08217
Internet Programcılığı I

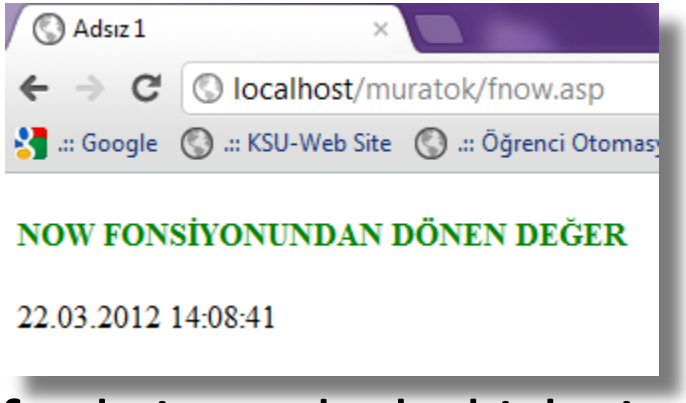
How to ASP Language

Elbistan Meslek Yüksek Okulu
2015– 2016 Güz Yarıyılı

Öğr. Gör. Murat KEÇECİOĞLU

Dönen Değer

- Fonksiyonlar, kendilerini göreve çağıran VBScript komutlarına ve işlemlerine bir değer sunarak karşılık verirler.
- Buna fonksiyondan dönen değer denir. Diyelim ki Now() fonksiyonunu göreve çağırdınız.



- Bu fonksiyon derhal işletim sisteminden saati ve tarihi öğrenerek kendisini göreve çağıran işleme bildirir.

Karar Süreçleri: Select Case

- VBScript'in bir diğer duruma bakarak karar verme ifadesi, Select Case (Durum Seç) yapısıdır. Bu kontrol öğesi;

- Durum Seç
  - Durum 1 : Yapılacak işler
  - Durum 2: Yapılacak işler
  - Durum 3: Yapılacak işler
  - ….
  - Durum n: Yapılacak işler
- Seçmeyi Bitir

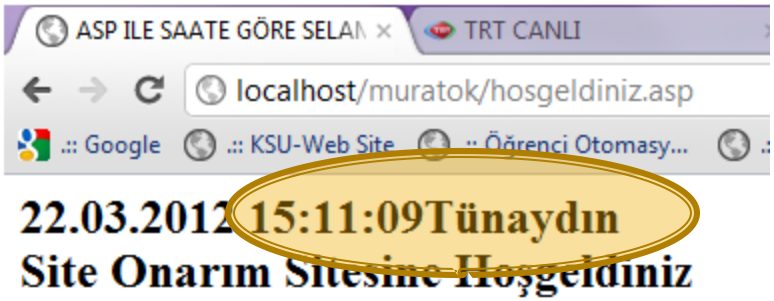
Karar Süreçleri: Select Case

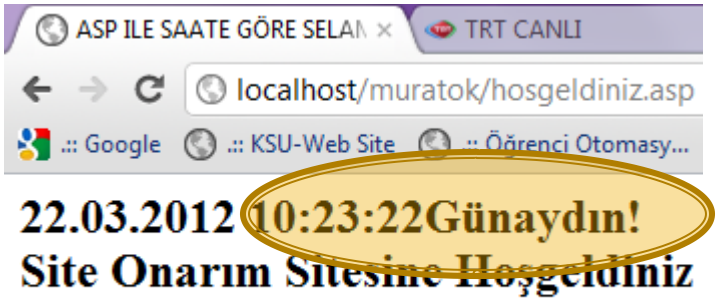
- VBScript, verdiğiniz durum listesine veya içinde çeşitli değerler bulunan değişkene bakacaktır.
- Değişkenin her bir değerini bir “durum” sayacak ve verdiğiniz durumlardan hangisini tutuyorsa, ona ait komut dizisini icra edecektir.
- Şimdi asp sayfamızı bu kez bu yapıyı kullanarak yazalım (hosgeldiniz.asp):

Karar süreçleri: Select Case

```
<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><%Response.Write now()
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"
%></H2>
</BODY>
</HTML>
```

Karar Süreçleri: Select Case





Sistem saatine göre
Fonksiyon değer üretecektir.

