# ASP NEDIR?

# ICINDEKILER

#### [ASP NIN MANA VE EHEMMIYETI](#_bookmark0) 1

[GIRIS](#_bookmark1) 1

[ISTE SIZE ASP](#_bookmark2) 1

#### [PERSONAL WEB SERVER NE KI?](#_bookmark3) 3

[PWS YI KURDUN MU ?](#_bookmark4) 3

[GENEL MANZARA](#_bookmark5) 3

[GELISMIS'TEKI SIRLAR](#_bookmark6) 4

[SONUÇ](#_bookmark7) 4

#### [DEGISKENLERIMI KIMSEYLE DEGISMEM](#_bookmark8) 6

[ASP YI YAZMAYA BASLAYALIM](#_bookmark9) 6

#### [FORM'UNDA SAYFALAR YAZMAK](#_bookmark10) 8

[FORMDAN IÇERIK TALEBI](#_bookmark11) 8

#### [SORGU CÜMLECIKLERI KULLANMAK](#_bookmark12) 12

[SORGU CÜMLECIGI : QueryString](#_bookmark13) 12

.............................................................................................................................................................13

#### [AH VERITABANI AH!](#_bookmark15) 15

[ALANLARDA VARSAYILAN DEGER MESELESI](#_bookmark16) 16

[NEYIN VERISINI TABANLAYACAGIZ](#_bookmark17) 16

[BASLIYORUZ ORNEK VERITABANINA](#_bookmark18) 17

#### [ASP ILE VERITABANI BAGLANTISI](#_bookmark19) 20

[BAGLANMAK.](#_bookmark20) 20

[SORU ISARETLERI.](#_bookmark21) 22

[HIZLI SQL LAFDANLIGI](#_bookmark22) 23

[POPÜLER SQL LAFLARI](#_bookmark23) 24

#### [HOST'ES MESELESI](#_bookmark24) 26

[BEN HOST'ES ISTEREM](#_bookmark25) 26

[BELES OLSUN HA!](#_bookmark26) 26

[EWEBCITY NASIL BIR VARLIK, ANLATIR MISIN BANA?](#_bookmark27) 27

[DOMAINDLX: BIR NEFRET ABIDESI](#_bookmark28) 27

[HAYDI SAMBA'YA!](#_bookmark29) 28

#### [BIRAZ VBSCRIPT (I)](#_bookmark30) 30

[VBSCRIPT NEDIR?](#_bookmark31) 30

1. [SCRIPT DILI BILDIRIM SATIRI](#_bookmark32) 30
2. [HARF DURUMLARI](#_bookmark33) 30
3. [DEGISKEN BILDIRIMI](#_bookmark34) 31
4. [DEGISKENLERDE VERI TÜRLERI](#_bookmark35) 31

[VARIANT VE ALT TIPLERI](#_bookmark36) 32

[ALT TIPLER ARASINDA DONÜSÜMLER](#_bookmark37) 32

[CInt](#_bookmark38) 32

[CStr](#_bookmark39) 33

[ICINDEKILER](#_TOC_250000)

[CDate](#_bookmark40) 33

[BIRAZ VBSCRIPT (II)](#_bookmark41) 34

[DURUMLAR VE DAVRANISLAR](#_bookmark42) 34

[VBSCRIPT AKIS KONTROLLERI](#_bookmark43) 34

[IF...THEN...ELSE KONTROLÜ](#_bookmark44) 34

[SELECT CASE KONTROLÜ](#_bookmark45) 35

[DO...LOOP DONGÜ YAPISI](#_bookmark46) 35

[FOR..NEXT DONGÜ YAPISI](#_bookmark47) 36

[FOR EACH...NEXT DONGÜ YAPISI](#_bookmark48) 37

[WHILE...WEND DONGÜ YAPISI](#_bookmark49) 37

[SITEDE OTURANLARIN SAYISI](#_bookmark50) 38

[EY ORADAKILER! KAÇ KISISINIZ... TEKER TEKER GELIN](#_bookmark51) 38

[GLOBAL.ASA](#_bookmark52) 38

[SITEDE OTURANLARIN SAYISI](#_bookmark53) 40

[FORMDAN VERITABANI ISTIKAMETINE GIDEN KAYITLAR](#_bookmark54) 42

[VERITABANINDAKI TABLO VE ALANLARIN DURUMU](#_bookmark55) 42

[KAYIT EKLEYEN ASP DOSYASI](#_bookmark56) 43

[SORU ISARETLERI](#_bookmark57) 46

[SUB RUTINLER](#_bookmark58) 47

[FORMDAN ISTENEN IÇERIK VE DE KONTROLLERI](#_bookmark59) 47

[SIFIR UZUNLUK > NULL](#_bookmark60) 48

[VT BAGLANTISI](#_bookmark61) 48

[KAYIT SETI](#_bookmark62) 48

[ARTIK YENI BIR KAYIT EKLEYELIM HA?](#_bookmark63) 48

[VBSCRIPT ILE STRING DÜZENLEME](#_bookmark64) 50

[LEN FONKSIYONU](#_bookmark65) 50

[UCASE, LCASE FONKSIYONLARI](#_bookmark66) 51

[INSTR FONKSIYONU](#_bookmark67) 51

[REPLACE FONKSIYONU](#_bookmark68) 51

[LEFT, RIGHT FONKSIYONLARI](#_bookmark69) 52

[MID FONKSIYONU](#_bookmark70) 52

[TRIM, LTRIM, RTRIM FONKSIYONU](#_bookmark71) 53

[STRREVERSE FONKSIYONU](#_bookmark72) 53

[STRING FONKSIYONU](#_bookmark73) 53

[SPLIT FONKSIYONU](#_bookmark74) 53

[JOIN FONKSIYONU](#_bookmark75) 54

[VERITABANINDAKI KAYITLARI GÜNCELLEMEK](#_bookmark76) 55

[VERITABANINDAKI KAYITLARI GÜNCELLEMEK](#_bookmark77) 56

[KAYIT\_YENI.ASP](#_bookmark78) 58

[KAYIT\_SIL.ASP](#_bookmark79) 59

[ASRIN FIYAKASI: MAIL GONDERMEÇ!](#_bookmark80) 61

[DERS BASLIYOR, ONCE AMAÇ!](#_bookmark81) 62

ICINDEKILER

[FORM'UM](#_bookmark82) 62

[CDO ILE YOLLA.ASP](#_bookmark83) 63

[PERSIST ILE YOLLA.ASP](#_bookmark84) 64

[JMAIL ILE YOLLA.ASP](#_bookmark85) 65

[BIR ONERI](#_bookmark86) 66

[ASP VE KABUKLU YEMISLER](#_bookmark87) 69

[NERELERDE KULLANILABILIR?](#_bookmark88) 70

[KAYITLARI SAYFALAMAK (DB PAGING)](#_bookmark89) 71

[KAYITLAR ÇOK FAZLAYSA](#_bookmark90) 71

[BIR TAKIM AYRINTILAR.](#_bookmark91) 74

[HARIÇTEN GAZEL OKUMAK](#_bookmark92) 76

[INCLUDE ILE DOSYA EKLEME](#_bookmark93) 78

[BIR MENU ORNEGI](#_bookmark94) 79

[SUB'INA KADAR ASP](#_bookmark95) 82

[NEDEN ALT PROGRAM?](#_bookmark96) 82

[KIMIN DEGISKENI, KIMIN DEGISMEZKENI](#_bookmark97) 84

[SUB RUTIN VE PARAMETRELER](#_bookmark98) 86

[EXIT SUB](#_bookmark99) 89

[ACTIVEX DLL OLAYI](#_bookmark100) 91

[BASLIYORUZ.](#_bookmark101) 92

[PROJE.](#_bookmark102) 93

[DERLERKEN.](#_bookmark103) 95

[KAYIT.](#_bookmark104) 95

# ASP NIN MANA VE EHEMMIYETI

## GIRIS

Yetenekler yavas yavas yeryüzüne çikiyor arkadaslar. Artik internet herkesin, hepimizin oluyor farkinda misiniz? Her client(istemci) ayni düzeyde erisim ve gezinim hakkina sahip. Soz ve düsünce ozgürlügü var. Kendinizi sinif arkadasiniza bile ifade etmekte güçlük çekerken, karsimiza dev gibi dikilen bu zahiri alemde en kolay seyin bir “kendimi ifade edeyim” alani (homepage derler) yaratmak oldugunu gorünce sasirmistik.

Insanlar HTML kullaniyorlardi eski çaglarda. Yapilan webolojik çalismalar ve derin darbeler, Frig medeniyetinde “hypertext markup language” teriminin bilindigini, Urartu taraflarinda da HTML adli bir canavarin varligindan süphe edildigi gerçeklerini ortaya çikarmistir.

Meshurlugu eski çaglara kadar dayanan HTML neydi peki? Sadece insanlara yazdiklarimizi, çizdiklerimizi ve begendigimiz site koprülerini yerlestirdigimiz, envai çesit editorü bulunan, utanilmasa “Nesneye Yonelik Programlama Dili” gurubuna alinacak bir çesit kod akaidi. Excel’in içinden çikan, Word’de tek tusla ciziliveren tablolar HTML’de de var. Form ogeleri koyabiliyorsunuz. Yazi yazabiliyorsunuz (pes yani).

“Bu ask fazla sana”

Askimiz evrim geçirecek. Insanlar daha dogrusu webmimarlar, HTML yi tek tarafli metin ve grafik ogesi olmaktan çikarmak için çabaladilar. Microsoft’un evliyalari devreye girdiler. NT4 isletim sistemi mimarisine dayanan, Internet Information Server ‘ de çalisip, yeni bir ekmek kapisi olacak olan ASP yi gelistirdiler.

ASP artik Türkiye’nin yeni yüzü olacak! Herkes ASP kullanacak, herkes rahatlayacak.

## ISTE SIZE ASP

Neden hep ASP diyoruz. Bunu bir açmak gerek. ASP Ingilizce “Active Server Pages” tümcesinin kisaltmasidir. Ingilizce bilenler bunun “Aktif Sunucu Sayfalari” manasina geldigini anlamistir. Aslinda Türkçesi de mantikli geliyor insana. Sunucu tarafinda islenen sayfalar demek oluyor ASP.

Sunucu ne peki? Internette bilgi alisverisi bakkalla müsterisi arasindaki alisveristen çok da farkli degil. Istemci (Client) talip olandir; sunucu (Server) arz edendir. Genelde internete bagli bilgisayarlar (sizinki de muhtemelen bu genellemeye dahildir) istemci konumundadir. Bilgisayarinizin diger insanlara, yani dünyaya sundugu, arz ettigi bir meta yoktur. Sunucularin arz ettigi web sitelerini gezersiniz. Kendi web sitenizi olusturursunuz, ancak kendi sitenize yine istemci olarak girersiniz. Sunan kimdir? (Hypermart, Xoom... ama asla siz degil!)

ASP NIN MANA VE EHEMMIYETI 2 ASP NEDIR?

Bu sunucu−istemci ayrintisindan sonra sunu belirtelim: ASP sunucularda çalisir. Olusturdugunuz HTML dosyalarini üzerine tiklayarak browserle gorüntüleyebiliyordunuz. Çünki HTML belgeleri sadece browser tarafindan yorumlanmakta. Ancak HTML nin içine yeni kodlar yazmaya baslayacagiz bundan sonra. Bu kodlar ise oyle babadan kalma üzerine tiklama yontemiyle çalisamayacak maalesef.

Simdi gerekleri siralayalim:

Bilgisayarinizda ASP ile çalismak istiyorsaniz bilgisayarinizda ASP yi destekleyen web server programi yüklü olmalidir. (yani bilgisayariniz server davranisi gosterebilmelidir.) Soyle güzel bir web−browser gerekli

. (mümkünse IE tedarik edin) Metin Editorü veya herhangi bir HTML editorü bulunsun. Simdi bu maddeleri irdeleyelim :

Birinci maddede bahsedilen server programini sizi düsündürmesin, temin edilmesi çok kolay.

Alternatifleri denenebilir yalniz ben isletim sistemi olarak Win98 ve web server olarak da Microsoft Personal Web Server (PWS) üzerinde çalismanizi istiyorum. Amacimiz ASP ogrenmek ise boylesi yeter de artar bile. Peki nerde bu PWS. Win98 CD niz var degil mi? Orada iste. CD nin ana dizinindeki addons/pws klasoründen setup.exe ile PWS yi bilgisayariniza kurabilirsiniz. (Tabi bu islem win98 yüklendikten sonra yapiliyor.) Artik program kura kura setup kurdu olduguzdan size kurulumu anlatmaya gerek yok. Normal kurulum isinizi gorecektir.

Ikinci madde çok açik. Internet Explorer 4−5 tavsiyemdir. (“Microsoft’un dilinden ancak yine bir Microsoft anlar” mantigi...) Üçüncü maddeye ekleyecegim bir sey var. Her ne kadar basit bir text editorü ile HTML veya ASP kodlari yazilabilse de, güzel ve kullanisli programlar kullanmak bizim de hakkimiz. Ben HTML insasinda Allaire Homesite 4 (http://www.allaire.com) ve ASP kodu yazarken de EditPlus 2 (http://www.editplus.com) kullaniyorum. Homesite HTML hamalligimi en aza indiriyor ama onda ASP yazmak çok zevksiz geliyor bana. EditPlus 2 de bana gore çok güzel ve anlasilir bir renklendirme var. Bazi yetenekleri de çok kullandiginiz kod parçalarini tekrarlamaktan kurtariyor.

Evet. Artik alet edevat hazir olduguna gore baslayabiliriz demektir. Unutmayin, ASP ile çalismak için sadece Personal Web Server yeterli olacaktir. Biz extradan kod yazacagimiz editor tavsiye ettik.

Bir sonraki makalede PWS yi inceleyecegiz.

# PERSONAL WEB SERVER NE KI?

## PWS YI KURDUN MU ?

Bu makalede Microsoft Windows 98 CD si içinde gelen Personal Web Server'dan bahsedecegiz. Oncelikle bu programi sisteminize kurmaniz gerekiyor. Basit bir kuruluma sahip. Normal kurulum bize yeterli olacaktir.

Kurulumu tamamladiginizi varsayip basliyoruz.

## GENEL MANZARA

**Baslat > Programlar > Donatilar > Internet Araçlari > Personal Web Server > Personal Web Manager** ile veya **C:\WINDOWS\SYSTEM\INETSRV\pws.exe** ile ***Personal Web Manager*** programini açalim bakalim.

Bu program bizim Web sunucumuzu yonetmekte kulllanacagimiz "manager" dir diyecegiz. Neler yapabiliyoruz peki bununla?

Once programin açildiktan sonraki manzarasi ve endamini gorelim. Ilgili resim asagidadir.

Yukaridaki ekran gorüntüsünde ***"dur"*** yazan butonun ne ise yaradigini açiklayalim:

Bu buton serverinizin çalismasi veya durdurulmasi durumlari arasinda size geçis yaptiriyor. Su anda butonda dur yaziyor. Demek ki dügmeye tiklayinca serverin isleyisi duracak; bilgisayariniz sadece client olacak.

Butonun üzerindeki metin ise ***"baslat"*** (bildigimiz baslat) a donüsecek. Bu durumda butona tekrar tiklanirsa server aktif hale gelecek. Istemci bilgisayariniz, fukara bir server haline gelecek tekrar. Butonda "dur" yazisi belirecek. Hayat bu sekilde devam edecek.

[***"http://mucit"***](http://mucit/)ifadesi sunucunuzun aktif ana dizinidir. Web sitenizin giris sayfasi bu dizinde olmalidir. "mucit" benim sunucumun adidir. Buraya tiklarsaniz browser açilir ve [http://mucit](http://mucit/) adresini bulmaya çalisir. Sunucunuz aktif ise bulacaktir olagan olarak. Ve giris sayfasina bakacaktir. Eger giris sayfaniz varsa (ornegin default.asp) onu açacaktir. Burada "mucit" yerine localhost yazarsaniz ayni sonuçla karsilasirsiniz.

***"E:\inetpub"*** [ifadesi, sunucunuzun aktif dizininin "fiziksel yol" udur. Yani http://localhost](http://localhost/) adresi aslinda "e:\inetpub" a isaret etmektedir ve buradaki giris sayfasini açmaktadir. Web server kurulurken c:\inetpub\wwwroot yolunu sizin ana dizininiz yapar. Bunu "gelismis" seçenegiyle kolayca degistirebilirsiniz.

## GELISMIS'TEKI SIRLAR

1. **Sanal Dizinler** : Burada gordügünüz sanal dizin agacidir. Sanal dizinler, bilgisayarinizin fiziksel dizinlerinden farklidir. Adi üstünde sanal dizinlerdir. Burada birbirinden çok uzakta bulunan iki fiziksel klasorü, "home" anadizini altinda sanal dizin olarak belirleyebilirsiniz. "home" ifadesi sunucunuzun ismine tekabül eder. Benim sunucumun ismi "mucit" olduguna gore buradaki sanal dizinler, browserden <http://mucit/sanal_dizin_adi> seklinde yazilarak çagrilabilir. (veya <http://localhost/sanal_dizin_adi)>

Buradaki sanal dizin agacina istediginiz kadar sanal dizin ekleyebilirsiniz. Isterseniz ekleme isine bir goz atalim.

1. EKLE tusuyla yandaki pencereye ulasirsiniz. Ve buradan yeni bir sanal dizin ekleme firsatina kavusursunuz. Ben burada ornek olarak bilgisayarimdaki "C:\Belgelerim\cahitzarifoglu" klasorümü, "zarifoglu" adinda bir sanal dizin olarak atiyorum. Erisim bolümündeki izinlere hiç dokunmuyorum. Bizim için bunlar yeterli. Tamam a tiklayip ekle islemini bitiriyorum. Yeni ekledigimiz klasorün, sanal dizin agacinda gorünmesi lazim.

Su notu eklemek lazim. Sanal dizin olarak atadiginiz bir fiziksel klasorün alt klasorleri, otomatik olarak sanal dizin olacaktir. Adlari mi? Kendi adlariyla ayni olacaktir. Ornegin yukarida ekledigimiz klasorün alt dizini olan "C:\Belgelerim\cahitzarifoglu\resimler" klasorü, otomatik olarak kendini sanal dizin hissedecektir ve [sanal adresi de "http://mucit/zarifoglu/resimler"](http://mucit/zarifoglu/resimler) olacaktir.

Ozellikler tusu ile seçtiginiz sanal dizinin ozelliklerini degistirmek, Kaldir tusuyla da seçili sanal dizini kaldirmak mümkündür.

1. **Varsayilan Belgeyi Etkinlestirmek** [: Varsayilan belge ne demek? Browserden siteniz (http://mucit/)](http://mucit/)) veya bir alt dizini [(http://mucit/zarifoglu)](http://mucit/zarifoglu)) çagrildiginda sunucu o dizinde belirli isme sahip belgeler aramaya baslar. Eger bulursa onu browsere gonderir. Iste sunucunun otomatik olarak açmak istedigi dosyalara PWS, "varsayilan belge" diyor. Yüzyillardan beri kullanilan "varsayilan belge" lerden en meshurlari: "index.html, index.asp, default.asp, default.html, main.asp, main.html" vs... "Varsayilan belgeyi etkinlestir" seçenegi tiklanmis ise sunucu, belirtilen isimdeki dosyalari otomatik açmaya çalisir. Eger tiklanmamis ise, boyle bir arayis içinde olmayacaktir sunucu. Biz tikliyoruz. Altta da bir metin kutusu gozünüze çarpti degil mi?

Iste bu metin kutusuna, caninizin istedigi, yazmak isteyip de yazamadiginiz varsayilan belge türlerini yazabilirsiniz. Tabi kurala uyarak: Ornegin Default.htm; Default.asp; Main.asp; anasayfa.html; babasayfa.html; paramparca.asp gibi bir varsayilan belge silsilesi yazabilirsiniz.

1. **Dizinde Goz atma** : Bu seçenek tiklanirsa, ziyaretçiler web sitenizin klasorlerinin içeriklerini gozlemleme imkanina kavusur. Tabi o klasorde varsayilan belgenin yoklugu veya varsayilan belgenin etkin omadigi durumdan bahsediyoruz. Diger seçenek ise sunucunun günlük kaydetmesini denetliyor. (lazim olmacak yani)

## SONUÇ

Nihayet PWS yi tanidik. Kabaca PWS nin bilgisayarinizi bir WWW sunucusu haline getirdigini tekrarlayalim.

Bizim amacimiz ASP ogrenmek ve uygulamak. Tercihimizi bu yüzden ASP den anlayan bir web server programindan yana yaptik. ASP Microsoft'un ozgün web programlama teknolojisi, PWS ise yine Microsoft'un Win98 için gelistirdigi bir server programi (NT deki IIS'in bir beden küçügü).

Artik bilgisayarimiz ASP'den anlayan ve web dokümanlarimizi dünyaya sunan minik bir sunucu haline geldi. Bunu gozlerinizle gormek isterseniz...

Internet'e baglandiginiz zaman IP numaranizi tespit edin. Ornegin IP niz 212.64.197.51 olsun.

Browsere 212.64.197.51 yazdiginiz zaman, kendi sunucunuza konuk olursunuz. Ve varsa ana sayfaniz gelir. Bunu o anda internete bagli tüm insanlar denediginde ayni sonucu alirlar. Yani sizin sunucunuza konuk olurlar ve ana sayfaniza ulasirlar. Ne kadar global düsündügümüzün farkina varin.

Bunu kullanarak, bilgisayarinizda olusturdugunuz ama henüz web e yerlestirmediginiz bir sitenizi, arkadasiniza gosterip onun fikrini alabilirsiniz. Veya arkadasinizin, sizin bilgisayarinizdan bir dosyayi (mesela bir mp3), download etmesini saglayabilirsiniz.

PWS muhabbeti yeterli saniyorum.

# DEGISKENLERIMI KIMSEYLE DEGISMEM

## ASP YI YAZMAYA BASLAYALIM

HTML bildiginiz varsayiliyor. Tablolama tasarim konusunda ayri bir oneme sahiptir. Bir de ileride veritabani ile iliskilerimizde arayüz olarak kullanacagimiz için formlari ve nesnelerini iyi bilmeniz lazim.

HTML belgelerinde, HTML taglari < ile > simgeleri arasina yazilmaktaydi. Bu sekilde diger metinlerden ayirt edilmekteydi. Yazacagimiz ASP kodlari ise benzer formda <% ile %> ayraçlari arasina yazilacak. Sunucu bu iki ayraç arasindaki ifadelerin ASP kodu oldugunu bildiginden bunlari yürütecek ve klullaniciya salt, ASP den arinmis HTML belgesi gonderilecek. Ornegi inceleyterek bunu daha iyi anlayacagiz. Asagidaki kodu bos bir metin belgesine kopyalayip "ilk.asp" diye kaydedin.

**1.sekil: ilk.asp nin ham kaynak kodu**

<html>

<body>

<p>Asagida simdinin zamani gorüntülenecek

**<%**

**Dim** simdi *' degisken tanimladik*

simdi = now *' degiskene simdi nin zaman degerini atadik.*

**Response**.**Write "**<p><font color=**""**red**""**>**" &** simdi **& "**</font>**"**

*' ve belgeye formatlayarak yaziyoz*

**%>**

</body>

</html>

ASP kodlarini ayirt etmissinizdir. **Dim** ile "simdi" diye bir degiskeni deklare ediyoruz. "simdi" degiskenine **now** hazir zaman fonksiyonunun degerini atiyoruz (VB SCRIPT). "Response.Write" komutuyla küçük bir formatlama yapip simdi yi belgeye yazdiriyoruz. Tek tirnak isaretinden satir sonuna kadar yazilan ifadeler, artik olmazsa olmaz olan comment (yorum) lardir. Yani sunucu buradaki deyimleri anlamaya ugrasmak yerine kaale bile almayacaktir; gülüp geçecektir. Ote yandan buradaki yorumlar istemci bilgisayara da ulasmayacagina gore webmimarin tapulu mali olarak kabul edilebilir. Webmimar içinden geçen her seyi satir

DEGISKENLERIMI KIMSEYLE DEGISMEM 7 ASP NEDIR?

aralarina commentler seklinde serpistirebilir. Kime ne? Degil mi... ASP nin bir avantajlar dünyasi oldugu oyle belli ki...

Sunucu tarafinda bu sekilde gorünen "ilk.asp" dosyasini kaydettiginiz sanal dizinden browser ile çagirin bakalim. Browserle çagirdiginiz "ilk.asp" nin kaynagini gorüntülediginizde asagidaki kodu goreceksiniz. (Burada tarih ve saat olarak sizin bu sayfayi sunucudan istediginiz "ân" in degeri gorünecektir )

**2.sekil: ilk.asp nin browserden çagrildiktan sonraki (yorumlanmis) kaynak kodu**

<html>

<body>

<p>Asagida simdinin zamani gorüntülenecek

<p><font color=**"**red**"**>08.07.2000 16:47:19</font>

</body>

</html>

Gordügünüz gibi istemci bilgisayara yazdigimiz hiç bir ASP kodu gitmedi. Sadece sunucu tarafindan üretilen zaman degeri geldi. Sanki biz, bu sayfa sunucudan istendiginde belgeyi açip elimizle o anki zamani yazdik ve de kaydetti, kullanici da bunu yedi:))) Hayir.. Iste ASP teknolojisi webmimara boyle kolayliklar getiriyor. Bir çok isi sunucuya yüklemis oluyorsunuz. Alet isler, el ovünür dostlar yalan mi?

Ornek olarak daha yukarida verdigim "ilk.asp" nin kaynak kodundaki sintaks renklendirmesi dikkatinizi çekmistir. Sizce ben bunlari teker teker elimle kirmizi, yesil renklere mi boyadim? Asla! Birakin sunucu yapsin arkadaslar. Isi ne? Ancak oyle "yap" deyince yapilan sey degil tabi. Sizin her kosulu ve istisnayi goz onüne alip bir plan yapmaniz lazim. Ve bu plani, sunucuya anlayacagi sekilde anlatmaniz lazim. Iste onun için ASP ye basvuruyoruz. Ileride deyim yerindeyse sunucuyu "kole" gibi kullanmayi da ogrenecez. Helal dostlar, helal...

# FORM'UNDA SAYFALAR YAZMAK

Form lar veritabani ile olan iliskilerimizde aramizdaki çopçatan gibi bir sey olacak. (ne benzetme!) Kimi zaman veritabanindan aldigimiz verileri düz metin haline dokmekten baska, formdaki alanlara da dokmek isteyecegiz. En onemlisi veritabanina web den veri basarken, formlari kullanacagiz. Elimiz mahkum.

Bütün bilgi girisleri formlar üzerinden gerçeklesiyor desek yalan omaz degil mi? Anketler, ziyaret defterleri, forumlar, arama sorgulari... Hepsini olusturacagimiz form nesneleriyle teker teker halledecegiz.

Bu makalede HTML formlarini ASP de nasil amacimiza yonelik kullaniriz; onu inceleyelecegiz. Bunu için HTML nin form unsurlarini bilmeniz gerekiyor. Asagi yukari her HTML editoründe bu unsurlar listelenerek hazir sekilde size sunuluyor.

## FORMDAN IÇERIK TALEBI

Ilk once formlardaki elemanlarin içerigini talep etmeyi ogrenelim. Bir formda çesitli elementler bulunabilir. Bunlardan bilgi girisleri için metin kutulari (input text), metin alanlari (text area), seçme kutulari (checkbox, input radio), seçim menüleri (select option) vs.. kullanilabilir. Kullanici karsisina dikilen bir formdaki gerekli alanlara bilgi girisi yapar ve “submit” tusuyla formu tüm unsurlariyla “action” ozelliginde belirtilen URL ye gonderir. “Action” ozelligine girilen bu URL bir ASP dosyasi olabilir. Çünkü ASP ile gonderilen form unsurlarindaki bilgilere ulasabiliriz. Iste burada, bir asp dosyasinda form elemanlarindaki içerigin nasil talep edilecegini gorecegiz. Bunun için bir form barindiran HTML dokümani olusturalim ve “form.html” olarak kaydedelim.

