfi "Fırat" kelimesine denk gelir.
- Mid** Bir String'in içinde başlangıç noktası ve karakter olarak boyu verilen alanda yer alan **String**'i verir.
 Ortada = Mid(Elazig, 20, 8)
 Ortada değişkeninin değeri "Elektrik" olacaktır; çünkü Elazig değişkeninin soldan 20'nci değişkeninden itibaren 8 karakterlik alanda "Elazig" karakterleri yer almaktadır.

Tablo 4.4 VBScript String Fonksiyonları

4.7.3 Test Fonksiyonları

VBScript'te kullanılan bazı değişkenlerin o andaki durumu, programın akışını kontrolde kullanılacak bilgiyi sağlayabilir. Sözelimi bir değişkenin değeri boş ise, ziyaretçinin formu tam olarak doldurmadığını düşünülebilir. VBScript değişkenlerin durumunun sınanması için bazı özel fonksiyonlar sağlar. Bu özel fonksiyonlardan dönen değer **True** (doğru) veya **False** (yanlış) olur; doğru sonucun değeri -1, yanlış sonucun değeri ise 0'dır:

- isArray** Bir değişkenin dizi-değişken (Array) olup olmadığını sınar.
- isDate** Bir değişkenin değerinin tarihe (Date) çevrilip çevrilemeyeceğini sınar.
- isEmpty** Bir değişkenin tanımlanıp değer atanmış olup olmadığını sınar.
- isNull** Bir değişkenin geçerli bir değer tutup tutmadığını sınar.
- isNumeric** Bir değişkenin sayı olarak işleme tabi tutup tutulamayacağını sınar
- isObject** Bir ifadenin geçerli bir ActiveX veya OLE nesnesine referansta bulunup bulunmadığını sınar.
- typeName** Bir değişkenin türünü belirtir.
- varType** Bir değişkenin türünü belirten sayıyı verir.

Tablo 4.5 VBScript Test Fonksiyonları

KAYNAKLAR:

1. Nesne Tabanlı Programlama, Faruk ÇUBUKÇU,2002
2. www.farukcubukcu-bt.com
3. www.15seconds.com
4. www.aspwire.com
5. asp.superexpert.com
6. ASP ile Web Yazılımı Geliştirmeye Giriş, Yrd.Doç.Dr.Hasan H.BALIK,2004
7. www.asp101.com