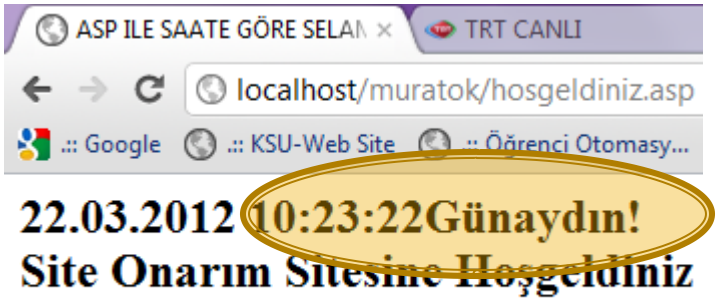
Karar Süreçleri: Select Case

- Select Case komutuna, içindeki değerleri “durum” sayacağı dizi veya değişken olarak VBScript’in kullanılmaya hazır fonksiyonlarından Hour(Now)’ı veriyoruz.

- Bu fonksiyondan, 0 ile 24 arasında bir değer dönecektir.

Karar Süreçleri: Select Case

- 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.



Karar Süreçleri: Select Case

- Eğer 24'den sonra ve 04'den önce ziyaretçinize "İyi geceler!" dilemek isterseniz, bu programı nasıl değiştirirdiniz?

Bunu da uygulama olarak yapalım.

Döngüler:

- Karar sınaması bir programın akışını kontrol için kullanacağımız birinci en önemli unsur ise, döngü de ikinci en önemli unsur sayılır.

- Hatta programcının tembellik katsayısına göre, belki de birinci en önemli unsuru bile sayılabilir!

Döngüler:

- Döngü (Loop) programa, bir işi biteviye yaptırmaya yarar.
- Tabiî bu iş sonsuza kadar sürecek olursa, buna Endless Loop (Sonsuz Döngü) denir.
- En iyi program ve Windows dondurma yöntemidir!

Döngüler: For..Next

- Programın bir işi belirli kere yapmasını istiyorsak, ona yapacağı işi bir sayaç değişkeniyle birlikte, For döngüsüyle bildiririz:

  - For sayaç = başlangıç To son Step adım

    yapılacak işler

  - Next

Döngüler: For..Next

- Burada;
  - "sayaç" yerine istediğiniz bir değişken adını,
  - "başlangıç" yerine sayacın başlamasını istediğiniz sayıyı,
  - "son" yerine sayacın durmasını istediğiniz sayıyı, ve
  - "adım" yerine, sayacın kaçar-kaçar artmasını istediğinizi yazarız.

Döngüler: For..Next

- En sondaki Next deyimi ise döngünün bir sonraki adıma geçmesini sağlar.
- Bu adımda sayaç, Step kelimesi varsa, karşısındaki değer kadar arttırılır ve yapılacak işler yeniden yapılır.
- Bir uygulama gunler.asp yapalım.