**Sekil1. Form.html kaynak kodu**

<html>

<body bgcolor=**"**#FFFFFF**"**>

<form method=post action=**"**getform.asp**"**>

<input

type=**"**hidden**"** name=**"**FormAdi**"** value=**"**Müsteri Anketi**"**>

Adiniz : <input type=**"**text**"** name=**"**ad**"**>

<<−−!−−−−−−−àrequest.form”ad” −−>>

<<br><br>

Cinsiyetiniz : <input

type=**"**radio**"** name=**"**cinsiyet**"** value=**"**bay**"**> Bay

<input

type=**"**radio**"** name=**"**cinsiyet**"** value=**"**bayan**"**>

Bayan

<br><br>

Ilgi Alanlariniz : <input

type=**"**checkbox**"** name=**"**ilgi**"** value=**"**muzik**"**> Müzik

<input

type=**"**checkbox**"** name=**"**ilgi**"** value=**"**spor**"**> Spor<br>

<input

type=**"**checkbox**"** name=**"**ilgi**"** value=**"**siyaset**"**> Siyaset

<input

type=**"**checkbox**"** name=**"**ilgi**"** value=**"**edebiyat**"**> Edebiyat

<br><br>

Yas Araliginiz :

<**Select** name=**"**yas**"**>

<option value=**"**çocuk**"**>01−15 </option>

<option value=**"**genç**"** selected>16−25</option>

<option value=**"**orta**"**>26−46</option>

<option value=**"**yasli**"**>47−85</option>

</select>

<br><br>

Yorumunuz : <textarea

name=**"**yorum**"** rows=**"**6**"** cols=**"**20**"**></textarea>>>

<br><br>

<input type=**"**submit**"**>

</form>

</body></html>

Sira form.html belgesindeki formdan bilgi talep etmeye geldim. Bu isi bir ASP dosyasiyla yapacagiz. Formdan bilgi talep etmeden once, formdaki elemanlarin kimliklerini (name) bilmemiz gerekir ki istedigimiz sekilde kullanalim. Bu isimleri bilmesek bile topyekün listeleyebiliriz. Ama biz su anda bildigimiz bir formdan bilgi çekecegiz. Ve elemanlarin “name” kimliklerini bildigimiz varsayiyoruz ki kullanicinin girdigi bilgileri (value) leri alalim.

Bir onceki sayfada verdigimiz form.html nin kaynak kodunda, her bir form elemanin anlamli bir adi olduguna dikkat edin. Sirayla elemanlarin isimleri ve tipleri : FormAdi (hidden input), ad (text input), cinsiyet (radio input), ilgi (checkbox input), yas (option select), yorum (textarea).

Bu isimleri bildikten sonra yapacagimiz sey, teker teker bu elemanlarin içerigini talep etmek. Ama nasil? Su ifadeyle:

Request.Form("eleman\_ismi")

Ornegin forma bilgi giren kisinin adini yazdigi "ad" kimlikli elemanin degerini Request.Form("ad") seklinde

isteriz. Tabi bu ifade bize bir deger getirecek, biz de bu degeri bir degiskene atayarak sayfalarda (veya oturumda, çerezde..) kullanabiliriz.

Simdi form.html deki formdan elemanlari degerlerini alan, formun "action" unda belirttigimiz "getform.asp" nin kaynak kodunu verelim. Bu asp belgesi form.html ile ayni dizinde olsun.

**Sekil2. Getform.asp kaynak kodu**

**<%**

*' Degiskenler resmi geçidi*

**Dim** FormAdi ‘burada teker teker o formdan gelecek degiskenler tanimlaniyor…

**Dim** Ad

**Dim** Cinsiyet **Dim** Ilgi **Dim** Yas **Dim** Yorum

* *Form dan taleplerimiz var sirada*
* *Aldigimiz degeri ilgili degiskene aktariyoruz*

FormAdi = **Request**.**form**(**"**FormAdi**"**) Ad = **Request**.**form**(**"**ad**"**)

Cinsiyet = **Request**.**form**(**"**cinsiyet**"**) Ilgi = **Request**.**form**(**"**ilgi**"**)

Yas = **Request**.**form**(**"**yas**"**) Yorum = **Request**.**form**(**"**yorum**"**)

* *Iste bu kadar. Su anda formdaki tüm degerleri sayfamiza dahil ettik.*
* *istedigimiz yerde kullanabiliriz.*
* *Ornegin bu bilgileri bir text dosyasina yazdirabiliriz,*

*'bir veritabanina ekleyebiliriz, 'bir mail adresine gonderebiliriz*

* *biz yine bu sayfa hudutlari içinde bir ornek yapalim*

**%>**

**<%**

**Dim** Ek *' Kisinin adinin sonuna cinsiyetine uygun bir hitap eki ilave edelim*

**If** Cinsiyet=**"**bay**" Then ‘eger cinsiyeti bay ise**

Ek = **"** Bey**" Else ‘degilse**

Ek = **"**Hanim**"**

‘bey

**‘hanim**

**End If**

**‘eger komutlarinin sonu..**

**%>**

<p>**<%**= FormAdi**%>**

<p>Sayin **<%**= Ad**%>&” “& <%**= Ek**%&”,”>**<br>

Ilgi alanlariniz : **<%**= Ilgi**%>**<br> Yas Araliginiz : **<%** = Yas**%>**<br> Yorumunuz : **<%**= Yorum**%>**

Yukarida kaynak kodu verilen getform.asp dosyasi ile formdaki elemanlarin içerdigi bilgileri kimliklerinin kullanarak (name) istedik ve de yien ayni sayfa içinde bir dogrulama (confirmation) orneginde kullandik. Eger bir kullanici form.html deki formu doldurup, gonder tusuna bastiginda forma girdigi bilgilerin gorüntülendigi sayfayla karsilacaktir. Ornek:

Müsteri Anketi

Sayin Hande Ataizi Hanim Ilgi alanlariniz : edebiyat Yas Araliginiz : orta Yorumunuz : Bayildim

# SORGU CÜMLECIKLERI KULLANMAK

## SORGU CÜMLECIGI : QueryString

ASP sayfalari kodlarken, bir sayfadan diger bir sayfaya bilgi aktarmanin degisik yollari vardir. Bunlardan biri olan Form kullanarak bilgi geçirmeyi gormüstük. Bir diger yol ise sorgu cümlecigi kullanmaktir. Sorgu cümlecigi, link verecegimiz sayfanin URL sinin sonuna eklenen çesitli string (katar) tipteki ifadelerdir.

[Ornek : http://www.x.com/deneme.asp?telefon\_numarasi=4121254](http://www.x.com/deneme.asp?telefon_numarasi=4121254)

Yukaridaki URL de belirtilen deneme.asp sayfasina “telefon\_numarasi” kimligiyle bir deger yolluyoruz. Burada deger olarak sayisal ifade yazdik gibi gorünüyor. Ancak hayir! Buraya istediginiz kadar rakam esitleyin, yazdiginiz bir string dir, yani matematiksel olarak bir anlami yok. Bildigimiz düz metin.

Evet, hernagi bir sayfanin URL sinin sonuna bu sekilde sorgu stringleri ekleyebiliriz. Birden fazla eklemek istersek aralara “&” isareti koyariz.

[Ornek: http://www.x.com/siparis.asp?musteri\_no=15&siparis\_no=52](http://www.x.com/siparis.asp?musteri_no=15&amp;siparis_no=52) gibi.

Boylece bir sayfaya sorgu stringi yoluyla bilgi geçirme ifadesinin ne kadar kolay hazirlandigini gorüyoruz. Bir de madalyanonu oteki tarafina geçelim. Yani sorgu cümlesiyle yollanan bilgileri talep edecek bir asp dosyasina ihtiyacimiz olacak. Bu asp dosyasinda sorgu stringini talep etmek için asagidaki kod kullanilir.

Request.QueryString(“eleman\_adi”)

Ornegin yukarida verdigimiz, URL deki sorgu cümlesinden deger çeken siparis.asp dosyasinda su kodlar kullanilabilir.

Dim musteri\_no, siparis\_no

Musteri\_no = Request.QueryString("musteri\_no") Siparis\_no = Request.QueryString("siparis\_no")

Bu kodun çalismasiyla Musteri\_no degiskenimize sorgu stringinden aldigimiz “15” degeri, Siparis\_no degiskenimize ise “52” degeri atanmistir. Tekrar edersek bu degiskenlerdeki degerler string tiptedir. String ve Sayisal tiplerin farki için asagidaki kullanimi inceleyelim: (yukaridaki koda devam ediyoruz)

Response.Write (Musteri\_no + Siparis\_no)

Yukaridaki ifade sonucu sayfaya 15 + 52 = 67 yazilacak diye beklemeyin. Bunlar string ifade olduklarindan aradaki toplama isareti matematiksel toplama islemi yapmak yerine iki stringi birlestirecektir. Ve sayfaya ‘1552’ yazilacaktir.

Bir asp belgesine sorgu cümlecigi yoluyla bilgi geçirmenin yolunu ogrenmis bulunuyoruz. Peki bu yontemi nerelerde kullanabiliriz? Tabi ki bu yontemi de daha çok veritabanlariyla ugrasirken kullanacagiz. Zaten anlatilan bu konularla hep veritabani kullanimina hazirlik yapiyoruz. Sadede geldigimizde bu yontemleri çok kullanacagiz. Ornegin veritabanindan tekil kayitlar seçerken sorgu stringi kullanmak en kullanisli yoldur.

Okudugunuz bu web dokümanindaki yazilar da bir veritabanindan çekilmistir. Ve çekilirken de kolaylikla

gorebileceginiz gibi sorgu stringi kullanilmistir.

#### BIR ORNEK

Sorgu stringiyle bilgi geçisine bir ornek yaparak bu bahsi kapatalim. Ilk olarak basit bir html dosyasi yaratacagiz. Adi “sorguver.html” olsun. Kaynak kodu asagidadir.

<html>

<body bgcolor=**"**#FFFFFF**"**>

BIR LINKE TIKLAYINIZ<br><br>

<a href=**"**sorgual.asp?id=01**"**>BIRINCI LINK</a><br>

<a href=**"**sorgual.asp?id=02**"**>IKINCI LINK</a><br>

<a href=**"**sorgual.asp?id=03**"**>ÜÇÜNCÜ LINK</a><br>

</body>

</html>

Simdi de bu html dosyasinin linklendigi sorgual.asp yi olusturalim.

**<%**

id = **Request**.**QueryString**(**"**id**"**) **Select Case** id

**Case "**01**"**

**Response**.**Write "**1. linke tiklamissiniz. Acaba aklinizdan ne geçti de tikladiniz?**"**

**Case "**02**"**

**Response**.**Write "**2. linke tiklamissiniz. Nerden bildim?**"**

**Case "**03**"**

**Response**.**Write "**4. linke

tiklamissiniz. saka saka 3. linke tiklamissiniz**"**

**Case Else**

**Response**.**Write "**ya sorgu stringi girilmemis, ya da girilmis de bizim istedigimiz tipte olmamis.**"**

**End Select**

**%>**

Çok basit bir sorgu stringi kullanimi gordük. Ancak onemli olan buradaki HTTP basligi ile bilgiyi gonderme ve basliktan alma yontemini kavramaktir.

# AH VERITABANI AH!

Sira geldi dilimizden düsmeyen veritabani kullanmayi ogrenmeye...

**Veritabani:** Veritabanlarini verilerin depolandigi tablolari barindiran dosyalar seklinde düsünebiliriz. Tablolar ise ad, soyad, telefon gibi alanlardan (field) olusur. Bu alanlarin tamami ise bir kayiti (record) olusturur. Kayit kavrami satir kavramiyla hemen hemen aynidir. Veritabani tablosundaki satirlara record, sütunlara field diyecegiz.

Veritabani dosyalari, farkli programlarla hazirlanabilir. Ancak biz burada yaygin olarak kullanilan Microsoft Access programiyla yaratilmis veritabanlarini

inceleyecegiz. Microsoft’un bu programiyla kolayca kendi spesifik veritabani dosyanizi hazirlayabilirsiniz. (Excel de tablo hazirlamak kadar kolaydir.) Veritabani dosyanizin uzantisi .mdb olacaktir.

Yeni bir veritabani olustururken yapacaginiz en onemli islem, amaciniza ve kayitlariniza uygun tablolar, sonra da tablolar için field (alan) lar olusturmaktir. Ve de alanlarin tutacagi verilerin tipini belirlemek gerekecektir.



Ornegin yukaridaki resimde ornek bir veritabanindaki bir tablonun kayitlari gorünüyor. (Access buna verisayfasi gorünümü demis). Tablonun adi da onemlidir. Diger tablolardan adiyla ayrilacaktir. Buradaki tablomuzun adi “OrnekTablo” dur. (Microsoft Access yazisinin yaninda yaziyor). Ornektablo adli tablomuzda id, AdSoyad, Adres, Telefon, Mail adlariyla olusturulmus 5 tane field (alan) bulunuyor. Resimde tabloya üç adet kayit girilmis. “id” alani birincil anahtar dir. Tabloya eklenen her yeni kayitta bu alan otomatik olarak bir sayi yazilir. Diger tablolarla, veritabanlariyla iliskilerde, ozellikle bizim kullancagimiz ASP sayfalarindaki sorgularda bu kayit seçmek için bu tür alanlari çok kullanacagiz. Diger alanlarda düz metin tipindeki alanlardir. Metin tipindeki alanlarda her kayit en fazla 255 karakter bilgi tutabilir. Bu okudugunuz yazi da veritabanindan geliyor. Sizce 255 karakteri çoktan geçmedi mi okudugunuz kisim.

Elbette geçti. Iste bu tür uzun metin verilerinde “not” tipi veri tutan fieldlar kullaniyoruz. Eger yukaridaki “adres” sütunu yerine “Ozgecmis” diye bir sütun olsaydi (255 karakteri asar degil mi bir ozgeçmis) ve biz buraya kisilerin ozgeçmislerini girseydik, Ozgecmis alaninin tutacagi veri tipini access deki tasarim gorünümüne gelip ilgili yerden “not” a çevirecektik. Ote yandan tarih veya saat türünde veriler girilecek bir alanin veritürünü de tarih/saat olarak seçecegiz. Ve sayisal kayitlar için sayi veri türü seçilmeli.

Yani olusturdugumuz alanlarin veri türlerini (ikinci resimde asagi−düser listeyle gorünen kisim) amacimiza uygun olarak belirleyecegiz.

## ALANLARDA VARSAYILAN DEGER MESELESI

Yeni bir kayit eklendiginde bazi alanlara otomatik olarak varsayilan deger atanmasini isteyebiliriz. Bunun için bir tablonun tasarim gorünümüne geçin. Varsayilan deger koymak istediginiz alani seçin. Asagidaki o alanla ilgili ozelliklerin ayarlandigi "genel" panelinde "varsayilan deger" yazisinin karsisina istediginiz varsayilan degeri girin. Ancak gireceginiz deger, o alanin veri türüne uygun olmalidir.

Ornegin sayi veri türüne sahip bir alanin varsayilan degeri "Sezen Aksu" olamaz her halde degil mi? Bu varsayilan deger ozelligi tarih/saat türü alanlarda isimize çok yarayacak. tarih/saat türünde bir alanin varsayilan degerine now() yazarsaniz, tabloya yeni bir kayit eklediginizde bu alanin degeri yani kaydi eklediginiz an otomatikmen yazilir. Date() yazarsaniz sadece eklenen tarih yazilir. Burada Visual Basic fonksiyonlari geçerlidir.

Sira ornek bir veritabani dosyasi hazirlamaya geldi. Hazirlayacagimiz bu veritabani dosyasini daha sonraki ASP ile veritabani baglantilarini ogrenecegimiz makalelerde kullanacagiz. Sayfa 2 ye geçin ve ornek Access veritabanimizi olusturmaya baslayalim.

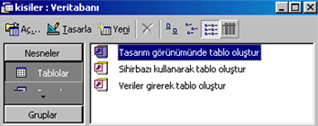
## NEYIN VERISINI TABANLAYACAGIZ

Bir fikir jimnastigiyle basliyrouz. Neyin veri tabanini yapacagiz? Bizim nasil bir veritabanina ihtiyacimiz olabilir? Bunun cevabi herkes için farkli olabilir. Golcü bir kardesimiz, attigi gollerin istatistiklerinin

tutulacagi bir vt isteyebilir. Ablamiz yemekler, malzemeleri, tarifleri vb., manyak kitap okuyan bir dostumuz kitaplarin, yazarlarinin, yayinevlerinin vb. depolanacagi bir veritabani isteyebilir. Biz burada genele hitap edecek bir veritabani ornegi hazirlayalim. Hazirlayalim ki siz de "Ahh.. ne de çok ihtiyacim vardi boyle bir veritabanina.. Ne iyi etti su Mucit.." diyesiniz. Evet... Sürpriz. Bu aksam aldigim karara gore ornek bir arkadas fihristi olusturacagiz. Hani yanimizda tasiriz, hiç aramadigim arkadaslarin telefon numaralari durur içinde. Dogum günleri, adresleri, e−kokoreç devrinde oldugumuzdan bir de e−mailleri... Ve eklenebilecek baska alanlarda arkasdaslarimizla, yakinlarimizla ilgili personal kayitlari barindiracagimiz sahsa ozel bir veritabani. Ne hos degil mi? Büyük ihtimal bu veritabaninin sadece kendi bilgisayarimizda durmasi pek isimize gelmeyecek. Bunu internete atacagiz. Eee.. Attiktan sonra nasil erisecegiz? Ama acele etmeye gerek

yok. Hepsine sira gelecek. Biz once eli yüzü düzgün bir veritabanimizi olusturalim degil mi? Hadi baslayalim.

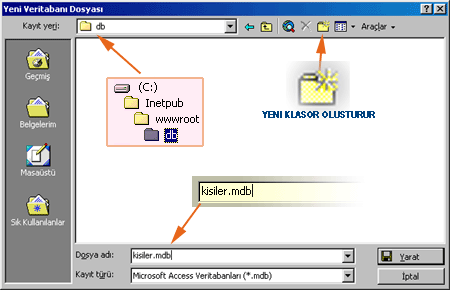
## BASLIYORUZ ORNEK VERITABANINA

Sifirdan bir veritabani olusturacagiz. Microsoft Baba'mizdan edindigimiz Access 2000 programini açiyoruz. Açinca karsimiza adet oldugu üzre bir takim seçenekler sunuluyor. Nazli gelinler gibi süzülmeden hemen amacimiza uygun olan ilk seçenegi (bos acces veritabani) isaretleyip tamam diyoruz. Bu islemin hemen ardindan Access bize "yeni veritabani dosyasi" basligiyla

zarif bir pencere çikartiyor. (ilgili resim asagidadir) Amaç yeni veriatabani dosyamiz için bir dizin ve bir de isim seçmek. Dizin olarak PWS nin ana dizini olan **"C:\inetpub\wwwroot"** a geçelim (pws yüklü degil mi makinanizda? sayet yüklü degilse personal web server ile ilgili [**yaziya**](http://evcilasp.mucit.gen.tr/home/makale.asp?id=90)gidin ve yükleyin). Bu dizin içinde

yeni bir alt dizin yaratalim, adini da ne verelim? Hemen heyecanlanmayin ad verme hakki sizde degil, bende! Hayir bende de degil maalesef :) Internet üzerindeki bulundurma hizmeti veren sunucular yalnizca belirli dizinlere yazma hakki veriyorlar. Güvenlik için tabi. Sunucun her tarafi bostan gibi ekilebilir bir alan olmuyor. Kendi bilgisayarimizda bütün dizinleri ekilebilir bostanlara çevirebiliriz. Ancak bizim amacimiz yaptigimiz database ye internet üzerinden erismek. Kendi bilgisayarimizda zaten üstüne iki tiklayip açabiliriz dosyayi. Onun için dosyalarimizi internetteki bir alana atarken geçerli "path" problemi yasamamak için veritabani dosyalarimizi belirli bir dizinde toplamaliyiz. Su anda okudugunuz yazi Ewebcity nin sunucusundaki benim olusturdugum veritabanindan geliyor. Bu veritabani ise /db klasoründe tutulmak zorunda. Biz de ayzma izni verilen klasor için db adini seçelim. Yani olusturacagimiz Access veritabani için c:\inetpub\wwwroot\db klasorünü yaratalim ve bu dizini açalim. Ve veritabani için bir isim belirleyelim: "kisiler.mdb" olsun. (Access veritabani dosyalari uzantisi = mdb) Yarat tusu bizi amacimiza ulastiracaktir.

Artik fiziksel yolu "c:\inetpub\wwwroot\db\kisiler.mdb" olan bos bir veritabanimiz var. Ne mutlu. Ama bos bir veritabani. Mutlu olmayi biraz ertelememiz daha iyi :)



Olusturdugumuz bos veritabanini doldurmaya sonraki sayfada baslayacagiz.

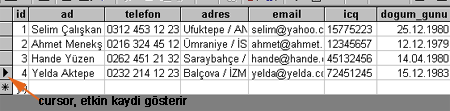
#### YENI VERI TABANIMIZ IÇIN YENI BIR TABLO VE ALANLARINI BELIRLEYELIM

Bir Access veritabaninin tablolar, sorgular, makrolar, formlar, raporlar, sayfalar gibi unsurlari var. Ancak veritabaninin tutacagi veriler tablolarda saklaniyor. Digerleri tablolardaki verilerin amaca uygun olarak kullanilmasi için gelistirilmis unsurlar. Bizim ilgilenecegimiz bolüm ise tabi ki tablolar. Bir veritabaninda birden fazla tablo bulunabilir. Bu tamamen sizin verileri nasil siniflandirdiginiza bagli olarak belirlenecek.

Biz veritabanimiz için simdilik tek tablo yaratacagiz. Tabloyu yaratmak için yanda resmi gorünen menüden "tasarim gorünümünde tablo olustur" u tikliyoruz. (diger seçenekler de kullanilabilir. biz bunu seçtik bakalim.) Karsimiza üç sütundan olusan bir tablo çikiyor. (1.sütun: alan adi; 2.sütun: veri türü; 3.sütun: tanim). Sizin de tahmin ettiginiz gibi bu tablo ile kendi tablomuzda kullanacagimiz alanlari (field) olusturup veri türlerini belirleyecegiz. Tablomuzda id, ad, telefon, adres, email, icq, dogum\_gunu olmak üzere 7 alanimiz olsun.

Bunlardan id adli alani niye olusturuyoruz diye sorarsaniz sabredin derim, onu nasil kullanacagimizi gorecegiz. SImdi bu alan adarini tasarim gorünümünde açtigimiz tablonun alan adlari bolümüne yazalim. Veri türlerini de id için otomatik sayi; ad, telefon, adres, email, icq için metin; dogum\_gunu için de tarih/saat olarak belirleyelim. id adli alanin bulundugu satira sag tiklayalim ve birincil anahtar komutu verelim. (satirin en saginda bir anahtar isareti belirecek) Yaptigimiz islemlerden sonraki gorüntü ikinci resimdedir.

Simdi sorun yoksa Access in araç çubugundaki kaydet butonuna basinca program bizden yeni olusturdugumuz tablo için bir isim soracaktir. Gelin biz buna da "kisiler" diyelim. Tama dedikten sonra tablomuz kaydedilecek. Simdi "kisiler" adli veritabanimizin nesnelerinin listelendigi menünün tablolar bolümünde yeni olusturdugumuz "kisiler" tablosunun simgesi gorünmekte. Üzerine çift tiklarsak "kisiler" tablomuzu "veri sayfasi gorünümünde" açacagiz ki bu bizim çocukluktan beri bildigimiz tablo gorüntüsüdür. Excel'deki gibi veri girecegiz. Yalniz tablomuzda hiç kayit olmadigindan sadece tek satir var ve imleç (cursor) da bu satir üzerinde. Vakit kaybetmeden ilk satirdaki alanlari doldurun. Tabi id alani kendi kendine bir numara alacak. Bu ilk kaydimiz oldugu için de bu numara "1" olacak. Ben tabloya ornek olmasi için 4 kayit ekledim. Resim asagida.



Eveeet. Veritabanimizi olusturduk. Simdi kaydet tusuna basip ortaligi temizleyelim. Artik c:\inetpub\wwwroot\db dizininde kisiler.mdb diye bir Access veritabani dosyamiz var. Ilerleyen makalelerde ASP ile bu ornek veritabanini kullanarak çesitli islemler yapacagiz. Eger bu ornek veritabanini hazirlamaya üsendiyseniz [**buraya**](http://evcilasp.mucit.gen.tr/ornekler/kisiler.zip)tiklayarak indirebilirsiniz (8,15 kb).

# ASP ILE VERITABANI BAGLANTISI

## BAGLANMAK...

Artik "yaz askimiz veritabanlariyla nasil temasa geçecegiz" mes'elesine geldik. Gerçekten olaganüstü basit ve kisa kodlamalarla dev gibi bir veritabanini onümüzde diz çoktürecek güçteyiz. Yasasin!

Hangi veritabanina baglanacagiz problemimiz yok çünkü onceki makalede ornek bir Access veritabani dosyasi hazirladik (kisiler.mdb). Bu dosyayi zip'ik halini [**buraya**](http://evcilasp.mucit.gen.tr/ornekler/kisiler.zip)tiklayarak indirebilirsiniz (8,15 kb). Baglanti muhabbetimiz bu dosya çerçevesinde olacak.

Veritabani dosyaniz "c:\inetpub\wwwroot\db" dizininde olsun. Simdi "c:\inetpub\wwwroot" dizininde yeni bir asp dosyasi yaratacagiz. Kullandiginiz editorü açin ve asagidaki kaynak kodunu kopyalayin. Sonra da vtbaglanti.asp adiyla wwwroot dizinine kaydedin.

**Sekil1. vtbaglanti.asp kaynak kodu**

**<%**

* *vtbaglanti.asp*
* *bu sayfada db alt dizinindeki kisiler.mdb veritabani ' dosyasiyla baglanti denemesi yapiliyor.*
* *çevreye verecemiz rahatsizlik ' umurumuzda deeeldir.*

**%>**

**<%**

*'' BAGLANTI KISMI BASLIYOR \*\*\*\*\**

* *baglanmadan once baglanti nesnesi olusturmamiz lazim.*
* *adini da ismail koyalim. elbet bir hatirasi var.* **Set** ismail = **Server**.**CreateObject**(**"**ADODB.Connection**"**)
* *Artik nurtopu gibi bir baglantimiz var. Adi da ismail.*
* *sonra ismail adini verdigimiz baglanti nesnemizle kisiler.mdb veritabani*
* *dosyamizi açalim.*

ismail.**Open** (**"**DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=**"** &\_ **Server**.**MapPath**(**"**db/kisiler.mdb**"**))

* *kisiler.db de bulunan kisiler tablosundaki tüm kayitlari almak için*
* *bir SQL lafi gelistiriyoruz. Adini da ismailinSQLi yapalim. Hatira sürüyor.*

**Dim** ismailinSQLi

ismailinSQLi = **"Select** \* FROM kisiler**"**

* *ismailinSQL i ni kullanarak ismail adli baglantimizi ' kayitlar adiyla çalistiriyoruz. kayitlar adini biz salladik.*
* *siz buna hatiranin sürmesi için ismailintablosu da diyebilirdiniz.*

**Set** kayitlar = ismail.**Execute** (ismailinSQLi)

*'' BAGLANTI KISMI BITTI \*\*\*\*\**

**%>**

**<%**

* *ve kayitlari sayfaya dokelim*

**Dim** k k = 1

**Do While Not** kayitlar.eof

**%>**

<p>**<%**= k **%>**. kisinin<br>

Adi: **<%**= kayitlar(**"**ad**"**) **%>**<br>

Telefonu : **<%**= kayitlar(**"**telefon**"**) **%>**<br> E−maili : <a href=**"**mailto: **<%**= kayitlar(**"**email**"**)**%>"**>

**<%**= kayitlar(**"**email**"**) **%>**</a><br> ICQ nosu: **<%**= kayitlar(**"**icq**"**)**%>**<br> Adresi : **<%**= kayitlar(**"**adres**"**) **%>**<br>

Dogum Günü : **<%**= kayitlar(**"**dogum\_gunu**"**)

**%>**<br>

**<%**

k = k + 1 kayitlar.movenext **Loop**

**%>**

**<%**

*'' VE AYRILIK VAKTI GELDI*

* *En son açtigimiz nesneyi en once kapayarak ayriliyoruz..*

kisiler.**Close**

**Set** kisiler = **Nothing**

ismail.**Close**

**Set** ismail = **Nothing**

* *ve ayrildik. her sey bitti. sunucu buraya kadar sorunsuz geldiyse*
* *veritabanindaki kayitlar html formatina dokülüp istemciye*
* *yollanmis demektir.*

**%>**

Simdi sunucu kok dizininize kaydettiginiz bu asp dosyasini tarayici ile çagirin bakalim. [(http://localhost/vtbaglanti.asp)](http://localhost/vtbaglanti.asp)) Ne çikti karsiniza?

