

ASP NEDİR?

ICINDEKILER

<u>ASP NIN MANA VE EHEMMİYETİ.....</u>	1
<u>GİRİS.....</u>	1
<u>İSTE SIZE ASP.....</u>	1
<u>PERSONAL WEB SERVER NE Kİ?.....</u>	3
<u>PWS YI KURDUN MU ?.....</u>	3
<u>GENEL MANZARA.....</u>	3
<u>GELİSMİŞTEKİ SIRLAR.....</u>	4
<u>SONUÇ.....</u>	4
<u>DEĞİSKENLERİMİ KİMSEYLE DEĞİSMEM.....</u>	6
<u>ASP YI YAZMAYA BAŞLAYALIM.....</u>	6
<u>FORM'UNDA SAYFALAR YAZMAK.....</u>	8
<u>FORMDAN İÇERİK TALEBİ.....</u>	8
<u>SORGU CÜMLECİKLERİ KULLANMAK.....</u>	12
<u>SORGU CÜMLECİĞİ : QueryString.....</u>	12
<u>.....</u>	13
<u>AH VERİTABANI AH!.....</u>	15
<u>ALANLARDA VARSAYILAN DEĞER MESELESİ.....</u>	16
<u>NEYİN VERİSİNİ TABANLAYACAGIZ.....</u>	16
<u>BASLIYORUZ ÖRNEK VERİTABANINA.....</u>	17
<u>ASP İLE VERİTABANI BAĞLANTISI.....</u>	20
<u>BAGLANMAK.....</u>	20
<u>SORU İSARETLERİ.....</u>	22
<u>HIZLI SOL LAFDANLIĞI.....</u>	23
<u>POPÜLER SOL LAFLARI.....</u>	24
<u>HOST'ES MESELESİ.....</u>	26
<u>BEN HOST'ES İSTEREM.....</u>	26
<u>BELES OLSUN HA!.....</u>	26
<u>EWBCITY NASIL BİR VARLIK, ANLATIR MİSİN BANA?.....</u>	27
<u>DOMAINDLX: BİR NEFRET ABİDESİ.....</u>	27
<u>HAYDI SAMBA'YA!.....</u>	28
<u>BİRAZ VBSCRIPT (I).....</u>	30
<u>VBSCRIPT NEDİR?.....</u>	30
<u>1. SCRIPT DİLİ BİLDİRİM SATIRI.....</u>	30
<u>2. HARF DURUMLARI.....</u>	30
<u>3. DEĞİSKEN BİLDİRİMİ.....</u>	31
<u>4. DEĞİSKENLERDE VERİ TÜRLERİ.....</u>	31
<u>VARIANT VE ALT TIPLERİ.....</u>	32
<u>ALT TIPLER ARASINDA DONÜSÜMLER.....</u>	32
<u>CInt.....</u>	32
<u>CStr.....</u>	33

ICINDEKILER

<u>CDate</u>	33
<u>BIRAZ VBSCRIPT (II)</u>	34
<u>DURUMLAR VE DAVRANISLAR</u>	34
<u>VBSCRIPT AKIS KONTROLLERI</u>	34
<u>IF...THEN...ELSE KONTROLÜ</u>	34
<u>SELECT CASE KONTROLÜ</u>	35
<u>DO...LOOP DONGÜ YAPISI</u>	35
<u>FOR...NEXT DONGÜ YAPISI</u>	36
<u>FOR EACH...NEXT DONGÜ YAPISI</u>	37
<u>WHILE...WEND DONGÜ YAPISI</u>	37
<u>SITEDE OTURANLARIN SAYISI</u>	38
<u>EY ORADAKILER! KAÇ KISISINIZ... TEKER TEKER GELIN</u>	38
<u>GLOBAL.ASA</u>	38
<u>SITEDE OTURANLARIN SAYISI</u>	40
<u>FORMDAN VERITABANI ISTIKAMETINE GIDEN KAYITLAR</u>	42
<u>VERITABANINDAKI TABLO VE ALANLARIN DURUMU</u>	42
<u>KAYIT EKLEYEN ASP DOSYASI</u>	43
<u>SORU ISARETLERİ</u>	46
<u>SUB RUTINLER</u>	47
<u>FORMDAN ISTENEN İÇERİK VE DE KONTROLLERİ</u>	47
<u>SIFIR UZUNLUK > NULL</u>	48
<u>VT BAĞLANTISI</u>	48
<u>KAYIT SETİ</u>	48
<u>ARTIK YENİ BİR KAYIT EKLEYELİM HA?</u>	48
<u>VBSCRIPT İLE STRING DÜZENLEME</u>	50
<u>LEN FONKSİYONU</u>	50
<u>UCASE, LCASE FONKSİYONLARI</u>	51
<u>INSTR FONKSİYONU</u>	51
<u>REPLACE FONKSİYONU</u>	51
<u>LEFT, RIGHT FONKSİYONLARI</u>	52
<u>MID FONKSİYONU</u>	52
<u>TRIM, LTRIM, RTRIM FONKSİYONU</u>	53
<u>STRREVERSE FONKSİYONU</u>	53
<u>STRING FONKSİYONU</u>	53
<u>SPLIT FONKSİYONU</u>	53
<u>JOIN FONKSİYONU</u>	54
<u>VERITABANINDAKİ KAYITLARI GÜNCELLEMEK</u>	55
<u>VERITABANINDAKİ KAYITLARI GÜNCELLEMEK</u>	56
<u>KAYIT_YENI.ASP</u>	58
<u>KAYIT_SIL.ASP</u>	59
<u>ASRIN FIYAKASI: MAIL GONDERMEÇ!</u>	61
<u>DERS BASLIYOR, ÖNCE AMAÇ!</u>	62

ICINDEKILER

FORM'UM	62
CDO ILE YOLLA.ASP	63
PERSIST ILE YOLLA.ASP	64
JMAIL ILE YOLLA.ASP	65
BIR ONERI	66
ASP VE KABUKLU YEMISLER.....	69
NERELERDE KULLANILABILIR?	70
KAYITLARI SAYFALAMAK (DB PAGING)	71
KAYITLAR ÇOK FAZLAYSA	71
BIR TAKIM AYRINTILAR	74
HARIÇTEN GAZEL OKUMAK.....	76
INCLUDE ILE DOSYA EKLEME	78
BIR MENU ORNEGI	79
SUB'INA KADAR ASP.....	82
NEDEN ALT PROGRAM?	82
KIMIN DEĞİSKENİ, KIMIN DEĞİSMEZKENİ	84
SUB RUTIN VE PARAMETRELER	86
EXIT SUB	89
ACTIVEX DLL OLAYI.....	91
BASLIYORUZ	92
PROJE	93
DERLERKEN	95
KAYIT	95

ASP NIN MANA VE EHEMMİYETİ

GIRIS

Yetenekler yavas yavas yeryüzüne çikiyor arkadaşlar. Artık internet herkesin, hepimizin oluyor farkında misiniz? Her client(istemci) aynı düzeyde erişim ve gezinim hakkına sahip. Söz ve düşünce özgürlüğü var. Kendinizi sınıf arkadaşınıza bile ifade etmekte güçlük çekerken, karsımıza dev gibi dikilen bu zahiri alemde en kolay şeyin bir “kendimi ifade edeyim” alanı (homepage derler) yaratmak olduğunu görünce sasirmistik.

İnsanlar HTML kullanıyorlardı eski çağlarda. Yapılan webolojik çalışmalar ve derin darbeler, Frig medeniyetinde “hypertext markup language” teriminin bilindiğini, Urartu taraflarında da HTML adlı bir canavarın varlığından şüphe edildiği gerçeklerini ortaya çıkarmıştır.

Meshurluğu eski çağlara kadar dayanan HTML neydi peki? Sadece insanlara yazdıklarımızı, çizdiklerimizi ve beğendığımız site koprülerini yerleştirdiğimiz, envai çeşit editorü bulunan, utanılmasa “Nesneye Yonelik Programlama Dili” gurubuna alınacak bir çeşit kod akaidi. Excel’in içinden çıkan, Word’de tek tustla ciziliveren tablolar HTML’de de var. Form oğeleri koyabiliyorsunuz. Yazı yazabiliyorsunuz (pes yani).

“Bu ask fazla sana”

Askımız evrim geçirecek. İnsanlar daha doğrusu webmimarlar, HTML yi tek taraflı metin ve grafik ogesi olmaktan çıkarmak için çabaladılar. Microsoft’un evliyalari devreye girdiler. NT4 işletim sistemi mimarisine dayanan, Internet Information Server ‘de çalışıp, yeni bir ekmeç kapisi olacak olan ASP yi geliştirdiler.

ASP artık Türkiye’nin yeni yüzü olacak!

Herkes ASP kullanacak, herkes rahatlayacak.

ISTE SIZE ASP

Neden hep ASP diyoruz. Bunu bir açmak gerek. ASP İngilizce “Active Server Pages” tümcesinin kısaltmasıdır. İngilizce bilenler bunun “Aktif Sunucu Sayfaları” manasına geldiğini anlamıştır. Aslında Türkçesi de mantıklı geliyor insana. Sunucu tarafında işlenen sayfalar demek oluyor ASP.

Sunucu ne peki? İnternette bilgi alışverişi bakkalla müşterisi arasındaki alışveristen çok da farklı değil. İstemci (Client) talip olandır; sunucu (Server) arz edendir. Genelde internete bağlı bilgisayarlar (sizinki de muhtemelen bu genellemeye dahildir) istemci konumundadır. Bilgisayarınızın diğer insanlara, yani dünyaya sunduğu, arz ettiği bir meta yoktur. Sunucuların arz ettiği web sitelerini gezersiniz. Kendi web sitenizi oluşturursunuz, ancak kendi sitenize yine istemci olarak girersiniz. Sunan kimdir? (Hypermart, Xoom... ama asla siz değil!)

Bu sunucu–istemci ayrıntısından sonra sunu belirtelim: ASP sunucularda çalışır. Oluşturduğunuz HTML dosyalarını üzerine tıklayarak browserle görüntüleyebiliyordunuz. Çünkü HTML belgeleri sadece browser tarafından yorumlanmakta. Ancak HTML nin içine yeni kodlar yazmaya başlayacağız bundan sonra. Bu kodlar ise oyle babadan kalma üzerine tıklama yöntemiyle çalışamayacak maalesef.

Şimdi gerekleri sıralayalım:

Bilgisayarınızda ASP ile çalışmak istiyorsanız bilgisayarınızda ASP yi destekleyen web server programı yüklü olmalıdır. (yani bilgisayarınız server davranisi gosterebilmelidir.) Soyle güzel bir web–browser gerekli . (mümkünse IE tedarik edin) Metin Editorü veya herhangi bir HTML editorü bulunsun. Şimdi bu maddeleri irdelleyelim :

Birinci maddede bahsedilen server programını sizi düşündürmesin, temin edilmesi çok kolay.

Alternatifleri denenebilir yalnız ben işletim sistemi olarak Win98 ve web server olarak da Microsoft Personal Web Server (PWS) üzerinde çalışmanızı istiyorum. Amacımız ASP öğrenmek ise boylesi yeter de artar bile. Peki nerde bu PWS. Win98 CD niz var değil mi? Orada iste. CD nin ana dizinindeki addons/pws klasöründen setup.exe ile PWS yi bilgisayarınıza kurabilirsiniz. (Tabi bu işlem win98 yükledikten sonra yapılıyor.) Artık program kura kura setup kurdu olduğuzdan size kurulumu anlatmaya gerek yok. Normal kurulum isinizi görecektir.

İkinci madde çok açık. Internet Explorer 4–5 tavsiyemdir. (“Microsoft’un dilinden ancak yine bir Microsoft anlar” mantığı...) Üçüncü maddeye ekleyeceğim bir şey var. Her ne kadar basit bir text editorü ile HTML veya ASP kodları yazılabilir de, güzel ve kullanışlı programlar kullanmak bizim de hakkımız. Ben HTML insasında Allaire Homesite 4 (<http://www.allaire.com>) ve ASP kodu yazarken de EditPlus 2 (<http://www.editplus.com>) kullanıyorum. Homesite HTML hamalligimi en aza indiriyor ama onda ASP yazmak çok zevksiz geliyor bana. EditPlus 2 de bana göre çok güzel ve anlaşılır bir renklendirme var. Bazı yetenekleri de çok kullandığınız kod parçalarını tekrarlamaktan kurtarıyor.

Evet. Artık alet edevat hazır olduğuna göre başlayabiliriz demektir. Unutmayın, ASP ile çalışmak için sadece Personal Web Server yeterli olacaktır. Biz extradan kod yazacağımız editor tavsiye ettik.

Bir sonraki makalede PWS yi inceleyeceğiz.

PERSONAL WEB SERVER NE KI?

PWS YI KURDUN MU ?

Bu makalede Microsoft Windows 98 CD si içinde gelen Personal Web Server'dan bahsedeceğiz. Öncelikle bu programı sisteminize kurmanız gerekiyor. Basit bir kurulumla sahip. Normal kurulum bize yeterli olacaktır. Kurulumu tamamladığınızı varsayıp başlıyoruz.

GENEL MANZARA

Baslat > Programlar > Donatılar > Internet Araçları > Personal Web Server > Personal Web Manager ile veya **C:\WINDOWS\SYSTEM\INETSRV\pws.exe** ile *Personal Web Manager* programını açalım bakalım.

Bu program bizim Web sunucumuzu yönetmekte kullanacağımız "manager" dir diyeceğiz. Neler yapabiliyoruz peki bununla?

Önce programın açıldıktan sonraki manzarası ve endamini görelim. İlgili resim aşağıdadır.

Yukarıdaki ekran görüntüsünde "**dur**" yazan butonun ne işe yaradığını açıklayalım:

Bu buton serverinizin çalışması veya durdurulması durumları arasında size geçiş yaptırıyor. Su anda butonda **dur** yazıyor. Demek ki düğmeye tıklayınca serverin işleyisi duracak; bilgisayarınız sadece client olacak. Butonun üzerindeki metin ise "**baslat**" (bildiğimiz baslat) a dönüşecek. Bu durumda butona tekrar tıklanırsa server aktif hale gelecek. İstemci bilgisayarınız, fukara bir server haline gelecek tekrar. Butonda "dur" yazısı belirecek. Hayat bu şekilde devam edecek.

"**http://mucit**" ifadesi sunucunuzun aktif ana dizinidir. Web sitenizin giriş sayfası bu dizinde olmalıdır. "mucit" benim sunucumun adıdır. Buraya tıklarsanız browser açılır ve **http://mucit** adresini bulmaya çalışır. Sunucunuz aktif ise bulacaktır olagan olarak. Ve giriş sayfasına bakacaktır. Eğer giriş sayfanız varsa (örneğin default.asp) onu açacaktır. Burada "mucit" yerine localhost yazarsanız aynı sonuçla karşılaşırsınız.

"**E:\inetpub**" ifadesi, sunucunuzun aktif dizininin "fiziksel yol" udur. Yani **http://localhost** adresi aslında "e:\inetpub" a işaret etmektedir ve buradaki giriş sayfasını açmaktadır. Web server kurulurken **c:\inetpub\wwwroot** yolunu sizin ana dizininiz yapar. Bunu "gelismis" seçeneğiyle kolayca değiştirebilirsiniz.

GELISMIS'TEKI SIRLAR

1. Sanal Dizinler : Burada gördüğünüz sanal izin ağacıdır. Sanal izinler, bilgisayarınızın fiziksel izinlerinden farklıdır. Adı üstünde sanal izinlerdir. Burada birbirinden çok uzakta bulunan iki fiziksel klasörü, "home" ana dizini altında sanal izin olarak belirleyebilirsiniz. "home" ifadesi sunucunuzun ismine tekabül eder. Benim sunucumun ismi "mucit" olduğuna göre buradaki sanal izinler, browserden http://mucit/sanal_dizin_adi şeklinde yazılarak çağrılabilir. (veya http://localhost/sanal_dizin_adi)

Buradaki sanal izin ağacına istediğiniz kadar sanal izin ekleyebilirsiniz. İsterseniz ekleme işine bir göz atalım.

2. EKLE tusuyla yandaki pencereye ulaşırsınız. Ve buradan yeni bir sanal izin ekleme fırsatına kavuşursunuz. Ben burada örnek olarak bilgisayarımdaki "C:\Belgelerim\cahitzarifoglu" klasörümü, "zarifoglu" adında bir sanal izin olarak atıyorum. Erişim bölümündeki izinlere hiç dokunmuyorum. Bizim için bunlar yeterli. Tamam'a tıklayıp ekle işlemini bitiriyorum. Yeni eklediğimiz klasörün, sanal izin ağacında görünmesi lazım.

Su notu eklemek lazım. Sanal izin olarak atadığınız bir fiziksel klasörün alt klasörleri, otomatik olarak sanal izin olacaktır. Adları mı? Kendi adlarıyla aynı olacaktır. Örneğin yukarıda eklediğimiz klasörün alt dizini olan "C:\Belgelerim\cahitzarifoglu\resimler" klasörü, otomatik olarak kendini sanal izin hissedecektir ve sanal adresi de "http://mucit/zarifoglu/resimler" olacaktır.

Özellikler tusu ile seçtiğiniz sanal dizinin özelliklerini değiştirmek, Kaldır tusuyla da seçili sanal dizini kaldırmak mümkündür.

3. Varsayılan Belgeyi Etkinleştirmek : Varsayılan belge ne demek? Browserden siteniz (<http://mucit/>) veya bir alt dizini (<http://mucit/zarifoglu>) çağrıldığında sunucu o dizinde belirli isme sahip belgeler aramaya başlar. Eğer bulursa onu browser'e gönderir. İste sunucunun otomatik olarak açmak istediği dosyalara PWS, "varsayılan belge" diyor. Yüzyıllardan beri kullanılan "varsayılan belge" lerden en meshurları: "index.html, index.asp, default.asp, default.html, main.asp, main.html" vs... "Varsayılan belgeyi etkinleştir" seçeneği tıklanmış ise sunucu, belirtilen isimdeki dosyaları otomatik açmaya çalışır. Eğer tıklanmamış ise, böyle bir arayış içinde olmayacaktır sunucu. Biz tikiyoruz. Altta da bir metin kutusu gözünüze çarptı değil mi? İste bu metin kutusuna, caninizin istediği, yazmak isteyip de yazamadığınız varsayılan belge türlerini yazabilirsiniz. Tabi kurala uyarak: Örneğin Default.htm; Default.asp; Main.asp; anasayfa.html; babasayfa.html; paramparca.asp gibi bir varsayılan belge silsilesi yazabilirsiniz.