Döngüler: For..Next

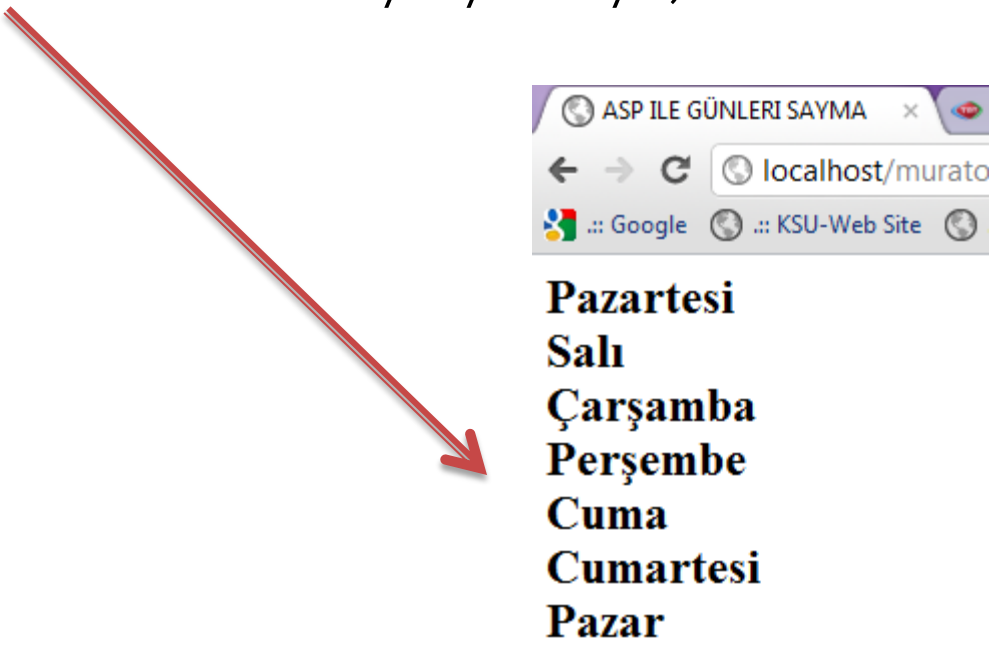
```
<HTML>
<HEAD>
<TITLE>ASP ILE GÜNLERI SAYMA</TITLE>
<META http-equiv="content-type" content="text/html; charset=ISO-8859-9">
<META http-equiv="Content-Type" content="text/html; charset=windows-1254">
</HEAD>
<BODY>
<H2>
<%
Dim Gunler
Gunler = Array("Pazartesi", "Salı", "Çarşamba", "Perşembe", "Cuma", "Cumartesi", "Pazar")
For sayac = 0 to 6
    Response.Write Gunler(sayac)
    Response.Write "<BR>"
Next
%>
</H2>
</BODY>
</HTML>
```

Sonucunu elde ederiz.

Döngüler: For..Next

Bu ASP kodunda, Gunler adıyla bir dizi-değişken oluşturuyoruz ve bu değişkenin yedi hanesine, günlerin adlarını atıyoruz.

Sonra, sayac adlı sayacı o'dan 6'ya kadar arttırıyoruz (Bir sayaç birer birer artsın istersek, Step bölümüne adım sayısı yazmayız).



Döngüler: While...Wend

- Ne var ki, program mantığı bazen bize böyle açık ve seçik bir sayaç kurma imkanı vermez.
- Sayaç olarak kullanacağımız değer, programın başka bir bölümü tarafından üretiliyor olabilir.
- Bu değer ziyaretçi tarafından belirlenmiş olabilir.

Döngüler: While...Wend

- Yapılmasını arzu ettiğimiz işin ancak sayaç bir değerden azsa, çoksa veya eşitse yapılmasını, bu durum değişirse durmasını isteyebiliriz.
- Bunu While (..iken) komutuyla yapabiliriz.
- While döngüsünü kullandığımız zaman sayacı bizim arttırmamız gerekir.

Döngüler: While...Wend

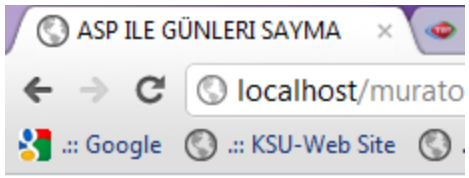
- Bir önceki örneğimizde 7 günün tümünü ekrana yazmasını değil de, mesela gün sayısı 5'den küçük ise yazmasını istiyor olabiliriz.
- Bu durumda kodumuzda For.. Next arasında kalan bölümde şu değişikliği yapabiliriz:

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

- Burada While döngüsünün Wend kelimesiyle sonlandırıldığına dikkat edin.