Gordügünüz gibi basit bu isler. Baglanti bolümünde üçü geçmeyen kod emriyle bir veritabanindaki kayitlara ulasma imkanina kavusuyoruz. Ve SQL (structured query language) sorgu lafimizla sepetimize kisiler tablosundaki tüm kayitlari dolduruyoruz.

Sonra da bu kayitlari sepet bosalincaya kadar çalisacak bir dongüyle sayfamiza HTML taglarinin arasina doküyoruz. Ve sepetten dokme islemi tamamlaninca sepeti sunucuya iade etmek gerekiyor (racon geregi, yoksa bi sey olmaz da). BUrada kullanilan ASP kodlariyla ilgili ayrintilar sonraki sayfada sizi bekliyor.

## SORU ISARETLERI..

Onceki sayfada bir ornek veritabani baglanti sayfasi olusturduk. Ancak baglanti yaparken karsimiza yeni ifadeler çikti. Simdi bunlari inceleyelim.

set ismail = Server.CreateObject("ADODB.Connection")

Bu standart kod dizimiyle sunucu tarafinda yeni bir nesne lusturuyoruz. ismail ifadesi disindaki her sey standarttir. siz olusturacaginiz baglanti nesnelerinde ismail yerine baska isimler kullanabilirsiniz. Ancak ASP gelistiriciler boyle baglanti yaptiklari zaman ObjConn (object connection laflarinin kisasi) gibi kendilerince anlamli ifadeler kullaniyorlar. Biz de NsnBag diyebiliriz. Amann. Neyse iste.. Isin geyigine giriyor bu. Ne isim verirseniz verin (unutmayin s,g,ü,I,i,ç,o vb.. karakterler kullanilmiyor) çalissin yeter.

ismail.Open ("DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=" &\_ Server.MapPath("db/kisiler.mdb"))

ismail adindaki baglanti nesnemizden bir veritabani dosyasini bizim için açmasi emrini veriyoruz. Dosyayi açarken bir veritabani sürücüsü kullaniyor, o da Access. Ve dosyamizin goreceli adresini de veriyoruz.

Data Source Name less (DSN siz) bir baglanti kurduk. Yani kullandigimiz veritabaninin ODBC sürücülerinde takma isimle taninir halde olmasina gerek yok. Yani ek hiç bir ayar yapmadan bu sayfa belirtilen alt dizindeki kisiler.mdb veritabani dosyasina rahatlikla ulasacak. Yolunu kesenin vay haline!

ismailinSQLi = "SELECT \* FROM kisiler"

Burada bir ismailinSQLi degiskenimize bir string (karakter dizisi) atamasi yapiyoruz. Ancak atadigimiz karakter dizisi Ankarali Turgut'un sarki sozlerinden farkli olarak bir anlam ifade ediyor. Degiskenimize çok çesitli cümleler atayabilirken biz gittik "kisilerden ne var ne yok seç" mealinde bir laf dedik. Bu laf spesifik bir laf arkadaslar. Elalem buna **SQL lafi** diyor. Peki SQL de ne? Bu konu ayri bir makale konusu. (bekleyin). Buradaki SQL lafimiz bir komut içeriyor: "kisiler tablosundan tüm alanlari seç".

set kayitlar = ismail.Execute (ismailinSQLi)

ismail adli veritabani baglantimizi onceden belirledigimiz SQL lafimizin direktiifi dogrultusunda çalistiriyoruz. Biz SQL lafimizla kayit toplulugu seçtik. Bu kayitlara ulasmak için de çalistirdigimiz

veritabani için kayitlar nesnesi olusturuyoruz.

Artik kisiler.mdb veritabanindaki kisiler tablosundaki tüm kayitlar elimizin altinda. Ve cursor da ilk kaydin üzerinde. kayitlar("ad") dedigimiz zaman cursorun bulundugu kaydin (satirin), ad alanina (field) ait degerini kastetmis oluyoruz. response.write kayitlar("email") derseniz tablomuzdaki ilk kaydin email alanindaki degerini (selim@yahoo.com) sayfaya yazdirmis olursunuz.

Peki diger kayitlarin içerigini nasil yazdiracagiz. Tabi ki kayitlar arasinda hareket ederek. Su anda cursorumuz ilk kayit üzerinde deil mi?

kayitlar.movenext komutu cursoru bir sonraki kayda atlatacak. Su anda ikinci kayita geldik. Yazdiracagimiz alan degerleri ikinci kayita ait olacak. Ancak boyle tek tek cursor atlatarak kayit degerlerini yazdirirsaniz sizi dokuz koyden kovarlar :P Çünkü az sonra gokten elma düsecek ve içinden VB SCRIPT çikacak. Size bir onden kontrollü dongü yapisi verecek.

Do While <KOSUL>

<YAPILACAK ISLER>

Loop

Biz de "saol vb script" diyerek dongümüzü yapilandiracagiz. Kosulumuz ne olacak? Kayit toplulugu sonunda degilsek, yani cursor dosya sonunda degilse. Bunu anlamak için kayitlar.eof ozelligini kullanacagiz. Bu ozellik, eger kayit toplulugu sonuna gelmissek **true** degeri verir bize. Aksinde false. Bizim dongümüz ise bu ozellik false oldugunda yani kayit toplulugu sonuna gelinmediginde çalisacak. O zaman on kontrolümüz Do While Not kayitlar.eof olacak. Dongü içinde etkin kayda ait bilgilerin yazilmasi için gerekli ifadeleri yazacagiz. (Orn: kayitlar("ad")). Etkin kayittaki içerik yazildiktan sonra bir sonraki kayda atlamamiz lazim. Kaytilar.movenext bu ise yariyordu. Ve dongü blogunu da Loop ile nihayetlendirecegiz.

Artik sayfamiza kayitlarin içerigini doktügümüz için veritabaniyla isimiz kalmadi. Sunucuda açtigimiz nesneleri son açtigimizdan baslayarak kapiyoruz. Soyle ;

kisiler.close

set kisiler = nothing

ismail.close

set ismail = nothing

Nasil? Kaptik mi bu isi...

## HIZLI SQL LAFDANLIGI

Ahmet Savas'in tanimi: "SQL (Structured Query Language) veri tabanlarindaki verileri islemek için kullanilan yapisal sorgulama dilidir.". Evet dogru soylüyor Ahmet Savas. SQL dili çok fukara bir dildir. Ancak güzel isler yapan laflari vardir. Hani bazen bir çin atasozünde buluruz kendimizi. SQL da her türlü veritabani dosyalariyla evcil'lik oynamamizda birbirinden enfes laflariyla bize yardimci olan bir iletisim dilidir. Yani bir veritabani dosyasina "Ey veritabani! Hemen bana kisiler tablondan adi "a" ile baslayan adamlarin listesini uzat bakalim... Hadi ulan. Hemen simdi. Basarim tokadi vallaa.." derseniz elde geçen tek sey avcunuzu yalamak olur. Ne der atalar? "SQL veritabanini deliginden çikarir". Iste tatli dilimiz SQL ile

veritabanina emirler yagdiracagiz.

SQL laflari çesitli temel komutlardan olusur. Bunlar SELECT, FROM, WHERE, ORDER BY, GROUP BY, HAVING, UPDATE, DELETE, INSERT dir. Gordügünüz gibi ne kadar da fakir bir dil. Biz bunlardan isimize yarayacak olanlari inceleyecegiz.

## POPÜLER SQL LAFLARI

#### TABLODAN TÜM KAYITLARI TÜM ALANLARIYLA SEÇMEK

Yani tablonun tamamini seçmek için kullanacagimiz SQL lafi sudur: sozdizimi:

"SELECT \* FROM <tablo\_adi>"

#### TABLODAN KAYITLARI ISTEDIGIMIZ ALANLARI SEÇMEK

soz dizimi:

"SELECT <alan\_adi\_1>, <alan\_adi\_2>, .. , <alan\_adi\_n> FROM <tablo\_adi>"

#### TABLODAN BELIRLI KAYITLARI SEÇMEK (SÜZGEÇLEME)

Ornegin tablomuzda kent diye bir alanimiz var. Ve mevcut kayitlar arasindan kenti "ankara" olanlari seçmek istiyoruz.

SQL lafimiz:

"SELECT \* FROM tablomuz WHERE kent = 'ankara' "

Birden fazla kosul da süzgeç de koyabiliriz. Tablomuzda yas diye bir alan olsun. Kenti ankara olan ve yasi 30 dan büyük olanlari seçmek için SQL lafi:

"SELECT \* FROM tablomuz WHERE kent='ankara' AND yas < 37"

Gordügünüz gibi WHERE ifadesi bize kayitlari süzgeçleme imkani veriyor. Burada = , > , < , <> gibi mukayese operatorleri kullaniliyor.

#### TABLODAN KAYITLARI SIRALI HALDE SEÇMEK

Ornegin tablomuz da kenti 'ankara' olanlari yas sirasinda seçmek istiyoruz. Bunun için ORDER BY alan\_adi yazimini kullaniriz. Sunun gibi :

"SELECT \* FROM tablomuz WHERE kent = 'ankara' ORDER BY yas"

Eger azalan sirada seçeceksek **ORDER BY alan\_adi DESC** yazimini kullanacagiz.

SQL hakkinda bilgiler sonraki sayfada devam ediyor.

#### TABLODAN KAYIT SILMEK

Ornegin tablomuzdan yasi 18 den küçük olanlari silmek için SQL lafi: "DELETE FROM tablomuz WHERE yas < 18"

#### LIKE KULLANARAK KAYIT SEÇIMI YAPMAK

LIKE kullanarak tam karsilastirma yapamadigimiz alanlarla kayit seçme imkani elde ederiz. WHERE ifadesinde alan adindan sonra kullanilan karsilastirma operatorü yerine LIKE yazilir. Bu ifadeyle veritabanimizda arama motoru gibi bir sey bile yapabilecegiz ileride. Simdi ornek kullanimlar gorelim.

Tablomuzdan adi "A" ile baslayan müsterileri seçecegiz. "SELECT \* FROM tablomuz WHERE musteri\_adi LIKE 'A%' "

Tablomuzdan yazi alaninda "asp" içeren kayitlari seçecegiz. (bir arama motoru misali) "SELECT \* FROM tablomuz WHERE yazi LIKE '%asp%' "

Evet bu temel komutlar ile yolumuza devam edecegiz. Artik bu laflar ile veritabanimizdan istedigimiz rafinelikte kayit seçebiliriz, silebiliriz, arayabiliriz.

# HOST'ES MESELESI

## BEN HOST'ES ISTEREM

Evet, yerinde, son derece de makul bir istek. ASP ile gelistirdiginiz web sayfalarinizi internete tasimak artik her yeni dogan çocugun oldugu gibi sizin de en dogal hakkiniz. Keske size "alin hepsi sizin olsun" diyebilecegim bostan gibi bir sunucum olsaydi. Ama farkettiginiz üzre biz de boyle koprü alti sunuculardan ulasiyoruz size. Beles etin yahnisine de katlanmak durumunda kaliyoruz.

## BELES OLSUN HA!

Internet gibi sinirsiz bir alemde bedava site alani veren yerler azimsanamayacak kadar çok biliyorsunuz. Hepsi degisik metotlarla sunucusundan bedava alan dagitiyor. Bir nevi amme hizmeti yapiyor gibi gorünüyorlar. En azindan onceden boyle soylenebilirdi. Ancak simdi gitgide azittilar. Bedava verdikleri sayfalari niye aldiniz der gibi bir sürü reklam basiyorlar. Bir kovmadiklari kaliyor. Hatta bazilari kovuyor: DomainDLX.com. Bu arkadaslar size Microsoft IIS 5 ile kosan (run) sunucularindan taze bir yer verdiklerinde güle oynaya FTP nizi açip belgelerinizi yerlestiriyorsunuz. Ahh... Reklami da yokmus derken iki gün sonra FTP niz den "kovuldun!" gibi abuk sabuk iletiler çikmaya basliyor. Saygideger ardadaslar sayfalarinizin yukarisina ve de asagisina (ortasi size kaldi, ne yaparsaniz yapin) koyacaginiz kodlarla reklamlarini yayinlamanizi istiyorlarmis megerse. Ve altinda da bir yazi : "Failure to do so WILL result in account termination!". Evet.. Sikiysa reklamimizi yapmayin demek bu resmen. Iste günümüzdeki bedava sunucularin çogunlugu bu dingil ve embesil reklam anlayisi içerisinde. Güya reklamlardan para kazaniyorlar. Ancak kulanicilarin nefretini kazanmak, kazandiklari parayi bereketli kilmayacak, eminim! :)

Evet, derin host yaralarina fazla girmeden esas meselemiz ASP hosting e gelelim. Artik adimiz gibi biliyoruz ki ASP sunucuyu kolelestiren hâkim bir teknoloji. Iste bu nedenle bu yüzden bedava ASP sunumu yapan servis hayli azinlikta. Çogu yer HTMLnin yaninda, CGI, FRONTPAGE SE destekli bedava alanlar vadediyor. Yalniz ASP gibi bir "bela"ya yanasmiyorlar. Karamsarliga kapilmayin. ASP sayfalarinizi yayinlayabileceksiniz mutlaka. Sonuçta bu sitede bedava bir yerden sunuluyor. Ancak ben sunu belirtmek istedim: "Dünyada ASP teknolojisini kullanarak olusturdugunuz sayfalari sundurabileceginiz babayigit servisler çok sinirli." Ancak bu babayigitler tedbiri de elden birakmamakta israr ediyolar. Kisitlamalar yapiyorlar çesitli. Ancak bu sizi yildirmasin. Çünkü beni yildirmadi. Bakin bu site ayaktaysa ben yikilmamisim demektir. :)

ASP sayfalarimizi dünyada bulunan üç beles servisle sunabiliriz. Bunlardan ilki, bu sitenin de sunuldugu ewebcity dir. Ikincisi DomainDLX! (gicik herifler). Ve bir de websamba.com var. Bir tane webhostme den bahsediyorlar ancak herifler katilimi durdurmuslardi en son baktigimda. Neyse degerlendirmelere ilerleyen sayafada baslayalim.

## EWEBCITY NASIL BIR VARLIK, ANLATIR MISIN BANA?

Elbette arkadasim. Bak simdi Ewebcity ASP scriptleri çalistirmana izin veren bedava alan saglayan bir servistir. Ewebcity nin en goze çarpan artisi (benim tercihim olmasina sebep) reklamsiz olmasidir. Evet, gerçekten Ewebcity'den sundugunuz sayfalarda reklam gorünmüyor. Ha. Yukaridaki banneri soruyorsaniz o benim kendi tesis ettigim degisken banner kusagi. Yani Ewebcity size sayfalarin browserle gorünen bütün alanini tahsis ediyor simdilik. Yani sitenizin façasi bozulmasina engel oluyor bu bir nebze de olsa. Reklamsiz host'esimiz ewebcity deki ozelliklere daha yakindan bakalim.

**ASP scriptleri**: Evet, dedik ya izin veriyor. standart ASP nesnelerini dilediginiz gibi kullanin.

**Database Destegi**: Ewebcity sadece MS ACCESS türü veritabanlariyla islem yapmaniza izin veriyor. Bizim için yeter de artar bile. Veritabanlari kok dizinin /db adli alt dizininde çalisabiliyor ancak. Ve bir ayrinti: db klasoründe Access veritani dosyalari çalisabiliyor ancak ASP scriptleri içeren sayfalara izin verilmiyor. Yani ASP sayfalarinizi "/db" dizini disinda her yerde çalistirabiliyorsunuz, veritabani dosyalariniz da sadece "/db" klasorü içinde kullanilabiliyor. Ote yandan asp ile yazdiracaginiz text dosyalari da "/db" dizinine yazilmali. Çünkü tahmin ettiginiz gibi yazma izni sadece o klasore verilmis.

**FTP Destegi**: Maalesef. Ewebcity nin size en kotü sürprizlerinden biri olsa gerek bu. BU bedava servis size her ne hikmetse alaniniza FTP ile erisim hakki vermiyor. E peki nasil yüklenecek sayfalar? Nasil mi? Ee golaaay! Ewebcity size hizli mi hizli çalisan bir file upload (doya bindirme) sayfasi takdim ediyor. Bu sayfadan sunucuya bindireceginiz dosyalari besli postalar halinde gonderiyorsunuz. Çok dosyaniz varsa sizi biraz ugrastiracak ancak hizinin da FTP den daha makul oldugunu farkedeceksiniz.

**MAIL BILESENI**: Yine size kotü haber. Bu servis reklam koymuyor ama Web'in yaygin nimetlerinden de bizi bayagi mahrum birakiyor. Ewebcity deki sayfalariniz mail gonderemeyen sayfalar oluyor. ASP sayfalariyla mail gondermek sunucudaki bilesen yardimiyle oluyor. Ve size verilen bu beles dünyada, sunucudaki mail bilesenini kullanim hakki taninmiyor.

**SIZE VERDIGI URL**[: http://wwwX.ewebcity.com/uye\_adi/](http://wwwX.ewebcity.com/uye_adi/) (buradaki X yerine size o anda müsait olan bir numara geliyor ornegin www4.e....)

Size goren, bakan, duyan ama konusamayan utangaç bir site sunmak için ideal bir servis olarak tavsiye ediyorum Ewebcity yi. Adresi : [**http://www.ewebcity.com/**](http://www.ewebcity.com/)

## DOMAINDLX: BIR NEFRET ABIDESI

Istemeyerek de olsa bu manyak heriflerden bahsedecegim. Çünkü "beles" ASP host ettiklerini iddia ediyorlar. ASP yi free host eden servisler daha fazla olsaydi bu servisi goz ardi edebilirdim.

Domaindlx [**ana sayfasi**](http://www.domaindlx.com/)' na konuk olun bakalim. Sizi gayet itici bir tasarim karsilayacak. Microsoft IIS 5 kullandiklarini kocaman yazilarla haber veren bu adamlar çagdas bir sunucu olduklarini saniyorlar herhalde. Ancak once basit bir HTML düzenine getirdikleri çagdisi yorumu güncellemeleri gerekiyor.

DomainDlx size bedava bir alan sunuyor. Bu alan içinde ASP nin yaninda PHP de çalistirabiliyormussunuz. Size verilen dizine FTP ile erisebiliyorsunuz. Veritabani dosyalarinizi /database, sunucu tarafli script içeren sayafalarinizi da /script klasorüne yerlestirmeniz gerekiyor.

DomainDlx sizden para istemiyor ancak verecegi reklam kodlarini (iki tane, biri üste, biri alta konulacak) sayfalariniza yerlestirmenizi "rica" ediyor. Eger ricasini yerine getirmezseniz sizi tarlasindan kovuyor.

Domaindlx de ASP nin sunucu nesnesi FileSystemObject i kullanamiyorsunuz. Bu text dosyalariyla islem yapamayacaksiniz anlamina geliyor. Mecburen verilerinizi tabanlayacaksiniz.

Domaindlx in sasirtici bir ozelligi var. Size sunucusundan e−mail bileseni (persist) tahsis ediyor. Ve smtp mail gondermeç sunucusunu keyfinizce kullanabiliyorsunuz.

[Ve son olarak Domaindlx in size bagisladigi kosenize tarayicidan http://media.domaindlx.com/evcilasp/](http://media.domaindlx.com/evcilasp/) gibi bir URL ile ulasiyorsunuz. Eger arzu edenler olursa bu beles servis hakkindaki malumatlarimizi genisletebiliriz.

## HAYDI SAMBA'YA!

[**Websamba**](http://www.websamba.com/)da bedava ASP sunumu yapan servislerden. Ancak su ana kadar inceleme firsatina erisemedim. Aldigim hesapta bir sorun olustu FTP ile ulasamdim. Yeni hesap açtirma müracatima da henüz yanit gelmedi. Benim için bir muamma. Ancak sizi bu servis konusunda da aydinlatmak vazifemiz. O yüzden bu servisin kritigini [**Bedavamaster**](http://www.bedavamaster.com/)'i hazirlayan arkadastan rica ettim. Sonraki sayfada arkadasin Websamba hakkindaki degerlendirmesini bulacaksiniz. Beni kirmadigi ve açiklayici bir yazi gonderdigi için kendisine çok tesekkür ederim.

#### WEBSAMBA VE DÜSÜNDÜRDÜKLERI

BedavaMaster'dan... "

Oncelikle bu isi amatorce yaptiklari belli. Bu nedenle pek de ustlerine gitmek istemiyorum.

Once artilarini soyleyeyim:

+ 30 MB web alani veriyorlar.

+ Hicbir banner koymuyorlar ,

+ Dosyalarinizi FTP ile upload edebiliyorsunuz.

+ Ewebcity gibi database dosyalarini illa su dizine koyacaksin diye sart

kosmuyorlar.

+ CDO mail destegi veriyor.

+ XOOM gibi hergun junk mail yollamiyor. Hatta hic mail yollamiyorlar.

+ Hizli bir servere sahipler. Bazen gercekten adi gibi samba yapiyor.

Gelelim bu arkadaslarin eksilerine.

* Tamamen amatorce (yani para kazanma amaciyla degil) bir hizmet.
* Cok kotu servis. Maillerinize cevap verirlerse kendinizi sansli sayin.
* Cogu zaman coken DNS server. "500 internal error" mesajina asina oluyorsunuz.

Diyelimki cumartesi aksami server coktu. Hapi yuttunuz. Bilinki pazartesi

sabahi oraya

biri gelip "aaaa server gumlemis yaw" deyip el atana kadar duzelmeyecektir.

* Duzenli yedek aliyoruz diyorlar, server cokunce 6 ay oncesinin yedeklerini

servere geri yukleyip sizi cildirtiyorlar.

* Yaptigim gorusmede (buyuk basariydi !) istenirse bedava domain hosting

hizmetide

verdiklerini belirttiler. Ancak bu yuzden 1 hafta offline kalmamiza ragmen

beceremedik.

* Kendi global.asa dosyaniza izin vermiyor.

\* Sonucta veri kayiplari sizin icin onemli degilse, "Ben kendi yedegimi kendim alirim"

diyorsaniz ve ikidebir coken server sizi delirtmezse Websamba su an en

ideal

cozumlerden biri. Tabii uye olabilirseniz. Duyduguma gore aylardir bekleyen arkadaslar

varmis. :−)

"

Mucit'ten not: Websamba dan bir hesap aldim arkadaslar. FTP ile hesaba ulasmama ragmen bos olan dizinime bon bon bakiyorum. Çünkü permission um yok. Dosya koyamiyorum. Mükemmel! Sahiden de mükemmel...

# BIRAZ VBSCRIPT (I)

Sevgili arkadaslarim. ASP kodlarimizi yazarken, ister VBScript, ister Jscript olsun istedigimiz script dilini kullanabilecegimizi biliyoruz. Yani sunucu sistemin destekledigi her türlü dil kullanilabilir. Istemci tarafindaki kardesimizin kullandigimiz dille hiçbir ilgisi olmayacagindan onun için hiçbir sorun teskil etmeyecek bu. Yani kullandigi tarayici Netscape olmasina ragmen bizim yazdigimiz VBScript ifadeleri çalisacak. Çünkü bu kodlar Netscape’in yüce huzurlarina gitmeden, sunucu tarafindaki gorevli memurlar tarafindan icra edilecek. Ve bu ozgürlük, Netscape’nin “Ben .asp uzantili dosyalari çagirmayacagim bundan boyle, prensiplerimi çigneyemem” diyerek cingar çikarmasina kadar devam edecek.

Burada ASP sayfalari olustururken gereksinim duyacagimiz temel VBScript konularina deginecegiz. Sunu belirtmek gerekir ki bu konu hakkinda soylenecek bir sey birakmadi Hakki Ocal Agabey. Onun yazdigi ve sizin de aldiginizi tahmin ettigim ASP kitapçiginda mükemmel bir VBScript referansi mevcut. Yalniz biz burada bir ASP bahçesinin bahçivanligini yapiyorsak elbette VBScript’ten de bahsedecegiz.

## VBSCRIPT NEDIR?

En basit cevap: Microsoft’un “Visual Basic” den türettigi Javascript gibi bir scripting dilidir.

“Microsoft Visual Basic Scripting Edition, the newest member of the Visual Basic family of programming languages, brings active scripting to a wide variety of environments, including Web client scripting in Microsoft Internet Explorer and Web server scripting in Microsoft Internet Information Server.” deniliyor Microsoft’un referansinda. Biz burada belirtilen ikinci ortamdaki VBSCRIPT kullanimina deginecegiz. Yani “Web server scripting in Microsoft Internet Information Server”…

## SCRIPT DILI BILDIRIM SATIRI

ASP sayfalarinin en üstünde sunucuya, sayfada kullanilan Script dili hakkinda bilgi verilir. VBScript kullanildigi zaman sayfalarin en üstüne

<%@ Language = "VBScript" %> yazilir. Microsoft sunucularinda varsayilan script dili olarak VBScript’i kabul eder. O yüzden bu satiri yazmasaniz bile PWS veya IIS sayfalarinizda VBScript dilini kullandiginizi varsayar ve buna gore davranir. Ancak JScript kullaniyorsaniz <%@ Language = "JScript" %> satirini ekleyin.

## HARF DURUMLARI

VBScript’in “letter−case” kaprisi yoktur. Yani büyük−küçük harf duyarligina sahip degildir. “musTaFa” da

yazsaniz, “MUStafA” da yazsaniz VBScript ayni çocuktan bahsettiginizi anlar. Asagidaki ifadeler ayni çiktiyi verir:

<% Response.Write 12 div 3 %>

<% RESPONSE.WRITE 12 dIv 3 %>

## DEGISKEN BILDIRIMI

VBScript'te degiskenler için esneklik had safhadadir. Degisken bildirim zorunlulugu bulunmamaktadir. Ve buna bagli olarak degisken türü bildirimi de yoktur. Degiskenlerin bildirim zorunlulugu olmamasi programciya ilk basta garip gelecekken, sonralari VBScript'e alistikça diger programalama dillerinde de tanimlama yapma konsunda unutkanlasacaktir. Yani biraz tembellige itmekte bu esneklik bizi. Ben degiskenlerimi tanimlamak istiyorum diyorsaniz siz bilirsiniz. VBscript saygi duyacaktir. diyerek degiskeninizi tanimlayabilirsiniz. Ancak hâlâ bir zorunluluk yoktur. Bu zorunluluk, sayfaya language bildiriminden sonra ekleyeceginiz <% Option Explicit %> satiriyla saglanir. Artik Dim ile bir degiskeni deklare etmeden kullanmaya yeltenirseniz VBScript kulaginizi çekme hakki kazanmistir.

## DEGISKENLERDE VERI TÜRLERI

Temelde kullandigimiz degisken türleri sayisal, alfa−sayisal ve mantiksal olmak üzere üçe ayrilir. Ancak her programlama dilinde karsimiza çikan integer, real, string, boolean vs.. türü temel veri tipi tanimlamalarindan yoksun bir dildir VBScript. Peki VBScript bir degiskenin hangi türde oldugunu nasil anlar? Bunu o degiskene yaptiginiz atama ile anlar. Asagidaki atamalari inceleyin.

degisken\_1 = 15

degisken\_2 = "15"

degisken\_3 = "Zeytinyagli yiyemem amaan, basma da fistan giyemem aman..." degisken\_4 = true

degisken\_5 = Sen Ali Munzur, daglarin kartali, bir yanim biçak yarasi bir\_ yanim sevda ve emanet kalmistir bir koylü kizinda kalbimin yarisi...

Yukarida degisken\_1'e sayisal türde bir veri atandigi için omrünün sonuna kadar sayisal tipte degisken olacaktir. Sayilara uyguladiginiz her türlü muameleye "hay hay" diyecektir. Bu atamada degiskene 15 sayisi ataniyor. Sayinin tirnak arasinda olmamasina dikkat edin. Durum degisken\_2 de ise biraz farkli. Bu sefer yine 15 sayisi çift tirnaklar arasinda degiskene atandi. degisken\_2 boylece sayisal degil, alfa−sayisal bir tipte degisken oldu. Yani matematiksel islemlere sirt çevirmis, string fonksiyonlariyla islenebilir bir degisken. degisken\_3 ün de ayni karakterde oldugunu kestirebildiniz. degisken\_4 e de mantiksal ifadelerden true degeri atandi. (digeri de false biliyorsunuz). Bundan boyle degisken\_4 boolean tipte bir degisken olarak gezecek.

Peki degisken\_5 de durum ne? Bu vatan evladi, yanlis bir soz dizimi nedeniyle VBScript'in dikkatini çekecektir. Ve geçerli bir atama olmadigi için sayfa hata verecektir. String ifadelerin çift tirnak içinde belirtilmesi gerekir.