4. Dizinde Göz atma : Bu seçenek tıklanırsa, ziyaretçiler web sitenizin klasörlerinin içeriklerini gözlemleme imkanına kavuşur. Tabi o klasörde varsayılan belgenin yokluğu veya varsayılan belgenin etkin olmadığı durumdan bahsediyoruz. Diğer seçenek ise sunucunun günlük kaydetmesini denetliyor. (lazım olacak yani)

SONUÇ

Nihayet PWS'yi tanıdık. Kabaca PWS'nin bilgisayarınızı bir WWW sunucusu haline getirdiğini tekrarlayalım.

Bizim amacımız ASP öğrenmek ve uygulamak. Tercihimizi bu yüzden ASP den anlayan bir web server programından yana yaptık. ASP Microsoft'un özgün web programlama teknolojisi, PWS ise yine Microsoft'un Win98 için geliştirdiği bir server programı (NT'deki IIS'in bir beden küçüğü).

Artık bilgisayarımız ASP'den anlayan ve web dokümanlarımızı dünyaya sunan minik bir sunucu haline geldi. Bunu gözlerinizle görmek isterseniz...

Internet'e bağlandığınız zaman IP numaranızı tespit edin. Örneğin IP'niz 212.64.197.51 olsun.

Browsere 212.64.197.51 yazdığınız zaman, kendi sunucunuza konuk olursunuz. Ve varsa ana sayfanız gelir.

Bunu o anda internete bağlı tüm insanlar denediğinde aynı sonucu alırlar. Yani sizin sunucunuza konuk olurlar ve ana sayfanıza ulaşırlar. Ne kadar global düşündüğümüzün farkına varın.

Bunu kullanarak, bilgisayarınızda oluşturduğunuz ama henüz web'e yerleştirmediniz bir sitenizi, arkadaşınıza gösterip onun fikrini alabilirsiniz. Veya arkadaşınızın, sizin bilgisayarınızdan bir dosyayı (mesela bir mp3), download etmesini sağlayabilirsiniz.

PWS muhabbeti yeterli sanıyorum.

DEGISKENLERIMI KIMSEYLE DEGISEMEM

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 nesnelarini 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** hazır 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 almayacaktır; gülüp geçecektir. Ote yandan buradaki yorumlar istemci bilgisayara da ulasmayacagina gore webmimarın tapulu mali olarak kabul edilebilir. Webmimar içinden geçen her seyi satir

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

Sunucu tarafinda bu sekilde gorunen "ilk.asp" dosyasini kaydettiginiz sanal dizinden browser ile çağirin bakalim. Browserle çağirdiginiz "ilk.asp" nin kaynakini görüntülediginizde asagidaki kodu goreceksiniz. (Burada tarih ve saat olarak sizin bu sayfayi sunucudan istediginiz "ân" in degeri görünecektir)

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

```
<html>
<body>
<p>Asagida simdinin zamani
görüntülenecek

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

</body>
</html>
```

Gördüğünüz gibi istemci bilgisayara yazdigimiz hiç bir ASP kodu gitmedi. Sadece sunucu tarafından üretilen zaman degeri geldi. Sanki biz, bu sayfa sunucudan istendiginde belgeyi açıp elimizle o anki zamani yazdik ve de kaydetti, kullanıcı 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 istisnayı goz önüne alip bir plan yapmanız lazim. Ve bu plani, sunucuya anlayacağı sekilde anlatmanız lazim. Iste onun için ASP ye basvuruyoruz. Ileride deyim yerindeyse sunucuyu "kole" gibi kullanmayı da ogrenecez. Helal dostlar, helal...

FORM'UNDA SAYFALAR YAZMAK

Form lar veritabanı ile olan ilişkilerimizde aramızdaki çöpçatan gibi bir şey olacak. (ne benzetme!) Kimi zaman veritabanından aldığımız verileri düz metin haline dokmekten baska, formdaki alanlara da dokmek isteyeceğiz. En önemlisi veritabanına web den veri basarken, formları kullanacağız. Elimiz mahkum.

Bütün bilgi girişleri formlar üzerinden gerçekleşiyor desek yalan olmaz değil mi? Anketler, ziyaret defterleri, forumlar, arama sorguları... Hepsini oluşturalarımız form nesneleriyle teker teker halledeceğiz.

Bu makalede HTML formlarını ASP de nasıl amacımıza yönelik kullanırız; onu inceleyeceğiz. Bunu için HTML nin form unsurlarını bilmeniz gerekiyor. Aşağı yukarı her HTML editoründe bu unsurlar listelenerek hazır şekilde size sunuluyor.

FORMDAN İÇERİK TALEBİ

İlk önce formlardaki elemanların içeriğini talep etmeyi öğrenelim. Bir formda çeşitli elementler bulunabilir. Bunlardan bilgi girişleri için metin kutuları (input text), metin alanları (text area), seçme kutuları (checkbox, input radio), seçim menüleri (select option) vs.. kullanılabilir. Kullanıcı karşısına dikilen bir formdaki gerekli alanlara bilgi girişi yapar ve “submit” tuşuyla formu tüm unsurlarıyla “action” özelliğinde belirtilen URL ye gönderir. “Action” özelliğine girilen bu URL bir ASP dosyası olabilir. Çünkü ASP ile gönderilen form unsurlarındaki bilgilere ulaşabiliriz. İşte burada, bir asp dosyasında form elemanlarındaki içeriğin nasıl talep edileceğini göreceğiz. Bunun için bir form barındıran HTML dokümanı oluşturalım 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="Müşteri
Anketi">
Adınız : <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>
```

Sıra form.html belgesindeki formdan bilgi talep etmeye geldim. Bu işi bir ASP dosyasıyla yapacağız. Formdan bilgi talep etmeden önce, formdaki elemanların kimliklerini (name) bilmemiz gerekir ki istediğimiz şekilde kullanalım. Bu isimleri bilmesek bile topyekün listeleyebiliriz. Ama biz şu anda bildiğimiz bir formdan bilgi çekeceğiz. Ve elemanların “name” kimliklerini bildiğimiz varsayıyoruz ki kullanıcının girdiği bilgileri (value) lerini alalım.

Bir önceki sayfada verdığımız form.html nin kaynak kodunda, her bir form elemanının anlamlı bir adı olduğuna dikkat edin. Sırayla elemanların isimleri ve tipleri : FormAdi (hidden input), ad (text input), cinsiyet (radio input), ilgi (checkbox input), yas (option select), yorum (textarea).

Bu isimleri bildikten sonra yapacağımız şey, teker teker bu elemanların içeriğini talep etmek. Ama nasıl? Şu ifadeyle:

```
Request.Form("eleman_ismi")
```

Örneğin forma bilgi giren kişinin adını yazdığı "ad" kimlikli elemanın değerini Request.Form("ad") şeklinde

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" 'bey  
Else  
'degilse  
Ek = "Hanim" '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 kullanıcı form.html deki formu doldurup, gonder tusuna bastiginda forma girdigi bilgilerin görüntülandigi sayfayla karsilacaktır. Ornek:

Müsteri Anketi

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

SORGU CÜMLECİKLERİ KULLANMAK

SORGU CÜMLECİĞİ : QueryString

ASP sayfaları kodlarken, bir sayfadan diğer bir sayfaya bilgi aktarmanın değişik yolları vardır. Bunlardan biri olan Form kullanarak bilgi geçirmeyi görmüştük. Bir diğer yol ise sorgu cümlecığı kullanmaktır. Sorgu cümlecığı, link vereceğimiz sayfanın URL sinin sonuna eklenen çeşitli string (katar) tipteki ifadelerdir.

Ornek : http://www.x.com/deneme.asp?telefon_numarasi=4121254

Yukarıdaki URL de belirtilen deneme.asp sayfasına “telefon_numarasi” kimliğiyle bir değer yolluyoruz. Burada değer olarak sayısal ifade yazdık gibi görünüyor. Ancak hayır! Buraya istediğiniz kadar rakam eşitleyin, yazdığınız bir string dir, yani matematiksel olarak bir anlamı yok. Bildiğimiz düz metin.

Evet, herhangi bir sayfanın URL sinin sonuna bu şekilde sorgu stringleri ekleyebiliriz. Birden fazla eklemek istersek aralara “&” işareti koyarız.

Ornek: http://www.x.com/siparis.asp?musteri_no=15&siparis_no=52 gibi.

Boylece bir sayfaya sorgu stringi yoluyla bilgi geçirme ifadesinin ne kadar kolay hazırlandığını görüyoruz. Bir de madalyanonu oteki tarafına geçelim. Yani sorgu cümlesiyle yollanan bilgileri talep edecek bir asp dosyasına ihtiyacımız olacak. Bu asp dosyasında sorgu stringini talep etmek için aşağıdaki kod kullanılır.

```
Request.QueryString("eleman_adi")
```

Orneğin yukarıda verdiğimiz, URL deki sorgu cümlesinden değer çeken siparis.asp dosyasında şu kodlar kullanılabilir.

```
Dim musterino, siparisno
```

```
Musterino = Request.QueryString("musterino")
```

```
Siparisno = Request.QueryString("siparisno")
```

Bu kodun çalışmasıyla Musterino değişkenimize sorgu stringinden aldığımız “15” değeri, Siparisno değişkenimize ise “52” değeri atanmıştır. Tekrar edersek bu değişkenlerdeki değerler string tiptedir. String ve Sayısal tiplerin farkı için aşağıdaki kullanımı inceleyelim: (yukarıdaki koda devam ediyoruz)