Döngüler: While...Wend

```
<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>
<%
Dim Gunler
Gunler = Array("Pazartesi", "Salı", "Çarşamba", "Perşembe", "Cuma", "Cumartesi", "Pazar")
For sayac = 0 to 6
    While sayac <= 5
    Response.Write Gunler(sayac)
    Response.Write "<BR>"
sayac = sayac + 1
Wend
Next
%>
</H2>
</BODY>
</HTML>
```



Pazartesi
Salı
Çarşamba
Perşembe
Cuma
Cumartesi

Döngüler: While...Wend

- While satırındaki sayacı değiştirdik, programın sayaç 5'den küçük veya 5'e eşit iken işlemesini sağladık.
-
- For'dan farklı bir diğer ifade ise sayacı arttıran "sayac = sayac + 1" ifadesidir.
- Bu ifade, ilk bakışta garip görünebilir. Fakat bilgisayar açısından bu "sayac'ın o andaki değerini al, 1 ile topla ve bulduğun yeni değeri sayacın mevcut değerinin yerine yaz!" demektir.

Döngüler: While...Wend

- 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.

Döngüler: Do..Loop

- Do (Yap) komutu ile kuracağımız döngüler iki ayrı türdür.
- Bu döngü ile bir dizi komutu, bir koşul doğru iken veya doğru oluncaya kadar yaptırabiliriz.
- Bu yöntemlerden her biri iki ayrı şekilde yazılabilir.

Döngüler: Do..Loop

- Bir koşul doğru iken bazı işlerin sürekli yapılmasını istiyorsak, Do While yöntemini kullanırız:

```
Do While koşul
    koşul doğru iken yapılacak işler
Loop
```

  - Bu ifade ile VBScript koşul doğru olduğu sürece istediğimiz işi yapacaktır.
  - Buradaki Loop kelimesi, döngünün başa dönmesini sağlar.

Döngüler: Do..Loop

- Bu yöntemden şu şekilde de yararlanabiliriz:
  - Do

    koşul doğru iken yapılacak işler
  - Loop While koşul

- Burada, Loop komutu şartın hâlâ doğru olup olmadığını sınar ve doğru ise verilen işleri yapar; koşul doğru değilse bir sonraki satıra geçer.

Döngüler: Do..Loop

- Döngünün bir şart gerçekleşinceye kadar bir işi yapmasını ise Do Until yöntemiyle sağlarız.

```
Do Until koşul
    koşul gerçekleşinceye kadar yapılacak işler
Loop
```

Döngüler: Do..Loop

- Bu ifade ile VBScript koşul doğru oluncaya kadar istediğimiz işi yapacaktır.
- Buradaki Loop kelimesi, döngünün başa dönmesini sağlar.

Döngüler: Do..Loop

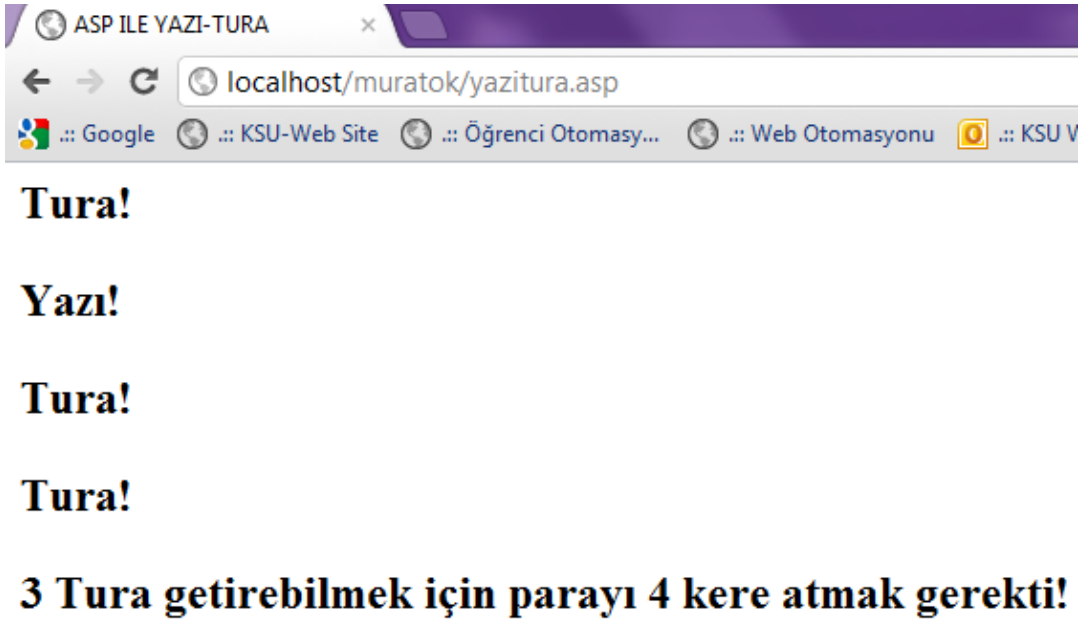
- Bu yöntemden şu şekilde de yararlanabiliriz:
  - Do