*Veri türlerine sonraki sayfada devam edelim.*

3. DEGISKEN BILDIRIMI 31 ASP NEDIR?

## VARIANT VE ALT TIPLERI

VBScript dili kendince tek veri türüne sahiptir, adi ise Variant! Variant, kullanimina gore degisik bilgi türleri içerebilen ozel bir veri türüdür. Onceki sayfada orneklerini ordügümüz gibi, degiskenin hangi alt veri türünde oldugunu sizin kullaniminiz belirleyecektir. Variant'in alt veri tipleri sunlardir: Empty, Null, Boolean, Byte, Integer, Currency, Long, Single, Double, Date, String, Object, Error. Bir degiskenin türünü ogrenmek istiyorsaniz VarType fonksiyonunu kullanabilirsiniz. Bu fonksiyona degisken adini parametre olarak gonderiyorsunuz. Ornek:

<%

degisken\_1 = 12

degisken\_2 = "Ben sendeyim sen bendesin Rüveyda..." degisken\_3 = #08−12−2000#

response.write vartype(degisken\_1) *'sayfaya 2 yazar* response.write vartype(degisken\_2) *'sayfaya 8 yazar* response.write vartype(degisken\_3) *'sayfaya 7 yazar*

%>

Gordügünüz gibi bu fonksiyon bize anlamli sayisal degerler veriyor. Bu sayisal degerler ne anlama geliyor? Asagida VBScript Variant Tip sabitlerinin listesi var.

vbEmpty 0 Uninitialized (default) ; vbNull 1 Contains no valid data ; vbInteger 2 Integer subtype ; vbLong 3 Long subtype ; vbSingle 4 Single subtype ; vbSingle 5 Double subtype ; vbCurrency 6 Currency subtype ; vbDate 7 Date subtype ; vbString 8 String subtype ; vbObject 9 Object ; vbError 10 Error subtype ; vbBoolean 11 Boolean subtype ; vbVariant 12 Variant (used only for arrays of variants) ; vbDataObject 13 Data access object ; vbDecimal 14 Decimal subtype ; vbByte 17 Byte subtype ; vbArray 8192 Array

Ornek kodumuzda sayfaya yazilan 2,7,8 rakamlarini buraya bakarak anlamlandirabilirsiniz.

## ALT TIPLER ARASINDA DONÜSÜMLER

Evet, buna mutlaka ihtiyacimiz olacak. Variant'in alt türleri arasinda donüsüm yapmamiz için VBScript bize bir dizi fonksiyon tahsis etti. Bunlar nerede isimize yarayabilir peki? Ozellikle String ve Integer tipleri arasinda hizir gibi yetisecekler. Ornegin bir sorgu cümlesinden aldiginiz (Request.QueryString ile) degeri matematiksel islemlere sokamazsiniz, çünkü string tiptedir. Diyelim ki sorgu cümleciginden sayfa numarasi aliyorsunuz (okudugunuz bu sayfanin çagrilamsinda bunu kullaniyoruz). Aldiginiz sayfa numarasiyla da bir takim matematiksel islemler yapmak zorundasiniz. E o zaman Cint gibi bir fonksiyonla aldiginiz string tipteki degeri, sayisal bir türe çeviriyorsunuz. Iste size baslica Conversion (Donüsüm) fonksiyonlari ve kullanim biçimleri.

### CInt

Ne is yapar: Herhangi tipteki geçerli bir ifadeyi integer alt tipine donüstürür. Soz dizimi: CInt(ifade)

Ornek:

mystr = "12"

myint = CInt(mystr)/3 *'myint degiskeninin degeri : 4*

### CStr

Ne is yapar: Herhangi tipteki geçerli bir ifadeyi integer alt tipine donüstürür. Soz dizimi: CStr(ifade)

Ornek: myint = 41

mystr = CStr(myint) + " kere masallah" *'mystr degeri : "41 kere masallah"*

### CDate

Ne is yapar: Herhangi tipteki geçerli bir ifadeyi integer alt tipine donüstürür. Soz dizimi: CDate(ifade)

Ornek:

mystr = "Ekim 12, 1998" *' sunucu sistemin saat/tarihi Ingilizce ise degisir.*

mydate = CStr(mystr) *'mydate degeri : 12.10.1998*

Bunlarla ayni soz dizimiyle kullanilabilecek Asc, CBool, CByte, CCur, CDbl, Chr, CLng, CSng, Hex, Oct fonksiyonlari da elimizin altindadir. Gerekirse aklinizda bulunsun.

# BIRAZ VBSCRIPT (II)

## DURUMLAR VE DAVRANISLAR

Programlarin en temel kabiliyeti karar verme mekanizmasiyla donatilmalaridir. Insan kendi kararlarini kendi verebilen bir canlidir. Ancak bir program tasarliyorsak, o programin karsilasacagi durumlarda ondan karar vermesini bekleyemeyiz. Durumlari goz onüne alip, bu durumlara gore programin verecegi karari, takinacagi tutumu biz belirlemeliyiz. Belirlemeliyiz ki yaptigimiz sey program olsun.

Iste programamlara bu isleri akis kontrolleriyle yaptiririz. Akis kontrolleri, programin isaret edilen durum gerçeklestiginde veya gerçeklesmediginde herhangi bir davranis sergilemesini temin eder. Bu dilbilgisinde ogrendiginiz, üniversite giris sinavinda karsiniza çikan sart−kosul cümle türleri gibi bir seydir. "Sunlar dogruyken sunlari yapadur", "Sunlar dogru degilken sunlari yap", "sunlar olana kadar sunlari yap", vs..

## VBSCRIPT AKIS KONTROLLERI

VBScript'in baslica akis kontrolleri genel yazimlariyla sunlardir: **Do...Loop, For...Next, For Each...Next, If...Then...Else, Select Case, While...Wend.** Simdi bunlari inceleyelim.

### IF...THEN...ELSE KONTROLÜ

Bir ifade grubunun, bir durumun mantiksal sonucuna gore yürütülüp yürütülmemesini saglar. Soz Dizimi:

If [durum] Then [Ifadeler(if)]

ElseIf [baskadurum] Then [ifadeler(elseif)]

Else [ifadeler(else)] End If

Ornek kullanim:

If Request.Form("Parola") = "Ben Parolayim" Then Response.Write "Evet.. Parolayi Bildiniz.. "

Else

Response.Write "Bilinemedi. Heyhât!" End If

Ornegimizde formdan "parola" kimlikli elemanin degerinin "Ben Parolayim" string ifadesine esit olup

olmadigi kontrol ediliyor. Eger esitse parolanin bilindigini, esit degilse bilinemedigini anlatan mesajlar yazdiriliyor.

### SELECT CASE KONTROLÜ

Bir ifadenin kontrol edilecek durumu ikiden fazla oldugu vakit bu kontrolün kullanilmasi kolaylik saglar. Yaptigi is IF..THEN..ELSE ile aynidir.

Soz Dizimi:

Select Case [kontrol\_edilecek\_ifade] Case [durum1]

[ifadeler1] Case [durum2] [ifadeler2] Case [durumn] [ifadelern]

Case Else [ifadeler(else)]

End Select

Ornek kullanim: select case hour(now) case 8,9,10,11,12

Response.Write "Hos Geldiniz, sitemiz emrinizdedir." case 13

Response.Write "Kapaliyiz. Ogle yemegine çiktik. 2 den sonra gelin" case 14,15,16,17

Response.Write "Hosgeldiniz, downloadlar sabah 9−10 arasi yapilabilir" case else

Response.Write "Sitemiz kapalidir. Gece mesaisi yoktur. 8 de gelin." end select

Ornegimizde kendine mesai saatleri disinda çalismama karari almis bir web sitesinde kullanilabilecek SELECT CASE kontrolü yapiliyor. Saatin durumuna gore gelen ziyaretçiye bir seyler soyleniyor. Ornek gelistirilip, saatin durmuna gore farkli sayfalara yünlendirilmesi saglanabilir. (response.redirect "sayfa\_ismi.html" metoduyla)

### DO...LOOP DONGÜ YAPISI

Bir kosul dogruyken veya dogru olana dek belli bir kod blogunun tekrarlanarak yürütülmesini saglar. "Durum kontrollü dongü" de denilebilir.

Soz Dizimi:

Do While|Until [kontrol\_edilecek\_durum] [yapilacak\_isler]

Loop

Dongü içindeki ifadeler, "Do" ifadesi "While" ile kullanilirsa kosul dogruyken, "Until" ile kullanilirsa kosul dogru olana dek yürütülür.

Ornek kullanim:

Do While Not KayitKumesi.EOF

Response.write "<p>" & KayitKumesi("email") & "</p>" KayitKumesi.Movenext

Loop

Veritabanindan seçtigimiz kayitlarla olusturdugumuz kayitkümesinin "email" adli alanindaki tüm kayitlara ait degerleri bu dongü yapisiyla sayfaya yazdirabiliriz. Burada koydugumuz kosul: "KayitKümesi sonunda olunmamasi" (Not KayitKUmesi.EOF). Bu kosul dogru oldugu sürece dongü isleyecektir. Ancak dongü içindeki ifadede KayitKumesi.Movenext metodu gozünüze çarpacak. Bu kayitkümesinde bir sonraki kayda geçmemizi saglayan metoddur. Eger bunu yazmazsaniz dongü sayfaya hep ayni kaydin "email" alanindaki degeri yazacak, yani sonsuz dongüye girilecektir. Çünkü kayitkümesi hiç bir zaman dosya sonuna gelmeyecektir. Dongülerde bu tür durumlara dikkat etmeliyiz.

Isterseniz bir de "Until" ile ornek yapalim. Do Until a>=10

Response.Write a & "<br>" a = a +1

Loop

UNTIL deki tek fark, dongünün giriste degil çikista kontrol yapmasidir. Yani burada "a" degiskenindeki deger 10'a büyük−esit olsa bile programin akisi dongüye bir kez girer girer. Yani ifadeler bir kez islenir, dongü çikisinda kontrol yapilir.

### FOR..NEXT DONGÜ YAPISI

Bu dongü yapisi DO..LOOP un ozel bir seklidir. Bir ifade blogunun, belirlenen sayi kadar yürütülmesini saglar. Durum kontrolü yoktur. Bu dongüyle yapilan her sey, DO..LOOP ile de yapilabilir.

Soz Dizimi:

For [sayaç\_degiskeni] = [ilk\_deger] To [son\_deger] Step [adim\_sayisi] [yapilacak\_isler]

Next

"Step" den dongümüzde sayacin artis adimini belirliyoruz. Eger sayaç birer birer artacaksa buraya "Step 1" yazmaya gerek yoktur.

Ornek kullanim:(Ornegi Hakki Agabey'den aldim) Dim Gunler

Gunler = Array("Pazartesi", "Sali", "Çarsamba", "Persembe", "Cuma", "Cumartesi", "Pazar") For sayac = 0 to 6

Response.Write Gunler(sayac) Response.Write "<BR>"

Next

Bir dizi tanimlayip, FOR..NEXT dongüsüyle, dizinin elemanlarini yazdiriyoruz. Yazdirma isleminde sayaç olarak belirledginiz "sayac" degiskenini Gunler dizisinin elemanlarina ulasmak için indis olarak kullaniyoruz.

### FOR EACH...NEXT DONGÜ YAPISI

Bir ifade blogunu, belli bir dizi veya kolleksiyondaki herbir eleman için tekrar eder. Soz Dizimi:

For Each [eleman\_degiskeni] In [topluluk] [yapilacak\_isler]

Next

Eleman degiskenini siz belirlersiniz. Topluluk yerine bir dizi veya kolleksiyon adi gelir.

Ornek kullanim (Onceki sayfadaki For..Next orneginden uyarlandi ): Dim Gunler

Gunler = Array("Pazartesi", "Sali", "Çarsamba", "Persembe", "Cuma", "Cumartesi", "Pazar") For Each gun In Gunler

Response.Write gun Response.Write "<BR>"

Next

Bu dongü, FOR..NEXT ornegindeki kullanimla ayni sonucu verecektir. Sayfaya yine dizideki elemanlar sirayla yazdirilacaktir.

Dizilerden sonra kolleksiyonlarla kullanimina ornek yapalim. For Each eleman in Request.form

Response.Write eleman & " : "

Response.Write Request.Form(eleman) & "<br>" Next

Bir HTML formunun yonlendigi sayfada kullanilabilecek bir kullanim. ASP nin Request.Form kolleksiyonundaki her elemanin adini ve degerini sayfaya yaziyor.

### WHILE...WEND DONGÜ YAPISI

Bu dongü yapisi da DO..LOOP un ozel ve basit seklidir. Soz Dizimi:

While [durum] [yapilacak\_isler] Wend

# SITEDE OTURANLARIN SAYISI

## EY ORADAKILER! KAÇ KISISINIZ... TEKER TEKER GELIN

Evet, sitenizin ASP ile akil küpü haline geldigine en güzel orneklerden birisi sitede gezinen insanlari saymaktir. Bunu saymak ASP için çok basit bir istir. ASP adeta sunucunun nazli gelini oldugundan istedigi her türlü ayricaligi elde etmektedir.

ASP nin yerlesik nesneleri arasinda iki tane yakisikli nesne vardir: **Application** ve **Session**. ASP nin oyle bir gozü vardir ki bizim gordüklerimizi o farkli gormeye çalisir. Ornegin biz bir web sitesine "web sitesi" gozüyle bakarken, ASP onun bir uygulama (application) − yani bir program − oldugunu savunur. Ve siteye girenler de ona gore soz edilen uygulamayi kullananlardir. Ote yandan ASP, siteyi giren her kisi için ayri bir "oturum" (session) baslatir. Ve çikarken de bu oturumu sonlandirir. Ve kullanicinin bu oturum boyunca yaptigi her isten haberi olur. Adeta onun arkasinda dolasip "Ne yapiyor bu çocuk, ogrenmezsem çatlarim ayy" diyen bir Merakli Melahat profili çizer. ASP nin bu yenetegi, onu çok çekici kilar. Site tasarimcisinin yüreginde yanginlar çikarir bu ASP...

"Application" nesnesi bir uygulamadaki "oturum"lar arasinda bilgi paylasimi yapabilir. "Session" nesnesi ise bir oturum boyunca bilgi tutabilir. ASP nin bu iki yerlesik nesnesi ile yapilabilecek bir çok güzellik mevcuttur. Biz bu yazida "sitedekileri saymak"tan bahsedelim.

Sitedeki adamlari saymak için "Application" la muhabbet etmemiz lazim. Bu muhabbeti de

**global.asa** yerlesik dosyasinda yapmamiz lazim.

## GLOBAL.ASA

Microsoft der ki:

" Global.asa dosyasi, içinde olay komut dosyalarini belirtebileceginiz ve oturum veya uygulama kapsami içeren nesneleri bildirebileceginiz, seçime bagli bir dosyadir. Bu dosya kullanicilara gorüntülenen bir içerik dosyasi degildir; uygulama tarafindan genel olarak kullanilan olay bilgilerini ve nesneleri saklar. Dosyanin Global.asa olarak adlandirilmasi ve uygulamanin kok dizininde saklanmasi gerekir. Bir uygulamanin tek bir Global.asa dosyasi olabilir. "

Bu dosya, sitenize giren "oturum" cu için oturum baslamadan once çalistirilan uygulamadir. (Active Server Application). PWS ile olusturulan C:\inetpub\wwwroot dizinine baktiginizda bu dosya size goz kirpacaktir. PWS sisteminiz global.asa kullanimini destekler. Ancak her yer insanin evi gibi degildir. Gittiginiz bedava hizmet veren bir sunucu size **global**'ini birak da gel derse sasirmayin. Nitekim bu yaziyi okudugunuz sitenin ev ve yurt sahibi Ewebcity, global.asa kullanimini desteklememektedir. Ancak ücretli ASP bulundurmasi yapan yerlerde bu kullanima izin vardir. Bu isi sitenizde denemeden once sunucunuzun global.asa ile barisik oldugundan emin olun.

Dedik ya global.asa lazim bize diye. C:\inetpub\wwwroot\global.asa dosyasini düzenlemek için açtiginizda karsinizda (eger daha once degistirmediyseniz) asagidaki içerik gelecek.

**Orijinal global.asa kaynak kodu**

<OBJECT RUNAT=Server

SCOPE=Session ID=MyInfo PROGID=**"**MSWC.MyInfo**"**>

</OBJECT>

Simdi bu dosyaya yeni bir koid grubu ekleyecegiz. Dosyanin son hali asagidaki gibi olacak. Ve kaydedecegiz. (Bu islemleri kendi kisisel sunucunuzda denediginizi varsayiyorum.)

**Degismis global.asa kaynak kodu**

<OBJECT RUNAT=Server SCOPE=Session ID=MyInfo PROGID=**"**MSWC.MyInfo**"**>

</OBJECT>

<SCRIPT

LANGUAGE=**"**VBScript**"** RUNAT=**"**Server**"**>

**Sub** Application\_OnStart

*'Oturum baslarken sayaç 0 olsun.*

**Application**(**"**aktif\_oturan\_sayisi**"**) = 0

**End** Sub

**Sub** Session\_OnStart *' bir oturum baslarken*

* *Oturum bitme süresi belirle.. ihtiyaç varsa*

Session.Timeout = 20

* *Oturum basladigi an baslasin... (ne cümle oldu ama)*

Session(**"**Start**"**) = Now

* *Uygulamadki aktif adam sayisini artir* Application.Lock *'kontrolü ele al, istemciler için uygulamayi kilitle*

**Application**(**"**aktif\_oturan\_sayisi**"**) = **Application**(**"**aktif\_oturan\_sayisi**"**) + 1 Application.UnLock

**End** Sub

**Sub** Session\_OnEnd *' bir oturum biterken ' Ugulamadaki aktif adam sayisini düsür*

Application.Lock

**Application**(**"**aktif\_oturan\_sayisi**"**) = **Application**(**"**aktif\_oturan\_sayisi**"**) − 1 Application.UnLock

**End** Sub

</SCRIPT>

## SITEDE OTURANLARIN SAYISI

Evet.. Yapacagimiz en onemli ve en büyük is buydu. Artik global.asa sayesinde siteye oturmak için diz çoken biri gorülürse uygulamada tutulan sayaç bir artirilacak. Siteden ayrilan bir vatandas farkedilirse sayaç bir azaltilacak. Bu sekilde sitede oturanlarin sayisi dinamik olarak tespit edilecek. Dikkat ederseniz sayaç degiskenimizi Application nesnesi içinde tutuyoruz.

Sayaç degiskenimiz: Application("sitede\_oturan\_sayisi") . Boylelikle bu degiskendeki deger, o sitedeki tüm oturumlarda kullanilabilir hale getirilmis oluyor. Yani oturumlar üstü bir degisken oluyor bu. Simdi sitedeki herhangi bir sayfada (ki bu genelde ana sayfadir) bu degiskenin degerini gorüntülemek gerekecek. Iste test.asp. Bu dosyayi herhangi bir konuma kaydedin.

**Test.asp kaynak kodu**

<HTML>

<HEAD>

<TITLE>Active Sessions</TITLE>

</HEAD>

<BODY>

<B>Su anda sitemizde <FONT COLOR=**"**#CC0000**"**>

**<%**= **Application**(**"**aktif\_oturan\_sayisi**"**)

**%>**</FONT> kisi oturuyor.</B>

</BODY>

</HTML>

[Simdi bu sayfayi tarayiciniz ile çagirin. (Ornek: http://localhost/test.asp).](http://localhost/test.asp)) Karsiniza su metin çikacak: "Su anda sitemizde 1 kisi oturuyor." .

Ve bu kisi de sizsiniz. Durumu internete bagli oldugunuz zaman hatta olan bir arkadasinizin da bu sayfaya

girmesini saglayarak tekrar deneyebilirsiniz. (Bir not. global.asa destegi veren sunucunuzda denemek istiyorsaniz: global.asa yi sitenizin ilk açilan sayfasinin oldugu dizine atin.)

Incir çekirdegini doldurmayan bir kaç islemle sitedeki adamlari sayabilir konuma geldik. Ziyaretçilerini sasirtmayi seven webmimarlar için... ASP hepimize yeter.

# FORMDAN VERITABANI ISTIKAMETINE GIDEN KAYITLAR

Evet, bugüne degin bir çok konu hakkinda soylemler yansittik. Daha once veritabanindaki kayitlari sayfamiza nasil dokecegimizi gormüstük. Simdi bu isin daha ileri kismi olan, veritabanina kayit geçirmeye bakalim.

Once bize kayit ekleyecegimiz bir veritabani dosyasi gerekecek. Tahmin ettiginiz gibi (ve her yemek tarifi programinda oldugu gibi), vakit ve emek kazanmak açisindan bu dosyamiz hazir bir sekilde bizi bekliyor. Bu Access dosyasini daha onceki "Ah veritabani..Ah.. " baslikli makale boyunca hazirlamistik (kisiler.mdb). Bu dosyayi zip'ik halini [**buraya**](http://evcilasp.mucit.gen.tr/ornekler/kisiler.zip)tiklayarak indirebilirsiniz (8,15 kb). (bu dosyayi wwwroot/db/ konumuna yerlestirin.) Bu bir ornek olacaktir. Buradaki kayit ekleme mantigini kendi olusturdugunuz veritabani dosyalarina kolaylikla tasiyabilirsiniz.

## VERITABANINDAKI TABLO VE ALANLARIN DURUMU

Veritabanina kayit ekleyecegimize gore, veritabanimizdaki tablo(lar) ve alanlari hakkinda bigi sahibi olmamiz gerekir. Yoksa veriyi nereye gonderecegimizden habersiz bir kisi oluruz. Elinizde bulunan

kisiler.mdb dosyasinda bir adet tablo var. Adi: "kisiler". Kisiler tablosundaki alanlar ise sunlar: id, ad, telefon, adres, email, icq, dogum\_gunu. Kayit eklerken bu alanlardan hepsini doldurmak zorunda olmayabiliriz.

Yalniz "id" kismi birincil anahtardir. Buraya veri giremeyiz. Çünkü bu haylaz bolümdeki sayilar kendi kendine basina buyruk olarak artacaktir. Yeni bir kayit eklendiginde, bu alana mevcut kayit için otomatik sayi atanacak ve baska hiç bir kayit bu sayiyi alamayacaktir. Yani bu kayitlarin benzersiz olmasini saglayacaktir. Biz suçlulari kulagindan tutarken, kayitlari da 'id' sinden tutup kaldiracagiz. Karsimiza bir kayit çikarsa: "Once id ni soyle, sonra konus... yoksa seni dinleyemem" diye agirligimizi da koyacagiz. Neticede "id" alani disinda tüm alanlara veri ekleyebiliriz.

Evet.. Artik on muhabbeti tamamladigimizi saniyorum. Simdi kollari sivayin bakalim. Dogustan edindiginiz HTML bilgilerinizle form barindiran bir doküman üretin. Uzantisi simdilik .html olsun. Ileride içinde ASP kodu kullanmak isterseniz .asp yaparsiniz. HTML formunu olustururken, seçtiginiz elemanlar, tablodaki kayitlarin veri türüne uygun olsun. Ornegin metin, sayi, tarih türündeki alanlar için text input, not türündeki alanlar için textarea, evet−hayir tipindeki alanlar için de checkbox form elemanlarini

kullanabilirsiniz. Ornegimizde tüm alanlar için text input kullanmak uygun. Bakalim su form.html ye. (dosya konumu: wwwroot/form.html olsun)

**Form.html kaynak kodu**

<html>

<head>

<title> FORM.HTML </title>

</head>

<body bgcolor=**"**#FFFFFF**"**>

<h2>YENI KAYIT EKLEME FORMU</h2>

<form method=post action=**"**kayit\_ekle.asp**"**>

Ad : <input

type=**"**text**"** name=**"**ad**"**><br><br>

Telefon : <input

type=**"**text**"** name=**"**telefon**"**><br><br>

Adres : <input

type=**"**text**"** name=**"**adres**"**><br><br>

ICQ : <input

type=**"**text**"** name=**"**icq**"**><br><br>

Email : <input

type=**"**text**"** name=**"**email**"**><br><br> Dogum Günü (gg/aa/yyyy) : <input type=**"**text**"** name=**"**dgunu**"**><br><br>

<br>

<input

type=**"**submit**"** value=**"**EKLE**"**>

</form>

</body>

</html>

Form sayfamizi olusturduk. Dikkat ederseniz formumuzdaki elemanlarin hepsi text input (metin kutusu) ve kimlikleri de (name) tablomuzdaki alanlarla es isimler. Boyle olmak zorunda degil elbette. Siz ilk metin kutusunun kimligini "ad" yerine "ey\_sevdigim\_bir\_gun\_bana\_yar\_demedin" derseniz sizin bileceginiz is. Kolaylik, kolayliktir.

Formun "action" ozelliginde "kayit\_ekle.asp" ye isaret ediliyor. Oysa boyle bir dosya yok ki. Birazadan olacak. Sayfayi çevirin.

## KAYIT EKLEYEN ASP DOSYASI

Formdaki bilgilerin teslim edilecegi kayit\_ekle.asp yi insa edecegiz. Bui dosya içinde ne gibi islemler yapacagiz sizce? Isimiz formdan kimlikleriyle elemanlarin içerdigi bilgileri isteyecegiz. Bu bilgileri sayfa içindeki degiskenlerimizde tutacagiz tabi ki. Sonra eger gerekli gorülüyorsa bir takim deger kontorlleri yapacagiz. Ornegin ad kismina mutlaka bir sey girilmesini istiyorsak veya geçerli bir email adresi girilsin sarti varsa... Tüm bunlari scriptimiz içinde kontrol isini tamamladiktan sonra asil onemli kisma gelecegiz. Veritabani baglantisi yapilip bir kayit kümesi olusturulacak. Yeni kayit eklenip alanlara formdan alinan uygun veriler doldurulacak ve de tablo güncellenecek. Her sey bu kadar. Sonuçta veritabanina formdan alinan bilgiler eklenmis olacak. Ki bu da isimizi hallettigimiz anlamina gelecek.

#### kayit\_ekle.asp

**<%**

* *Eger bu sayfaya birisi direk URL yi yazarak ulasmaya çalistiysa*
* *bunu form dan bilgi gonderilip gonderilmedigine bakarak anlayabiliriz.*

**Dim** metod metod =

**Request**.ServerVariables(**"**REQUEST\_METHOD**"**)

**If** strcomp(metod, **"**POST**"**, vbtextcompare) <> 0 **Then** mesaj = **"**Hayir.. Bu dosya formundan baskasina is yapmaz. **"**

Vatandasi\_Uyar mesaj *' uyari mesaji ver.*

**Response**.end *'yaniti kes*

#### End If

* *degiskenlerimiz kundakta...*

**Dim** ad, adres, telefon, icq, email, dgunu

* *formdaki elemanlarin degerleri degiskenlerimize ataniyor*

ad = trim(**Request**.**form**(**"**ad**"**)) icq = trim(**Request**.**form**(**"**icq**"**))

adres = trim(**Request**.**form**(**"**adres**"**)) telefon = trim(**Request**.**form**(**"**telefon**"**)) email = trim(**Request**.**form**(**"**email**"**)) dgunu = trim(**Request**.**form**(**"**dgunu**"**))

* *bir kontrol yapalim ha.. ne dersiniz?*
* *vatandas bazi alanlari doldurmazsa biz de devam etmeyiz yolumuza*
* *eger ad, adres veya dogum günü belirtilmemisse..*

If **Len**( ad)=0 **Or Len**( dgunu)=0 **Or Len**( adres)=0

**Or Not** isdate(dgunu) **Then Dim** mesaj

mesaj = **"**Lütfen ad, adres ve dogum günü kutularini adam gibi doldurun**"**

mesaj = mesaj **& "**<br>Zorluk çikartmayin lütfen. Bakin ciddiyim!**"**

Vatandasi\_Uyar mesaj *' uyari mesaji ver.* **Response**.end *' isleyisi durdur. (veya sunucundan gelen yaniti kes)*

End **If**

* *diger üç form alanina bilgi girisi yapilmamis olabilir. ' tablodaki alanlara sifir uzunluk izni verilmediginden (varsayilan budur)*
* *sifir uzunluktaki degiskenlerimize bos deger (null) atayacagiz.*

If **Len**( icq)=0 **Then**

icq = null

**ElseIf Len**( telefon)=0 **Then**

telefon = null

**ElseIf Len**( email)=0 **Then**

email = null End **If**

* *veri tabani yolu ve baglanti cümlesi bildirimleri*

**Dim** Veri\_yolu, bag\_cumle

Veri\_yolu = **Server**.**MapPath**(**"**db/kisiler.mdb**"**) bag\_cumle = **"**DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=**" &** Veri\_yolu

* *ADO baglanti nesnesi yarat, nesneyi baglanti cümlesiyle aç*

**Set** bag = **Server**.**CreateObject**(**"**ADODB.Connection**"**) Bag.**Open** (bag\_cumle)

* *bir kayit seti yarat, mevut tablodan sorgu yap, kayit setini aç*

**Set** kayitseti = **Server**.**CreateObject**(**"**ADODB.Recordset**"**) kayitseti.**Open "Select** \* from kisiler**"**, bag, 1, 3

* *kayit setine gicir bir kayit ekle*

kayitseti.addnew

* *alanlara formdan aldigin verileri tika* kayitseti(**"**ad**"**) = ad kayitseti(**"**telefon**"**) = telefon kayitseti(**"**icq**"**) = icq kayitseti(**"**email**"**) = email kayitseti(**"**adres**"**) = adres kayitseti(**"**dogum\_gunu**"**) = dgunu
* *kayit setini güncelle*

kayitseti.update

* *dayinin sozünü dinle, nesneleri kapat*

kayitseti.**Close**

**Set** kayitseti = **Nothing**

bag.**Close**

**Set** bag = **Nothing**

*' is bitmistir. ufak bir bilgilendirme yapsak hos olmaz mi?*

Basariyi\_kutla()

**%>**

**<%** *'\*\*\*\*\*\*\*\** ***%>***

**<% Sub** Vatandasi\_Uyar (mesaj) **%>**

<h1>EY VATANDAS, HATA!</h1>

<font color=**"**red**"**><b>**<%**= mesaj **%>**</b></font>

<p><a href=**""** onclick=**"**history.back(1)**"**>GERI</a>

<p>© Benim Hata Denetim Mekanizmam versiyon bir

<br>Soran olursa 2000 yilinda yaptim :)

**<% End Sub %>**

**<%** *'\*\*\*\*\*\*\*\** ***%>***

**<%** *'\*\*\*\*\*\*\*\** ***%>***

**<% Sub** Basariyi\_kutla **%>**

<h1>TEBRIKLER</h1>

Veritabanina kayit eklem islemi gayet basarili bir sekilde tamamlandi.