```
Response.Write (Musterino + Siparisno)
```

Yukarıdaki ifade sonucu sayfaya $15 + 52 = 67$ yazılacak diye beklemeyin. Bunlar string ifade olduklarından aradaki toplama işareti matematiksel toplama işlemi yapmak yerine iki stringi birleştirecektir. Ve sayfaya ‘1552’ yazılacaktır.

Bir asp belgesine sorgu cümlecığı yoluyla bilgi geçirmenin yolunu öğrenmiş bulunuyoruz. Peki bu yöntemi nerelerde kullanabiliriz? Tabii ki bu yöntemi de daha çok veritabanlarıyla uğraşırken kullanacağız. Zaten anlatılan bu konularla hep veritabanı kullanımına hazırlık yapıyoruz. Sadece geldiğimizde bu yöntemleri çok kullanacağız. Örneğin veritabanından tekil kayıtlar seçerken sorgu stringi kullanmak en kullanışlı yoldur. Okuduğunuz bu web dokümanındaki yazılar da bir veritabanından çekilmiştir. Ve çekilirken de kolaylıkla

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 linklendiği 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!

Alan Adı	Veri Türü
İd	Otomatik Sayı
AdSoyad	Metin
Adres	Metin
Telefon	Metin
Mail	Metin
	Metin
	Not
	Sayı
	Tarih/Saat
	Para Birimi
	Otomatik Sayı
	Evet/Hayır
	OLE Nesnesi
	Köprü
	Arama Sihirbazı...

Sıra geldi dilimizden düsmeyen veritabani kullanmayı öğrenmeye...

Veritabani: Veritabanlarını verilerin depolandığı tabloları barındıran dosyalar şeklinde düşünebiliriz. Tablolar ise ad, soyad, telefon gibi alanlardan (field) oluşur. Bu alanların tamamı ise bir kayıta (record) oluşturur. Kayıt kavramı satır kavramıyla hemen hemen aynıdır. Veritabani tablosundaki satırlara record, sütunlara field diyeceğiz.

Veritabani dosyaları, farklı programlarla hazırlanabilir. Ancak biz burada yaygın olarak kullanılan Microsoft Access programıyla yaratılmış veritabanlarını

inceleyeceğiz. Microsoft'un bu programıyla kolayca kendi spesifik veritabani dosyanızı hazırlayabilirsiniz. (Excel de tablo hazırlamak kadar kolaydır.) Veritabani dosyanızın uzantısı .mdb olacaktır.

Yeni bir veritabani oluştururken yapacağınız en önemli işlem, amacınıza ve kayıtlarınıza uygun tablolar, sonra da tablolar için field (alan) lar oluşturmaktır. Ve de alanların tutacağı verilerin tipini belirlemek gerekecektir.

Id	AdSoyad	Adres	Telefon	Mail
22	Muazzez Güçlü	Hayat Ap. İskitler ANKARA	0312 456 23 12	muaz@muaz.com
23	K< bu bir kayittir (record) >	Kızılay	0312 123 12 12	kemal@kemal.com
24	Murat Kaya	Murat Ap. Keçiören	0312 254 23 45	murat@murat.com

Örneğin yukarıdaki resimde örnek bir veritabanındaki bir tablonun kayıtları görünüyor. (Access buna verisayfasi görünümü demis). Tablonun adı da önemlidir. Diğer tablolardan adıyla ayrılacaktır. Buradaki tablomuzun adı "OrnekTablo" dur. (Microsoft Access yazısının yanında yazıyor). Ornektablo adlı tablomuzda id, AdSoyad, Adres, Telefon, Mail adlarıyla oluşturulmuş 5 tane field (alan) bulunuyor. Resimde tabloya üç adet kayıt girilmiş. "id" alanı birincil anahtar dir. Tabloya eklenen her yeni kayıta bu alan otomatik olarak bir sayı yazılır. Diğer tablolarda, veritabanlarıyla ilişkilerde, özellikle bizim kullandığımız ASP sayfalarındaki sorgularda bu kayıt seçmek için bu tür alanları çok kullanacağız. Diğer alanlarda düz metin tipindeki alanlardır. Metin tipindeki alanlarda her kayıt en fazla 255 karakter bilgi tutabilir. Bu okuduğunuz yazı da veritabanından geliyor. Sizce 255 karakteri çoktan geçmedi mi okuduğunuz kısım. Elbette geçti. İşte bu tür uzun metin verilerinde "not" tipi veri tutan fieldlar kullanıyoruz. Eğer yukarıdaki "adres" sütunu yerine "Özgeçmiş" diye bir sütun olsaydı (255 karakteri asar değil mi bir özgeçmiş) ve biz buraya kişilerin özgeçmişlerini girseydik, Özgeçmiş alanının tutacağı veri tipini access deki tasarım görünümüne gelip ilgili yerden "not" a çevirecektik. Ote yandan tarih veya saat türünde veriler girilecek bir alanın veritürünü de tarih/saat olarak seçeceğiz. Ve sayısal kayıtlar için sayı veri türü seçilmeli.

Yani olusturdugumuz alanlarin veri türlerini (ikinci resimde asagi–düser listeyile görünen kısım) amacimiza uygun olarak belirleyecegiz.

ALANLARDA VARSAYILAN DEGER MESELESİ

Yeni bir kayıt eklendiğinde bazı alanlara otomatik olarak varsayılan deger atanmasını isteyebiliriz. Bunun için bir tablonun tasarım görünümüne geçin. Varsayılan deger koymak istediğiniz alanı seçin. Asagidaki o alanla ilgili özelliklerin ayarlandığı "genel" panelinde "varsayılan deger" yazısının karsısına istediğiniz varsayılan degeri girin. Ancak gireceğiniz deger, o alanın veri türüne uygun olmalıdır.

Örneğin sayı veri türüne sahip bir alanın varsayılan degeri "Sezen Aksu" olamaz her halde değil mi? Bu varsayılan deger özelliği tarih/saat türü alanlarda isimize çok yarayacak. tarih/saat türünde bir alanın varsayılan degerine now() yazarsanız, tabloya yeni bir kayıt eklediğinizde bu alanın degeri yani kaydı eklediğiniz an otomatikmen yazılır. Date() yazarsanız sadece eklenen tarih yazılır. Burada Visual Basic fonksiyonları geçerlidir.

Sıra örnek bir veritabanı dosyası hazırlamaya geldi. Hazırlayacağımız bu veritabanı dosyasını daha sonraki ASP ile veritabanı bağlantılarını öğreneceğimiz makalelerde kullanacağız. Sayfa 2 ye geçin ve örnek Access veritabanımızı oluşturmaya başlayalım.

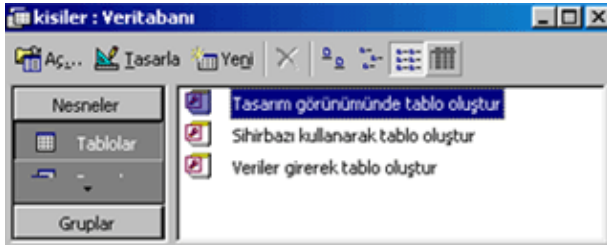
NEYİN VERİSİNİ TABANLAYACAGIZ

Bir fikir jimnastigiyle başlayalım. Neyin veri tabanını yapacağız? Bizim nasıl bir veritabanına ihtiyacımız olabilir? Bunun cevabı herkes için farklı olabilir. Golcü bir kardesimiz, attığı gollerin istatistiklerinin

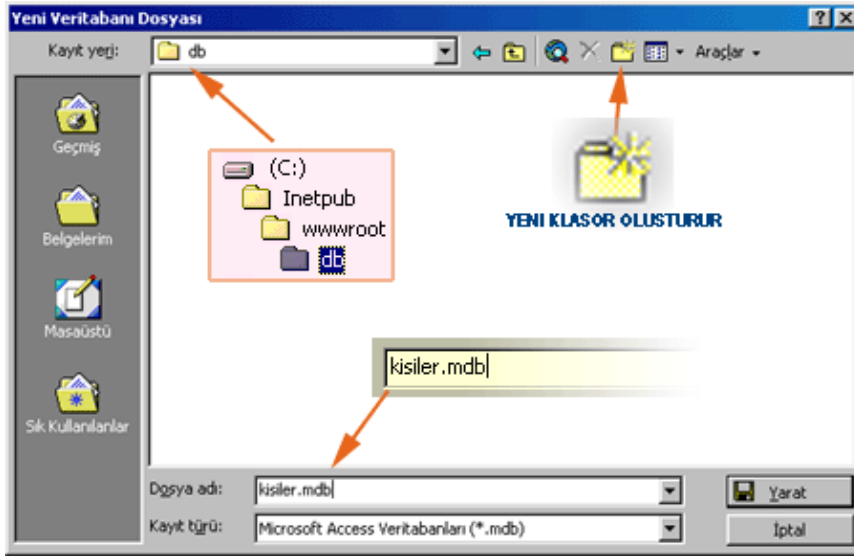
tutulacağı bir vt isteyebilir. Ablamız yemekler, malzemeleri, tarifleri vb., manyak kitap okuyan bir dostumuz kitapların, yazarlarının, yayinevlerinin vb. depolanacağı bir veritabanı isteyebilir. Biz burada genele hitap edecek bir veritabanı örneği hazırlayalım. Hazırlayalım ki siz de "Ahh.. ne de çok ihtiyacım vardı böyle bir veritabanına.. Ne iyi etti su Mucit.." diyebilirsiniz. Evet... Sürpriz. Bu akşam aldığım karara göre örnek bir arkadaş fihristi oluşturalım. Hani yanımızda tasiriz, hiç aramadığım arkadaşların telefon numaraları durur içinde. Doğum günleri, adresleri, e–kokoreç devrinde olduğumuzdan bir de e–mailleri... Ve eklenebilecek başka alanlarda arkadaşlarımızla, yakınlarımızla ilgili personal kayıtları barındıracağımız şahsa özel bir veritabanı. Ne hoş değil mi? Büyük ihtimal bu veritabanının sadece kendi bilgisayarımızda durması pek isimize gelmeyecek. Bunu internete atacacağız. Eee.. Attıktan sonra nasıl erişeceğiz? Ama acele etmeye gerek yok. Hepsine sıra gelecek. Biz önce eli yüzü düzgün bir veritabanımızı oluşturalım değil mi? Hadi başlayalım.

	Alan Adı	Veri Türü
id		Otomatik Sayı
ad		Metin
telefon		Metin
adres		Metin
email		Metin
icq		Metin
dogum_gunu		Tarih/Saat

BASLIYORUZ ORNEK VERITABANINA



Sifirdan bir veritabanı oluşturalım. Microsoft Baba'mızdan edindiğimiz Access 2000 programını açıyoruz. Açınca karşımıza adet olduğu üzere bir takım seçenekler sunuluyor. Nazlı gelinler gibi süzülmeden hemen amacımıza uygun olan ilk seçeneği (bos acces veritabanı) işaretleyip tamam diyoruz. Bu işlemin hemen ardından Access bize "yeni veritabanı dosyası" başlığıyla zarif bir pencere çıkartıyor. (ilgili resim aşağıdadır) Amaç yeni veritabanı dosyamız için bir dizin ve bir de isim seçmek. Dizin olarak PWS nin ana dizini olan "C:\inetpub\wwwroot" a geçelim (pws yüklü değil mi makinanızda? sayet yüklü değilse personal web server ile ilgili [yaziya](#) gidin ve yükleyin). Bu dizin içinde yeni bir alt dizin yaratalım, adını da ne verelim? Hemen heyecanlanmayın ad verme hakkı sizde değil, bende! Hayir bende de değil maalesef :) Internet üzerindeki bulundurma hizmeti veren sunucular yalnızca belirli dizinlere yazma hakkı veriyorlar. Güvenlik için tabi. Sunucun her tarafı bostan gibi ekilebilir bir alan olmuyor. Kendi bilgisayarımızda bütün dizinleri ekilebilir bostanlara çevirebiliriz. Ancak bizim amacımız yaptığımız database ye internet üzerinden erismek. Kendi bilgisayarımızda zaten üstüne iki tıklayıp açabiliriz dosyayı. Onun için dosyalarımızı internetteki bir alana atarken geçerli "path" problemi yaşamamak için veritabanı dosyalarımızı belirli bir dizinde toplamalıyız. Su anda okuduğunuz yazı Ewebcity nin sunucusundaki benim oluşturduğum veritabanından geliyor. Bu veritabanı ise /db klasöründe tutulmak zorunda. Biz de ayzma izni verilen klasör için db adını seçelim. Yani oluşturalım Access veritabanı için c:\inetpub\wwwroot\db klasörünü yaratalım ve bu dizini açalım. Ve veritabanı için bir isim belirleyelim: "kisiler.mdb" olsun. (Access veritabanı dosyaları uzantısı = mdb) Yarat tusu bizi amacımıza ulaştıracaktır. Artık fiziksel yolu "c:\inetpub\wwwroot\db\kisiler.mdb" olan bos bir veritabanımız var. Ne mutlu. Ama bos bir veritabanı. Mutlu olmayı 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 ilgilenecemiz bolüm ise tabi ki tablolar. Bir veritabaninda birden fazla tablo bulunabilir. Bu tamamen sizin verileri nasıl siniflandirdiginiza bagli olarak belirlenecek.

Biz veritabanimiz için simdilik tek tablo yaratacagiz. Tabloyu yaratmak için yanda resmi gorünen menüden "tasarim görünümünde tablo olustur" u tikliyoruz. (diger seçenekler de kullanılabilir. biz bunu seçtik bakalim.) Karsimiza üç sütundan olusan bir tablo çıkıyor. (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 alanı niye olusturuyoruz diye sorarsanız sabredin derim, onu nasıl kullanacagimizi gorecegiz. Simdi bu alan adarini tasarim görünümünde açtigimiz tablonun alan adlari bolümüne yazalım. 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 bulunduğu satira sag tiklayalım ve birincil anahtar komutu verelim. (satirin en saginda bir anahtar isareti belirecek) Yaptigimiz islemlerden sonraki görüntü ikinci resimdedir.

Simdi sorun yoksa Access in araç çubugundaki kaydet butonuna basinca program bizden yeni olusturdugumuz tablo için bir isim soracaktır. Gelin biz buna da "kisiler" diyelim. Tama dedikten sonra tablomuz kaydedilecek. Simdi "kisiler" adli veritabanimizin nesnelerrinin listelendiđi menünün tablolar bölümünde yeni olusturdugumuz "kisiler" tablosunun simgesi görünmekte. Üzerine çift tıklarsak "kisiler" tablomuzu "veri sayfası görünümünde" açacağız ki bu bizim çocukluktan beri bildiğimiz tablo görüntüsüdür. Excel'deki gibi veri gireceğiz. Yalnız tablomuzda hiç kayıt olmadıđından sadece tek satır var ve imleç (cursor) da bu satır üzerinde. Vakit kaybetmeden ilk satırdaki alanları doldurun. Tabi id alanı kendi kendine bir numara alacak. Bu ilk kaydımız olduđu için de bu numara "1" olacak. Ben tabloya örnek olması için 4 kayıt ekledim. Resim aşağıda.

id	ad	telefon	adres	email	icq	dogum_gunu
1	Selim Çalışkan	0312 453 12 23	Ufuktepe / AN	selim@yahoo.c	15775223	25.12.1980
2	Ahmet Menekş	0216 324 45 12	Ümraniye / İS	ahmet@ahmet.	12345657	12.12.1979
3	Hande Yüzen	0262 451 21 32	Saraybahçe /	hande@hande.	45132456	14.04.1980
4	Yelda Aktepe	0232 214 12 23	Balçova / İZM	yelda@yelda.cc	72451245	15.12.1983

* cursor, etkin kaydı gösterir

Eveet. Veritabanımızı oluşturduk. Simdi kaydet tusuna basıp ortalığı temizleyelim. Artık c:\inetpub\wwwroot\db dizininde kisiler.mdb diye bir Access veritabanı dosyamız var. İlerleyen makalelerde ASP ile bu örnek veritabanını kullanarak çeşitli işlemler yapacağız. Eger bu örnek veritabanını hazırlamaya üsendiyseniz [buraya](#) tıklayarak indirebilirsiniz (8,15 kb).

ASP İLE VERİTABANI BAĞLANTISI

BAGLANMAK...

Artık "yaz askimiz veritabanlariyla nasıl temasa geçeceğiz" mes'elesine geldik. Gerçekten olaganüstü basit ve kısa kodlamalarla dev gibi bir veritabanını önümüzde diz çöktürecek güçteyiz. Yasasın!

Hangi veritabanına bağlanacağız problemimiz yok çünkü önceki makalede örnek bir Access veritabanı dosyası hazırladık (kisiler.mdb). Bu dosyayı zip'ik halini [buraya](#) tıklayarak indirebilirsiniz (8,15 kb). Bağlantı muhabbetimiz bu dosya çerçevesinde olacak.

Veritabanı dosyanız "c:\inetpub\wwwroot\db" dizininde olsun. Şimdi "c:\inetpub\wwwroot" dizininde yeni bir asp dosyası yaratacağız. Kullandığınız editörü açın ve aşağıdaki kaynak kodunu kopyalayın. Sonra da vtbaglanti.asp adıyla wwwroot dizinine kaydedin.

Sekil1. vtbaglanti.asp kaynak kodu

```
<%  
' vtbaglanti.asp  
' bu sayfada db alt dizinindeki kisiler.mdb veritabanı  
' dosyasıyla bağlantı denemesi yapılıyor.  
' çevreye vereceğimiz rahatsızlık  
' umurumuzda değildir.  
%>  
  
<%  
" BAGLANTI KISMI BASLIYOR *****  
  
' bağlanmadan önce bağlantı nesnesi oluşturmamız  
lazım.  
' adını da ismail koyalım. elbet bir hatırası var.  
Set ismail =  
Server.CreateObject("ADODB.Connection")  
' Artık nurltopu gibi bir bağlantımız var. Adı da ismail.  
  
' sonra ismail adını verdiğimiz bağlantı nesnemizle  
kisiler.mdb veritabanı  
' dosyamızı açalım.  
ismail.Open ("DRIVER={Microsoft Access Driver  
(* .mdb)};DBQ=" &  
Server.MapPath("db/kisiler.mdb"))  
  
' kisiler.db de bulunan kisiler tablosundaki tüm  
kayıtları almak için
```



```
' bir SQL lafi geliştiriyoruz. Adini da ismailinSQLi
yapalım. Hatıra sürüyor.
Dim ismailinSQLi
ismailinSQLi = "Select * FROM kisiler"

' ismailinSQL i ni kullanarak ismail adli bağlantımızı
' kayıtlar adıyla çalıştırıyoruz. kayıtlar adini biz
salladık.
' siz buna hatıranın 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 açtığımız nesneyi en önce kapayarak
ayriliyoruz..

kisiler.Close
Set kisiler = Nothing

ismail.Close
Set ismail = Nothing

' ve ayrıldık. her şey bitti. sunucu buraya kadar
sorunsuz geldiye
```

```
' veritabanındaki kayıtlar html formatına doküüp
istemciye
'yollanmış demektir.
%>
```

Şimdi sunucu kök dizininize kaydettiğiniz bu asp dosyasını tarayıcı ile çağırın bakalım. (http://localhost/vtbağlantı.asp) Ne çıktı karşınıza?

Gördüğünüz gibi basit bu işler. Bağlantı bölümünde üçü geçmeyen kod emriyle bir veritabanındaki kayıtlara ulaşma imkanına kavuşuyoruz. Ve SQL (structured query language) sorgu dilimizle sepetimize kişiler tablosundaki tüm kayıtları dolduruyoruz.

Sonra da bu kayıtları sepet boşalınca kadar çalışacak bir döngüyle sayfamıza HTML taglerinin arasına doküyoruz. Ve sepetten dokme işlemi tamamlanınca sepeti sunucuya iade etmek gerekiyor (racon gereği, yoksa bir şey olmaz da). Burada kullanılan ASP kodlarıyla ilgili ayrıntılar sonraki sayfada sizi bekliyor.

SORU ISARETLERİ..

Önceki sayfada bir örnek veritabanı bağlantı sayfası oluşturduk. Ancak bağlantı yaparken karşımıza yeni ifadeler çıktı. Şimdi bunları inceleyelim.

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

Bu standart kod dizimiyle sunucu tarafında yeni bir nesne oluşturuyoruz. ismail ifadesi dışındaki her şey standarttır. siz oluşturacağınız bağlantı nesnelerinde ismail yerine başka isimler kullanabilirsiniz. Ancak ASP geliştiriciler böyle bağlantı yaptıkları zaman ObjConn (object connection laflarının kısası) gibi kendilerince anlamlı ifadeler kullanıyorlar. Biz de NsnBag diyebiliriz. Amann. Neyse iste.. İşin geyigine giriyor bu. Ne isim verirseniz verin (unutmayın s,g,ü,I,i,ç,o vb.. karakterler kullanılmıyor) çalışsın yeter.

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

ismail adındaki bağlantı nesnemizden bir veritabanı dosyasını bizim için açması emrini veriyoruz. Dosyayı açarken bir veritabanı sürücüsü kullanıyor, o da Access. Ve dosyamızın göreceli adresini de veriyoruz. Data Source Name less (DSN siz) bir bağlantı kurduk. Yani kullandığımız veritabanının ODBC sürücülerinde takma isimle tanınır halde olmasına gerek yok. Yani ek hiç bir ayar yapmadan bu sayfa belirtilen alt dizindeki kisiler.mdb veritabanı dosyasına rahatlıkla ulaşacak. Yolunu kesenin vay haline!

```
ismailinSQLi = "SELECT * FROM kisiler"
```

Burada bir ismailinSQLi değişkenimize bir string (karakter dizisi) ataması yapıyoruz. Ancak atadığımız karakter dizisi Ankaralı Turgut'un sarkisi sözlerinden farklı olarak bir anlam ifade ediyor. Değişkenimize çok çeşitli cümleler atayabilirken biz gittik "kisilerden ne var ne yok seç" mealinde bir laf dedik. Bu laf spesifik bir laf arkadaşlar. Elalem buna **SQL lafi** diyor. Peki SQL de ne? Bu konu ayrı bir makale konusu. (bekleyin). Buradaki SQL lafımız bir komut içeriyor: "kisiler tablosundan tüm alanları seç".

```
set kayıtlar = ismail.Execute (ismailinSQLi)
```

ismail adlı veritabanı bağlantımızı önceden belirlediğimiz SQL dilimizin direktifi doğrultusunda çalıştırıyoruz. Biz SQL dilimizle kayıt topluluğu seçtik. Bu kayıtlara ulaşmak için de çalıştırdığımız

veritabani için kayıtlar nesnesi oluşturuyoruz.

Artık kisiler.mdb veritabanındaki kisiler tablosundaki tüm kayıtlar elimizin altında. Ve cursor da ilk kaydın üzerinde. kayitlar("ad") dedigimiz zaman cursorun bulunduğu kaydın (satirin), ad alanına (field) ait degerini kastetmis oluyoruz. response.write kayitlar("email") dersiniz tablomuzdaki ilk kaydın email alanındaki degerini (selim@yahoo.com) sayfaya yazdirmis olursunuz.

Peki diger kayitların içeriğini nasıl yazdiracagiz. Tabi ki kayitlar arasında hareket ederek. Su anda cursorumuz ilk kayıt üzerinde deil mi?

kayitlar.movenext komutu cursoru bir sonraki kayda atlatacak. Su anda ikinci kayita geldik. Yazdiracagimiz alan degerleri ikinci kayita ait olacak. Ancak böyle tek tek cursor atlatarak kayıt degerlerini yazdirirsiniz sizi dokuz koyden kovarlar :P Çünkü az sonra gokten elma düşecek ve içinden VB SCRIPT çıkacak. Size bir onden kontrollü döngü yapisi verecek.

```
Do While <KOSUL>  
  <YAPILACAK ISLER>  
Loop
```

Biz de "saol vb script" diyerek döngümüzü yapılandıracağız. Kosulumuz ne olacak? Kayıt topluluğu sonunda degilsek, yani cursor dosya sonunda degilse. Bunu anlamak için kayitlar.eof özelligini kullanacağız. Bu özellik, eger kayıt topluluğu sonuna gelmissek **true** degeri verir bize. Aksinde false. Bizim döngümüz ise bu özellik false olduğunda yani kayıt topluluğu sonuna gelinmediğinde çalışacak. O zaman on kontrolümüz Do While Not kayitlar.eof olacak. Döngü içinde etkin kayda ait bilgilerin yazılması için gerekli ifadeleri yazacağız. (Orn: kayitlar("ad")). Etkin kayittaki içerik yazıldıktan sonra bir sonraki kayda atlamamız lazım. Kayitlar.movenext bu ise yariyordu. Ve döngü blogunu da Loop ile nihayetlendireceğiz.

Artık sayfamıza kayıtların içeriğini doktüğümüz için veritabanıyla isimiz kalmadı. Sunucuda açtığımız nesnelere son açtığımızdan başlayarak kapiyoruz. Soyle ;

```
kisiler.close  
set kisiler = nothing
```

```
ismail.close  
set ismail = nothing
```

Nasil? Kaptik mi bu isi...

HIZLI SQL LAFDANLIGI

Ahmet Savas'ın tanımı: "SQL (Structured Query Language) veri tabanlarındaki verileri işlemek için kullanılan yapısal sorgulama dilidir.". Evet doğru söylüyor Ahmet Savas. SQL dili çok fukara bir dildir. Ancak güzel işler yapan lafları vardır. Hani bazen bir çin atasözünde buluruz kendimizi. SQL da her türlü veritabanı dosyalarıyla evcil'lik oynamamızda birbirinden enfes laflarıyla bize yardımcı olan bir iletişim dilidir. Yani bir veritabanı dosyasına "Ey veritabanı! Hemen bana kisiler tablonun adı "a" ile başlayan insanların listesini uzat bakalım... Hadi ulan. Hemen şimdi. Basarım tokadı vallaa.." dersiniz elde geçen tek şey avcunuzu yalamak olur. Ne der atalar? "SQL veritabanını deliginden çıkarır". İşte tatlı dilimiz SQL ile

veritabanına emirler yagdiracagiz.

SQL laflari cesitli temel komutlardan olusur. Bunlar SELECT, FROM, WHERE, ORDER BY, GROUP BY, HAVING, UPDATE, DELETE, INSERT dir. Gordüğünüz 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"
```

Gordüğünüz 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 hakkında bilgiler sonraki sayfada devam ediyor.