    koşul gerçekleşinceye kadar yapılacak işler
  - Loop Until koşul

- Burada, Loop komutu şartın henüz gerçekleşip gerçekleşmediğini sınar ve henüz gerçekleşmemişse verilen işleri yapar; koşul gerçekleşmişse bir sonraki satıra geçer.

Döngüler: Do..Loop

- Bu döngüye verilen klasik örnek, bilgisayara yazı-tura attırmaktır! Biz de ASP sayfamıza yazı-tura attırabiliriz.
- Bunun için sonraki slayttaki kodu yazın ve yazitura.asp adıyla kaydedip tarayıcıda çalıştırın.



Her Refresh Yaptığınızda 3 kere tura gelmesi İçin atış sayısı da Değişecektir.

Döngüler: Do..Loop

```
<% 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>
<%
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
%>
Yazı!<P>
<%
    Yazi = Yazi + 1
    Else
%>
Tura!<P>
<%
    Tura = Tura + 1
    End If
Loop
%>
3 Tura getirebilmek için parayı <%=Atis%> kere atmak gerekti!
</H2></BODY>
</HTML>
```

Döngüler: Do..Loop

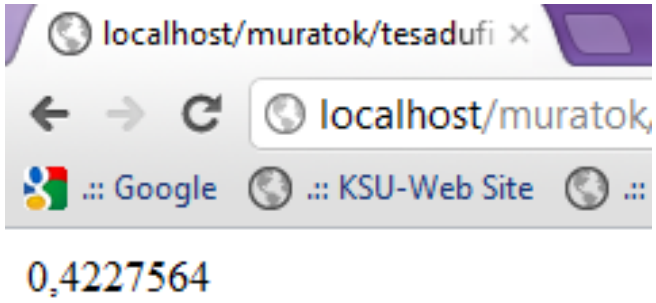
- Programımızın bütün işlemi Do döngüsü bölümünde yapılıyor ve bilgisayarın bir tesadüfi sayı üretmesi esasına dayanıyor.
- Bunu Randomize ve Rnd fonksiyonları ile yapıyoruz.
- Rnd'un verdiği tesadüfi rakamı, iki ile çarpıyor ve çıkan sayıyı 1 ile topluyoruz.

Döngüler: Do..Loop

- Böylece ortaya 1'den büyük 3'den küçük bir kesirli rakam çıkmış oluyor (Neden?).
- Bu rakamı Int() fonksiyonundan geçirerek, kesirinden kurtarıyoruz.

Randomize sayı üretmek :

- Daha önce bu uygulamayı yapmıştık şimdi bir kez daha yapalım. Randomize ve Rnd fonksiyonları



Her Refresh Yaptığınızda farklı bir ondalık basamaklı sayı gelecek.

tesadufisayi.asp*

```
1 <% OPTION EXPLICIT %>
2 <HTML>
3 <%
4 Dim TesadufiSayi
5 Randomize
6 TesadufiSayi = Rnd
7 %>
8 <%=TesadufiSayi%>
9 </HTML>
```

Tam Sayı Elde Etmek için: Int ve Round

- Rnd fonksiyonu ile ilgili örneği yaptığımızda, üretilen (dönen) sayının daima 0 ile 1 arasında, yani kesirli olduğunu görürüz.

- Bazen bizim sayfalarımızdaki hesaplamalar veya veritabanından alınan değerler de kesirli olabilir.