**<% End Sub %>**

**<%** *'\*\*\*\*\*\*\*\** ***%>***

Yukaridaki kulanilan kodlarla ilgili ayrintilar sonraki sayfada.

## SORU ISARETLERI

Sayfamizin basinda bir kontrol yapisi goze çarpiyor. Hani olur ya.. Akillinin biri tarayiciyi açar, kayit\_ekle.asp nin yolunu yazar. Bu durumda kayit\_ekle.asp ye formdan bilgi gitmemistir. Ve kayit\_ekle.asp nin de isi gücü oldugundan boyle amaçsiz istemlere aninda yanit vermesini sagliyoruz. Diyoruz ki istem metodunu içeren ifadede "POST" metni yoksa (ki bu, sayfamiza form ile ulasilmamis demektir) bir hata mesaji gorüntüle.

metod = Request.ServerVariables("REQUEST\_METHOD")

If strcomp(metod, "POST", vbtextcompare) <> 0 Then

mesaj = "Hayir.. Bu dosya formundan baskasina is yapmaz. " Vatandasi\_Uyar mesaj *' uyari mesaji ver.*

response.end *'yaniti kes*

End If

## SUB RUTINLER

Yukaridaki kodda hata mesajini gorüntülemek için bir sub rutine bir degiskenle parametre gonderip çagiriyoruz. Sub VBScript te bir alt kod blogudur. Bu kullanimi yapabilmemiz için çagirdigimiz SUB sayfa içerisinde tanimlanmis olmalidir. Sayfanin en asagisinda ise bu SUB i tanimliyoruz. (yeri mühim degil, en basta da olabilir)

<% Sub Vatandasi\_Uyar (mesaj) %>

<h1>EY VATANDAS, HATA!</h1>

<font color="red"><b><%= mesaj %></b></font>

<p><a href="" onclick="history.back(1)">GERI</a>

<p>© Benim Hata Denetim Mekanizmam versiyon bir

<br>Soran olursa 2000 yilinda yaptim :)

<% End Sub %>

SUB rutinimizin adi "Vatandasi\_uyar". Bu sub, bir disaridan parametre aliyor. Parametremize mesaj adini verdik. Aldigi mesaj degerini formatlayarak ekrana yaziyor. SUB blogunu END SUB deyimi ile sonlandirmamiz gerekiyor.

## FORMDAN ISTENEN IÇERIK VE DE KONTROLLERI

ad = trim(Request.form("ad")) tipindeki formdaki bir elemanin içerigini istemeyi biliyorsunuz.

Burada farkli olarak TRIM fonksiyonunu kullandik. Bu fonksiyon VBScript de bir karakter dizisi isleme fonksiyonudur. Metnin sag ve solundaki bosluk karakterlerini temizler. Yani vatandas formun ad kutusuna killik yapip " Abuzer " yazdiysa kullandigimiz TRIM fonksiyonu bunu "Abuzer" seklinde kirpip

bize geri verecektir. Bu sekilde formdaki diger elemanlarin içerigini de isteyip belirledigimiz degiskenlerimize atama yapiyoruz. Sira geldi bir takim kontrollere. Bazi alanlarin mutlaka doldurulmasini isteyebiliriz. Bunu için sifir uzunluk kontrolü yapacagiz. LEN(degisken\_adi) fonksiyonu bize degiskenin uzunlugunu verir. Eger bu uzunluk 0 ise degiskende deger yok demektir.

If Len( ad)=0 Or Len( dgunu)=0 Or Len( adres)=0 Or Not isdate(dgunu) Then Dim mesaj

mesaj = "Lütfen ad, adres ve dogum günü kutularini adam gibi doldurun" mesaj = mesaj & "<br>Zorluk çikartmayin lütfen. Bakin ciddiyim!" Vatandasi\_Uyar mesaj *' uyari mesaji ver.*

response.end *' isleyisi durdur. (veya sunucundan gelen yaniti kes)*

End If

Yazdigimiz kontrolde eger ad, adres ve dogum gününü tutan degiskenlerimizin uzunlugunun yaninda bir de ISDATE(degisken) fonksiyonu ile dogum günü alanina girilen degerin tarih/saat türüne uygun olup olmadigi denetleniyor. Eger sartlardan en az biri (OR baglaci bunu saglar) gerçeklesmezse Vatandasi\_uyar alt rutini bir hata mesaji gorüntülenmek için çagriliyor. response.end ile sunucunun sayfayi icrasi durduruluyor.

## SIFIR UZUNLUK > NULL

Eger tablomuzdaki alanlar için sifir uzunluk izni verilmediyse, formda doldurulmasi gerekli gormedigimiz ama degre girilmemis alanlarin degeri kayitsetine yazilirken hata verecektir. Soz konusu tablo alanina sifir uzunlugunda bir deger atanamayacagini soyleyecektir. Biz de sifir uzunluklu degiskenlerimize NULL (bos) degeri atayarak bu hatanin onüne geçiyoruz. Ornek:

If Len( icq)=0 Then icq = null

End If

## VT BAGLANTISI

Bunu daha onceki VERITABANI BAGLANTISI konulu makaleden biliyorsunuz. kayit\_ekle.asp dosyasina gore db/kisiler.mdb konumunda olan veritabani dosyamizla bir baglanti açiyoruz.

Veri\_yolu = Server.MapPath("db/kisiler.mdb")

bag\_cumle = "DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=" & Veri\_yolu Set bag = Server.CreateObject("ADODB.Connection")

Bag.Open (bag\_cumle)

## KAYIT SETI

Bir ADO kayit seti nesnesi olusturuyoruz. Nesnemizi "SELECT \* FROM kisiler" sorgu lafiyla açiyoruz. Kayit setimize kisiler tablosundaki tüm kayitlar tüm satirlariyla alinmis oluyor.

Set kayitseti = Server.CreateObject("ADODB.Recordset") kayitseti.open "Select \* from kisiler", bag, 1, 3

## ARTIK YENI BIR KAYIT EKLEYELIM HA?

Artik isimizin asil kismi basliyor. kayitseti.AddNew metodu kayit setine yeni bir satir (kayit) eklemis oluyor. (Yalniz henüz tablomuza yansimamis bir kayit bu). Kayittaki alanlara gerekli deger atamalarini yapiyoruz.

Ornek: kayitseti("ad") = ad. Atamalar bittikten sonra bu kayit setindeki degisikligi tablomuza yansitacagiz. Metodumuz: kayitset.UPDATE. Artik tablomuz güncellenmistir.

Açtigimiz nesneleri kapatarak bir de basarili islem yapildigi için mesaj gorüntülemek istiyoruz. Basariyi\_kutla adiyla tanimladigimiz (parametre istemeyen) subimizi da çagirarak vazifemizi kivançla bitiriyoruz.

# VBSCRIPT ILE STRING DÜZENLEME

#### STRING NEDIR BABA?

Microsoft'taki Türkçe kelime motorundan sorumlu arkadaslar "dizgi" diyorlar "string"e. Takdir ediyorum, güzel bir kelime. Onceleri anlamadan geçiyordum bu kelimeyi. Daha sonra metinden çikardigim anlama gore bu kelimenin String yerine kullanildigini farkettim. O an ne kadar mutlu oldum bilemezsiniz. Ah...

Aslinda String deyimi, gerçke anlamiyla da ortüsüyor. "Sicim, ip" karsiligindaki bu deyim, karakterlerin pes pese siralandigi veri türünü ifade ediyor. Burada karakterler için bir ayrim yok. Rakam, harf, sembol..

Yeryüzünde mevcut karakterlerin olusturdugu bir zincir. Pascal'da dizileri kavradiktan sonra string türü verilerin de "katakter dizisi" olarak adlandirilmasinda hiç bir sakinca olmadigini gormüstük. Yine de biz "dizgi" diyelim de hem sizi fazla kizdirmadan bu anlamsiz geyigi keseyim, hem de Microsoft "yerlesikligine" riayet edelim.

Peki dizgilere ve onlari biçimleyecegimiz fonksiyonlara neden gereksinim duyuyoruz? Neden? Çünkü biz "dizgi" siz bir sey yapamayiz. (Üff... sikildim bu kelimeden. Ben "string" diye devam edecegim, kusura bakmayin.) Ornegin okudugunuz bu yazi bir string türü veridir. Formla yolladiginiz her sey string türdedir. Sorgu cümleciginden aldiginiz ivir ve zivir her sey stringdir. E bu kadar stringlerle iç içeyiz, o zaman onlara daha yakindan bakalim degil mi? Stringlerin onemli bir ozelligi matematiksel bir kiymet tasimamalaridir.

String türünde bir degiskende rakamsal veriler varsa, bunlari ancak CINT gibi bir convertion (donüsüm) fonksiyonu yardimiyle sayisal veri türüne çevirip matematiksel kavgalara sokabilirsiniz.

String lere matematik anlatamiyoz diye onlari basi bos birakacaz demedik. Programlama dillerinde bir takim metin düzenleme fonksiyonlari bulunuyor. VBScript ise Visual Basic dayisindan aldigi string formatlama fonksiyonlari ile karsimiza çikiyor. Bunlar: **Asc, AscB, AscW, Chr, ChrB, ChrW, Filter, InStr, InStrB, InStrRev, Join, Len, LenB, LCase, UCase, Left, LeftB, Mid, MidB, Right, RightB, Replace, Space, Split, StrComp, String, StrReverse, LTrim, RTrim, Trim.** Uff... Bunlarin hepsini degil de islerimizde çok fazla ihtiyaç duyacagimiz belli baslilarini inceleyecegiz.

## LEN FONKSIYONU

Length kelimesinden alir adini. Bir string ifade içindeki karakter sayisini dondürür. Len("Ankara−Izmit") ifadesi 380 (km) degil de ifadedeki karakter sayisi olan 12 yi dondürür. Ornek:

Dim mystr

mystr = "Ikimiz bir kuslar gibi yuva kuralim. Ayirmasin Mevlam bizi omür boyunca.. " Response.write LEN(mystr) *' sayfaya 74 yazar*

## UCASE, LCASE FONKSIYONLARI

Harflerin büyük−küçük donüsümünü yaparlar. UCASE("ahMet") > "AHMET", LCASE("MUCIt") > "mucit" orneklerinden anlasilacagi gibi UCASE uppercase ye, LCASE lowercaseye donüsüm yapar. Yalniz dikkat edilmesi gereken bir husus:

UCASE("isil") > "iSiL" ; LCASE("ISIL") > "isil". Artik bu istenmeyen durumu kendi yazacaginiz ozel büyüt−küçült fonksiyonlariyla asacaksiniz. (Merak etmeyin, yazariz bir tane)

## INSTR FONKSIYONU

Bu güzel fonksiyon bize arama türü bir islev saglar. Uzun bir metin içerisinde daha kisa bir ifadenin indis olarak baslangiç pozisyonunu verir. Soz dizimi: INSTR(index, uzun\_string, aranacak\_string, karsilastirmakodu) seklindedir. Burada index ise uzun string içindeki aramanin hangi siradaki karakterden baslayacagini bildirir. Bu siradan onceki metne bakilmaz. Eger index parametresi geçirilmezse 1 olarak kabul edilir. Aranilan ifadeye rastlanmazsa fonksiyon 0 degeri dondürür. Karsilastirma kodu 0 yazildiginda VbBinaryCompare yapilir, 1 yazildiginda ise VBTextCompare yapilir. Bunun ayrimina büyük küçük harf duyarliginda varirsiniz. Asagidaki ornekleri inceleyin.

Ornek:

Dim mystr

mystr = "Bilene her devir aslinda Lâle Devri / Akliniz varsa Nedim gibi yasayin..(Ü.Y.O)" Response.write Instr(1, mystr, "Bilene", 0) *'sayfaya 1 yazar*

Response.write Instr(1, mystr, "Bilene", 1) *'sayfaya 1 yazar* Response.write Instr(1, mystr, "bilene", 0) *'sayfaya 0 yazar* Response.write Instr(1, mystr, "bilene", 1) *'sayfaya 1 yazar* Response.write Instr(mystr, "Lâle") *'sayfaya 26 yazar* Response.write Instr(mystr, "lâle") *'sayfaya 0 yazar* Response.write Instr(1, mystr, "lâle", 1) *'sayfaya 26 yazar*

Response.write Instr(mystr, "Kizilcahamam Maden Sulari") *'sayfaya 0 yazar*

## REPLACE FONKSIYONU

Gayet faydali bir fonksiyondur. Uzun bir metin içerisindeki belirlediginiz ifadeyi baska ifadeyle degistirmenizi saglar.

Soz dizimi: REPLACE(string, degisecek\_olan, yerine\_konulacak, baslangiç\_indexi, degisiklik\_sayisi, karsilastirmakodu)

Parametreler:

String: Içinde degisiklik yapacagimiz metindir.

Degisecek\_olan: Metin içinde degistirmek istedigimiz alt−dizgi (sub−string) dir. Yerine\_konulacak: Belirtilen alt−dizginin yerine konulacak ifadedir.

Baslangiç\_indexi: Degistirme isleminin, stringin kaçinci karakterinden baslayacagidir. Varsayilan: 1 Degisiklik\_sayisi: Çok açik. −1 yazilirsa bulunan tüm alt−dizgiler degistirilir.

karsilastirmakodu: VBTextCompare için 1, VBBinaryCompare için 0 yazilabilir. Varsayilan 0.

Replace fonksiyonu su sekilde de islev yapar. REPLACE(string, degisecek, yerinekonulacak). Ancak fonksiyonun daha verimli kullanilabilmesi için tüm parametlerinin belirtilmesi gerekir. Ornekleri inceleyin, anlayacaksiniz.

Ornek:

Dim mystr

mystr = "Agir agir çikacaksin bu merdivenlerden" Response.write replace(mystr, "Agir", "Hizli", 1, −1, 0)

* *sayfaya "Hizli agir çikacaksin bu merdivenlerden" yazar.*

Response.write replace(mystr, "Agir", "Hizli", 1, −1, 1)

* *sayfaya "Hizli Hizli çikacaksin bu merdivenlerden" yazar.*

Response.write replace(mystr, "a", "A", 1, −1, 1)

* *sayfaya "Agir Agir çikAcAksin bu merdivenlerden" yazar.*

Response.write replace(mystr, "a", "A", 1, 2, 1)

* *sayfaya "Agir Agir çikacaksin bu merdivenlerden" yazar.*

Replace fonksiyonu bir çok yerde yardimimiza yetisir. Ornegin bir formdan alinan içerikte yer almasi muhtelemel istenmeyen ifadelerin ayiklanmasi saglanabilir. HTML taglerinin kullanilmasi onlenebilir. Ve daha bir çok yerde REPLACE size büyük fayda saglayacaktir.

## LEFT, RIGHT FONKSIYONLARI

Left bir metnin sol bogründen, Right bir metnin sol bogründen belirtilen sayida karakter alir. Sozdizimi: LEFT|RIGHT(string, uzunluk).

Ornek:

Dim mystr

mystr = "Yüz bin elle dokunurum sana, Istanbul a..." Response.write LEFT(mystr, 7) *' sayfaya "Yüz bin" yazar* Response.write RIGHT(mystr, 13) *' sayfaya "Istanbul a..." yazar*

## MID FONKSIYONU

Left fonksiyonunun biraz gelismis seklidir. Burada uzunluk parametresinden once bir de index parametresi gonderilir.

sozdizimi: MID(string, index, uzunluk) Ornek:

Dim mystr

mystr = "Yüz bin gozle seyrederim seni, Istanbul u" Response.write MID(mystr, 5, 3) *' sayfaya "bin" yazar*

## TRIM, LTRIM, RTRIM FONKSIYONU

Bosluk (space) karakterlerini kirpan fonksiyonlar. TRIM her iki yandan, LTRIM sol yandan, RTRIM sag yandan bosluklari kirpar.

Ornek:

Dim mystr

mystr = " KIRP BENI "

Response.write TRIM(mystr) *' sayfaya "KIRP BENI" yazar* Response.write LTRIM(mystr) *' sayfaya "KIRP BENI " yazar* Response.write RTRIM(mystr) *' sayfaya " KIRP BENI" yazar*

## STRREVERSE FONKSIYONU

Biraz eglenceli, biraz da fuzuli geliyor bana. Metni ters çeviriyor. Bunun için hazir fonksiyon yazmaya ne gerek vardi? Ornege bakin.

Ornek:

Dim mystr mystr = "kitap"

mystr2 = "ey edip adanada pide ye"

Response.write StrReverse(mystr) *' sayfaya "patik" yazar*

Response.write StrReverse(mystr2) *' sayfaya "ey edip adanada pide ye" yazar.. ehu ehu*

## STRING FONKSIYONU

VBScript ten yine evlere senlik bir fonksiyon. Belirtilen sayida karakterin tekrarini içeren string olusturuyor. Ornek:

Response.write STRING(5, "\*") *' sayfaya "\*\*\*\*\*" yazar*

## SPLIT FONKSIYONU

Bir metinden belirlenmis ayraçlar (delimiter) vasitasiyla belirtilen sayida alt−dizgi içeren 0 tabanli, tek boyutlu dizi üretir.

Sozdizimi: SPLIT(ifade, ayraç, sayi, karsilastirmakodu).

(Sayi yerine −1 yazilirsa tüm belirlenen tüm alt−dizgiler isleme tabi tutulur.) Ne is yaptigini anlamak için ornek yapalim.

Ornek:

Dim mystr, dizi(3)

mystr = "EvcilASP|Türkiye nin|ASP Bostani" dizi = SPLIT(mystr, "|", −1, 1)

Reponse.Write dizi(0) *' sayfaya "EvcilASP" yazar* Reponse.Write dizi(1) *' sayfaya "Türkiye nin" yazar* Reponse.Write dizi(2) *' sayfaya "ASP Bostani" yazar*

## JOIN FONKSIYONU

Split fonksiyonuna gore zir islemi yapar. Bir dizinin elemanlarini belirtilen ayraç ile birlestirip bir string üretir.

Sozdizimi: JOIN(ifade, ayraç) Ornek:

Dim dizi(3), str, str2 dizi(0) = "Ben" dizi(1) = "Sekizinci" dizi(2) = "Henri"

str = JOIN(mystr, "|") str2 = JOIN(mystr, " ")

Reponse.Write str *' sayfaya "Ben|Sekizinci|Henri" yazar*

Reponse.Write str2 *' sayfaya "Ben Sekizinci Henri" yazar*

Bu kadar. Bu ogrendigimiz fonksiyonlar bize simdilik yeter de artar bile. Ancak gelisen kosullara gore diger fonksiyonlara da ileride goz atmak gerekebilir. Haydi sagicakla kalin. Her string biçimlendirme

sirasinda EvcilASP yi hatirlayin. Eyvallah...

# VERITABANINDAKI KAYITLARI GÜNCELLEMEK

Veritabani muhabbetimiz tam gaz devam ediyor. Bu yazida bir çok veritabani operasyonunu birlikte gorecegiz. Kayit ekleme, silme, güncelleme gibi. Kullanacagimiz veritabani dosyasi [**burada**](http://evcilasp.mucit.gen.tr/ornekler/kisiler.zip)(8,15 kb). Bu dosya C:\inetpub\wwwroot\db konumunda olsun. Sira geldi script dosyalarimiza.

#### KAYITLAR.ASP

Oncelikle kayitlari listeleyecegimiz bir arabirim insa ettik. kayitlar.asp ye goz atalim. Bu ve diger ASP dosyalari C:\inetpub\wwwroot\ konumunda olsun.

**Kod1. kayitlar.asp**

**<%**

Veri\_yolu = **Server**.**MapPath**(**"**db/kisiler.mdb**"**) Bcumle = **"**DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=**" &** Veri\_yolu

**Set** bag = **Server**.**CreateObject**(**"**ADODB.Connection**"**) bag.**Open** (Bcumle)

**Set** kset = bag.**Execute**(**"Select** \* FROM kisiler**"**)

**%>**

**<%** i=1 **%>**

<p><a href=**"**kayit\_yeni.asp**"**>Yeni Kayit</a></p>

<table border=1>

<tr>

<th>#</th>

<th>Ad</th>

<th colspan=2>Eylemler</th>

</tr>

**<%** Do **While Not** kset.eof **%>**

<tr><td>**<%**= i **%>**</td>

<td>**<%**= kset(**"**ad**"**) **%>**</td>

<td><a href=**"**kayit\_duzenle.asp?id=**<%**= kset(**"**id**"**)

**%>"**>düzenle</a></td>

<td><a href=**"**kayit\_sil.asp?id=**<%**= kset(**"**id**"**)

**%>"**>sil</a></td></tr>

**<%** kset.movenext **%>**

**<%** i = i + 1 **%>**

**<%** Loop **%>**

</table>

**<%**

kset.**Close**

**Set** kset = **Nothing**

bag.**Close**

**Set** bag = **Nothing**

**%>**

Bu script, basit bir sekilde /db alt dizinindeki kisiler.mdb ile baglanti kuruyor. kisiler tablosundan aldigi tüm kayitlarin "ad" alanindaki degerlerini yazdiriyor. Ve de her bir kayit için düzenleme ve silme linkleri olusturuyor. Linkler olusturulurken sorgu stringi içinde eylemi gerçeklestirecek script dosyasina "id" anahtariyla kaydin "id" alanindaki degeri gonderiliyor. Tabi bunlar kayitseti sonunda duracak bir dongü içerisinde yapiliyor. Veriler tablo hücrelerine dongüyle doküldügüne de dikkat ediniz.

## VERITABANINDAKI KAYITLARI GÜNCELLEMEK

#### KAYIT\_DUZENLE.ASP ve KAYIT\_GUNCELLE.ASP

Kayitlarin listelendigi kayitlar.asp de her kaydin yaninda bir düzenle baglantisi var. Bu baglanti ile duzenle.asp dosyasina isaret ediliyor. Bir de sorgu cümlesi konuluyor yanina: "duzenle.asp?id=XX" Buradaki XX yerine geçerli kaydin id alanindaki degeri yaziliyor. Simdi kayit\_duzenle.asp ye bakalim.

#### Kod2. kayit\_duzenle.asp

**<%**

id = **Request**.**QueryString**(**"**id**"**)

**If Not** isnumeric(id) **Or Len**( id)=0 **Then**

mesaj **"**Yanlis Sorgu Cümlesi**"**

#### End If

*'−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−*

Veri\_yolu = **Server**.**MapPath**(**"**db/kisiler.mdb**"**) Bcumle = **"**DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=**" &** Veri\_yolu

**Set** bag = **Server**.**CreateObject**(**"**ADODB.Connection**"**) bag.**Open** (Bcumle)

**Set** kset = bag.**Execute**(**"Select** \* FROM kisiler where id =**" &** id)

**If** kset.eof **Then**

mesaj **"**Olmayan Kayit Istendi**" End If**

#### %>

<a href=**"**kayitlar.asp**"**>Kayitlar</a>

<form method=post action=**"**kayit\_guncelle.asp**"**>

<table border=1>

<tr>

<td>Ad</td>

<td><input type=**"**text**"** name=**"**ad**"** value=**"<%**= kset(**"**ad**"**)

**%>"**></td>

</tr>

<tr>

<td>Teleon</td>

<td><input type=**"**text**"** name=**"**telefon**"** value=**"<%**= kset(**"**telefon**"**) **%>"**></td>

</tr>

<tr>

<td>Email</td>

<td><input type=**"**text**"** name=**"**email**"** value=**"<%**= kset(**"**email**"**) **%>"**></td>

</tr>

<tr>

<td>ICQ</td>

<td><input type=**"**text**"** name=**"**icq**"** value=**"<%**= kset(**"**icq**"**)

**%>"**></td>

</tr>

<tr>

<td>Adres</td>

<td><input type=**"**text**"** name=**"**adres**"** value=**"<%**= kset(**"**adres**"**) **%>"**></td>

</tr>

<tr>

<td>Dogum Günü</td>

<td><input type=**"**text**"** name=**"**dgunu**"** value=**"<%**= kset(**"**dogum\_gunu**"**) **%>"**></td>

</tr>

<tr>

<td></td>

<td><input type=**"**submit**"** value=**"**güncelle**"**></td></tr>

</table>

<input type=**"**hidden**"** name=**"**id**"** value=**"<%**= kset(**"**id**"**) **%>"**>

</form>

**<%**

kset.**Close**

**Set** kset = **Nothing**

bag.**Close**

**Set** bag = **Nothing**

**%>**

**<%***'−−−−−−−−−−−−−−−−−−−−−−−−−−****%>***

**<% Sub** mesaj(msg) **%>**

<p>**<%**= msg **%>**</p>

**<% Response**.end **%>**

**<% End Sub %>**

**<%***'−−−−−−−−−−−−−−−−−−−−−−−−−−****%>***

Aslinda bu scriptin de oncekinden pek farki yok. sadece sorgu cümlesinden "id" anahtarindaki degeri aliyor. Ve bu id degerini kullanarak tablodan tek kayit seçiyor. ("SELECT \* FROM kisiler where id =" & id).

Seçtigi kayda ait degerleri ise form input alanlarina doküyor. Formun action ozelligine ise kayit\_guncelle.asp yazilmis. Güncelleme isini bu dosyaya havale ediyoruz.

#### Kod3. kayit\_guncelle.asp

**<%**

ad = **Request**.**form**(**"**ad**"**) id = **Request**.**form**(**"**id**"**)

telefon = **Request**.**form**(**"**telefon**"**) dgunu = **Request**.**form**(**"**dgunu**"**) icq = **Request**.**form**(**"**icq**"**)

email = **Request**.**form**(**"**email**"**) adres= **Request**.**form**(**"**adres**"**)

*'−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−*

Veri\_yolu = **Server**.**MapPath**(**"**db/kisiler.mdb**"**) Bcumle = **"**DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=**" &** Veri\_yolu

**Set** bag = **Server**.**CreateObject**(**"**ADODB.Connection**"**) bag.**Open** (Bcumle)

**Set** kset = **Server**.**CreateObject**(**"**ADODB.Recordset**"**) sql = **"Select** \* FROM kisiler WHERE id = **" &** id kset.**Open** sql, bag, 1, 3

kset(**"**ad**"**) = ad kset(**"**telefon**"**) = telefon kset(**"**email**"**) = email kset(**"**adres**"**) = adres kset(**"**icq**"**) = icq kset(**"**dogum\_gunu**"**) = dgunu kset.update

kset.**Close**

**Set** kset = **Nothing**

bag.**Close**

**Set** bag = **Nothing**

**Response**.**Write "**<p>Kayit Yapildi.. Baska bir arzunuz? **" Response**.**Write "**<p><a href=**""**kayitlar.asp**""**>Kayitlar</a>**"**

#### %>

## KAYIT\_YENI.ASP

kayitlar.asp de Yeni Kayit linkinde isaret edilen kayit\_yeni.asp... Yaptigi is su. Veritabanina baglanip ilgili tablodan olusturdugu kayit seti içine yeni bir kayit ekliyor. Kayitsetini güncelliyor, "id" alanindaki degeri bir degiskende sakliyor. nesneleri kapatiyor. Ve de kayit\_duzenle.asp dosyasina "id" anahtariyla yeni eklenen kaydin "id" alanindaki degeri gonderiyor. Yani bu sayfa sadece islem yapiyor. Kullaniciya gozükmeden islemi tamamlayip düzenleme sayfasina yoneliyor.