TABLODAN KAYIT SILMEK

Orneğin tablomuzdan yasi 18 den küçük olanlari silmek için SQL lafi:

```
"DELETE FROM tablomuz WHERE yas < 18"
```

LIKE KULLANARAK KAYIT SEÇİMİ YAPMAK

LIKE kullanarak tam karsilastirma yapamadigimiz alanlarla kayit seçme imkani elde ederiz. WHERE ifadesinde alan adindan sonra kullanılan 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 MESELESİ

BEN HOST'ES İSTEREM

Evet, yerinde, son derece de makul bir istek. ASP ile geliştirdiğiniz web sayfalarınızı internete tasımak artık her yeni doğan çocuğun olduğu gibi sizin de en doğal hakkınız. Keske size "alin hepsi sizin olsun" diyebileceğim bostan gibi bir sunucum olsaydı. Ama farkettiğiniz üzere biz de böyle köprü altı sunuculardan ulaşıyoruz size. Beles etin yahnisine de katlanmak durumunda kalıyoruz.

BELES OLSUN HA!

İnternet gibi sınırsız bir alemde bedava site alanı veren yerler azımsanamayacak kadar çok biliyorsunuz. Hepsini değişik metodlarla sunucusundan bedava alan dağıtıyor. Bir nevi amme hizmeti yapıyor gibi görünüyorlar. En azından önceden böyle soylenebilirdi. Ancak şimdi gitgide azittiler. Bedava verdikleri sayfaları niye aldınız der gibi bir sürü reklam basıyorlar. Bir kovmadıkları kalıyor. Hatta bazıları kovuyor: DomainDLX.com. Bu arkadaşlar size Microsoft IIS 5 ile kovan (run) sunucularından taze bir yer verdiklerinde güle oynaya FTP nizi açıp belgelerinizi yerleştiriyorsunuz. Ahh... Reklamı da yokmuş derken iki gün sonra FTP nizi den "kovuldun!" gibi abuk sabuk iletiler çıkmaya başlıyor. Saygıdeğer arkadaşlar sayfalarınızın yukarisına ve de asagısına (ortası size kaldı, ne yaparsanız yapın) koyacağınız kodlarla reklamlarını yayınlamanızı istiyorlarmış meğerse. Ve altında da bir yazı : "Failure to do so WILL result in account termination!". Evet.. Sikiysa reklamimizi yapmayın demek bu resmen. İşte günümüzdeki bedava sunucuların çoğunluğu bu dingil ve embesil reklam anlayışı içerisinde. Güya reklamlardan para kazanıyorlar. Ancak kullanıcıların nefretini kazanmak, kazandıkları parayı bereketli kilmayacak, eminim! :)

Evet, derin host yaralarına fazla girmeden esas meselemiz ASP hosting e gelelim. Artık adımız gibi biliyoruz ki ASP sunucuyu kolelestiren hâkim bir teknoloji. İşte bu nedenle bu yüzden bedava ASP sunumu yapan servis hayli azınlıkta. Çoğu yer HTMLnin yanında, CGI, FRONTPAGE SE destekli bedava alanlar vadediyor. Yalnız ASP gibi bir "bela"ya yanasmiyorlar. Karamsarlığa kapılmayın. ASP sayfalarınızı yayınlatabileceksiniz mutlaka. Sonuçta bu sitede bedava bir yerden sunuluyor. Ancak ben sunu belirtmek istedim: "Dünyada ASP teknolojisini kullanarak oluşturduğunuz sayfaları sundurabileceğiniz babayıgit servisler çok sınırlı." Ancak bu babayıgitler tedbiri de elden bırakmamakta ısrar ediyorlar. Kısıtlamalar yapıyorlar çeşitli. Ancak bu sizi yıldırmaz. Çünkü beni yıldırmadı. Bakın bu site ayaktaysa ben yıkılmamışım demektir. :)

ASP sayfalarımızı dünyada bulunan üç beles servisle sunabiliriz. Bunlardan ilki, bu sitenin de sunulduğu ewebcity dir. İkincisi DomainDLX! (gıcık herifler). Ve bir de websamba.com var. Bir tane webhostme den bahsediyorlar ancak herifler katilimi durdurmuşlardı en son baktığımda. Neyse değerlendirmelere ilerleyen sayfadaki başlayalım.

EWEBCITY NASIL BIR VARLIK, ANLATIR MISIN BANA?

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

ASP scriptleri: Evet, dedik ya izin veriyor. standart ASP nesnelere 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 klasöründe 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" klasörü içinde kullanilabiliyor. Ote yandan asp ile yazdiracaginiz text dosyalari da "/db" dizinine yazilmali. Çünkü tahmin ettiginiz gibi yazma izni sadece o klasore verilmiş.

FTP Destegi: Maalesef. Ewebcity nin size en kötü sürprizlerinden biri olsa gerek bu. BU bedava servis size her ne hikmetse alanınıza FTP ile erisim hakki vermiyor. E peki nasıl yüklenecak sayfalar? Nasil mi? Ee golaaaay! Ewebcity size hizli mi hizli çalisana bir file upload (doya bindirme) sayfası 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 kötü haber. Bu servis reklam koymuyor ama Web'in yaygin nimetlerinden de bizi bayagi mahrum birakiyor. Ewebcity deki sayfalariniz mail gonderemeyen sayfalar oluyor. ASP sayfalarıyla 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/ (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/>

DOMAINDLX: BIR NEFRET ABIDESI

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

Domainlx [ana sayfası](#)' 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 önce basit bir HTML düzenine getirdikleri çagdası yorumu güncellemeleri gerekiyor.

DomainDlx size bedava bir alan sunuyor. Bu alan içinde ASP nin yanında PHP de çalıştırabiliyormussunuz. Size verilen dizine FTP ile erisebiliyorsunuz. Veritabanı dosyalarınızı /database, sunucu taraflı script içeren sayfalarınızı da /script klasörüne yerleştirmeniz gerekiyor.

DomainDlx sizden para istemiyor ancak vereceği reklam kodlarını (iki tane, biri üste, biri alta konulacak) sayfalarınıza yerleştirmenizi "rica" ediyor. Eğer ricasını yerine getirmezseniz sizi tarlasından kovuyor.

DomainDlx de ASP nin sunucu nesnesi FileSystemObject i kullanamıyorsunuz. Bu text dosyalarıyla işlem yapamayacaksınız anlamına geliyor. Mecburen verilerinizi tabanlayacaksınız.

DomainDlx in sasirtici bir özelliği var. Size sunucusundan e-mail bileşeni (persist) tahsis ediyor. Ve smtp mail göndermeç sunucusunu keyfinizce kullanabiliyorsunuz.

Ve son olarak DomainDlx in size bağladığı kosenize tarayıcıdan <http://media.domaindlx.com/evcilasp/> gibi bir URL ile ulaşıyorsunuz. Eğer arzu edenler olursa bu beles servis hakkındaki bilgilerimizi genişletebiliriz.

HAYDI SAMBA'YA!

[Websamba](#) da bedava ASP sunumu yapan servislerden. Ancak su ana kadar inceleme fırsatına erisemedim. Aldığım hesapta bir sorun oluştu FTP ile ulaşamadım. Yeni hesap açtırma müracaatına da henüz yanıt gelmedi. Benim için bir muamma. Ancak sizi bu servis konusunda da aydınlatmak vazifemiz. O yüzden bu servisin kritiğini [Bedavamaster](#)'i hazırlayan arkadaştan rica ettim. Sonraki sayfada arkadaşın Websamba hakkındaki değerlendirmesini bulacaksınız. Beni kirmadığı ve açıklayıcı bir yazı gönderdiği için kendisine çok teşekkür ederim.

WEBSAMBA VE DÜŞÜNDÜRDÜKLERİ

BedavaMaster'dan...

"

Oncelikle bu işi amatorce yaptıkları belli. Bu nedenle pek de ustlerine gitmek istemiyorum.

Önce artılarını söyleyeyim:

- + 30 MB web alanı veriyorlar.
- + Hiçbir banner koymuyorlar ,
- + Dosyalarınızı FTP ile upload edebilirsiniz.
- + Ewebcity gibi database dosyalarını illa su dizine koyacaksın diye şart koşmuyorlar.
- + CDO mail desteği veriyor.
- + XOOM gibi hergün junk mail yollamıyor. Hatta hiç mail yollamıyorlar.

+ 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"

diyorsanız ve ikidebir coken server sizi delirtmezse Websamba su an en

ideal

cozumlerden biri. Tabii uye olabilirsiniz. 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 arkadaşlarım. ASP kodlarımızı yazarken, ister VBScript, ister Jscript olsun istediğimiz script dilini kullanabileceğimizi biliyoruz. Yani sunucu sistemin desteklediği her türlü dil kullanılabilir. İstemci tarafındaki kardesimizin kullandığımız dille hiçbir ilgisi olmayacağından onun için hiçbir sorun teskil etmeyecek bu. Yani kullandığı tarayıcı Netscape olmasına rağmen bizim yazdığımız VBScript ifadeleri çalışacak. Çünkü bu kodlar Netscape'in yüce huzurlarına gitmeden, sunucu tarafındaki görevli memurlar tarafından icra edilecek. Ve bu özgürlük, Netscape'nin "Ben .asp uzantılı dosyaları çağırmayacağım bundan böyle, prensiplerimi çiğneyemem" diyerek cingar çıkarmasına kadar devam edecek.

Burada ASP sayfaları oluştururken gereksinim duyacağımız temel VBScript konularına değineceğiz. Sunu belirtmek gerekir ki bu konu hakkında soylenecek bir şey bırakmadı Hakki Ocal Agabey. Onun yazdığı ve sizin de aldığınızı tahmin ettiğim ASP kitapçığında mükemmel bir VBScript referansı mevcut. Yalnız biz burada bir ASP bahçesinin bahçivanlığını yapıyorsak elbette VBScript'ten de bahsedeceğiz.

VBSCRIPT NEDİR?

En basit cevap: Microsoft'un "Visual Basic" den türettiği 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 referansında. Biz burada belirtilen ikinci ortamdaki VBSCRIPT kullanımına değineceğiz. Yani "Web server scripting in Microsoft Internet Information Server"...

1. SCRIPT DILI BILDIRIM SATIRI

ASP sayfalarının en üstünde sunucuya, sayfada kullanılan Script dili hakkında bilgi verilir. VBScript kullanıldığı zaman sayfaların en üstüne

<% @ Language = "VBScript" %> yazılır. Microsoft sunucularında varsayılan script dili olarak VBScript'i kabul eder. O yüzden bu satırı yazmasanız bile PWS veya IIS sayfalarınızda VBScript dilini kullandığınızı varsayar ve buna göre davranır. Ancak JScript kullanıyorsanız <% @ Language = "JScript" %> satirini ekleyin.

2. HARF DURUMLARI

VBScript'in "letter-case" kaptisi yoktur. Yani büyük-küçük harf duyarlığına sahip değildir. "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 %>
```

3. DEGISKEN BILDIRIMI

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

4. DEGISKENLERDE VERI TÜRLERİ

Temelde kullandigimiz degisken türleri sayisal, alfa-sayisal ve mantiksal olmak üzere üçe ayrilir. Ancak her programlama dilinde karsimiza çıkan integer, real, string, boolean vs.. türü temel veri tipi tanımlamalarından yoksun bir dildir VBScript. Peki VBScript bir degiskenin hangi türde oldugunu nasil anlar? Bunu o degiskene yaptiginiz atama ile anlar. Asagidaki atamaları 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 bıçak 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 olacaktır. Sayılara 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 fonksiyonlariyle 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.

VARIANT VE ALT TIPLERİ

VBScript dili kendince tek veri türüne sahiptir, adı ise Variant! Variant, kullanımına göre değişik bilgi türleri içerebilen özel bir veri türüdür. Önceki sayfada örneklerini ordüğümüz gibi, değişkenin hangi alt veri türünde olduğunu sizin kullanımınız belirleyecektir. Variant'ın alt veri tipleri şunlardır: Empty, Null, Boolean, Byte, Integer, Currency, Long, Single, Double, Date, String, Object, Error. Bir değişkenin türünü öğrenmek istiyorsanız VarType fonksiyonunu kullanabilirsiniz. Bu fonksiyona değişken adını parametre olarak gönderiyorsunuz. Örnek:

```
<%
degisken_1 = 12
degisken_2 = "Ben sendeyim sen bendesin Rûveyda..."
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
%>
```

Gördüğünüz gibi bu fonksiyon bize anlamlı sayısal değerler veriyor. Bu sayısal değerler ne anlama geliyor?

Aşağıda 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 ; vbDouble 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
Örnek kodumuzda sayfaya yazılan 2,7,8 rakamlarını buraya bakarak anlamlandırabilirsiniz.

ALT TIPLER ARASINDA DONÜSÜMLER

Evet, buna mutlaka ihtiyacımız olacak. Variant'ın alt türleri arasında dönüşüm yapmamız için VBScript bize bir dizi fonksiyon tahsis etti. Bunlar nerede isimize yarayabilir peki? Özellikle String ve Integer tipleri arasında hizir gibi yetisecekler. Örneğin bir sorgu cümlesinden aldığınız (Request.QueryString ile) değeri matematiksel işlemlere sokamazsınız, çünkü string tiptedir. Diyelim ki sorgu cümleciginden sayfa numarası alıyorsunuz (okuduğunuz bu sayfanın çağrılımsında bunu kullanıyoruz). Aldığınız sayfa numarasıyla da bir takım matematiksel işlemler yapmak zorundasınız. E o zaman Cint gibi bir fonksiyonla aldığınız string tipteki değeri, sayısal bir türe çeviriyorsunuz. İşte size baslıca Conversion (Dönüşüm) fonksiyonları ve kullanım biçimleri.

CInt

Ne iş yapar: Herhangi tipteki geçerli bir ifadeyi integer alt tipine dönüştürür.

Soz dizimi: CInt(ifade)

Örnek:

```
mystr = "12"
```

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

CStr

Ne is yapar: Herhangi tipteki geçerli bir ifadeyi integer alt tipine donüştürür.

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 donüştürür.

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, CDb1, Chr, CLng, CSng, Hex, Oct fonksiyonlari da elimizin altindadir. Gerekirse aklinizda bulunsun.

BIRAZ VBSCRIPT (II)

DURUMLAR VE DAVRANISLAR

Programların en temel kabiliyeti karar verme mekanizmasıyla donatılmalarıdır. İnsan kendi kararlarını kendi verebilen bir canlıdır. Ancak bir program tasarlıyorsak, o programın karşılaşacağı durumlarda ondan karar vermesini bekleyemeyiz. Durumları göz önüne alıp, bu durumlara göre programın vereceği kararı, takınacağı tutumu biz belirlemeliyiz. Belirlemeliyiz ki yaptığımız şey program olsun.

İste programalara bu işleri akis kontrolleriyle yaptırırız. Akis kontrolleri, programın işaret edilen durum gerçekleştiğinde veya gerçekleşmediğinde herhangi bir davranış sergilemesini temin eder. Bu dilbilgisinde öğrendiğiniz, üniversite giriş sınavında karşınıza çıkan şart-kosul cümle türleri gibi bir şeydir. "Sunlar doğruyken sunları yap", "Sunlar doğru değilken sunları yap", "sunlar olana kadar sunları yap", vs..

VBSCRIPT AKIS KONTROLLERİ

VBScript'in baslıca akis kontrolleri genel yazımlarıyla şunlardır: **Do...Loop, For...Next, For Each...Next, If...Then...Else, Select Case, While...Wend**. Şimdi bunları inceleyelim.

IF...THEN...ELSE KONTROLÜ

Bir ifade grubunun, bir durumun mantıksal sonucuna göre yürütülüp yürütülmemesini sağlar.

Soz Dizimi:

```
If [durum] Then
  [ifadeler(if)]
ElseIf [baskadurum] Then
  [ifadeler(elseif)]
Else
  [ifadeler(else)]
End If
```

Ornek kullanım:

```
If Request.Form("Parola") = "Ben Parolayim" Then
  Response.Write "Evet.. Parolayi Bildiniz.. "
Else
  Response.Write "Bilinemedi. Heyhât!"
End If
```

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

olmadığı kontrol ediliyor. Eğer eşitse parolanın bilindiğini, eşit değilse bilinemediğini anlatan mesajlar yazdırılıyor.

SELECT CASE KONTROLÜ

Bir ifadenin kontrol edilecek durumu ikiden fazla olduğu vakit bu kontrolün kullanılması kolaylık sağlar. Yaptığı iş IF..THEN..ELSE ile aynıdır.

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 almıs bir web sitesinde kullanilabilecek SELECT CASE kontrolü yapiliyor. Saatin durumuna gore gelen ziyaretçiye bir seyler soyleneiyor. 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 kullanilrsa kosul dogruyken, "Until" ile kullanilrsa kosul dogru olana dek yürütülür.

Ornek kullanım:

```
Do While Not KayitKumesi.EOF
```

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

```
  KayitKumesi.Movenext
```

Loop

Veritabanından seçtiğimiz kayıtlarla oluşturdugumuz kayıtkümesinin "email" adlı alanındaki tüm kayıtlara ait değerleri bu dongü yapisiyla sayfaya yazdirabiliriz. Burada koydugumuz kosul: "KayitKümesi sonunda olunmaması" (Not KayitKümesi.EOF). Bu kosul dogru oldugu sürece dongü isleyecektir. Ancak dongü içindeki ifadeye KayitKümesi.Movenext metodu gozünüze çarpacak. Bu kayıtkümesinde bir sonraki kayda geçmemizi saglayan metoddur. Eger bunu yazmazsaniz dongü sayfaya hep ayni kaydın "email" alanındaki degeri yazacak, yani sonsuz dongüye girilecektir. Çünkü kayıtkümesi hiç bir zaman dosya sonuna gelmeyecektir. Dongülerde bu tür durumlara dikkat etmeliyiz.

Isterseniz bir de "Until" ile ornek yapalım.