Tam Sayı Elde Etmek için: Int ve Round

- Örneğin öğrencilerin not ortalamalarını hesaplarken VBScript size sonu gelmez kesirler verecektir.
- Oysa çoğu zaman bu rakamların ya yukarı "yuvarlanması", ya da sadece tam sayı bölümü gerekir.

Tam Sayı Elde Etmek için: Int ve Round

- VBScript'te Int() fonksiyonu, bize bir sayının tam sayı bölümünü verir.
- Diyelim ki elimizdeki KesirliSayi değişkeninin değeri 123,234567 olsun.
  - Tamsayi = Int(KesirliSayi)
- işleminden sonra Tamsayi değişkenin değeri 123 olur.

Tam Sayı Elde Etmek için: Int ve Round

- Fakat kimi zaman bir sayının kesirli bölümünü böyle kesip atmak işimize gelmeyebilir.

- Round() fonksiyonu, kesirli bir sayıyı yukarı veya aşağı "yuvarlayarak" tam sayı haline getirir.

Tam Sayı Elde Etmek için: Int ve Round

- Bu kez elimizdeki KesirliSayi değişkeninin değeri 5,6 olsun.
  - Tamsayi = Round(KesirliSayi)
- işleminden sonra Tamsayi değişkenin değeri 6 olur.
- Kesirli sayı 5,2 ise, Round() fonksiyonu bize 5 değerini verir.

Dizi değişkenler için döngü: For Each..Next

- For..Next gibi çalışan bu özel döngü, sayaç değeri kullanmaz, fakat bir dizi değişkenin bütün değerleri için bir kere icra edilir.
- Dizi-değişkenler, VBScript ile yapacağımız işlemlerde önemli bir yer tutar.
- Örneğin bir sınıftaki öğrencilerin veya müşterilerimizin listesi bir dizi değişkenin elemanları olabilirler.

Dizi değişkenler için döngü: For Each..Next

- Yapmak istediğimiz işlem, dizi-değişkenin bütün elemanları için tekrar edilecekse, For Each..Next döngüsü daha elverişli olabilir.

- Bir dizi-değişkenin eleman sayısı ilerde değişirse ve siz döngüyü For..Next ile kurmuşsanız döngünün sayacı için verdiğiniz için alt ve üst sınırı değiştirmek zorunda kalırsınız.

- Oysa For Each, kaç kere tekrar edeceğine ilişkin değeri her zaman dizi-değişkenin elemanların sayısından alır.

Dizi değişkenler için döngü: For Each..Next

- Örneğin, bütün öğrencilerin listesini tutan Ögrenciler dizi-değişkeninin bütün elemanlarının değerini ekrana yazdıralım:

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

- Dizinin her elemanı için bir kez işlem yapılacağından, dizi içeriği artsa veya eksilse de dizi içerisindeki tüm elemanlar tarayıcıya yazdırılacaktır.

Döngüyü durdurmak istersek:

- Bir döngüden belirlediğiniz koşul gerçekleşsin veya gerçekleşmesin çıkmanız gerekebilir.
- Bu durumu bir başka değişkendeki değişiklik zorunlu kılabilir.
- Bir döngüden çıkmak için Exit (çık) ifadesini kullanabilirsiniz.
- Bu ifade, döngünün yaptığı işler arasında, genellikle bir If deyimi ile birlikte yer alır.

Döngüyü durdurmak istersek:

```
For sayac = 1 to 10
            [..bir takım işler yap..]
            If Degisken1 > Degisken 2 Then Exit For
            [..bir takım işlere devam et..]
Next
```

- Bu durumda For..Next döngüsü, Degisken1'in değerinin Degisken2'den yüksek olduğunu belirlerse, derhal döngüyü durdurarak, Next'ten sonraki satıra gidecektir.

- Do döngüsünden ise Exit Do ile çıkababiliriz. Bu ifadenin kullanımı da Exit For gibi olur.

08217 Internet Programcılığı I

Procedures and ETC. …Next on Week

Elbistan Meslek Yüksek Okulu

Öğr. Gör. Murat KEÇECİOĞLU