**Kod4. kayit\_yeni.asp**

**<%**

**Response**.buffer=true

Veri\_yolu = **Server**.**MapPath**(**"**db/kisiler.mdb**"**) Bcumle = **"**DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=**" &** Veri\_yolu

**Set** bag = **Server**.**CreateObject**(**"**ADODB.Connection**"**) bag.**Open** (Bcumle)

**Set** kset = **Server**.**CreateObject**(**"**ADODB.Recordset**"**) sql = **"Select** \* FROM kisiler**"**

kset.**Open** sql, bag, 1, 3 kset.addnew kset.update

id = kset(**"**id**"**) kset.**Close**

**Set** kset = **Nothing**

bag.**Close**

**Set** bag = **Nothing**

**Response**.**Redirect "**kayit\_duzenle.asp?id=**" &** id

**%>**

## KAYIT\_SIL.ASP

Sorgu cümlesindeki "id" anahtarindan alinan id degeri kullanilarak su SQL ifadesi olusturuluyor.

DELETE FROM kisiler WHERE id = " & id . Ve bu ifade baglanti tarafindan yürütülüyor (execute). Sonuçta id si alinan kayit tablodan silinmis oluyor. Kullanici da geldigi sayafaya yonleniyor.

#### Kod5. kayit\_sil.asp

**<%**

**Response**.buffer = **True**

id = **Request**.**QueryString**(**"**id**"**)

*'−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−*

Veri\_yolu = **Server**.**MapPath**(**"**db/kisiler.mdb**"**) Bcumle = **"**DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=**" &** Veri\_yolu

**Set** bag = **Server**.**CreateObject**(**"**ADODB.Connection**"**) bag.**Open** (Bcumle)

)

sql = **"**DELETE FROM kisiler WHERE id = **" &** id

**Set** kset = bag.**Execute**(sql) **Set** kset = **Nothing** bag.**Close**

**Set** bag = **Nothing**

**Response**.**Redirect** (**Request**.ServerVariables(**"**HTTP\_REFERER**"**)

**%>**

# ASRIN FIYAKASI: MAIL GONDERMEÇ!

Neden bahsedecegimizi baslikta çok açik bir sekilde belirttik. Internet’in en kullanisli ve insani boyutlarindan biri de maildir bence. Çok zaman geçmedi mektup zarflarinin kapaklarini dilimizle isladiktan sonra büyük bir heyecan ve “ümit” içerisinde yapistirip güle oynaya Posta Teskilati binasinin yolunu tuttugumuz günlerin üzerinden. Mektup yazmayi çok severdik biz günesin çocuklari olarak. Ogretmenler once karsinin hal ve hatirini sormayi ogütlediler bize. Biz de hep “orada havalar nasil” diye baslardik test.mektuplar’a. Ve kendimizden bahsetmeye gelince de “Burada havalar iyi” en büyük vurgumuz olurdu. Sonra amcalara, teyzelere yazilan mektuplarin yerini Leylâ’ya yazilan mektuplar aldi. Ozenle yazdigimiz “sevda” kokulu cümleler, postaci amcanin eline ne büyük güvenle teslim edilirdi! Postaci amcalar adresi hiç sasirmazdi, sevda menzile ererdi. O zaman cep telefonu olmadigi için karsidaki arkadas aninda “mektubun geldi arkadasim, haziran da geldi” içerikli kisa bir mesaj atamazdi. Siz Leylâ’ya mektubun ulastigini, ancak Postaci

amcanin karsi sokaktan elinde isildayan bir zarfla sizin kapiya dogru yonelisine tanik oldugunuzda anlardiniz. Hangi heyecanlar salinirdi içinizde kim bilir. Hangi yildizlara kosardi gozleriniz...

Artik mektuplardan çok uzaklastik. Çünkü artik internete, onun getirilerine çok yaklastik. Posta Teskilati’na giden, yürüdükçe uzayan yolun yerini, basit “send” tuslari aldi. Iadeli taahhütlü mailler iletisimin agir yükünü kolaylikla sirtladi.

Simdi herkesin neredeyse kendi Posta teskilati var. Mail sunucular haril haril çalisiyor. Bu hizli degisim, bizi mail iletisiminin üzerine daha ciddi egilmeye itiyor. Deminden beri “ne diyor yav bu, biz buraya ASP türküleri dinlemeye geldik” dediniz. Sabrettiginize degecek umarim. Bu kisa muhabbetten sonra ASP teknolojisiyle nasil postacilik yapacagimiza deginecegiz. Artik buraya kadar okudunuz, gerisine katlanirsiniz degil mi :)

#### POSTACI OLUYORUZ.

ASP bize kolay bir sekilde mail gonderme imkani sunmakta. Yalniz bu kolayligi çogu zaman verimli bir sekilde kullanamiyoruz. Çünkü bu mail ileti sistemi, mail sunucunuzu kullanmanizi tesis eden hazir sunucu (mail component) bilesenleriyle saglanmakta. Bunlar da sunucudan sunucuya degisiyor maalesef. Kimi sunucunun tercihi CDO, kimininki Persist, kimininki de JMail. Piyasada envai çesit mail bileseni mevcut. Genelde bedava oluyorlar. Bunlardan en yaygini saydigimiz üç tanesi. Bu bilesenleri ASP ile kullanarak mail gondermek için bu bilesenlerin, sunucu sisteme kayit edilmesi (register) gerekir. Bazi bilesenlerin kendi kurulum programlari, bu isi size zahmet vermeden yapmaktadir. Bazilarini ise siz, bir zahmet kayit edeceksiniz (c:\windows\system\regsvr32.exe [bilesen] komutuyla olabilir.)

Ancak bizim büyük ihtimal kendi sunucumuz yok. Bilesen kayitlariyla da isimiz yok diyebilirsiniz. O zaman mail gondermeyi hala istiyorsaniz dosyalarinizi “publish” ettiginiz web sunucuda herhangi bir mail bileseni oldugundan emin olunuz. Bunun için sunucu serifiyle iletisime geçin. Web sunucunuzdaki mail bilesenini mutlaka ogrenin. Çünkü mail gonderme scriptinizde kullanacaginiz kodlar, bilesenden bilesene az da olsa fark gosterecektir. Her bilesen bazi kendine ozgü metodlarla mail gondermeye programlanmis. Yaptiklari is ayni da, niye metodlari farkli dersiniz? Valla bilmem. Biri obj.send ile mail gonderiyor, digeri obj.execute ile ornegin. Bileseni yazan adamlarin kendi tercihleri bu. Sikâyet hakkimizin oldugunu sanmiyorum. Ne güzel, is çikiyor iste.

## DERS BASLIYOR, ONCE AMAÇ!

Amacimiz ne? Mesela kullanicilarin sitedeki bir formu doldurarak site yoneticisine mail gondermesini saglamak. Veya “yakin bir arkadasina bu siteyi ziyaret etmesini tembihle“ düzenegi kurmak. Biz, daha “mantikli” oldugundan birinciye ornek verelim.

## FORM'UM

Once form içeren bir HTML belgesi üretmemiz lazim. Dosyayi üretip adina da ne diyelim? Evet, bildiniz: form.html..

#### Sekil1. Form.html

<html>

<head>

<title> SITE SERIFINE MAIL

GONDER </title>

</head>

<body bgcolor=**"**#FFFFFF**"**>

<form method=post action=**"**yolla.asp**"**>

Isim : <input

type=**"**text**"** name=**"**isim**"**><br>

Mail : <input

type=**"**text**"** name=**"**mail**"**><br>

Konu : <input

type=**"**text**"** name=**"**konu**"**><br>

Mesaj : <textarea

name=**"**mesaj**"** rows=**""** cols=**""**></textarea><br>

<input

type=**"**submit**"** value=**"**yolla**"**>

</form>

</body>

</html>

Iste meshur form.html. Belgenin tarayicida gorünümünden siz sorumlusunuz tabi ki. Burada “estetik bir form tasarlayalim da karizmayi titretmeyelim” kaygisi tasimadigimizi belirtelim. Ornegimizde onemli olan islevdir. Formdaki elemanlarin kimlikleri (name) de dikkat edecegimiz noktadir.

Insa edilen bu formu dolduran ziyaretçi, olagan sartlarda çevrede bir “submit” dügmesi bulup tiklayacaktir. Ve tarayici tetiklenip formun “action” ozelliginde belirtilen “yolla.asp” ye kosacak. Bu, yolla.asp diye bir ASP komut dosyasi yazacagimiz anlamina geliyor. Yolla.asp yi yazarken bir mail bilesen ornegi yaratip, belirli komutlari kullanarak mail gonderme islemini tesis edecegiz.

## CDO ILE YOLLA.ASP

En yaygin mail gonderme bileseni olan CDO dan bahsediyoruz. Bu bilesen NT Option Pack 4 ile geliyor. Yani ayaginiza kadar geliyor. Yayginligi da buradan geliyor tabi ki. Eger web sunucunuzda CDO mail bileseni yüklüyse burada verecegimiz kodlari kullanabilirsiniz.

**Sekil2. yolla.asp (CDO ile)**

**<%**

* *formdan içerik al*

isim = **Request**.**form**(**"**isim**"**) mail = **Request**.**form**(**"**mail**"**) konu = **Request**.**form**(**"**konu**"**) mesaj = **Request**.**form**(**"**mesaj**"**)

* *mektup kime gidecek abisi*

kime = **"**[admin@sevdilsevilmedim.com](mailto:admin@sevdilsevilmedim.com)**"**

* *bir CDO Newmail nesne ornegi yarat*

**Set** mektup = **Server**.**CreateObject**(**"**CDONTS.NewMail**"**) *' çok açik, geleneksel atamalar*

mektup.From = isim **& "** <**" &** mail **& "**>**"**

mektup.To = kime mektup.Subject = konu mektup.Body = mesaj

* *hata olursa mizikçilik yapmadan devam et*

on error resume next *' ve sadede geldik.* mektup.Send

* *hata olusursa*

**If** err <> 0 **Then**

vukuat = err.description

**Response**.**Write "**Vukuat var: **" &** vukuat

**Else** *' is temizse*

tebrik = **"**Tebrik ederiz, maili yolladik**" Response**.**Write** tebrik

**End If**

*' açtigin nesneyi kapa...* **Set** mektup = **Nothing** *' bitti*

**%>**

## PERSIST ILE YOLLA.ASP

Bir de Persist Mail gonderme bileseniyle deneyelim bu isi. Maksat çesitlilik olsun. Karsimiza hangisi çikarsa çiksin, defterini dürelim.

**Sekil2. yolla.asp (Persist Ile)**

**<%**

* *formdan içerik al*

isim = **Request**.**form**(**"**isim**"**) mail = **Request**.**form**(**"**mail**"**) konu = **Request**.**form**(**"**konu**"**) mesaj = **Request**.**form**(**"**mesaj**"**)

* *mektup kime gidecek abisi...*

kime = **"**[admin@sevdimsevilmedim.com](mailto:admin@sevdimsevilmedim.com)**"**

* *bir Persist Mailsender nesnesi...*

**Set** mektup =

**Server**.**CreateObject**(**"**Persits.MailSender**"**)

* *mektup yollanirken hangi SMTP sunucu kullanilacak*

mektup.Host = **"**mail.sevdimsevilmedim.com**"**

* *gonderenin mail adresi*

mektup.From = mail

* *gonderenin adi* mektup.FromName = isim *' gidecek adres..* mektup.AddAddress kime *' karsilik yazilacak adres*

mektup.AddReplyTo kime mektup.Subject =konu mektup.Body = mesaj

On Error Resume Next mektup.Send

*' hata varsa*

**If** err <> 0 **Then**

vukuat = err.description

**Response**.**Write "**Vukuat var: **" &** vukuat

**Else** *' is temizse*

tebrik = **"**Tebrik ederiz, maili yolladik**" Response**.**Write** tebrik

**End If**

*' açtigin nesneyi kapa...* **Set** mektup = **Nothing** *' bitti*

**%>**

## JMAIL ILE YOLLA.ASP

Son olarak Jmail kullanimina bakalim.

**Sekil4. Yolla.asp (Jmail Ile)**

**<%**

* *formdan içerik al*

isim = **Request**.**form**(**"**isim**"**) mail = **Request**.**form**(**"**mail**"**) konu = **Request**.**form**(**"**konu**"**) mesaj = **Request**.**form**(**"**mesaj**"**)

* *mektup kime gidecek abisi...*

kime = **"**[admin@sevdimsevilmedim.com](mailto:admin@sevdimsevilmedim.com)**"**

* *bir JMail SMTPMail nesnesi...*

**Set** mektup = **Server**.**CreateObject**(**"**JMail.SMTPMail**"**) *' SMTP sunucusu*

mektup.ServerAddress = **"**mail.**Server**.here**"**

* *gonderen*

mektup.Sender = isim **& "** <**" &** mail **& "**>**"**

* *konu*

mektup.Subject = konu

* *aliciyi belirle* mektup.AddRecipient = kime mektup.Body = mesaj

On Error Resume Next mektup.**Execute**

* *hata varsa*

**If** err <> 0 **Then**

vukuat = err.description

**Response**.**Write "**Vukuat var: **" &** vukuat

**Else** *' is temizse*

tebrik = **"**Tebrik ederiz, maili yolladik**" Response**.**Write** tebrik

**End If**

**Set** mektup = **Nothing**

**%>**

Scriptlerimiz ornek olmasi hasebiyle basit tutuldu. Siz isterseniz (ki gerekebilir) bu scriptlere, formdaki alanlarin durumunu, geçerli mail adresi girilip girilmedigini kontrol eden, hatali durumlarda daha fiyakali uyarilar verip, adami tekrar forma gotüren daha kullanisli ozellikler ekleyebilirsiniz.

Üç ayri mail bileseniyle, üç farkli yolla.asp çikarttik ortaya. Siz kendi web sunucunuzun kosullarina gore istediginizle is yapabilirsiniz. Ancak web sunucusu size giciklik yapip, mail gonderme bileseni tahsis etmiyorsa boyle yolla.asp lerle bosuna ugrasmayin. Mail gonderme gibi masum bir isteginizi olumlu karsilayan sunucular aramaya koyulun. Mektup yazmayi da ihmal etmeyin. Kendi kendinize de olsa mektup yazin, zarfa koyun. Bir kosede dursun oyle. Duygularimizi baska ne türlü muhafaza edecegiz? Elbette yazarak.

Yazarak!

## BIR ONERI

Mail gonderme isi, karsimiza muhtemelen çok yerde çikacak. Bunu goz onünde bulundurarak kendimize bir iyilik yapabiliriz. Mail gonderme isini bir fonksiyona yükleriz, bu fonksiyon harici bir dosyada sakli kalir. Ve mail gonderme isi yapacagimiz bir script içerisinde de soz konusu dosyada bulunan mail gonderen fonksiyon, bir takim parametreler gonderilerek çalistirilir. Boylece mail gonderme scripti üzerinde yapilmasi gereken en ufak degisiklik, sadece bu dosyadaki fonksiyon üzerinde yapilir. Alt yordamlar ve fonksiyon kullanimi, programatik eylemlerin isleyisini ve akisini kontrol etmek açisindan çok onemli bir yere sahiptir. Dilerseniz simdi mail gonderen bir fonksiyon ornegi yazalim. Bu fonksiyonu yazdigimiz belgeyi mailf.asp olarak kaydedelim.

**Sekil5. mailf.asp**

**<%**

**Function** mailgonder(kimden\_isim, kimden\_mail, kime, konu, mesaj)

**Set** mektup = **Server**.**CreateObject**(**"**CDONTS.NewMail**"**) mektup.From = isim **& "** <**" &** mail **& "**>**"** mektup.To = kime

mektup.Subject = konu mektup.Body = mesaj on error resume next mektup.Send

**If** err <> 0 **Then** mailgonder = **False Else**

mailgonder = **True End If**

**Set** mektup = **Nothing End Function**

**%>**

Ve son olarak baska islerin arasinda mail gonderme ihtiyacimizin dogdugu bir deneme.asp farzedelim. deneme.asp içerisinde bu fonksiyonu kullana parametreleri gonderip çalistiracagiz. Ancak once bu fonksiyonu içeren mailf.asp dosyasini, sayfamiza "include" (dahil) etmemiz lazim ki fonksiyon sayfamiz hudutlari içerisinde çagrilabilir olsun.

**Sekil6. Deneme.asp**

<!−−#INCLUDE

FILE=**"**mailf.asp**"**−−>

**<%**

*' −−−−−−*

*' baska isler ' −−−−−− ' ....*

gonderildi = mailgonder(isim, mail, kime, konu, mesaj)

**If** gonderildi **Then Response**.**Write "**maili yolladik**" Else**

**Response**.**Write "**hata olustu, maili yollayamadik**"**

**End If**

*' −−−−−−*

*' baska isler ' −−−−−− ' ....*

**%>**

Bu gibi rutinlesen, çok tekrarlanan isleri alt program bloklarina yüklemek, her zaman rahat, estetik, kullanisli ve profesyonel çozüm olcaktir. Zaten bu isin raconu da budur. En olmadik seyleri bile sub rutinlere simdiden yükleyerek, bu isin havasina girmek faideli olacaktir nitekim :) Eyvallah der Mucit.

# ASP VE KABUKLU YEMISLER

Siteye giriyordunuz. Adam size "sayfayi kisisellestir" diye bir link koymus. Tiklayip içinizi doküyorsunuz. Siteyi bir sonraki ziyaretinizde site size adiniza hitaben "merhaba koçum, seni taniyorum, bana kendini anlatma, çok iyi bilirim ben seni ... ismin su, sevdigin kalem markasi Rotring, basin agriyinca onceden Novalgin alirdin simdi modaya uyup Aspirin Forte.. monitorün philips marka ve sen simit seklindeki power dügmesine gicik oluyorsun ... haa bir de bu siteyi 2. ziyaretin ... nerden bildim ama... hahaha" türünden saklabanliklar yapiyor. Peki bunu nasil beceriyor? Hepiniz çerezlerle oldugunu biliyorsunuz.

Biz yine de çerezlerin üzerinde kisaca duralim. Çerezler istemci bilgisayar yollanan kabuklardir. Siteniz bir yemis olarak düsünülürse, cookie denilen seyler de yemisin kabuklaridir. Siz yemisi yersiniz, kabuklarini bilgisayariniza atarsiniz. Yalniz bu kabuklu yemis kullanan siteler için geçerli. Bir de tarayicinizin yemis teknolojisini destekleme kabiliyeti ve "yemis kabuklarini kabul etme" modunda olmasi gerekir. Yani oz olarak kabuklu yemis dagitan bir siteye gireceksiniz, bir de yemis yemeyi bileceksiniz. Anlastik mi? Büyük ihtimalle MS Internet Explorer kullaniyorsunuz. Tarayiciniza cookie kabul etmesini de tembihledikten sonra ise koyulalim.

ASP ile istemci bilgisayarlara çerez yollamak çok basit bir istir. Gerçekten çok basit. ASP nin yerlesik delikanlilarindan RESPONSE nin bu is için ozel bir metodu var: COOKIES. Istemciye çerez yollamak için bu metodla birlikte anahtar degerler (bir veya iki) gonderiyoruz. (bir ayrinti: ASP de iki anahtar sadece cookies kolleksiyonunda kullaniliyor). Simdi ornek kodu inceleyelim:

Response.cookies("kabuk")= "Evet, kabuk" Response.cookies("kabuklarim")("adi") = "Mucit"

Response.cookies("kabuklarim")("sevdigi\_icecek") = "Kizilcahamam Maden Suyu ve Sodasi" Response.cookies("kabuklarim")("medeni\_durumu") = "Bekar!" Response.Cookies("kabuklarim").Expires = Now() + 90

Gordügünüz üzre birinci satirda tek anahtar açiyoruz ve bir deger atamasi yapiyoruz. Ikinci satirda "kabuklarim" anahtari altinda "adi" alt anahtari açiyoruz ve "Mucit" degerini koyuyoruz. Ilk anahtar sitemizden yollanan çerezleri gruplandirmak için. Dizinleme türünde bir sey yapiyoruz nitekim. Ornegin bir anketimiz var ve oylandigi zaman bilgisayara çerez yollamak istiyoruz. O zaman "anket" diye yeni bir anahtar altina alt anahtarlar açabiliriz. Boylece sitedemizden yollanan farkli amaçli çerezlerin gruplandirilmasi saglanmis olur. Yolladigimiz çerezlerin ikinci anahtarinda sakliyoruz bilgiyi. Ornegin ilk satirdaki "adi" anahtarinda bir deger gonderiliyor istemciye. Ayni sekilde üçüncü ve dordüncü satirlarda da gereksiz bilgiler gonderilmeye çalisilmis. Yalniz son satirda kismen degisik bir kullanim var. "kabuklarim" a kadar ayni her sey. Ancak "kabuklarim" dan sonra bir alt deger anahtari açilmamis da EXPIRES alt metodu kullanilmis. Tahmin ettiginiz gibi yemisin ne zaman çürüyüp kullanilmaz hale gelecegine (expired olacagina) karar veriyoruz. Buraya tarih/saat yani ejnebi dilde date/time veri türünde deger atamasi yapiliyor. Atamaya bakarsaniz Now() + 90 denmis. Yani simdiki andan tam 90 gün sonra bu yemis çürüsün, kokussun emri verilmis. Genelde bunu 365 yaparlar. Neden derseniz bir yil 365 günden olusuyormus...:)

Peki çerezi yolladik tarayiciya. Yani kabuklari. Peki bu kabuklari gormek istedigimiz zaman ne yapmamiz gerekiyor. Iste o zaman ASP nin diger delikanli nesnesi REQUEST talip oluyor goreve. Çünkü bir seyleri talep etmek onu gorevi. Biz de ona bir emir verecegiz. "Çerezler arasindan su anahtardaki degeri bana bir getiriver bakayim" denildigi anda size o degeri getirecektir. Bu metodu kullanmak, response ile çerez yollamaya çok benzer. Ancak bir fark vardir. Response de ilgili anahtara atama yapilir, burada ise request zaten bize bir deger getirir, biz de bu degeri isimize geldigi gibi kullaniriz. Asagida REQUEST ile çerezden bilgi almayi gorüyorsunuz. Yalniz ben burda yukaridaki çerez yollama orneginin icra edildigini farzediyorum.

Adim = Request.cookies("kabuklarim")("adi")

ne\_icerim = Request.cookies("kabuklarim")("sevdigi\_icecek") medeni\_halim = Request.cookies("kabuklarim")("medeni\_durumu")

Response.Write Adim & " / " & ne\_icerim & " / " & medeni\_halim

*' sayfadaki çikti: Mucit / Kizilcahamam Maden Suyu ve Sodasi / Bekar!*

Iste bu orneklerde çerezle bilgi yollama ve de çerezden bilgi alma tekniginin ne kadar kolay oldugunu gorüyorsunuz.

## NERELERDE KULLANILABILIR?

Aslinda kabuklu yemisler, windows'un registry teknolojisinin ufak modelidir. Bilgisayariniza yüklediginiz bir çok program Windows'un registry veritabaniyla birtakim münasebetlerde bulunur. Programda varsayilan ayarlarda yaptiginiz degisiklikler Windows Registry veritabaninda belirlenen anahtarlarla kaydedilir.

Programi daha sonra tekrar açtiginizda, degistirdiginiz ayarlar Registry den okunur ve programa yansitilir. Bunlara en basit ornek pencere boyutlaridir. Programin pencere boyutlarinda yaptiginiz düzenlemeler, bir sonraki açisinizda korunur. Bu ayarlar harici dosyalarda da saklanabilir, ancak Windows Registry etkili, hizli ve profesyonel çozümdür.

Iste yüklediginiz programlarin registry münasebeti ile web sitenizin cookie münasebeti arasinda pek fark yok. Site kisisellestirilmesi iste bu çerezlerle yapiliyor. Çerezler, yani kabuklu yemisler tasarladiginiz, gelistirdiginiz ASP tabanli bir sitede ne isinize yarayabilir? Bu sorunun cevabi sizin gereksinimlerinize bagli. Ornegin siz bir portallik yapmaya soyunduysaniz kisisellestirilebilir bir site iddianiz da olabilir.

Kullanicinizin hangi alanlara ilgi duydugunu ogrenirsiniz. Bu bir çerez olarak kullanici tarafina kaydedilir. Kullanici sonraki ziyaretlierinde ilgi alanlarina yonelik haberler, linkler ve bilgiler bulur. Ve sevinir. Çünkü kendini taniyan, ona deger veren, ilgilerini, tercihlerini goz onünde bulunduran bir site vardir karsisinda. "Müsteri memnuniyeti" diye bir seyi çok sik duyar olduk. Müsteriye insani boyutta deger veren bankalar reklam yapti. Pamukbank müsterilerinin dogum günlerinde kart gonderdigini duyurdu. Sizce bunlar zor seyler miydi? Hayir! Is sadece yaratici düsünmekte. ASP size bu firsati sonuna kadar veriyor. Çerezlerle neler yapabileceginizi kendiniz düsünmelisiniz. Biz çok kisir bir ornek verelim, siz kendi ozgün uygulamalarinizi gelistirin. Bu ornek, ornek olmasi nedeniyle basit ve de klasik olcak. Amaç su: Adam siteye gircek, sitemiz adamin siteye onceden gelip gelmedigini, geldiyse kaç kez geldigini felan tespit edecek. Akilli bir sey olcak yani. Biz de "Aa.. nerden bildi, bu sitede gelecek var" diyerek sasiracagiz.

**Sekil1. Kisisel Ziyaret Sayaci**

<html>

<head>

<title> KISISEL ZIYARET SAYACI </title>

<meta

http−equiv=**"**Content−Type**"** content=**"**text/html; charset=windows−1254**"**>

<meta name=**"**Generator**"** content=**"**EditPlus**"**>

<meta name=**"**Author**"** content=**"**Mucit**"**>

</head>

<body bgcolor=**"**#FFFFFF**"**>

**<%**

ziyaret = **Request**.**Cookies**(**"**sitem**"**)(**"**ziyaret**"**)

**If** ziyaret = **"" Or Not** isnumeric(ziyaret) **Then Response**.**Cookies**(**"**sitem**"**)(**"**ziyaret**"**) = **"**0**" Response**.**Cookies**(**"**kabuklarim**"**).**Expires** = Now() + 365

**Response**.**Write "**Siteye ilk defa geldiniz. Saolun. Bir daha gelin haaa..**"**

**Else**

ziyaret = ziyaret + 1 **Response**.**Cookies**(**"**sitem**"**)(**"**ziyaret**"**) = ziyaret **Response**.**Cookies**(**"**kabuklarim**"**).**Expires** = Now() + 365

**Response**.**Write "**Siteye **" &** ziyaret **& "**. gelisiniz, aferin hep gelin haa.. **"**

**End If**

**%>**

</body>

</html>

Bitti. Bundan sonra cookie, çerez yahut kabuklu yemis kullanimi sorun olmasa gerek. Haydi kolay gelsin.

## KAYITLARI SAYFALAMAK (DB PAGING)

Merak etmeyin bu sefer uzatmadan direk konuya girecegim. Deliop diyor ki "uyutuyorsun anlatirken".. Hakli olabilir. Simdi uyutmadan sadede girelim hadi..

## KAYITLAR ÇOK FAZLAYSA

Diyelim kayitlariniz oyle yenilir yutulur sayida degil. Bes adam zor tasiyor. :−] Tek sayfada 500 tane kaydi gorüntülemek hem estetik hem de kullanim açisindan dezavantajli oluyor biliyoruz. Bunun için kayitlari, kendi belirledigimiz paketlere bolerek gorüntülemek isteriz. Ve de her sayafada ileri−geri navigasyonla diger kayitlara ulasmamizi saglayan linkler olsun.