```
Do Until a>=10
```

```
  Response.Write a & "<br>"
```

```
  a = a +1
```

Loop

UNTIL deki tek fark, dongünün giriste degil çikista kontrol yapmasıdır. Yani burada "a" degiskenindeki deger 10'a büyük-esit olsa bile programın akisi dongüye bir kez girer girer. Yani ifadeler bir kez islenir, dongü çikisinde kontrol yapılır.

FOR..NEXT DONGÜ YAPISI

Bu dongü yapisi DO..LOOP un özel bir seklidir. Bir ifade blogunun, belirlenen sayı kadar yürütülmesini saglar. Durum kontrolü yoktur. Bu dongüyle yapılan her sey, DO..LOOP ile de yapılabilir.

Soz Dizimi:

```
For [sayaç_degiskeni] = [ilk_deger] To [son_deger] Step [adim_sayisi]
```

```
  [yapilacak_isler]
```

Next

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

Ornek kullanım:(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 tanımlayıp, FOR..NEXT dongüsüyle, dizinin elemanlarını yazdiriyoruz. Yazdırma isleminde sayaç olarak belirlediginiz "sayac" degiskenini Gunler dizisinin elemanlarına ulasmak için indis olarak kullaniyoruz.

FOR EACH...NEXT DONGÜ YAPISI

Bir ifade blogunu, belli bir dizi veya koleksiyondaki 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 koleksiyon adi gelir.

Ornek kullanım (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 yazdirilacaktır.

Dizilerden sonra koleksiyonlarla 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 koleksiyonundaki 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 GELİN

Evet, sitenizin ASP ile akıl küpü haline geldigine en güzel örneklerden birisi sitede gezinen insanları saymaktır. Bunu saymak ASP için çok basit bir iştir. ASP adeta sunucunun nazlı gelini olduğundan istediği her türlü ayrıcalığı elde etmektedir.

ASP nin yerleşik nesneleri arasında iki tane yakışıklı nesne vardır: **Application** ve **Session**. ASP nin öyle bir gözü vardır ki bizim gördüklerimizi o farklı görmeye çalışır. Örneğin biz bir web sitesine "web sitesi" gözüyle bakarken, ASP onun bir uygulama (application) – yani bir program – olduğunu savunur. Ve siteye girenler de ona göre söz edilen uygulamayı kullananlardır. Öte yandan ASP, siteyi giren her kişi için ayrı bir "oturum" (session) başlatır. Ve çıkarken de bu oturumu sonlandırır. Ve kullanıcının bu oturum boyunca yaptığı her işten haberi olur. Adeta onun arkasında dolasıp "Ne yapıyor bu çocuk, öğrenmezsem çatlarım ayy" diyen bir Meraklı Melahat profili çizer. ASP nin bu yeteneği, onu çok çekici kılar. Site tasarımcısının yüreğinde yangınlar çıkarır bu ASP...

"Application" nesnesi bir uygulamadaki "oturum"lar arasında bilgi paylaşımı yapabilir. "Session" nesnesi ise bir oturum boyunca bilgi tutabilir. ASP nin bu iki yerleşik nesnesi ile yapılabilecek bir çok güzellik mevcuttur. Biz bu yazıda "sitedekileri saymak"tan bahsedelim.

Sitedeki adamları saymak için "Application" la muhabbet etmemiz lazım. Bu muhabbeti de **global.asa** yerleşik dosyasında yapmamız lazım.

GLOBAL.ASA

Microsoft der ki:

" Global.asa dosyası, içinde olay komut dosyalarını belirtebileceğiniz ve oturum veya uygulama kapsamı içeren nesneleri bildirebileceğiniz, seçime bağlı bir dosyadır. Bu dosya kullanıcılara görüntülenen bir içerik dosyası değildir; uygulama tarafından genel olarak kullanılan olay bilgilerini ve nesneleri saklar. Dosyanın Global.asa olarak adlandırılması ve uygulamanın kök dizininde saklanması gerekir. Bir uygulamanın tek bir Global.asa dosyası olabilir. "

Bu dosya, sitenize giren "oturum" cu için oturum başlamadan önce çalıştırılan uygulamadır. (Active Server Application). PWS ile oluşturulan C:\inetpub\wwwroot dizinine baktığınızda bu dosya size göz kırpacaktır. PWS sisteminiz global.asa kullanımını destekler. Ancak her yer insanın evi gibi değildir. Gittığınız bedava hizmet veren bir sunucu size **global**'ini bırak da gel derse sasırmayın. Nitekim bu yazıyı okuduğunuz sitenin ev ve yurt sahibi Ewebcity, global.asa kullanımını desteklememektedir. Ancak ücretli ASP buldurması yapan yerlerde bu kullanıma izin vardır. Bu işi sitenizde denemeden önce sunucunuzun global.asa ile barışık olduğundan emin olun.

Dedik ya global.asa lazim bize diye. C:\inetpub\wwwroot\global.asa dosyasini düzenlemek için açtiginizda karsinizda (eger daha once degistirmediyse) 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
    ' Uygulamadaki 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 dusür
    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. Artık global.asa sayesinde siteye oturmak için diz çöken biri görülürse uygulamada tutulan sayaç bir artirilacak. Siteden ayrılan bir vatandas farkedilirse sayaç bir azaltılacak. Bu sekilde sitede oturanların sayısı 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 kullanılabilir hale getirilmis oluyor. Yani oturumlar üstü bir degisken oluyor bu. Simdi sitedeki herhangi bir sayfada (ki bu genelde ana sayfadir) bu degiskenin degerini görüntülemek 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 çağirin. (Ornek: <http://localhost/test.asp>). Karsinize su metin çıkacak: "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 istiyorsanız: global.asa yi sitenizin ilk açılan sayfasinin olduğu dizine atın.)

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

FORMDAN VERITABANI ISTIKAMETINE GIDEN KAYITLAR

Evet, bugüne degin bir çok konu hakkında soylemler yansittik. Daha once veritabanindaki kayitlari sayfamiza nasil dokecegimizi gormüstük. Simdi bu isin daha ileri kısmi 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 açısından bu dosyamiz hazır bir şekilde bizi bekliyor. Bu Access dosyasini daha önceki "Ah veritabani..Ah.." baslikli makale boyunca hazirlamistik (kisiler.mdb). Bu dosyayi zip'ik halini [buraya](#) tiklayarak indirebilirsiniz (8,15 kb). (bu dosyayi wwwroot/db/ konumuna yerlestirin.) Bu bir örnek olacaktır. Buradaki kayit ekleme mantigini kendi olusturdugunuz veritabani dosyalarina kolaylıkla tasiyabilirsiniz.

VERITABANINDAKI TABLO VE ALANLARIN DURUMU

Veritabanina kayit ekleyecegimize göre, veritabanimizdaki tablo(lar) ve alanlari hakkında bilgi sahibi olmamız gerekir. Yoksa veriyi nereye gonderecegimizden habersiz bir kişi 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. Yalnız "id" kısmi birincil anahtardir. Buraya veri giremeyiz. Çünkü bu haylaz bölümdeki sayilar kendi kendine basina buyruk olarak artacaktır. Yeni bir kayit eklendiginde, bu alana mevcut kayit için otomatik sayi atanacak ve baska hiç bir kayit bu sayiyi alamayacaktır. Yani bu kayitlari benzersiz olmasını sağlayacaktır. Biz suçlulari kulagindan tutarken, kayitlari da 'id' sinden tutup kaldiracagiz. Karsimiza bir kayit çıkarsa: "Once id ni soyle, sonra konus... yoksa seni dinleyemem" diye agirligimizi da koyacagiz. Neticede "id" alani disinda tüm alanlara veri ekleyebiliriz.

Evet.. Artık on muhabbeti tamamladigimizi saniyorum. Simdi kollari sivayin bakalim. Dogustan edindiginiz HTML bilgilerinizle form barindirir bir doküman üretin. Uzantisi simdilik .html olsun. Ileride içinde ASP kodu kullanmak isterseniz .asp yaparsiniz. HTML formunu olustururken, seçtiginiz elemanlar, tablodaki kayitlari veri türüne uygun olsun. Örneğin metin, sayi, tarih türündeki alanlar için text input, not türündeki alanlar için textarea, evet-hayir tipindeki alanlar için de checkbox form elemanlarini kullanabilirsiniz. Örneğimizde tüm alanlar için text input kullanmak uygun. Bakalım 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 sayfamızı oluşturduk. Dikkat ederseniz formumuzdaki elemanların hepsi text input (metin kutusu) ve kimlikleri de (name) tablomuzdaki alanlarla eş isimler. Böyle olmak zorunda değil elbette. Siz ilk metin kutusunun kimliğini "ad" yerine "ey_sevdigim_bir_gun_bana_yar_demedin" dersanız sizin bileceğiniz iş. Kolaylık, kolaylıktır.

Formun "action" özelliğinde "kayit_ekle.asp" ye işaret ediliyor. Oysa böyle bir dosya yok ki. Birazdan olacak. Sayfayı çevirin.

KAYIT EKLEYEN ASP DOSYASI

Formdaki bilgilerin teslim edileceği kayit_ekle.asp yi inşa edeceğiz. Bui dosya içinde ne gibi işlemler yapacağız sizce? İsimiz formdan kimlikleriyle elemanların içerdiği bilgileri isteyeceğiz. Bu bilgileri sayfa içindeki değişkenlerimizde tutacağız tabii ki. Sonra eğer gerekli görülüyorsa bir takım değer kontrolleri yapacağız. Örneğin ad kısmına mutlaka bir şey girilmesini istiyorsak veya geçerli bir email adresi girilsin şartı varsa... Tüm bunları scriptimiz içinde kontrol işini tamamladıktan sonra asıl önemli kısma geleceğiz. Veritabanı bağlantısı yapıp bir kayıt kümesi oluşturulacak. Yeni kayıt eklenip alanlara formdan alınan uygun veriler doldurulacak ve de tablo güncellenecek. Her şey bu kadar. Sonuçta veritabanına formdan alınan bilgiler eklenmiş olacak. Ki bu da işimizi hallettiğimiz anlamına 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 çıkartmayin 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 kayıt seti yarat, mevcut tablodan sorgu yap, kayıt  
setini aç  
Set kayitseti =  
Server.CreateObject("ADODB.Recordset")  
kayitseti.Open "Select * from kisiler", bag, 1, 3  
  
' kayıt setine gicir bir kayıt 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  
  
' kayıt setini güncelle  
kayitseti.update  
  
' dayinin sözünü dinle, nesnelere 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>Sorun olursa 2000 yilinda yaptim :)
```

```
<% End Sub %>
```

```
<% '***** %>
```

```
<% '***** %>
```

```
<% Sub Basariyi_kutla %>
```

```
<h1>TEBRIKLER</h1>
```

```
Veritabanina kayıt eklem islemi gayet basarili bir sekilde  
tamamlandı.
```

```
<% End Sub %>
```

```
<% '***** %>
```

Yukaridaki kulanilan kodlarla ilgili ayrıntılar sonraki sayfada.

SORU ISARETLERİ

Sayfamizin basında bir kontrol yapısı goze çarpıyor. Hani olur ya.. Akillinin biri tarayıcıyı açar, kayıt_ekle.asp nin yolunu yazar. Bu durumda kayıt_ekle.asp ye formdan bilgi gitmemistir. Ve kayıt_ekle.asp nin de işi gücü olduğundan böyle amaçsız istemlere anında yanıt vermesini sağlıyoruz. Diyoruz ki istem metodunu içeren ifadede "POST" metni yoksa (ki bu, sayfamıza form ile ulaşılmamış demektir) bir hata mesajı görüntüle.

```
metod = Request.ServerVariables("REQUEST_METHOD")
```

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

```
mesaj = "Hayir.. Bu dosya formundan baskasına iş yapmaz. "
```

```
Vatandasi_Uyar mesaj 'uyari mesajı ver.
```

```
response.end 'yaniti kes
```

```
End If
```

SUB RUTINLER

Yukarıdaki kodda hata mesajını görüntülemek için bir sub rutine bir değişkenle parametre gönderip çağırıyoruz. Sub VBScript te bir alt kod bloğudur. Bu kullanımı yapabilmemiz için çağırdığımız SUB sayfa içerisinde tanımlanmış olmalıdır. Sayfanın en aşağısında ise bu SUB i tanımlıyoruz. (yeri mühim değil, en başta 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>Sorun olursa 2000 yılında yaptım :)
<% End Sub %>
```

SUB rutinimizin adı "Vatandasi_uyar". Bu sub, bir dışarıdan parametre alıyor. Parametremize mesaj adını verdik. Aldığı mesaj değerini formatlayarak ekrana yazıyor. SUB bloğunu END SUB deyiimi ile sonlandırmamız gerekiyor.

FORMDAN İSTENEN İÇERİK VE DE KONTROLLERİ

ad = trim(Request.form("ad")) tipindeki formdaki bir elemanın içeriğini istemeyi biliyorsunuz. Burada farklı olarak TRIM fonksiyonunu kullandık. Bu fonksiyon VBScript de bir karakter dizisi işleme fonksiyonudur. Metnin sağ ve solundaki boşluk karakterlerini temizler. Yani vatandaş formun ad kutusuna killik yapıp " Abuzer " yazdıysa kullandığımız TRIM fonksiyonu bunu "Abuzer" şeklinde kirpip bize geri verecektir. Bu şekilde formdaki diğer elemanların içeriğini de isteyip belirlediğimiz değişkenlerimize atama yapıyoruz. Sıra geldi bir takım kontrollere. Bazı alanların mutlaka doldurulmasını isteyebiliriz. Bunu için sıfır uzunluk kontrolü yapacağız. LEN(değişken_adi) fonksiyonu bize değişkenin uzunluğunu verir. Eğer bu uzunluk 0 ise değişkende değer 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 doğum günü kutularını adam gibi doldurun"
mesaj = mesaj & "<br>Zorluk çıkartmayın lütfen. Bakın ciddiylim!"
Vatandasi_Uyar mesaj 'uyari mesajı ver.
response.end ' işleyisi durdur. (veya sunucundan gelen yanıtı kes)
End If
```

Yazdığımız kontrolde eğer ad, adres ve doğum gününü tutan değişkenlerimizin uzunluğunun yanında bir de ISDATE(değişken) fonksiyonu ile doğum günü alanına girilen değer tarih/saat türüne uygun olup olmadığı denetleniyor. Eğer şartlardan en az biri (OR bağlacı bunu sağlar) gerçekleşmezse Vatandasi_uyar alt rutini bir hata mesajı görüntülenmek için çağırılıyor. response.end ile sunucunun sayfayı icrası 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 önüne geçiyoruz. Ornek:

```
If Len( icq)=0 Then  
icq = null  
End If
```

VT BAGLANTISI

Bunu daha önceki 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 kısmi basliyor. kayitseti.AddNew metodu kayit setine yeni bir satir (kayit) eklemis oluyor. (Yalnız henüz tablomuza yansimamis bir kayit bu). Kayittaki alanlara gerekli deger atamalarını yapıyoruz.

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

Açtigimiz nesnelere kapatarak bir de basarili islem yapildigi için mesaj görüntülemek istiyoruz.
`Basariyi_kutla` adıyla tanımladigimiz (parametre istemeyen) subimizi da çağırarak vazifemizi kivançla bitiriyoruz.

VBSCRIPT ILE STRING DÜZENLEME

STRING NEDİR BABA?

Microsoft'taki Türkçe kelime motorundan sorumlu arkadaşlar "dizgi" diyorlar "string"e. Takdir ediyorum, güzel bir kelime. Önceleri anlamadan geçiyordum bu kelimeyi. Daha sonra metinden çıkardığım anlama göre bu kelimenin String yerine kullanıldığını farkettim. O an ne kadar mutlu oldum bilemezsiniz. Ah...

Aslında String deyimi, gerçekte anlamıyla da örtüşüyor. "Sicim, ip" karşılığindeki bu deyim, karakterlerin pes pes sıralandığı veri türünü ifade ediyor. Burada karakterler için bir ayırım yok. Rakam, harf, sembol.. Yeryüzünde mevcut karakterlerin oluşturduğu bir zincir. Pascal'da dizileri kavradıktan sonra string türü verilerin de "karakter dizisi" olarak adlandırılmasında hiç bir sakınca olmadığını görmüştük. Yine de biz "dizgi" diyelim de hem sizi fazla kızdırmadan bu anlamsız geyiği keseyim, hem de Microsoft "yerlesikliğine" riayet edelim.

Peki dizgilere ve onları biçimleyeceğimiz fonksiyonlara neden gereksinim duyuyoruz? Neden? Çünkü biz "dizgi" siz bir şey yapamayız. (Üff... sikildim bu kelimedem. Ben "string" diye devam edeceğim, kusura bakmayın.) Örneğin okuduğunuz bu yazı bir string türü veridir. Formla yolladığınız her şey string türüdür. Sorgu cümleciginden aldığınız ivir ve zivir her şey stringdir. E bu kadar stringlerle iç içeyiz, o zaman onlara daha yakından bakalım değil mi? Stringlerin önemli bir özelliği matematiksel bir kıymet tasımamalarıdır. String türünde bir değiskende rakamsal veriler varsa, bunları ancak CINT gibi bir conversion (dönüşüm) fonksiyonu yardımıyla sayısal veri türüne çevirip matematiksel kavgalara sokabilirsiniz.

String lere matematik anlatamıyoz diye onları bası bos bırakacak demedik. Programlama dillerinde bir takım metin düzenleme fonksiyonları bulunuyor. VBScript ise Visual Basic dayısından aldığı string formatlama fonksiyonları ile karsımıza çıkıyor. 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**. Üff... Bunların hepsini değil de işlerimizde çok fazla ihtiyaç duyacağımız belli başlılarını inceleyeceğiz.

LEN FONKSİYONU

Length kelimesinden alır adını. Bir string ifade içindeki karakter sayısını döndürür. Len("Ankara-Izmit") ifadesi 380 (km) değil de ifadedeki karakter sayısı olan 12'yi döndürür.