Evet, simdi bu isi ornegiyle gorecegiz. Kayitlari sayafya dokmeyi ogrendiyseniz buradaki anlatilanlar size hiç

karisik gelmeyecek. Çünkü yaptigimiz sadece ufak bir ek islem. Sadece tek dosyamiz var: bolunmus\_kayitlar.asp . Veritabani olarak da, onceki makalelerde de kullandigimiz, artik EvcilASP ye mal olmus kisiler.mdb yi kullanacagiz. Bu ornek veritabanimizi henüz edinmediyseniz [**buyrun.**](http://evcilasp.mucit.gen.tr/ornekler/kisiler.zip)

#### bolunmus\_kayitlar.asp kaynak kodu

**<%**

* *Sorgu cümleciginden yon parametresi istiyoruz.*
* *Parametredeki degere gore o anki sayfa nosunu tutan ' oturum degiskenine uygun degeri atiyoruz.*

yon = **Request**.**QueryString**(**"**yon**"**) **Select Case** yon

**Case "**ileri**"** *' ileri gidilmisse*

Session(**"**simdiki\_sayfa**"**) = Session(**"**simdiki\_sayfa**"**)

+ 1

**Case "**geri**"** *' geri gidilmisse*

Session(**"**simdiki\_sayfa**"**) = Session(**"**simdiki\_sayfa**"**)

− 1

**Case Else** *' hiç biri degilse* Session(**"**simdiki\_sayfa**"**) = 1 **End** select

* *Klasik veritabani baglantilari*

Veri\_yolu = **Server**.**MapPath**(**"**kisiler.mdb**"**) Bcumle = **"**DRIVER={Microsoft Access Driver (\*.mdb)};DBQ=**" &** Veri\_yolu

**Set** bag = **Server**.**CreateObject**(**"**ADODB.Connection**"**) bag.**Open** (Bcumle)

* *Kayit seti*

**Set** kset =

**Server**.**CreateObject**(**"**ADODB.Recordset**"**)

* *recordseti açalim. cursor type ozelligi olarak 1 (adopenkeyset) gonderdik*

kset.**Open "Select** \* FROM kisiler**"**, bag, 1 *' kayit setinin pagesize ozelligine, bir sayfada gorünecek maximum*

* *kayit sayisini atiyoruz. biz 3 dedik bakalim*

kset.pagesize = 3

* *bir takim kontroller.. kuralsiz geri veya ileri gidislerde ilk sayfa gorünsün.*

**If** Session(**"**simdiki\_sayfa**"**) > kset.pagecount **Then**

Session(**"**simdiki\_sayfa**"**) = 1 end **If**

**If** Session(**"**simdiki\_sayfa**"**) < 1 **Then**

Session(**"**simdiki\_sayfa**"**) = 1

end **If**

* *recordsete hangi sayfada oldugumuzu bildirelim..*

kset.absolutepage = Session(**"**simdiki\_sayfa**"**)

* *navigazyon rutinimizi çagiralim.. ' o da bize yon linklerini hazirlasin.*

**Call** yoneltgen (Session(**"**simdiki\_sayfa**"**), kset.pagecount, yon\_linkleri)

**%>**

**<%**= yon\_linkleri **%>** [ **<%**= Session(**"**simdiki\_sayfa**"**) **& "** of **" &** kset.pagecount**%>** ]

<br><br>

<table border=1>

<tr>

**<%** For k = 0 to kset.fields.count−1 **%>**

<th>**<%**= kset.fields(k).name **%>**</th>

**<%** next **%>**

</tr>

**<%** i=1 **%>**

**<%** Do **While Not** kset.eof

**And** i<=kset.pagesize **%>**

<tr>

**<%** For k = 0 to kset.fields.count−1 **%>**

<td>**<%**= kset.fields(k) **%>**</td>

**<%** next **%>**

</tr>

**<%** kset.movenext **%>**

**<%** i = i + 1 **%>**

**<%** Loop **%>**

</table>

<br>

**<%**= yon\_linkleri **%>**

**<%**

kset.**Close**

**Set** kset = **Nothing**

bag.**Close**

**Set** bag = **Nothing**

**%>**

**<%** *' \*\*\*\*\*\** ***%>***

**<%**

* *yon linklerini hazirlayan* ***Sub*** *rutinimiz...*

**Sub** yoneltgen(simdiki, hepsi, yon\_linkleri) makro = **Request**.servervariables(**"**THIS\_PAGE**"**)

If simdiki > 1 **Then**

str1 = **"**<a href=**""" &** makro **& "**?yon=geri**""** >[ <

Geri ]</a>**"** end if

If simdiki < hepsi **Then**

str2 = **"**<a href=**""" &** makro **& "**?yon=ileri**""** >[ Ileri

> ]</a>**"**

end **If**

yon\_linkleri = str1 **& " " &** str2

**End Sub**

**%>**

## BIR TAKIM AYRINTILAR...

Ilk olarak bir **SELECT...CASE** çok durumlu kontrolümüz ile scriptimizin URL siyle gonderilen yon parametresinin degerini inceliyoruz. Eger "ileri" degeri gonderildiyse **session("simdiki\_sayfa")** oturum degiskenimizin degeri bir artiriyoruz. "geri" degeri gelmisse tam tersini. Bu kosullara uymayan bir durumda (case else) sayfa sayisini tuttugumuz degiskene 1 degerini atiyoruz. Yani çagrilan sayfanin ilk sayfa olmasi saglanacak bu sekilde.

Sonra alisik olunan veritabani baglantimizi kuruyoruz. Baglanti nesnemizi açiyoruz. Hemen ardindan ADO nun Recordsetinden bir adet olusturuyoruz. Kurguladigimiz SQL cümlesiyle recordset içinde herhangi bir kayit toplulugu olusturuyoruz. Ve sonra zurnanin zirt dedigi yer (ulan bunu kim soylerdi be? Sanirim Fizik hocasiydi..) gelmekte. Recordset nesnemizin Pagesize adinda bir ozelligini kullaniyoruz. Bu ozellige tek sayfada maximum kaç kayit gorünmesini istiyorsak sayisini atiyoruz.

kset.pagesize = 3

Biz 3 tane kayit gorünsün diye buraya 3 yazdik. Siz 5 tane gorünmesini istiyorsaniz buraya 3 degil de 5 yazin olur mu? (Ehu EHu.. egleniyorum biraz, kusura bakmayin)

Ardindan iki adet denetim yapiyoruz. Adam son sayfadayken "ileri" yonünde istem yaparsa, (daha ilerisi olmayacagina gore hata olusacak) veya ilk sayfadayken "geri" yonünde istem yaparsa (ki yine hata olusacak) sayfa numarasini 1 yapiveriyoruz. Yani ilk sayfadaki kayitar gorünsün diyoruz. Buradaki denetimlerde (IF..THEN) recordset in **pagecount** ozelliginin dondürdügü degeri kullaniyoruz. kset.pagesize = 3 diyerek bolümledigimiz kayitarin kaç sayfadan olusacagi kset.pagecount ozelligine sayisal olarak ataniyor. Eger recordsette 8 kayit varsa, 3 erli dilimlere bolersek 3 sayfadan olusacak (pagecount 3 degeri dondürecek) demektir. Boylece simdiki sayfa ile toplam sayfa sayisi arasindaki iliskiye gore denetimimiz gereken islemi yapiyor.

Ve recordsetin baska bir ozelligi. kset.absolutepage = Session("simdiki\_sayfa")

Bu ozellik, cursoru hareket ettiriyor. Ama neye gore? Bulunan sayfa numarasina gore. O an ilk sayfadaysak cursor (imleç) 1 inci kayit diliminin ilk kaydina haraket ediyor. Ornegin 3. kayit sayfasina geldiysek cursor hopppa 3. dilimin en basina gidecek. Her sayafa maximum 3 kayit varsa, (3\*2)+1 = 7. kayda gidecek. Nasil hos degil mi?

Buraya kadar onemli islerin çogunu yaptik. Simdi sira kosula uyan kayitlari dokmeye geldi. Ancak once vatandaslarin sayfalar arasinda dolasmasini saglamak için yon linkleri koymaliyiz. Birinde "ileri" yazsin,

digerin de "geri". Ve tiklandiklarinda scripti uygun URL parametresiyle tekrar çagirsinlar.

Bu linkleri olusturmak için bir sub rutin kullandim. Ne gerek vardi? Demistim ya ufak seyleri bile sub layarak programlama havasindan kopmamis oluyorsunuz. Tavsiye ederim. Buradaki sub rutinimize

iki parametre (input) gonderiyoruz, bir parametre (output) aliyoruz. Birinci parametre simdiki sayfanin numarasi, ikinci toplam sayfanin numarasi. Aslinda gondermesek de sub rutinimiz global degiskenleri kullanabilecektir. Ama gondermeyi de ihmal etmeyin. Bu sub in ne islev yaptigini, hangi degiskenleri kullandigini takipte faydali olur. Gelen deger ise bir string. Yani olusturulan HTML yon linkleri. Sub rutinimiz simdiki sayfa ve toplam sayfa degerlerine gore kontrolleri yapip "ileri" ve/veya "geri" linki olusturuyor. Ve bunu bir string degiskenle ana program bloguna gonderiyor.

Ve kayitlar dongülerle tablolaniyor.

FOR...NEXT dongümüzle field (alan) lerin adlarini yazdiriyoruz. kset.fields.count, sütun (field) sayisini dondürür, kset.fields(index).name ise index.inci sütunun adini dondürür.

Do While Not kset.eof And i<=kset.pagesize diye bir on kontrolle dongüyle de sirayla kayitlar tabular data lara dokülüyor. ( kset.fields(index) ozelligi indexinci alandaki degeri dondürür. )

Açilan nesneleri kapatiyoruz. Evimize gidiyoruz. KIB.

# HARIÇTEN GAZEL OKUMAK

DISARIK DEGIL IÇERIK :)

Merhaba beyler, gorüsmeyeli nasilsiniz? Bu yazida ASP ile gelen #include emrinden bahsedecegiz. Bu islevi nasil kullanabiliriz... felan diye biraz düsünecegiz. Sonra normal insanlar gibi "uyku" moduna girecegiz.

Çünkü biz derslerine "late" kalan talebelerden degilek.. OK

Evet simdi ASP nin size tanidigi bir firsati soyeleyecegim: "Dosya Ekleme".. Sasirdiniz degil mi? ASP sayfalariniza, hariçten dosyalar (tabi fotosop dosyasi neyim anlamadiniz degil mi) eklemenize olanak veriyor. Bunu yeni duymadik, çünkü bir çok program gelistirme ortaminda bu islev mevcut. Biz ise ASP deki dosya ekleme yonergesinin kullanimini ve kazandirdiklarini bilmek durumundayiz. Ozellikle çok kod yazacagimiz büyük islerimizde "inkludasyon" lar bizi çok rahatlatacak.

Dosya ekleme nedir? Once buna açiklik getirelim. Diyelim bir site yaptik. Gelinlik kizlar gibi oldu. Surdan burdan "animated gif" ler çikiyor, maus imleci degisiyor, sag tusa basinca küfrediyor, yazilar neyim uçarak geliyor, reklama tiklamayinca girilmiyor.. Nadide.. Yeme de yaninda yat. Bu ozenle hazirladigimiz "masallah" li sitemizin haliyle birden çok sayfasi var. Ve biz istiyoruz ki bu kadar kargasa arasinda en altta ufak bir yazi yazsin:

"web dinayz Abuzer team (c) çalmayin, yamulturum!".

Bu yazi her sayfada olsun istiyoruz. Normal HTML sartlarinda her sayfanin altina bunu teker teker yazmamiz gerekiyor. Ancak yazinin bir tarafini degistirmek gerekirse, (ornegin "dinayz" sozcügünün yanlis oldugunu gordük, onu "dizayn" yapacaz) ne olcak?

250 tane html dosyasini edit mi edecez? Gidin Allah askina :) ASP biliyoruz ya artik. Çozüm ne peki: Alta yazacagimiz metni ayri bir dosyada tutacagiz elbet. Ve bu dosyayi sayfalarimiza "include" (dahil) edecegiz. Sonra ne olacak? Yapacagimiz degisiklik için sadece tek dosyayi muhatap alacagiz.

Verdigim bu ornek çok banaldi biliyorum. Yani gece saat 02:33 olmus... Benim de hosuma gitmedi bu saatte.. Bu isin en güzel ornegi sitelerde navigasyonu (dolasimi) saglayan menulerdir. Kabul edersiniz ki tüm sayfalarimizda, diger bolümlere linklerin oldugu standart bir menu olsun isteriz. Boylece yolcu kardesin kafasi karismaz ve aradigina daha kolay ulasir. Düsünceli webmimar olmak ne mutlu! Yoksa ne olurdu halimiz? Menulere donelim. Tüm sayfalarda gorünmesini istedigimiz menuyu ayri bir dosyaya koyariz.

Gerisini biliyonuz iste. Ancak koydugunuz düz metin dosyasinin uzantisinda dikkat edecegimiz bir husus var. Bu ayri dosyayi istediginiz uzantiyla kaydedebilirsiniz. Genelde "include" den gelen .inc uzantisi kullanilir. Ancak bu ve diger uzantilarla kaydettiginiz dosyalarda ASP kodlariniz varsa "elâleme" açik olacaktir.

Farzedin ki database sifrenizi bir harici dosyada sakliyorsunuz. Sifrelerin yanina ne sifreleri oldugunu da çekinmeden yazmissiniz diyelim.

<%

' data.mdb username ve passwordu

dbuser = "sana\_olan\_duygulari\_bir\_bilebilsem" dbpass = "belki\_severdin"

%>

Dosyanin adini da sifre.inc yaptiniz. Internet te akilli adam çok. Bu akillilardan biri sizin sitenize ugradi bir gün... Kardesimiz sifre.inc yazarsa ne olcak. Server "inc" uzantisinin üzerinde durma gereksinimi duymayacagindan içindeki "çok gizli" seylere bakmadan yollayacaktir. Browser dostumuz büyük bir hevesle download penceresi çikaracaktir. Ve nihayetinde yukaridaki kodlariniz çarsaf gibi serilecektir. Tüh. Ancak

dosya adi sifre.inc yerine sifre.asp olsaydi server'e ASP ogretildigi için, .asp uzantili dosyanin ASP kodlarini icra edecek, tarayiciya bos bir HTML gonderecekti. Bir noktayi atlamayalim. Include edeceginiz dosyalari ayri bir dizinde toplayip bu dizinin read iznini kaldirirsaniz etkili bir onlem almis olursunuz. Ancak server, elinizin altindaysa yapabilirsiniz bunu :)

Sanirim bu nüansi anlatabildim. Eger harici dosyalarda gizli kalmasini istediginiz ASP kodlari saklayacaksaniz uzantilarini .asp yapmaniz salik verilir. Ancak bunun disindaki harici dosyalarinizi .asp uzantisiyla kaydetmek gereksizdir. Serveri lüzumsuz yere mesgul etmekten baska bir ise yaramaz. Çünkü

.asp uzantili bir dosyanin içinde ASP kodu olmasa dahi server bu dosyaya ozel alaka gosterir. (Not: IIS 5 te bu konuyla ilgili bir yenilik var galiba. Su anda tam hatirlayamadim.)

Menumuzu olusturalim hadi. Dosyamizin adi: menu.inc (Burada ASP kodu kullanamayacagiz o yüzden .inc yaptik)

Kod1. menu.inc

<!−− Menu baslar −−>

<a href="mesaj.html">Cep telefonuna mesaj</a><br>

<a href="mesaj.html">Türksel e mesaj</a><br>

<a href="mesaj.html">Telsim e de mesaj</a><br>

<a href="mesaj.html">Orta Asya da hava durumu</a><br>

<a href="mesaj.html">temel fikralari, fikralar, dursun</a><br>

<a href="mesaj.html">!! eriskin bolümü !! (bebeler, girmeyin ulan)</a><br>

<!−− Menu biter −−>

Yukaridaki nadide menümüzü içeren menu.inc i simdi herhangi bir sayfamiza (belirledigimiz yerde) dahil edelim.

Kod2. test.asp

<!doctype html public "−//W3C//DTD HTML 4.0 Transitional//EN">

<html>

<head>

<title> Test Sayfamiz </title>

<meta http−equiv="Content−Type" content="text/html; charset=windows−1254">

</head>

<body bgcolor="#FFFFFF">

<p>hede hodo | abuzerin yeri | hos geldiniz<br> asagida menu olacak<br>

<!−− #include file ="menu.inc" −−>

Yukarida menu var.

</body>

</html>

Yukarida kaynak kodu verilen test.asp yi browserinizde çagirin bakalim. Sonuç:

Kod3. Yorumlanan test.asp

<!doctype html public "−//W3C//DTD HTML 4.0 Transitional//EN">

<html>

<head>

<title> Test Sayfamiz </title>

<meta http−equiv="Content−Type" content="text/html; charset=windows−1254">

</head>

<body bgcolor="#FFFFFF">

<p>hede hodo | abuzerin yeri | hos geldiniz<br> asagida menu olacak<br>

<!−− Menu baslar −−>

<a href="mesaj.html">Cep telefonuna mesaj</a><br>

<a href="mesaj.html">Türksel e mesaj</a><br>

<a href="mesaj.html">Telsim e de mesaj</a><br>

<a href="mesaj.html">Orta Asya da hava durumu</a><br>

<a href="mesaj.html">temel fikralari, fikralar, dursun</a><br>

<a href="mesaj.html">!! eriskin bolümü !! (bebeler, girmeyin ulan)</a><br>

<!−− Menu biter −−>

Yukarida menu var.

</body>

</html>

Nasil hos degil mi? Ancak bu #include emrinin çok basit bir uygulamasidir. Bu emri daha etkin sekilde kullanma isini de konusacagiz ancak once su #include lafini biraz irdeleyelim...

## INCLUDE ILE DOSYA EKLEME

ASP nin Include emri sunucu tarafinda dosya eklemeye olanak verir. Eklenen dosya, eklendigi dosya ile

sunucu tarafinda birlestirildikten sonra icra edilir. Sunucu tarafinda dosya eklemek için asagidaki soz dizimi kullanilir:

<!−− #include virtual | file ="dosya\_adi" −−>

\* Virtual ve File ekleyeceginiz dosyanin yol tipini belirtir. Sanal dizinlerle baslayan dosya yolu için virtual'i kullanin.

[Ornegin http://localhost/dosyalar/menu.inc](http://localhost/dosyalar/menu.inc) dosyasini su sekilde ayri bir sayfaya dahil edebiliriz:

<!−− #include virtual ="/dosyalar/menu.inc" −−>

Eger relative (izafi − goreli) bir yol belirtecekseniz file'yi kullanin. File kullandiginizda path, dosya eklediginiz sayfaya goredir.

Mesela: <!−− #include file ="../dosyalar/menu.inc" −−>

* ASP sayfaniza ayni dosyayi birden fazla eklemeniz mümkündür. Ornegin onceki sayfadaki basit ornekte menu.inc i sayfanin basinda ve sonunda ayri ayri ekleyip arada kalan yere de diger içerigi koyabilirdik.

Ancak soyle bir "cinslik" hataya neden olur: bir.asp de, iki.asp yi eklediniz, ancak iki.asp de bir.asp yi içeriyor. Ne derler buna.. Kimin eli kimin cebinde..

* Bir ayrinti da su: ASP ekleme emri verdiginiz dosyalari sayfaniza, diger komutlari icra etmeden once ekler. Bu yüzden

<% dosya\_adi = "menu.inc" %>

<!−− #include file ="<%= dosya\_adi %>" −−>

gibi bir ifade "halt" edecektir. Çünkü ASP sayfaniz islenirken once bu include emrine bakiyor ve dosya adi yerinde <%= dosya\_adi %> gibi bir ifade gorüyor. Haliyle bunu anlamlandiramiyor zavalli. Ve diyor ki: "Active Server Sayfalarý error 'ASP 0126' Ýçerik dosyasý bulunamadý Vavavava".

* Include emirleri ASP, Script, Object taglerinden arda kalan topsuz alanda (yani serbestçe html tagleri yazdigimiz alanlar) kullanilmalidir. Yoksa yine halt eder sayfamiz:

Ornek:

<% ' hatali kullanim ornegidir If menu\_olsun = true Then

<!−− #include file ="../dosyalar/menu.inc" −−> End if

%>

Yukaridaki ifade asagidaki gibi yazilsaydi halt etmeyecekti:

<% If menu\_olsun = true Then %>

<!−− #include file ="../dosyalar/menu.inc" −−>

<% End if %>

* Evet, dosya ekleme ayrintilarini bu sekilde inceledikten sonra biraz farkli bir ornek yapmayi deneyelim. Vira...

## BIR MENU ORNEGI

Asagidaki kodlari inceleyin. ASP ye doyum olmaz. Seyreyle gonlüm!

Kod4. menu.inc

<% Sub menuler(secim,ayrac) %>

<%

baslik = Array("Giris","Ozgeçmisim","Asklarim","Linklerim","Resimlerim") url = Array("giris.asp","ozgemis.asp","ask.asp","link.asp","resim.asp") For i=0 To Ubound(baslik) − 1 %>

<% If secim=i Then %><b><%= baslik(i) %></b><% Else %>

<a href="<%= url(i)%>"><%= baslik(i) %></a><% End If %>

<%= ayrac %>

<% Next %>

<% End Sub %>

Kod5. giris.asp

<!−−#INCLUDE FILE="menu.inc"−−>

<html>

<body bgcolor="#FFFFFF">

<h1>MENU DENEME</h1>

<% menuler 0," | " %>

</body>

</html>

Kod6. ask.asp

<!−−#INCLUDE FILE="menu.inc"−−>

<html>

<body bgcolor="#FFFFFF">

<h1>MENU DENEME</h1>

<% menuler 2," | " %>

</body>

</html>

# SUB'INA KADAR ASP

## NEDEN ALT PROGRAM?

Yapisal programlamada bir uygulamaya, onu belli kisimlara bolerek yaklasmak esastir. Bu kisimlara alt program denir. Alt programlar, ana programdan çagrilirlar. Belli bir isi yaparlar, deger isterler veya istemezler. Ana programa (veya baska bir alt programa) deger(ler) getirirler veya getirmezler.

Bu teorik bir geyikti. Biraz gevseyelim. Alt programlar, yani VBScript dilinde konusursak Sub rutinler belli islerimizi yükledigimiz script parçalarimizdir. Karsimiza bir is çikti: Ornegin bir databasenin yonetimi. Ne gibi operasyonlar olabilir uygulamamizda? Silme, güncelleme, ekleme, arama, listeleme, vs.. Bunlarin hepsi, temelde ayni database üzerinde bile gerçeklesse birbirinden bagimsiz islerdir. Iste programci, yani ASPci hayata yapisal dillerde top kosturan arkadaslari gibi baktigindan bu olaya da bol, parçala yok et taktigiyle yaklasacaktir. Bu bagimsiz isleri ayri ayri sub rutinlerde çozecek (çozüm kümesini bulacak), sonra da genel çozümü bulmak için bu alt çozümlerin dogrusal bir bagintisini yazacaktir. Yani [**BCR−Online.com**](http://www.bcr-online.com/)u hazirlayan Burak Erdem'in dedigi gibi, bu alt seviyede is bitiren elemanlar birlesip Voltran'i olusturacaklar. Ve büyük güç... Program tamam olacak.

Sub larin bize ne faydasi olacak? Dostlar, bu sublar kolumuz kanadimiz olacak bizim. Çünkü sub larin − en onemli − ozelliklerinden birisi tekrar tekrar çagrilabilir olmalaridir. Yani belli kod gruplarini birden fazla kullanacagimizi fark edersek bunlari bir sub ifadesi içinde tanimlayip gerektiginde ana program − veya baska bir alt program − içerisinden çagirmak akillicadir.

Alt programlarin bir takim faziletlerini saymak her programlama muhabbetinde adet haline geldiginden zikredelim.

1. Belli bir islemi yapan program parçasinin birden çok yerde degisik veriler için tekrar tekrar yazilmasini onler.
2. Düsük düzeydeki ayrintilari program mantiginin ana akisindan uzaklastirip ayri birimlere yerlestirerek okurlugu artirir. (bu onemli!)
3. Program yazma isinin birden fazla programci tarafindan paylasilmasina olanak saglar.

Simdi boylesi ehemmiyet arzeden Sub in soz dizimine bakalim:

Kod1. Sub blogu soz dizimi

<%

Sub prosedürün\_ismi (arguman1, arguman2, .. , argumanN) ' kodlar, kodlar, kodlar...

End Sub

%>

Evet bu kullanimin basitligin bakinca “Ulan bizi buraya bunun için mi çagirdin?” dediginizi duyar gibiyim. Bilin ki bu Sub i incelemeden gitmek yok. Oturun yerinize bakalim.

Simdi bir konuda anlasalim. Bu sub lar çagrildiklari yere deger getirirler veya getirmezler. Bu tamamen size baglidir. Sub lar degiskenlerle muhatap olmayan a−sosyal tipte de yazilabilir. Aha ornek:

Kod2. Degiskenlerle ilgilenmeyen Sub

<%

Sub yildirim\_gursesin\_anisina()

Response.write "leylaklar dokülür, güller aglasin" End Sub

%>

Gorüldügü gibi burada sub in sadece adi belirtilmis. Arguman listesinde bir sey yok (burayi da ayrintilayacagiz bekleyin). Sub in yaptigi tek marifet merhum Yildirim Gürses ustanin en güzel bestelerinden birini mirildanmak! Tabi bu is çagrildiginda gerçeklestirecek. Simdi sadece bekliyor. Peki yazdigimiz sub lari nasil çagiriyoruz? Soz dizimi:

Kod3. Bir Sub prosedürün isletilmesi (sozdizimi)

<%

' tercih 1

Call prosedur\_ismi ( arguman1, arguman2, .. , argumanN ) ' tercih 2

prosedur\_ismi arguman1, arguman2, .. , argumanN

%>

Herhangi bir sub rutini iki farkli sekilde çagirabiliriz. Ben bunlardan ilkini tercih ediyorum genelde. Zira parametreleri parantez içerisine alip gonderebiliyorsunuz. Hem de basinda "call" ifadesi oldugundan kolayca ayirt edilebiliyor. Argumanlarin iki veya daha fazla oldugu durumda call siz olan ikinci yontemi kullanayim dediginizde parantez kullanamiyorsunuz. VBScript yanasmiyor. O yüzden parantezleri kullanmayi adet haline getirmis biriyseniz siz de birinci çagirma metodunu tercih edin.

Bu sub yordami çagirma isini istediginiz yerde yapabilirsiniz. Ister ana programdan çagirin, ister ayri bir sub içerisinden çagirin. Ornegin yukarida Kod1 de ornek olarak verdigimiz sub'i çagiran ifadeyi yazalim.

Kod4. Degiskenlerle ilgilenmeyen kod1 deki sub'in çagrilmasi

<%

' baska isler...

Call yildirim\_gursesin\_anisina() ' veya

yildirim\_gursesin\_anisina

%>

Dikkat edilirse yildirim\_gursesin\_anisina() isimli prosedürümüzün arguman listesi bos. Boyle parametre istemeyen prosedürleri tanimlarken (ve dahi çagirirken) sonuna () isareti koymayabiliriz. Ancak isin raconu boyleymis. Siradan bir degisken adlari olmadigi için sonuna koyulan parantezler, âleme karsi fiyakalarini teyit ediyormus. Sira geldi parametre agrilarina... müteakip page...

## KIMIN DEGISKENI, KIMIN DEGISMEZKENI

Sub rutinleri anlamak demek sub rutinlerin çagrildigi yerdeki degiskenlerle olan iliskisini, muhabbetini anlamak demektir bir bakima. Oncelikle su teoremi verelim:

"Bir sub prosedür, ana programdaki bütün degiskenlere direkt müdahale edebilir."

Bu teorem âlemde Euler'in Evrensel Prosedür Yerine Koyma Teoremi olarak bilinir. Oldükten sonra, yastiginin altinda sakladigi günlük seklindeki ajandasinda bulunmustur. (Aslinda bundan yillar once bu teorem Anadolu'da halk arasinda bilinmekteydi. Ancak Batili kaynaklar her zaman yaptiklari gibi bizden bisikletimizi çaldilar, baska renge boyadilar ve mahallede gozümüzün onünde bizim bisikletimizle bize hava yaptilar. Aglamak istiyorum.)

Neyse biz ASP'ye donelim. Yukaridaki teoremin ispatini verelim de gonüller hos kalsin.