Örnek:

Dim mystr

mystr = "İkimiz bir kuşlar gibi yuva kuralım. Ayırmasın Mevlam bizi ömür boyunca.. "

Response.write LEN(mystr) ' sayfa 74 yazar

UCASE, LCASE FONKSIYONLARI

Harflerin büyük–küçük dönüşümünü yaparlar. UCASE("ahMet") > "AHMET", LCASE("MUCIt") > "mucit" örneklerinden anlaşılacağı gibi UCASE uppercase ye, LCASE lowercaseye dönüşüm yapar. Yalnız dikkat edilmesi gereken bir husus:

UCASE("isil") > "iSiL" ; LCASE("ISIL") > "isil". Artık bu istenmeyen durumu kendi yazacağınız özel büyük–küçük fonksiyonlarıyla asacaksınız. (Merak etmeyin, yazarız bir tane)

INSTR FONKSIYONU

Bu güzel fonksiyon bize arama türü bir işlev sağlar. Uzun bir metin içerisinde daha kısa bir ifadenin indisi olarak başlangıç pozisyonunu verir. Soz dizimi: INSTR(index, uzun_string, aranacak_string, karsilastirmakodu) şeklindedir. Burada index ise uzun string içindeki aramanın hangi sıradaki karakterden başlayacağını bildirir. Bu siradan önceki metne bakılmaz. Eger index parametresi geçirilmezse 1 olarak kabul edilir. Aranılan ifadeye rastlanmazsa fonksiyon 0 değeri döndürür. Karsilastirma kodu 0 yazıldığında VbBinaryCompare yapılır, 1 yazıldığında ise VBTextCompare yapılır. Bunun ayrimina büyük küçük harf duyarlığında varirsiniz. Aşağıdaki örnekleri inceleyin.

Örnek:

Dim mystr

mystr = "Bilene her devir aslında Lâle Devri / Akliniz varsa Nedim gibi yaşayın..(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 faydalı bir fonksiyondur. Uzun bir metin içerisindeki belirlediğiniz ifadeyi başka ifadeyle değiştirmenizi sağlar.

Soz dizimi: REPLACE(string, degisecek_olan, yerine_konulacak, baslangic_indexi, degisiklik_sayisi, karsilastirmakodu)

Parametreler:

String: İçinde değişiklik yapacağımız metindir.

Degisecek_olan: Metin içinde değiştirmek istediğimiz alt–dizgi (sub–string) dir.

Yerine_konulacak: Belirtilen alt–dizginin yerine konulacak ifadedir.

Baslangic_indexi: Değiştirme işleminin, stringin kaçınıcı karakterinden başlayacaktır. Varsayılan: 1

Degisiklik_sayisi: Çok açık. –1 yazılırsa bulunan tüm alt–dizgiler değiştirilir.

karsilastirmakodu: VBTextCompare için 1, VBinaryCompare için 0 yazılabilir. Varsayılan 0.

Replace fonksiyonu su şekilde de işlev yapar. REPLACE(string, degisecek, yerine konulacak). Ancak fonksiyonun daha verimli kullanılabilmesi için tüm parametrelerinin belirtilmesi gerekir. Örnekleri inceleyin, anlayacaksınız.

Örnek:

Dim mystr

mystr = "Agir agir çıkacaksın bu merdivenlerden"

Response.write replace(mystr, "Agir", "Hizli", 1, -1, 0)

' sayfaya "Hizli agir çıkacaksın bu merdivenlerden" yazar.

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

' sayfaya "Hizli Hizli çıkacaksın bu merdivenlerden" yazar.

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

' sayfaya "Agir Agir çıkAcAksın bu merdivenlerden" yazar.

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

' sayfaya "Agir Agir çıkacaksın bu merdivenlerden" yazar.

Replace fonksiyonu bir çok yerde yardımımıza yetisir. Örneğin bir formdan alınan içerikte yer alması muhtemelen istenmeyen ifadelerin ayıklanması sağlanabilir. HTML taglerinin kullanılması onlenebilir. Ve daha bir çok yerde REPLACE size büyük fayda sağlayacaktır.

LEFT, RIGHT FONKSIYONLARI

Left bir metnin sol başından, Right bir metnin sol başından belirtilen sayıda karakter alır. Sozdizimi: LEFT|RIGHT(string, uzunluk).

Örnek:

Dim mystr

mystr = "Yüz bin elle dokunurum sana, İstanbul a..."

Response.write LEFT(mystr, 7) ' sayfaya "Yüz bin" yazar

Response.write RIGHT(mystr, 13) ' sayfaya "İstanbul a..." yazar

MID FONKSIYONU

Left fonksiyonunun biraz gelişmiş şeklidir. Burada uzunluk parametresinden önce bir de index parametresi gönderilir.

sozdizimi: MID(string, index, uzunluk)

Örnek:

Dim mystr

mystr = "Yüz bin gözle seyredirim seni, İstanbul u"

Response.write MID(mystr, 5, 3) ' sayfaya "bin" yazar

TRIM, LTRIM, RTRIM FONKSİYONU

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

Örnek:

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 FONKSİYONU

Biraz eğlenceli, biraz da füzuli geliyor bana. Metni ters çeviriyor. Bunun için hazır fonksiyon yazmaya ne gerek vardı? Örneğe bakın.

Örnek:

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 FONKSİYONU

VBScript ten yine evlere senlik bir fonksiyon. Belirtilen sayıda karakterin tekrarını içeren string oluşturuyor.

Örnek:

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

SPLIT FONKSİYONU

Bir metinden belirlenmiş ayrıçlar (delimiter) vasıtasıyla belirtilen sayıda alt–dizgi içeren 0 tabanlı, tek boyutlu dizi üretir.

Sozdizimi: SPLIT(ifade, ayrıç, sayı, karşılaştırmakodu).

(Sayı yerine –1 yazılırsa tüm belirlenen tüm alt–dizgiler işleme tabi tutulur.)

Ne iş yaptığını anlamak için örnek yapalım.

Örnek:

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 FONKSİYONU

Split fonksiyonuna göre zir işlemi yapar. Bir dizinin elemanlarını belirtilen ayraç ile birleştirip bir string üretir.

Sozdizimi: JOIN(ifade, ayraç)

Örnek:

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 öğrendiğimiz fonksiyonlar bize simdilik yeter de artar bile. Ancak gelişen koşullara göre

- diğer fonksiyonlara da ileride göz atmak gerekebilir. Haydi sağcıkla kalın. Her string biçimlendirme sırasında EvcilASP'yi hatırlayın. Eyvallah...

VERİTABANINDAKİ KAYITLARI GÜNCELLEMEK

Veritabanı muhabbetimiz tam gaz devam ediyor. Bu yazıda bir çok veritabanı operasyonunu birlikte göreceğiz. Kayıt ekleme, silme, güncelleme gibi. Kullanacağımız veritabanı dosyası [burada](#)(8,15 kb). Bu dosya C:\inetpub\wwwroot\db konumunda olsun. Sıra geldi script dosyalarımıza.

KAYITLAR.ASP

Oncelikle kayitlari listeleyecegimiz bir arabirim insha 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 şekilde /db alt dizinindeki kisiler.mdb ile bağlantı kuruyor. kisiler tablosundan aldığı tüm kayıtların "ad" alanındaki değerlerini yazdırıyor. Ve de her bir kayıt için düzenleme ve silme linkleri oluşturuyor. Linkler oluşturulurken sorgu stringi içinde eylemi gerçekleştirecek script dosyasına "id" anahtarıyla kaydın "id" alanındaki değeri gönderiliyor. Tabi bunlar kayıtseti sonunda duracak bir döngü içerisinde yapılıyor. Veriler tablo hücrelerine döngüyle doküldüğüne de dikkat ediniz.

VERİTABANINDAKİ KAYITLARI GÜNCELLEMEK

KAYIT_DUZENLE.ASP ve KAYIT_GUNCELLE.ASP

Kayıtların listelendiği kayitlar.asp de her kaydın yanında bir düzenle bağlantısı var. Bu bağlantı ile duzenle.asp dosyasına işaret ediliyor. Bir de sorgu cümlesi konuluyor yanına: "duzenle.asp?id=XX" Buradaki XX yerine geçerli kaydın id alanındaki değeri yazılıyor. Şimdi kayit_duzenle.asp ye bakalım.

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 %>
<%'-----%>

```

Aslında bu scriptin de öncekinden pek farkı yok. sadece sorgu cümlesinden "id" anahtarındaki değeri alıyor. Ve bu id değerini kullanarak tablodan tek kayıt seçiyor. ("SELECT * FROM kisiler where id =" & id). Seçtiği kayda ait değerleri ise form input alanlarına doküyor. Formun action özelliğine ise kayıt_guncelle.asp yazılmış. Güncelleme isini bu dosyaya havale ediyoruz.

Kod3. kayıt_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 kayıt_yeni.asp... Yaptığı iş şu. Veritabanına bağlanıp ilgili tablodan oluşturduğu kayıt seti içine yeni bir kayıt ekliyor. Kayıtsetini güncelliyor, "id" alanındaki değeri bir değersiz hale getiriyor. nesnelere kapatıyor. Ve de kayıt_duzenle.asp dosyasına "id" anahtarıyla yeni eklenen kaydın "id" alanındaki değeri gönderiyor. Yani bu sayfa sadece işlem yapıyor. Kullanıcıya gözükmeden işlemi tamamlayıp düzenleme sayfasına yöneliyor.

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" anahtarından alınan id degeri kullanılarak su SQL ifadesi olusturuluyor. DELETE FROM kisiler WHERE id = " & id . Ve bu ifade baglanti tarafından yürütülüyor (execute). Sonuçta id si alınan kayıt tablodan silinmiş oluyor. Kullanici da geldiği sayfaya 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 bahsedeceğimizi başlıkta çok açık bir şekilde belirttik. İnternet'in en kullanışlı ve insani boyutlarından biri de maıldır bence. Çok zaman geçmedi mektup zarflarının kapaklarını dilimizle isladıktan sonra büyük bir heyecan ve "ümit" içerisinde yapıştırıp güle oynaya Posta Teskilati binasının yolunu tuttugumuz günlerin üzerinden. Mektup yazmayı çok severdik biz günesin çocukları olarak. Öğretmenler önce karsinin hal ve hatirini sormayı öğütlediler bize. Biz de hep "orada havalar nasıl" diye başlardık test.mektuplar'a. Ve kendimizden bahsetmeye gelince de "Burada havalar iyi" en büyük vurgumuz olurdu. Sonra amcalara, teyzelere yazılan mektupların yerini Leylâ'ya yazılan mektuplar aldı. Ozenle yazdığımız "sevda" kokulu cümleler, postacı amcanın eline ne büyük güvenle teslim edilirdi! Postacı amcalar adresi hiç sasırmazdı, sevda menzile ererdi. O zaman cep telefonu olmadığı için karsidaki arkadaş anında "mektubun geldi arkadaşım, haziran da geldi" içerikli kısa bir mesaj atamazdı. Siz Leylâ'ya mektubun ulaştığını, ancak Postacı amcanın karşı sokaktan elinde isildayan bir zarfla sizin kapiya doğru yönelisine tanık olduğunuzda anlardınız. Hangi heyecanlar salınırdı içinizde kim bilir. Hangi yıldızlara kosardı gözleriniz...

Artık mektuplardan çok uzaklastık. Çünkü artık internete, onun getirilerine çok yaklastık. Posta Teskilati'na giden, yürüdükçe uzayan yolun yerini, basit "send" tuşları aldı. İadeli taahhütlü mailler iletimin ağır yükünü kolaylıkla sıtladı.

Şimdi herkesin neredeyse kendi Posta teskilati var. Mail sunucular harıl harıl çalışıyor. Bu hızlı değişim, bizi mail iletiminin üzerine daha ciddi eğilmeye itiyor. Deminden beri "ne diyor yav bu, biz buraya ASP türkülleri dinlemeye geldik" dediniz. Sabrettiginize degecek umarım. Bu kısa muhabbetten sonra ASP teknolojisiyle nasıl postacılık yapacağımıza degeceğiz. Artık buraya kadar okudunuz, gerisine katlanırsınız değil mi :)

POSTACI OLUYORUZ.

ASP bize kolay bir şekilde mail gönderme imkanı sunmakta. Yalnız bu kolaylığı çoğu zaman verimli bir şekilde kullanamıyoruz. Çünkü bu mail iletisi sistemi, mail sunucunuzu kullanmanızı tesis eden hazır sunucu (mail component) bileşenleriyle sağlanmakta. Bunlar da sunucudan sunucuya değişiyor maalesef. Kimi sunucunun tercihi CDO, kimininki Persist, kimininki de JMail. Piyasada envai çeşit mail bileşeni mevcut. Genelde bedava oluyorlar. Bunlardan en yaygını saydığımız üç tanesi. Bu bileşenleri ASP ile kullanarak mail göndermek için bu bileşenlerin, sunucu sisteme kayıt edilmesi (register) gerekir. Bazı bileşenlerin kendi kurulum programları, bu işi size zahmet vermeden yapmaktadır. Bazılarını ise siz, bir zahmet kayıt edeceksiniz (c:\windows\system\regsvr32.exe [bileşen] komutuyla olabilir.)

Ancak bizim büyük ihtimal kendi sunucumuz yok. Bileşen kayıtlarıyla da isimiz yok diyebilirsiniz. O zaman mail göndermeyi hala istiyorsanız dosyalarınızı "publish" ettiğiniz web sunucuda herhangi bir mail bileşeni olduğundan emin olunuz. Bunun için sunucu serifiyle iletime geçin. Web sunucunuzdaki mail bileşenini mutlaka öğrenin. Çünkü mail gönderme scriptinizde kullanacağınız kodlar, bileşenden bileşene az da olsa fark gösterecektir. Her bileşen bazı kendine özgü metodlarla mail göndermeye programlanmış. Yaptıkları iş aynı da, niye metodları farklı dersiniz? Valla bilmem. Biri obj.send ile mail gönderiyor, diğeri obj.execute ile orneğin. Bileşeni yazan insanların kendi tercihleri bu. Sikâyet hakkımızın olduğunu sanmıyorum. Ne güzel, iş çıkıyor iste.

DERS BASLIYOR, ONCE AMAÇ!

Amacimiz ne? Mesela kullanıcıların sitedeki bir formu doldurarak site yöneticisine mail göndermesini sağlamak. Veya “yakın bir arkadaşına bu siteyi ziyaret etmesini tembihle” düzenegi kurmak. Biz, daha “mantikli” olduğundan birinciye örnek verelim.

FORM'UM

Önce form içeren bir HTML belgesi üretmemiz lazım. Dosyayı üretip adına 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 görünümünden siz sorumlusunuz tabi ki. Burada “estetik bir form tasarlayalım 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” düğmesi bulup tiklayacaktır. 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 bileсени olan CDO dan bahsediyoruz. Bu bileсени NT Option Pack 4 ile geliyor. Yani ayaginiza kadar geliyor. Yayginligi da buradan geliyor tabi ki. Eger web sunucunuzda CDO mail bileсени 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"  
  
' bir CDO Newmail nesne ornegi yarat  
Set mektup =  
Server.CreateObject("CDONTS.NewMail")  
' çok açık, 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 olursu
```

```

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 gönderme bileşeniyle deneyelim bu işi. Maksat çeşitlilik olsun. Karsımıza hangisi çıkarsa çıksın, 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"

' bir Persist Mailsender nesnesi...
Set mektup =
Server.CreateObject("Persits.MailSender")
' mektup yollanırken hangi SMTP sunucu kullanılacak
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 kullanımına bakalım.

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"

' 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 girilmediğini kontrol eden, hatali durumlarda daha fiyakali uyarilar verip, adami tekrar forma gotüren daha kullanisli ozellikler ekleyebilirsiniz.

Üç ayri mail bileseniyle, üç farkli yolla.asp çıkarttik 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 kosedede dursun oyle. Duygularimizi baska ne türlü muhafaza edecegiz? Elbette yazarak.

Yazarak!

BIR ONERI

Mail gonderme isi, karsimiza muhtemelen çok yerde çıkacak. Bunu goz önünde 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 çalıştirilir. 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 açısından 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 çalıştıracagız. Ancak önce bu fonksiyonu içeren mailf.asp dosyasını, sayfamıza "include" (dahil) etmemiz lazım ki fonksiyon sayfamız hudutları içerisinde çağrılabilir olsun.

Sekil6. Deneme.asp

```
<!--#INCLUDE  
FILE="mailf.asp"-->  
  
<%  
' -----  
' baska isler  
' -----  
' ....  
gonderildi = mailgonder(isim,  
mail, kime, konu, mesaj)  
If gönderildi Then  
Response.Write "maili yolladik"  
Else  
Response.Write "hata olustu,  
maili yollayamadik"
```

```
End If  
' -----  
' baska isler  
' -----  
' ....  
%>
```

Bu gibi rutinlesen, çok tekrarlanan isleri alt program bloklarına yüklemek, her zaman rahat, estetik, kullanışlı ve profesyonel çözüm olacaktır. Zaten bu işin raconu da budur. En olmadık şeyleri bile sub rutinelere simdiden yükleyerek, bu işin havasına girmek faydeli olacaktır 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 adınıza hitaben "merhaba koçum, seni taniyorum, bana kendini anlatma, çok iyi bilirim ben seni ... ismin su, sevdiğin kalem markasi Rotring, basin agriyinca onceden Novalgin alirdin simdi modaya uyup Aspirin Forte.. monitorün philips marka ve sen simit seklindeki power düğmesine gicik oluyorsun ... haa bir de bu siteyi 2. ziyaretin ... nerden bildim ama... hahaha" türünden saklabanliklar yapıyor. Peki bunu nasıl beceriyor? Hepiniz çerezlerle oldugunu biliyorsunuz.

Biz yine de çerezlerin üzerinde kısaca duralım. Çerezler istemci bilgisayar yollanan kabuklardır. Siteniz bir yemis olarak düşünülürse, cookie denilen seyler de yemisin kabuklarıdır. Siz yemisi yersiniz, kabuklarını bilgisayarınıza atarsınız. Yalnız bu kabuklu yemis kullanan siteler için geçerli. Bir de tarayicinizin yemis teknolojisini destekleme kabiliyeti ve "yemis kabuklarını kabul etme" modunda olması gerekir. Yani oz olarak kabuklu yemis dagitan bir siteye gireceksiniz, bir de yemis yemeyi bileceksiniz. Anlastik mi? Büyük ihtimalle MS Internet Explorer kullanıyorsunuz. Tarayiciniza cookie kabul etmesini de tembihledikten sonra ise koyulalım.

ASP ile istemci bilgisayarlara çerez yollamak çok basit bir istir. Gerçekten çok basit. ASP nin yerlesik delikanlılarından RESPONSE nin bu iş için özel bir metodu var: COOKIES. İstemciye çerez yollamak için bu metodla birlikte anahtar degerler (bir veya iki) gönderiyoruz. (bir ayrıntı: ASP de iki anahtar sadece cookies kolleksiyonunda kullanılıyor). Simdi örnek kodu inceleyelim:

```
Response.cookies("kabuk")= "Evet, kabuk"  
Response.cookies("kabuklarim")("adi") = "Mucit"  
Response.cookies("kabuklarim")("sevdiği_icecek") = "Kızılcahamam Maden Suyu ve Sodasi"  
Response.cookies("kabuklarim")("medeni_durumu") = "Bekar!"  
Response.Cookies("kabuklarim").Expires = Now() + 90
```

Gördüğünüz üzere birinci satırda tek anahtar açıyoruz ve bir deger ataması yapıyoruz. İkinci satırda "kabuklarim" anahtarı altında "adi" alt anahtarı açıyoruz ve "Mucit" degerini koyuyoruz. İlk anahtar sitemizden yollanan çerezleri gruplandırmak için. Dizinleme türünde bir sey yapıyoruz nitekim. Orneğin bir anketimiz var ve oylandığı zaman bilgisayara çerez yollamak istiyoruz. O zaman "anket" diye yeni bir anahtar altına alt anahtarlar açabiliriz. Böylece sitedemizden yollanan farklı amaçlı çerezlerin gruplandırılması sağlanmış olur. Yolladığımız çerezlerin ikinci anahtarında saklıyoruz bilgiyi. Orneğin ilk satırdaki "adi" anahtarında bir deger gönderiliyor istemciye. Aynı şekilde üçüncü ve dördüncü satırlarda da gereksiz bilgiler gönderilmeye çalışılmış. Yalnız son satırda kısmen değişik bir kullanım var. "kabuklarim" a kadar aynı her sey. Ancak "kabuklarim" dan sonra bir alt deger anahtarı açılmamış da EXPIRES alt metodu kullanılmış. Tahmin ettiğiniz gibi yemisin ne zaman çürüyüp kullanılmaz hale geleceğine (expired olacağına) karar veriyoruz. Buraya tarih/saat yani ejnebi dilde date/time veri türünde deger ataması yapıyor. Atamaya bakarsanız Now() + 90 denmiş. Yani simdiki andan tam 90 gün sonra bu yemis çürüsün, kokussun emri verilmiş. Genelde bunu 365 yaparlar. Neden dersenez bir yıl 365 günden oluşuyormus...:)

Peki çerezi yolladık tarayiciya. Yani kabukları. Peki bu kabukları gormek istedigimiz zaman ne yapmamız gerekiyor. İşte o zaman ASP nin diğer delikanlı nesnesi REQUEST talip oluyor göreve. Çünkü bir seyleri talep etmek onu görevi. Biz de ona bir emir vereceğiz. "Çerezler arasından su anahtardaki degeri bana bir getiriver bakayım" denildiği anda size o degeri getirecektir. Bu metodu kullanmak, response ile çerez yollamaya çok benzer. Ancak bir fark vardır. Response de ilgili anahtara atama yapılır, burada ise request zaten bize bir deger getirir, biz de bu degeri isimize geldiği gibi kullanırız. Asagıda REQUEST ile çerezden bilgi almayı görüyorsunuz. Yalnız ben burda yukarıdaki çerez yollama örneğinin icra edildiğini 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 çıktı: Mucit / Kizilcahamam Maden Suyu ve Sodasi / Bekar!
```

Iste bu örneklerde çerezle bilgi yollama ve de çerezden bilgi alma tekniğinin ne kadar kolay olduğunu görüyorsunuz.

NERELERDE KULLANILABİLİR?

Aslında kabuklu yemeler, windows'un registry teknolojisinin ufak modelidir. Bilgisayarınıza yüklediğiniz bir çok program Windows'un registry veritabanıyla birtakım münasebetlerde bulunur. Programda varsayılan ayarlarda yaptığınız değişiklikler Windows Registry veritabanında belirlenen anahtarlarla kaydedilir. Programı daha sonra tekrar açtığınızda, değiştirdiğiniz ayarlar Registry den okunur ve programa yansıtılır. Bunlara en basit örnek pencere boyutlarıdır. Programın pencere boyutlarında yaptığınız düzenlemeler, bir sonraki açışınızda korunur. Bu ayarlar harici dosyalarda da saklanabilir, ancak Windows Registry etkili, hızlı ve profesyonel çözümdür.

Iste yüklediğiniz programların registry münasebeti ile web sitenizin cookie münasebeti arasında pek fark yok. Site kişiselleştirilmesi iste bu çerezlerle yapılıyor. Çerezler, yani kabuklu yemeler tasarladığınız, geliştirdiğiniz ASP tabanlı bir sitede ne işinize yarayabilir? Bu sorunun cevabı sizin gereksinimlerinize bağlı. Örneğin siz bir portallık yapmaya soyunduysanız kişiselleştirilebilir bir site iddianız da olabilir. Kullanıcının hangi alanlara ilgi duyduğunu öğrenirsiniz. Bu bir çerez olarak kullanıcı tarafına kaydedilir. Kullanıcı sonraki ziyaretlerinde ilgi alanlarına yönelik haberler, linkler ve bilgiler bulur. Ve sevinir. Çünkü kendini tanıyan, ona değer veren, ilgilerini, tercihlerini göz önünde bulunduran bir site vardır karşısında. "Müşteri memnuniyeti" diye bir şeyi çok sık duyar olduk. Müşteriye insani boyutta değer veren bankalar reklam yaptı. Pamukbank müşterilerinin doğum günlerinde kart gönderdiğini duyurdu. Sizce bunlar zor şeyler miydi? Hayır! İs sadece yaratıcı düşünmekte. ASP size bu fırsatı sonuna kadar veriyor. Çerezlerle neler yapabileceğinizi kendiniz düşünmelisiniz. Biz çok kısır bir örnek verelim, siz kendi özgün uygulamalarınızı geliştirin. Bu örnek, örnek olması nedeniyle basit ve de klasik olacak. Amaç şu: Adam siteye gircek, sitemiz adamın siteye önceden gelip gelmediğini, geldiye kaç kez geldiğini felan tespit edecek. Akıllı bir şey olacak yani. Biz de "Aa.. nerden bildi, bu sitede gelecek var" diyerek sasıracağız.

Sekil1. Kişisel Ziyaret Sayacı

```
<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 görüntülemek hem estetik hem de kullanim açısından dezavantajli oluyor biliyoruz. Bunun için kayitlari, kendi belirledigimiz paketlere bolerek görüntülemek isteriz. Ve de her sayfada ileri-geri navigasyonla diger kayitlara ulasmamizi saglayan linkler olsun.

Evet, simdi bu isi ornegiyle gorecegiz. Kayitlari sayfaya dokmeyi ogrendiyerseniz buradaki anlatilanlar size hiç

karisik gelmeyecek. Çünkü yaptığımız sadece ufak bir ek işlem. Sadece tek dosyamız var: bolunmus_kayitlar.asp . Veritabanı olarak da, önceki makalelerde de kullandığımız, artık EvcilASP ye mal olmuş kisiler.mdb yi kullanacağız. Bu örnek veritabanımızı henüz edinmediyseniz [buyrun](#).

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 veritabanı bağlantıları  
Veri_yolu = Server.MapPath("kisiler.mdb")  
Bcumle = "DRIVER={Microsoft Access Driver  
(* .mdb)};DBQ=" & Veri_yolu  
Set bag =  
Server.CreateObject("ADODB.Connection")  
bag.Open (Bcumle)  
  
' Kayıt seti  
Set kset =  
Server.CreateObject("ADODB.Recordset")  
' recordseti açalım. cursor type özelliği olarak 1  
(adopenkeyset) gönderdik  
kset.Open "Select * FROM kisiler", bag, 1  
' kayıt setinin pagesize özelliğine, bir sayfada  
görünecek maximum  
' kayıt sayısını atıyoruz. biz 3 dedik bakalım  
kset.pagesize = 3  
' bir takım kontroller.. kuralsız geri veya ileri gidilerde  
ilk sayfa görünsün.  
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 olduğumuzu bildirelim..
kset.absolutepage = Session("simdiki_sayfa")

' navigasyon rutinimizi çağıralım.. ' o da bize yon
linklerini hazırlasın.
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 hazırlayan 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...

İlk olarak bir **SELECT...CASE** çok durumlu kontrolümüz ile scriptimizin URL siyle gönderilen yon parametresinin degerini inceliyoruz. Eger "ileri" degeri gönderildiyse **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 çağrılan sayfanin ilk sayfa olması saglanacak bu sekilde.

Sonra alisik olunan veritabani baglantimizi kuruyoruz. Baglanti nesnemizi açiyoruz. Hemen ardından ADO nun Recordsetinden bir adet olusturuyoruz. Kurguladigimiz SQL cümlesiyle recordset içinde herhangi bir kayıt topluluğu olusturuyoruz. Ve sonra zurnanın zirt dedigi yer (ulan bunu kim soyledi be? Sanirim Fizik hocasydi..) gelmekte. Recordset nesnemizin Pagesize adinda bir ozelligini kullaniyoruz. Bu ozellige tek sayfada maximum kaç kayıt görünmesini istiyorsak sayisini atiyoruz.

kset.pagesize = 3

Biz 3 tane kayıt görünsün diye buraya 3 yazdik. Siz 5 tane görünmesini istiyorsanız buraya 3 degil de 5 yazin olur mu? (Ehu EHu.. egleniyorum biraz, kusura bakmayın)

Ardından iki adet denetim yapıyoruz. Adam son sayfadayken "ileri" yönünde istem yaparsa, (daha ilerisi olmayacagina gore hata olusacak) veya ilk sayfadayken "geri" yönünde istem yaparsa (ki yine hata olusacak) sayfa numarasini 1 yapıveriyoruz. Yani ilk sayfadaki kayıtar görünsün diyoruz. Buradaki denetimlerde (IF..THEN) recordset in **pagecount** ozelliginin dondürdüğü degeri kullaniyoruz. kset.pagesize = 3 diyerek bölümledigimiz kayıtarin kaç sayfadan olusacağı kset.pagecount ozelligine sayisal olarak ataniyor. Eger recordsette 8 kayıt varsa, 3 erli dilimlere bolersenek 3 sayfadan olusacak (pagecount 3 degeri dondürecek) demektir. Boylece simdiki sayfa ile toplam sayfa sayisi arasindaki iliskiyeye gore denetimimiz gereken islemi yapıyor.

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 kayıt diliminin ilk kaydina hareket ediyor. Ornegin 3. kayıt sayfasina geldiysek cursor hopppa 3. dilimin en basina gidecek. Her sayfaya maximum 3 kayıt varsa, $(3*2)+1 = 7$. kayda gidecek. Nasil hos degil mi?

Buraya kadar onemli islerin çoğunu yaptık. Simdi sıra kosula uyan kayitlari dokmeye geldi. Ancak önce vatandaşların sayfalar arasında dolmasını sağlamak 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 dongulerle tablolaniyor.

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

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 dondurür.)

Açilan nesneleri kapatıyoruz. Evimize gidiyoruz. KIB.

HARIÇTEN GAZEL OKUMAK

DISARIK DEGIL İÇERİK :)

Merhaba beyler, görüşmeyeli nasilsiniz? Bu yazıda ASP ile gelen #include emrinden bahsedeceğiz. Bu islevi nasıl kullanabiliriz... felan diye biraz düşüneceğiz. Sonra normal insanlar gibi "uyku" moduna gireceğiz. Çünkü biz derslerine "late" kalan talebelerden degilek.. OK

Evet şimdi ASP nin size tanıdığı bir fırsatı soyeleyeceğim: "Dosya Ekleme".. Sasirdiniz degil mi? ASP sayfalarınıza, hariçten dosyalar (tabi fotosop dosyasi neyim anlamadiniz degil mi) eklemenize olanak veriyor. Bunu yeni duymadik, çünkü bir çok program gelistirme ortamında bu islev mevcut. Biz ise ASP deki dosya ekleme yonergesinin kullanimini ve kazandirdiklarini bilmek durumundayiz. Özellikle çok kod yazacağımız büyük islerimizde "inkludasyon" lar bizi çok rahatlatacak.

Dosya ekleme nedir? Önce buna açıklık getirelim. Diyelim bir site yaptik. Gelinlik kızlar gibi oldu. Surdan burdan "animated gif" ler çıkıyor, maus imleci degisiyor, sag tusa basınca küfrediyor, yazılar neyim uçarak geliyor, reklama tiklamayınca girilmiyor.. Nadide.. Yeme de yanında yat. Bu ozenle hazırladığımız "masallah" li sitemizin haliyle birden çok sayfası var. Ve biz istiyoruz ki bu kadar kargasa arasında en altta ufak bir yazı yazsin:

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

Bu yazı her sayfada olsun istiyoruz. Normal HTML sartlarında her sayfanın altına bunu teker teker yazmamız gerekiyor. Ancak yazının bir tarafını degistirmek gerekirse, (örneğin "dinayz" sozcüğünün yanlıs olduğunu gördük, onu "dizayn" yapamaz) ne olacak?

250 tane html dosyasını edit mi edecez? Gidin Allah askına :) ASP biliyoruz ya artık. Çözüm ne peki: Alta yazacağımız metni ayrı bir dosyada tutacağız elbet. Ve bu dosyayı sayfalarımıza "include" (dahil) edeceğiz. Sonra ne olacak? Yapacağımız degisiklik için sadece tek dosyayı muhatap alacağız.

Verdiğim bu örnek çok banaldı biliyorum. Yani gece saat 02:33 olmuş... Benim de hosuma gitmedi bu saatte.. Bu işin en güzel örneği sitelerde navigasyonu (dolasımı) sağlayan menulerdir. Kabul edersiniz ki tüm sayfalarımızda, diğer bölümlere linklerin olduğu standart bir menu olsun isteriz. Böylece yolcu kardesin kafası karismaz ve aradigina daha kolay ulasir. Düşünceli webmimar olmak ne mutlu! Yoksa ne olurdu halimiz? Menulere donelim. Tüm sayfalarda görünmesini istediğimiz menuyu ayrı bir dosyaya koyarız. Gerisini biliyoruz iste. Ancak koyduğunuz düz metin dosyasının uzantısında dikkat edeceğimiz bir husus var. Bu ayrı dosyayı istediğiniz uzantiyla kaydedebilirsiniz. Genelde "include" den gelen .inc uzantisi kullanilir. Ancak bu ve diğer uzantılarla kaydettiginiz dosyalarda ASP kodlarınız varsa "elâleme" açık olacaktır. Farzedin ki database sifrenizi bir harici dosyada sakliyorsunuz. Sifrelerin yanına ne sifreleri olduğunu da çekinmeden yazmissiniz diyelim.

```
<%  
' data.mdb username ve passwordu  
dbuser = "sana_olan_duygulari_bir_bilebilsem"  
dbpass = "belki_severdini"  
>%
```

Dosyanın adını da sifre.inc yaptınız. İnternet te akıllı adam çok. Bu akıllılardan biri sizin sitenize uğradı bir gün... Kardesimiz sifre.inc yazarsa ne olacak. Server "inc" uzantısının üzerinde durma gereksinimi duymayacağından içindeki "çok gizli" seylere bakmadan yollayacaktır. Browser dostumuz büyük bir hevesle download penceresi çıkaracaktır. Ve nihayetinde yukarıdaki kodlarınız 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 !! (bebelere, 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" -->
```

Yukarıda menu var.

```
</body>  
</html>
```

Yukarıda kaynak kodu verilen test.asp yi browserinizde çağırın bakalım. Sonuç:

Kod3. Yorumlanan test.asp

```
<!doctype html public "-//W3C//DTD HTML 4.0  
Transitional//EN">  
<html>  
<head>  
<title> Test Sayfamız </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 bölümü !! (bebeler, girmeyin  
ulan)</a><br>  
<!-- Menu biter -->  
  
Yukarıda 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 tarafında dosya eklemeye olanak verir. Eklenen dosya, eklendiği dosya ile

sunucu tarafında birleştirildikten sonra icra edilir. Sunucu tarafında dosya eklemek için aşağıdaki söz dizimi kullanılır:

```
<!-- #include virtual | file ="dosya_adi" -->
```

* Virtual ve File ekleyeceğiniz dosyanın yol tipini belirtir. Sanal dizinlerle başlayan dosya yolu için virtual'i kullanın.

Örneğin <http://localhost/dosyalar/menu.inc> dosyasını şu şekilde ayrı bir sayfaya dahil edebiliriz:

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

Eğer relative (izafi – görel) bir yol belirtecekseniz file'yi kullanın. File kullandığınızda path, dosya eklediğiniz sayfaya göredir.

Mesela:

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

* ASP sayfanıza aynı dosyayı birden fazla eklemeniz mümkündür. Örneğin önceki sayfadaki basit örnekte menu.inc i sayfanın başında ve sonunda ayrı ayrı ekleyip arada kalan yere de diğer içeriği koyabilirdik. Ancak şöyle 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 ayrıntı da şu: ASP ekleme emri verdığınız dosyaları sayfanıza, diğer komutları icra etmeden önce ekler. Bu yüzden

```
<% dosya_adi = "menu.inc" %>
```

```
<!-- #include file ="<%= dosya_adi %>" -->
```

gibi bir ifade "halt" edecektir. Çünkü ASP sayfanız islenirken önce bu include emrine bakıyor ve dosya adı yerinde `<%= dosya_adi %>` gibi bir ifade görüyor. Haliyle bunu anlamlandıramıyor zavallı. Ve diyor ki: "Active Server Sayfaları error 'ASP 0126' İçerik dosyası bulunamadı Vavavava".

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

Örnek:

```
<% ' hatali kullanım ornegidir
```

```
If menu_olsun = true Then
```

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

```
End if
```

```
%>
```

Yukarıdaki ifade aşağıdaki gibi yazılsaydı halt etmeyecekti:

```
<% If menu_olsun = true Then %>
```

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

```
<% End if %>
```

* Evet, dosya ekleme ayrıntılarını bu şekilde inceledikten sonra biraz farklı bir örnek yapmayı deneyelim. Vira...

BIR MENU ORNEGI

Aşağıdaki kodları inceleyin. ASP ye doyum olmaz. Seyreyle gönüm!

Kod4. menu.inc

```
<% Sub menuler(secim,ayrac) %>
<%
baslik =
Array("Giris","Ozgeçmişim","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?