Kod5. "Bir sub, ana programdaki degiskenlere müdahale eder" in ispati

<%

* hadi kurban diye bir degisken tanimlayalim. Hiç adetim deel ama

Dim kurban

* ve degiskenimizin bir degeri olsun. kurban = 5

response.write "kurban: " & kurban & "<br>"

* simdi prosedürümüzü isletelim Call kral\_ciplak()
* ve degiskeni tekrar yazdir

response.write "kurban: " & kurban & "<br>"

* Prosedürümüzü de yazalim... Sub kral\_ciplak()

' kurban adli degiskene darbe yapalim kurban = 15

End Sub

%>

Saydadaki Çikti:

kurban: 5

kurban: 15

Prosedürümüzün ana programin neresinde tanimlandiginin bir onemi yok. Burada ben en sona kral\_ciplak() adini verdigim prosedürü yazdim. Ve yukarisinda, yani ana program akisinda once kurban adli degiskenimize 5 degerini atiyoruz. Ana programda kurban adli bir degisken var ve de degeri 5; bunu ekrana yazdirdik. Sonra call kral\_ciplak() diyerek prosedürümüzü isletiyoruz. Programin isleyisi prosedür içine atliyor. Kral\_ciplak() subinda kurban adli degiskene 15 degerini verdik. Tahmin ettiginiz gibi ayni anda ana programdaki kurban adli degiskenin de içerigi degisti. Kurban'i tekrar yazdirirsak degismis sonucu gorecegiz. Bu demek oluyor ki, bir sub yordam, ana programda çagrildigi yere kadar kullanilan tüm degiskenlere müdahale edebiliyor.

Çagrildigi yerden sonrakileri zaten bilmiyor. Ancak bir sub, baska bir sub in içinden çagrildiginda, onun içindeki degiskenlere müdahale edemiyor. Asagidaki ornegi deneyin, sonucu temasa eyleyin:

Kod6. Sub in bir diger sub dan çagrilmasi

<%

Call bir() ' ~~~~~~ Sub bir() a = 3

Response.Write "a: " & a & "<br>" Call iki()

Response.Write "a: " & a & "<br>"

End Sub ' ~~~~~~ Sub iki() a = 9 End Sub

%>

Sayfadaki Çikti:

a: 3

a: 3

Ikinci bir teorem verelim: "Ana program, bir sub içinde degerine müdahale edilen degiskeni onu çagirmadan

once kullanmamissa, o degiskenin sub tarafindan degistirilen degerini de bilmez." Ispat:

Kod7. "Sub in ozge degiskenini ana program tanimaz..." in ispati

<%

Call afsin\_termik\_santrali() Response.Write "k: " & k & "<br>" ' ~~~~

Sub afsin\_termik\_santrali() k = 25

Response.Write "k: " & k & "<br>" End Sub

%>

Saydadaki Çikti:

k: 25

k:

Ne oldu? Sub içerisinde 25 degeri atadigimiz k degiskenini ana programda yazdiralim dedik bize bos bir deger (null) verdi. Bu demek oluyor ki, bir sub dan deger getirmenin baska bir yolu olmali. Yani olmasi gerekir, salak miyiz biz VBScript ogrenecegiz yoksa? Bize bu lüksü saglamalari gerekir.

Isterseniz degiskenlerle olan soylesimize "müteakip page" ye "click" eyleyerek devam edelim.

## SUB RUTIN VE PARAMETRELER

Onceden dedigimiz gibi bir sub rutine, ana programla bilgi (veri) alisverisi yapmadan kendi halinde "mütedeyyin" ve "münferit" olarak is düsüyorsa adindan sonra parametre listesine bir sey yazmiyorduk. Ancak artik bir sub rutinle daha profesyonel sekilde iletisim kurmanin vakti gelmedi mi? Evet, geldi..

Sub rutinin ana programdaki − tabi çagrildigi yere kadar olan − bütün degiskenlerin degerlerini kullanabildigi, hatta bu degiskenlerin degerlerini degistirebildigi gerçegine hep beraber sahit olmustuk. (euler prensibi) Ancak bu sakat bir durumdur. Istenmeyen sonuçlar dogurabilir. Çünkü programlari yazan yine insanlardir. Ana programda kullandiginiz bir degisken adini, bir sub rutin içerisinde de rasgele kullanirsaniz ve de degerini degistirirseniz ana programdaki degisken de degisecek ve belki ileriki islemlerde onemli yanlisliklara sebebiyet verecektir. Bunu kokten çozmenin, yani programin sagligini dalginliga kurban etmeyi onlemenin yolu degiskenleri Dim ile deklare edip sayfalarin basina da üsenmeyip <% OPTION EXPLICIT

%> yazarak bir degiskenin script içerisinde birden fazla deklare edilmesinin onüne geçmektir. Bu etkili

çozümdür. Artik seçim size kalmis: ben dikkatliyim, iyi araba sürerim.. felan diyorsaniz sozüm yok. Ama yanlis yapabilme ihtimali iliklerinizde soguk rüzgarlar estiriyorsa ( ho! Bu ne demek? ) belli metodu kullanabilirsiniz. Ama sunu belirtmeden geçmek istemiyorum: Ne kadar dikkatli olsaniz bile uzun kodlarda kafaniz allak bullak oluyor. Tecrübeyle sabittir vesselam.

Simdi gerçege yaklasalim. Bir sub rutinin ana programla deger alisverisi yapabilmesi için sub..end sub blogundaki isim tanimlamasindan hemen sonra parantez içinde kullanacagimiz degiskenleri yaziyoruz. Sub sub\_ismi ( degisken1, degisken2 ) gibisinden. Biraz teori daha. Prosedür (sub) ile çagrildigi yer arasindaki deger iletisimi tek yonlü veya iki yonlü olabilir. Bu yüzden parantez içindeki parametreleri iki gruba ayrilir:

1. formal parametre, 2. degisken parametre. . Bir prosedür çagrildigi yerden bir degiskenin ya degerini alir (value), ya da kaynagini alir (reference). Degerini alirsa – ki bunlar formal parametrelerdir – bu degiskenin içerigini degistiremez. Yani onceki sayfada Euler teoreminin ispatinda oldugu gibi ana programdaki degiskene müdahale edemez. Ancak parametre olarak o degiskenin kaynagini alirsa – bunlar da degisken parametrelerdir – o degiskenin hem içerigini kullanir, hem de içerigine müdahale eder. Bir sub tanimlarken hangi parametrenin formal parametre, hangisinin degisken parametre olacagini onüne yazdigimiz bir extra deyimle belirliyoruz. Asagidaki ornek dediklerimizi netlestirecek:

Kod8. Parametre cinsleri

<%

KazanToAnkara = "40 dk." KhamamToAnkara = "60 dk." AyasToAnkara = "unuttum" GolbasiToAnkara = "10 dk."

' ~~~~~~~~~~~~

Response.Write "Sub in icrasindan once...<br>"

Response.Write "KazanToAnkara: " & KazanToAnkara & "<br>" Response.Write "KhamamToAnkara: " & KhamamToAnkara & "<br>"

Response.Write "AyasToAnkara: " & AyasToAnkara & "<br>" Response.Write "GolbasiToAnkara: " & GolbasiToAnkara & "<br>"

Response.Write "======<br>" ' ~~~~~~~~~~~~

Call Ankara ( KazanToAnkara, KhamamToAnkara, AyasToAnkara, GolbasiToAnkara )

' ~~~~~~~~~~~~

Response.Write "Sub in icrasindan sonra ana programdan gorünen manzara...<br>"

Response.Write "KazanToAnkara: " & KazanToAnkara & "<br>" Response.Write "KhamamToAnkara: " & KhamamToAnkara & "<br>"

Response.Write "AyasToAnkara: " & AyasToAnkara & "<br>" Response.Write "GolbasiToAnkara: " & GolbasiToAnkara & "<br>"

' ~~~~~~~~~~~~

Sub Ankara ( kazan, byref khamam, byval ayas, byval golbasi) kazan = "yazin 100 bin TL idi"

khamam = "tam 1 milyon, yarim yani ogrenci 500 idi"

ayas = "valla bilmiyom iste ne diyim.."

golbasi = "belediye otobüsü gidiyo ya. ondan tam 300, yarim 200 bin"

Response.Write "Bir takim atamalardan sonra sub içinden gorünen manzara...<br>"

Response.Write "Kazan: " & kazan & "<br>" Response.Write "Khamam: " & khamam & "<br>" Response.Write "Ayas: " & ayas & "<br>" Response.Write "Golbasi: " & golbasi & "<br>" Response.Write "======<br>"

End Sub

%>

Sayfadaki Çikti:

Sub in icrasindan once... KazanToAnkara: 40 dk.

KhamamToAnkara: 60 dk. AyasToAnkara: unuttum GolbasiToAnkara: 10 dk.

======

Bir takim atamalardan sonra sub içinden gorünen manzara... Kazan: yazin 100 bin TL idi

Khamam: tam 1 milyon, yarim yani ogrenci 500 idi Ayas: valla bilmiyom iste ne diyim..

Golbasi: belediye otobüsü gidiyo ya. ondan tam 300, yarim 200 bin

======

Sub in icrasindan sonra ana programdan gorünen manzara... KazanToAnkara: yazin 100 bin TL idi

KhamamToAnkara: tam 1 milyon, yarim yani ogrenci 500 idi AyasToAnkara: unuttum

GolbasiToAnkara: 10 dk.

Yukaridaki ornegin kodlari çok açik. Ancak bazi yerlerin onemine deginelim.Burada üzerinde durulmasi gereken yer, sub isminden sonra parantez içine virgüllerle siraya konulup yazilan parametrelerdir. Ilk parametrenin (kazan) onünde hiçbir ifade yoktur. Onünde nasil bir parametre oldugu belirtilmeyen tüm parametreler varsayilan olarak degisken parametredir. Yani onüne byref yazmakla yazmamak ayni seydir. Ikinci parametre olan khamam in onünde byref var. Kazan ve khamam parametreleri degisken parametrelerdir. Prosedüre bu parametrelerle degiskenlerin hafizadaki adresleri geçirilir (byref). Üçüncü ve dordüncü parametrelerin (ayas, golbasi) onüne ise byval deyimi eklenmistir. Boylelikle prosedür içine bu parametrelerle sadece tek yonlü olarak deger gelmis olacaktir. Bir formal parametre olan degiskene prosedür içinde atadigimiz deger, sadece prosedür içinde geçerli olacaktir. Ana programdaki degisken bu degisiklikten etkilenmeyecektir. Zira prosedür bu degiskenin sadece içerdigi degeri kullanacak sekilde parametre tanimlamasina sahiptir.

Umarim yeterince açik bir sekilde anlatmisimdir bu mevzuyu. Ha bir de: eger prosedürünüz ana programdaki bir degiskenin degerini kullanacaksa, ve bu degiskenin degerinin hatayla da olsa degismesini engellemek için

sub in arguman listesinde onüne byval yazarak sadece deger aktarimina müsaade etmeyi ihmal etmeyin. Prosedürünüz, ana programdaki degiskenleri rasgele kullanmasin. Hangi degiskenlerle iliskideyse, hepsini arguman listesinde belirtmek racon geregidir. Boylelikle hangi prosedür olursa olsun baslik kismina bakarak hangi degiskenlerle ne sekilde is yaptigini anlarsiniz. Isiniz hizlanir.

## EXIT SUB

Son olarak <% exit sub %> deyimini yad ederek bitirelim. Bu deyimi bir sub blogu içinde kullanirsaniz program akisi, bu deyimi gorür gormez o sub blogundan çikar ve bu sub hangi satirdan çagrilmissa o satirdan sonraki kodlari icra etmeye koyulur. Buna ihtiyacmiz olur mu? Evet, olur. Zaten bu deyim, bir sartli sinama blogu içinde kullanilir genellikle. Belli bir kosul gerçeklestiginde veya gerçeklesmediginde sub içerisinde, diger deyimlerin islenmeden çikilmasini saglar. Asagidaki banal ornege bakalim:

Kod9. Sub dan ani çikis ornegi

<%

pay = 5

payda = 2

Call bol ( pay, payda ) pay = 16

payda = 0

Call bol ( pay, payda ) ' ~~~~~~~~

Sub bol ( x , y ) If y=0 Then

Response.Write x & " \ " & y & ": müjde! divijin bay ziro.. bumm.."

Exit Sub End If

bolum = x \ y ' tam bolme kalan = x mod y

Response.Write x & " \ " & y & " bolümü : "

Response.Write "bolüm: " & bolum & " : kalan: " & kalan & "<br>"

End Sub

%>

Sayfadaki Çikti:

5 \ 2 bolümü : bolüm: 2 : kalan: 1

16 \ 0: müjde! divijin bay ziro.. bumm..

Evet, yürüdük... hiç durmadan!

Sub'in hikmetlerine degindik. Bir sonraki yazida beraber olabilmek ümidiyle.:)

*(Bu arada vize sinavlarim umdugum gibi geçmedi. Pointerlerle ilgili arastirma odevim var. Kaynak ariyorum. Gelecek Ramazan bayraminizi kutluyorum. Euler teoremi uydurmaydi, sallayin. Talebelere basarilar. Ve saat: 01 56. )*

# ACTIVEX DLL OLAYI

Hani server tarafindaki bilesenlerden bahsediyoruz. Su bilesen olmali bunu yapmak için filan... ASP sitelerinde bin türlü component gorüyoruz. Hepsi kendi çaplarinda isler yapiyorlar. Kimisi mail gonderiyor, kimisi dosya transfer ediyor, kimisi dinamik gif dosyalari, grafikler üretiyor, vs.. Yani hepsi ASP scriptlerimizde yapamadigimiz komplex isleri halletmemiz için üretilmis, çogunlugu bedelli üçüncü parti sunucu bilesenleri.

Peki biz kendimiz programci kimligimizi on plana alip, bir an için ASPci oldugumuzu unutup bu sunucu bilesenlerinden yazabilir miyiz? Evet dostlar yazabiliriz. Elin adamindaki programlar bizde de oldugu sürece yazabiliriz nitekim.

Peki diyelim yazdik bir bilesen. Ne isimize yarayacak yazdigimiz bu bilesen? Bize getirisi ne olacak?

Oncelikle su temel bilgileri aklimizdan çikarmamaliyiz: ASP bir sunucu tarafli script teknolojisidir. Yani ASP ler derlenmeden, düz yazi dosyalarinda tutulur. Ve istemci tarafindan çagrildiginda; ASP den haberi olan bir Web Server yazilimi ile script kodlari islenir ve sonucu isteyen adama yollanir. Ama bir Activex DLL dedigimiz zaman bunun apayri bir boyutta oldugunu farkederiz. Çünkü aile terbiyesine gore EXE ve DLL gibi dosyalar, derlenmis dosyalardir. Yani "program" lafindan kastedilen onlardir esasinda. Bir derleyici tarafindan ( MS Visual Basic, Borland Delphi, Visual C++, vs.. ) makine kodlarina çevrilmis programlardir. DLL ler de EXE lerden farkli olarak yürütülmeyen derlenmis kütüphane dosyalaridir.

Peki DLL lerin avantaji nedir? DLL ler bir kere çagrildiktan sonra hafizada kalirlar. Yani oturur ve gitmezler. Bunun için sistem kaynaklarinda bagdas kurup gernesen bir DLL ye tekrar ihtiyaç duyuldugunda olay yerine müdahale etmesi daha kolay olur. Yani buna hiz kazanmak diyorlar. Ne kadar hiz kazandigimizi soyleyemeyecegim maalesef. ...'in her ay "Nasil Test Ettik?" diye sakal tirasi yapan Italyan kilikli adamlarina soyleyelim de bir ara bunu da test etsinler.

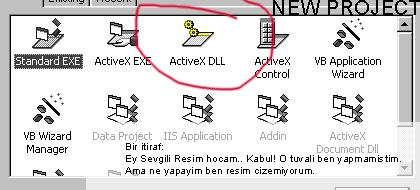
Ve hazirlayacagimiz sunucu DLL lerinin bir yarari daha var. Yazdiginiz kodlar derlenmis oldugundan "sakli" kalacaktir. Bir isi ASP scriptleriyle yaptiniz diyelim. Bilirsiniz ki ASP ler düz yazi dosyalari... Ve dünyada text editorü de bulmak kolay. Yani ozene bezene yazdiginiz kodlarin sakliligindan soz etmeniz mümkün olamiyor. "yaw ASP bir Server−side seysi.. kodlar gorünmez client in makinada.. " diye itiraza ragmen sunucu tarafinda kodlariniz açiktir. Yani bir DLL de is yapmak kodlarinizin mahremiyetini saglayacaktir.

Peki bu Activex DLL leri neylen yapariz, olustururuz? Her sabah ugradiginiz ekmek için bakkaldan yarin bir de Visual Basic benzeri son nesil bir derleyici isteyin. Eger "bende o dediginden yok" diye diretiyorsa israr etmeden bakkaldan çikin. Muhtemelen bir gün o da satmaya baslayacaktir. Neyse, elinize bir tane derleyici geçirmeniz lazim. Delphi, Visual Basic, Visual C++, Borland C++, felan filan.. Activex DLL yaratabilen seylerden bulun bir tane iste :) Biz burda ornek olarak ( çok kullanildigi için ) Visual Basic ile yaratacagiz DLL mizi.

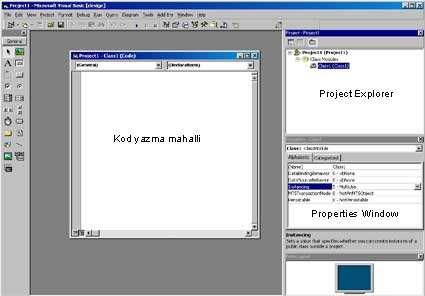
Peki DLL mizi olusturduk diyelim. Bunu kendi bilgisayarimizda sinayabildik ama sitemizi sundurdugumuz hostes sunucumuzda nasil çalistiracagiz? Valla orasini benimle degil de sunucunuzun serifiyle temasa geçerek halledersiniz. Biz burda bu DLL isini de evcillestirmeye çalisacagiz. Yani sirf zevk için. Bana eglence degil, reel yaklasimlar lazim derseniz yaziyi okumaniza gerek yok. Ama eger bir Activex DLL olusturup, kahve yudumlayaraktan safa eylemeyi yeglerseniz buyurun. Fazladan bilgi goz çikarmaz hem degil mi dostlar?

## BASLIYORUZ...

Evet, baslamak için Microsoft Visual Basic 5 veya yukarisindaki sürümü Baslat > Programlar > felan filan diyerek açin (saka yaw). Programi açinca artik her programin vazgeçilmezi olan çok seçenekli New/Open Project ekrani selamlayacak sizi. Sanki programi her zaman yeni bir sey yazmak için açiyoruz! Ya ben yeni bir program yapmak veya yaptigimi düzenlemek yerine programin about undaki yumurtalari seyredeceksem? Adami uyuz ediyorlar dostlar:) (Iyi ki altinda 'Don't Show bu gibi muhabbetleri' seçenegi var)



Karsimiza INTEGRATED DEVELOPMENT ENVIRONMENT diye adlandirilan bir ortam çikicak ( Burasi ScottJr ve Stalker içindi..). Bu "alisilmadik" ortamda bize yardimci olacak sagdan soldan boy gosteren edevati tanimaniz için size süre veriyorum :) Isiniz bitince müteakip sayfada devam edek.



## PROJE...

Once projemize bir ad verelim. Bunun için once sag tarafta üstte duran Project baslikli pencerede Project1 i seçiyoruz. Ve bu anda altta properties penceresinde Project1'e ait ozelliklerin listesi gorüntüleniyor. Burada sadece tek ozellik olarak gozümüze çarpan Name ozelligini hemen "evcilasp" diye degistiriyoruz. Lütfen siz de "evcilasp" diye degistirin. Illa kendi sitenizin adini yazacak degilsiniz ya. Yarin da sizin sitenin adini yazarsiniz :)

Project explorer e donelim. Yani oraya odaklanalim. Vusal Basic bizim için hazir bir Class Module koymus oraya. Adina da class1 demis. Biz bu ogeye tiklayip properties penceresinden name ozelligini ( artik alistik buna degil mi? ) 'fonksiyonlar' yapalim.

Evet... Bu kadar merasimden sonra artik kod yazmaya baslayalim ha? Eger bir kod penceresi gorünür degilse, Project Explorer deki fonksiyonlar modülü üzerine sag tiklayip View Code deyiniz. Basliginda 'evcilasp−fonksiyonlar (code)' yazan bir pencere peyda olacak. Kod mahalline asagidaki kodlari yaziniz dostlar.

Public Sub PanoyaYaz(str As String) Clipboard.SetText (str)

End Sub

Public Sub PanodanYaz(response) response.write Clipboard.GetText End Sub

Evet, haklisiniz, tek satirlik çok basit prosedürler yazdik. Ama burda gormek istedigimiz olayi orneklemesi açisindan boyle oldu. Simdi dikkat ederseniz ( ben VBasic bilmem ama anliyorum ) VBasic bilmeyen ASP cilerin anlayacagi basitlikte bir is var burda. PanoyaYaz isimli ilk sub imiz disaridan bir stringle çagriliyor, ve subimiz sunucu sistemin Clipboard'ina ( pano ) gonderilen stringi koyuyor. Ikinci subimiz ise disaridan bir nesne istiyor. Bu da ASP object modelinin iyi tanidigimiz delikanliliarindan "Response". Burada response yerine baska bir sey de yazabiliriz. Ama bosverin. Gül dalinda güzel. Subimizda ise panodan aldigimiz metni response.write metoduna gonderiyoruz. Ve panodaki metnin asp sayfamiza yazilmasini sagliyoruz.

Simdi bu projemizi kaydedelim de bir yerde dursun. File menusunden her zaman yaptiginiz kaydetme islemlerine bezner bir islemle bu kaynak dosyalarimizi saklayalim. Proje ve sinif isimleriyle sakladigimiz kaynak dosyalar ayni ismi tasimayabilir. Ama ben ugrasmadan ayni isimlerle kaydettim. Size bu konuda ozgürlük taniyorum. Istediginiz isimle kayit olanaginiz var.

Sonunda can alici noktaya geldik. Hazirladigimiz bu projenin neticesi bir dll olmali degil mi? Sira bu dll yi olusturmaya geldi. Hemen vakit kaybedip de bu eglenceyi sekteye ugratmadan File menusunde "Make evcilasp.dll ?" yazini bulalim. Ne güzel degil mi? Adam size dll ni yapayim mi diye soruyor.. Çok

yardimsever insanlar bu programcilar :) Biz de uzatilan bu dali tutuyoruz. Karsimiza bir pencere çikiyor. Dll mizi ne adla kaydedecegimizi sordugu her halinden belli. Kendisi de evcilasp.dll diye tercihini sunmus. Biz ona uymadan adini ilkdeneme.dll yapalim orayi. Hani projemize evcilasp ismini vermistik. Ikisinin de farkli olabilirligini gostermis olalim boylece. Sonra OK tusuyla bir devrim yasamaya basliyoruz. Visual Basic sizin için projenizi derlemeye basliyor.. Bir DLL dogacak.. Sabir.. Taslar yerine oturdu.. Sonra bu DLL yi gidip sizin yerinize Windows Registry veritabanina da kaydediyor.

Artik bir sunucu bileseniniz var.. Nasil bir duygu?

Peki her sey buraya kadar miydi diye soracak olursaniz Hayir! Sira geldi bu bilesenimizi (yani bize ait olan bileseni) ve sagladigi metodlari bir asp scriptinde keyifle kullanmaya. Hemen sevdigimiz veya hayran oldugumuz bir text editorü açiyoruz. Ve asagidaki kodlari kopyeliyoruz:

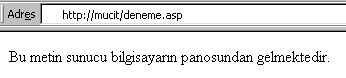
DENEME.ASP KOD MANZARASI

<%

Set BenimNesne = server.Createobject("evcilasp.fonksiyonlar") BenimNesne.PanoyaYaz("Bu metin sunucu bilgisayarin panosundan gelmektedir.") BenimNesne.PanodanYaz(response)

%>

Ve her server−side scripting teknolojisi bagimlisi gibi hemen kaydettigmiz bu scripti localhost tan çagirmaya kosuyoruz. Ve manzarayi temasa eyliyoruz.



Evet, arkaniza yaslanin simdi. "Elin adami yapmis" degil mi dostlar? Bu demek oluyor ki ASP bize sunucuyla sinirsiz etkilesim alani sunuyor. Yukaridaki çok basit uygulama bunun ozeti degil mi? Evet, ASP nin sinirli oz yetenegini sinirsiza donüstüren onun sunucu tarafindaki bilesenlerle is yapabilmesidir dostlar. Ve bu kapi sonuna kadar açiktir. Ve o yüzden dünyada bir sürü üçüncü parti bilesen üretip satan yazilim ekipleri türemis. Tabi Microsoft da geri duymuyor. Kendi amacina yonelik komponentler çikariyor. Eger ASP ile bir seyi yapamadiginizdan sikayetçiyseniz sizin isinizi halledebilecek bir compoenent elbette bir

yerde bekliyordur sizi. (Müsteri bekleyen bir çok sunucu componentlerine ulasmak için [**Aspin**](http://www.aspin.com/)sitesi idealdir.)

"Zeki Müren de bizi gorecek mi?" diye kaygilanmayin. Samimi duygularla iyi isler yapani herkes gorür, herkes duyar..

## DERLERKEN...

Bu isin gicik tarafindan da bahsedelim biraz. Hani dll dosyalarinin bir kez çagrildiktan sonra hafizada kaldigini, geri gitmedigini soyledik. Az once deneme.asp dosyamizi çalistirdigimizda farkinda olmadan ilkdeneme.dll kütüphanesini de sistem bellegine yüklemis olduk. Bu dll yi daha sonraki çagrislarinizda islemler daha hizli yürüyecektir haliyle.

Ancaaaak! Siz bir bilesen gelistiriyorsunuz ve dll yi tekrar derlemeniz gerekiyor. Ornegin bir onceki sayfadaki ornegi biraz kurcalayip tekrar derlemek istediniz. Ve aci manzara.. "Permission denied" denen yürek uçuklatici hata mesajini aliyorsunuz.. Niye? Hani bir vecize vardi hatirladiniz mi? "Windows kullandigi dll yi elletmeeez...". Yani ilkdeneme.dll dosyasi hafizada yüklü ise üstüne yazmaniz mümkün degildir. E.. peki bunun çaresi var mi? Ilkdenem.dll yi hafizafan kovmanin bir yolu var mi?

Var.. Hem de çok radikal!: Windows'un reboot edilmesi... Bu Win95/98−PWS sistemiyle çalisanlar için tek çozüm dostlar. Yani bunun disinda bu dll yi hafizadan kovamiyorlar.

Ama NT gençligi sansli. Zira onlarin bir IIS i var. Internet Information Server size PWS deki baslat−durdur seçeneginden çok daha genis servis yonetimi sunuyor. Hemen IIS yonetim (internet services manager) programinizi açiyorsunuz. Ve agaçta gordügünüz bilgisayar ismine sag tikliyorsunuz. Çikan menuden "Restart IIS..." gibisinden bir yazi var, ona tiklayinca karsiniza çesitli seçenekler içeren bir ekran çikiyor.

Burada Restart Internet Information Services demeniz yeterlidir. IIs yeniden basladiginde dll nizi derleyip tekrar deneyebilirsiniz.

Bunun yerine komut satiri da isinizi gorebilir. Start > Run diyip "cmd" yazin. Açilan kara ekrandaki komut satirina

net stop iisadmin

yazin. Yoneltilen soruyu Y (yes) diye yanitlayin. Size orayi burayi durdurdum diye verilen bilgilerden sonra komut satirina simdi de sunu yazin:

net start w3svc

Simdi dll nizi yeniden derleyebilirsiniz.

## KAYIT...

Bir yerden edindiginiz derlenmis bir dll yi sisteminize kayit etmek için (register) regsvr32.exe programini kullanabilirsiniz. Bu program Windows un system dizininde bulnmakta. Komut satirindan bu dizine gelip, soyle bir komut verebilirsiniz:

regsvr32.exe C:\belgelerim\ilkdeneme.dll

Bu komutla derlediginiz bir dll yi, baska bir bilgisayara kayit edebilirsiniz.

Sayet kayit olunmamis bir nesne ornegi yaratmaya çalisirsaniz su meshur ASP hatasiyla bulusursunuz: Sunucu nesnesi error 'ASP 0177 : 800401f3'

Server.CreateObject Baþarýsýz

/xxx.asp, line 2 800401f3

Evet, bu mevzuya burada ara verelim. Bir gün bu makalenin devamini da yazip veritabani islerini de bilesenlere yükleyip performans kazanmayi isleriz. 0555 hepinize hayirli olsun :)

\*\*\*\* Bu makale ile ilgili dosyalari [**buradan**](http://evcilasp.mucit.gen.tr/evcilasp/ornekler/activexolay.zip)alabilirsiniz.