Yapısal programlamada bir uygulamaya, onu belli kısımlara bölerek yaklaşmak esastır. Bu kısımlara alt program denir. Alt programlar, ana programdan çağırılırlar. Belli bir işi yaparlar, değer isterler veya istemezler. Ana programa (veya başka bir alt programa) değer(ler) getirirler veya getirmezler.

Bu teorik bir şeydi. Biraz gevşeyelim. Alt programlar, yani VBScript dilinde konuşursak Sub rutinler belli işlerimizi yüklediğimiz script parçalarımızdır. Karşımıza bir iş çıktı: Örneğin bir veritabanının yönetimi. Ne gibi operasyonlar olabilir uygulamamızda? Silme, güncelleme, ekleme, arama, listeleme, vs.. Bunların hepsi, temelde aynı database üzerinde bile gerçekleşse birbirinden bağımsız işlerdir. İşte programcı, yani ASPci hayata yapısal dillerde top kusturan arkadaşları gibi baktığından bu olaya da böl, parçala yok et taktiğiyle yaklaşacaktır. Bu bağımsız işleri ayrı ayrı sub rutinlerde çözecek (çözüm kümesini bulacak), sonra da genel çözümü bulmak için bu alt çözümlerin doğrusal bir bağıntısını yazacaktır. Yani BCR-Online.com u hazırlayan Burak Erdem'in dediği gibi, bu alt seviyede iş bitiren elemanlar birleşip Voltran'i oluşturacaklar. Ve büyük güç... Program tamam olacak.

Sub ların bize ne faydası olacak? Dostlar, bu sublar kolumuz kanadımız olacak bizim. Çünkü sub ların – en önemli – özelliklerinden birisi tekrar tekrar çağırılabilir olmalarıdır. Yani belli kod gruplarını birden fazla kullanacağımızı fark edersek bunları bir sub ifadesi içinde tanımlayıp gerektiğinde ana program – veya başka bir alt program – içerisinden çağırmak akıllıcadır.

Alt programların bir takım faziletlerini saymak her programlama muhabbetinde adet haline geldiğinden zikrederim.

1. Belli bir işlemi yapan program parçasının birden çok yerde değişik veriler için tekrar tekrar yazılmasını önler.
2. Düşük düzeydeki ayrıntıları program mantığının ana akisinden uzaklaştırıp ayrı birimlere yerleştirerek okurluğu artırır. (bu önemli!)
3. Program yazma işinin birden fazla programcı tarafından paylaşılmasına olanak sağlar.

Şimdi böyle bir önem arz eden Sub in söz dizimine bakalım:

Kod1. Sub bloğu söz dizimi

```
<%  
Sub prosedürün_ismi (arguman1, arguman2, .. , argumanN)  
' kodlar, kodlar, kodlar...  
End Sub  
>%
```

Evet bu kullanımın basitliğini bakınca “Ulan bizi buraya bunun için mi çağırdın?” dediğinizi duyar gibiyim. Bilin ki bu Sub i incelemeden gitmek yok. Oturun yerinize bakalım.

Simdi bir konuda anlasalim. Bu sub lar çağrildiklari 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  
>%
```

Görüldüğü 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 çağrildiginda gerçeklestirecek. Simdi sadece bekliyor. Peki yazdigimiz sub lari nasil çağiriyoruz? 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 çağirabiliriz. 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 çağirma metodunu tercih edin.

Bu sub yordami çağirma isini istediginiz yerde yapabilirsiniz. Ister ana programdan çağirin, ister ayri bir sub içerisinde çağirin. Ornegin yukarida Kod1 de ornek olarak verdigimiz sub'i çağiran ifadeyi yazalim.

Kod4. Degiskenlerle ilgilenmeyen kod1 deki sub'in çağirilmesi

```
<%  
' baska isler...  
Call yildirim_gursesin_anisina()  
' veya
```

```
yildirim_gursesin_anisina
%>
```

Dikkat edilirse yildirim_gursesin_anisina() isimli prosedürümüzün arguman listesi bos. Boyle parametre istemeyen prosedürleri tanımlarken (ve dahi çağırırken) sonuna () isareti koymayabiliriz. Ancak isin raconu böyleymiş. Sıradan bir değişken adları olmadığı için sonuna koyulan parantezler, âleme karşı fiyakalarını teyit ediyormuş. Sıra geldi parametre agrılarına... müteakip page...

KIMIN DEGISKENI, KIMIN DEGISMEZKENI

Sub rutinleri anlamak demek sub rutinlerin çağrıldığı yerdeki değişkenlerle olan ilişkisini, muhabbetini anlamak demektir bir bakıma. Öncelikle su teoremi verelim:

"Bir sub prosedür, ana programdaki bütün değişkenlere direkt müdahale edebilir."

Bu teorem âlemde Euler'in Evrensel Prosedür Yerine Koyma Teoremi olarak bilinir. Oldükten sonra, yastığının altında sakladığı günlük seklindeki ajandasında bulunmuştur. (Aslında bundan yıllar önce bu teorem Anadolu'da halk arasında bilinmekteydi. Ancak Batılı kaynaklar her zaman yaptıkları gibi bizden bisikletimizi çaldılar, başka renge boyadılar ve mahallede gözümüzün önünde bizim bisikletimizle bize hava yaptılar. Ağlamak istiyorum.)

Neyse biz ASP'ye donelim. Yukarıdaki teoremin ispatını verelim de gönüller hoş kalsın.

Kod5. "Bir sub, ana programdaki değişkenlere müdahale eder" in ispatı

```
<%
' hadi kurban diye bir degisken tanımlayalım. 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 yazdır
response.write "kurban: " & kurban & "<br>"

' Prosedürümüzü de yazalım...
Sub kral_ciplak()
```



```
' kurban adli degiskene darbe yapalim
kurban = 15
End Sub
%>
```

Saydadaki Çikti:

```
kurban: 5
kurban: 15
```

Prosedürümüzün ana programın neresinde tanımlandığının bir önemi yok. Burada ben en sona `kral_ciplak()` adını verdiğim prosedürü yazdım. Ve yukarıda, yani ana program akışında önce `kurban` adlı değişkenimize 5 değerini atıyoruz. Ana programda `kurban` adlı bir değişken var ve de değeri 5; bunu ekrana yazdırdık. Sonra `call kral_ciplak()` diyerek prosedürümüzü işletiyoruz. Programın isleyisi prosedür içine atlıyor. `kral_ciplak()` subunda `kurban` adlı değişkene 15 değerini verdik. Tahmin ettiğiniz gibi aynı anda ana programdaki `kurban` adlı değişkenin de içeriği değişti. `kurban`'i tekrar yazdırsak değişmiş sonucu göreceğiz. Bu demek oluyor ki, bir sub yordam, ana programda çağrıldığı yere kadar kullanılan tüm değişkenlere müdahale edebiliyor. Çağrıldığı yerden sonrakileri zaten bilmiyor. Ancak bir sub, başka bir sub in içinden çağrıldığında, onun içindeki değişkenlere müdahale edemiyor. Aşağıdaki örneği deneyin, sonucu temasa eyleyin:

Kod6. Sub in bir diğer sub dan çağrılması

```
<%
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
```

İkinci bir teorem verelim: "Ana program, bir sub içinde değerine müdahale edilen değişkeni onu çağırmadan

once kullanmamissa, o degiskenin sub tarafından degistirilen degerini de bilmez."

Ispat:

Kod7. "Sub in ozge degiskenini ana program tanimaz..." in ispatı

```
<%  
Call afsin_termik_santrali()  
Response.Write "k: " & k & "<br>"  
' ~~~~  
Sub afsin_termik_santrali()  
k = 25  
Response.Write "k: " & k & "<br>"  
End Sub  
>%
```

Saydadaki Çıktı:

```
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 olmalı. Yani olması 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 düsiyorsa adından sonra parametre listesine bir sey yazmiyorduk. Ancak artik bir sub rutinle daha profesyonel sekilde iletisim kurmanın vakti gelmedi mi? Evet, geldi..

Sub rutinin ana programdaki – tabi çağrildiği 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 önemli yanlisliklara sebebiyet verecektir. Bunu kokten çözenin, yani programin sagligini dalginliga kurban etmeyi onlemenin yolu degiskenleri Dim ile deklare edip sayfaların basına da üsenmeyip <% OPTION EXPLICIT %> yazarak bir degiskenin script içerisinde birden fazla deklare edilmesinin önüne geçmektir. Bu etkili

çözümdür. Artık seçim size kalmis: ben dikkatliyim, iyi araba sürerim.. felan diyorsanız sozüüm yok. Ama yanlis yapabilme ihtimali iliklerinizde soguk rüzgarlar estiriyorsa (ho! Bu ne demek?) belli metodu kullanabilirsiniz. Ama sunu belirtmeden geçmek istemiyorum: Ne kadar dikkatli olsanız bile uzun kodlarda kafanız allak bullak oluyor. Tecrübeyle sabittir vesselam.

Simdi gerçeğe yaklasalım. Bir sub rutinin ana programla deger alisverisi yapabilmesi için sub..end sub blogundaki isim tanımlamasından hemen sonra parantez içinde kullanacağımız degiskenleri yazıyoruz. Sub sub_ismi (degisken1, degisken2) gibisinden. Biraz teori daha. Prosedür (sub) ile çağrıldığı yer arasındaki deger iletisini tek yönlü veya iki yönlü olabilir. Bu yüzden parantez içindeki parametreleri iki gruba ayırılır: 1. formal parametre, 2. degisken parametre. . Bir prosedür çağrıldığı yerden bir degiskenin ya degerini alır (value), ya da kaynagini alır (reference). Degerini alırsa – ki bunlar formal parametrelerdir – bu degiskenin içerigini degistiremez. Yani önceki sayfada Euler teoreminin ispatında olduğu gibi ana programdaki degiskene müdahale edemez. Ancak parametre olarak o degiskenin kaynagini alırsa – bunlar da degisken parametrelerdir – o degiskenin hem içerigini kullanır, hem de içerigine müdahale eder. Bir sub tanımlarken hangi parametrenin formal parametre, hangisinin degisken parametre olacağını önüne yazdığımız bir extra deyimle belirliyoruz. Asagidaki örnek dediklerimizi netlestirecek:

Kod8. Parametre cinsleri

```
<%
KazanToAnkara = "40 dk."
KhamamToAnkara = "60 dk."
AyasToAnkara = "unuttum"
GolbasiToAnkara = "10 dk."
' ~~~~~
Response.Write "Sub in icrasindan önce...<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
görünen 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 = "yazın 100 bin TL idi"
khamam = "tam 1 milyon, yarım yani öğrenci 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.

```

Yukarıdaki örneğin kodları çok açık. Ancak bazı yerlerin önemine değinelim. Burada üzerinde durulması gereken yer, sub isminden sonra parantez içine virgüllerle siraya konulup yazılan parametrelerdir. İlk parametrenin (kazan) önünde hiçbir ifade yoktur. Önünde nasıl bir parametre olduğu belirtilmeyen tüm parametreler varsayılan olarak değişken parametredir. Yani önüne byref yazmakla yazmamak aynı şeydir. İkinci parametre olan khamam in önünde byref var. Kazan ve khamam parametreleri değişken parametrelerdir. Prosedüre bu parametrelerle değişkenlerin hafızadaki adresleri geçirilir (byref). Üçüncü ve dördüncü parametrelerin (ayas, golbasi) önüne ise byval deyiimi eklenmiştir. Böylelikle prosedür içine bu parametrelerle sadece tek yönlü olarak değer gelmiş olacaktır. Bir formal parametre olan degiskene prosedür içinde atadığımız değer, sadece prosedür içinde geçerli olacaktır. Ana programdaki değişken bu değişiklikten etkilenmeyecektir. Zira prosedür bu değişkenin sadece içerdiği değeri kullanacak şekilde parametre tanımlamasına sahiptir.

Umarım yeterince açık bir şekilde anlatmışımdır bu mevzuyu. Ha bir de: eğer prosedürünüz ana programdaki bir değişkenin değerini kullanacaksa, ve bu değişkenin değerinin hatayla da olsa değişmesini 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 çağrilmissa o satirdan sonraki kodlari icra etmeye koyulur. Buna ihtiyacimiz 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 tarafındaki bileşenlerden bahsediyoruz. Su bileşen olmalı bunu yapmak için filan... ASP sitelerinde bin türlü component görüyoruz. Hepsi kendi çaplarında işler yapıyorlar. Kimisi mail gönderiyor, kimisi dosya transfer ediyor, kimisi dinamik gif dosyaları, grafikler üretiyor, vs.. Yani hepsi ASP scriptlerimizde yapamadığımız kompleks işleri halletmemiz için üretilmiş, çoğunluğu bedelli üçüncü parti sunucu bileşenleri.

Peki biz kendimiz programcı kimliğimizi on plana alıp, bir an için ASPci olduğumuzu unutup bu sunucu bileşenlerinden yazabilir miyiz? Evet dostlar yazabiliriz. Elin adamındaki programlar bizde de olduğu sürece yazabiliriz nitekim.

Peki diyelim yazdık bir bileşen. Ne isimize yarayacak yazdığımız bu bileşen? Bize getirisi ne olacak?

Oncelikle su temel bilgileri aklımızdan çıkarmamalıyız: ASP bir sunucu taraflı script teknolojisidir. Yani ASP ler derlenmeden, düz yazı dosyalarında tutulur. Ve istemci tarafından çağrıldığında; ASP den haberi olan bir Web Server yazılımı ile script kodları işlenir ve sonucu isteyen adama yollanır. Ama bir Activex DLL dedğimiz zaman bunun apayrı bir boyutta olduğunu farkederiz. Çünkü aile terbiyesine göre EXE ve DLL gibi dosyalar, derlenmiş dosyalardır. Yani "program" lafından kastedilen onlardır esasında. Bir derleyici tarafından (MS Visual Basic, Borland Delphi, Visual C++, vs..) makine kodlarına çevrilmiş programlardır. DLL ler de EXE lerden farklı olarak yürütülmeyen derlenmiş kütüphane dosyalarıdır.

Peki DLL lerin avantajı nedir? DLL ler bir kere çağrıldıktan sonra hafızada kalırlar. Yani oturur ve gitmezler. Bunun için sistem kaynaklarında bağdas kurup gernesin bir DLL ye tekrar ihtiyaç duyulduğunda olay yerine müdahale etmesi daha kolay olur. Yani buna hız kazanmak diyorlar. Ne kadar hız kazandığımızı söyleyemeyeceğim maalesef. ...'in her ay "Nasil Test Ettik?" diye sakal tırası yapan İtalyan kilikli adamlarına söyleyelim de bir ara bunu da test etsinler.

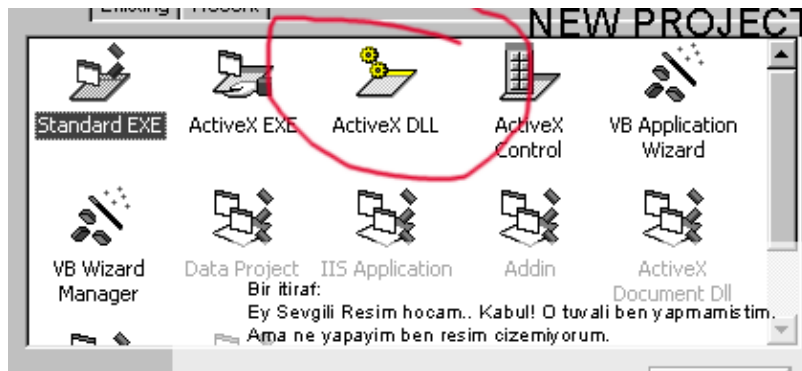
Ve hazırlayacağımız sunucu DLL lerinin bir yararı daha var. Yazdığınız kodlar derlenmiş olduğundan "saklı" kalacaktır. Bir işi ASP scriptleriyle yaptınız diyelim. Bilirsiniz ki ASP ler düz yazı dosyaları... Ve dünyada text editörü de bulmak kolay. Yani özene bezene yazdığınız kodların saklılığından söz etmeniz mümkün olamıyor. "yaw ASP bir Server-side seysi.. kodlar görünmez client in makinada.. " diye itiraza rağmen sunucu tarafında kodlarınız açıktır. Yani bir DLL de iş yapmak kodlarınızın mahremiyetini sağlayacaktır.

Peki bu Activex DLL leri neylen yaparız, oluşturunuz? Her sabah ugradığınız ekmeğe için bakkaldan yarın bir de Visual Basic benzeri son nesil bir derleyici isteyin. Eger "bende o dediginden yok" diye diretiyorsa ısrar etmeden bakkaldan çıkın. Muhtemelen bir gün o da satmaya başlayacaktır. Neyse, elinize bir tane derleyici geçirmeniz lazım. Delphi, Visual Basic, Visual C++, Borland C++, felan filan.. Activex DLL yaratabilen şeylerden bulun bir tane iste :) Biz burda örnek olarak (çok kullanıldığı için) Visual Basic ile yaratacağımız DLL mizi.

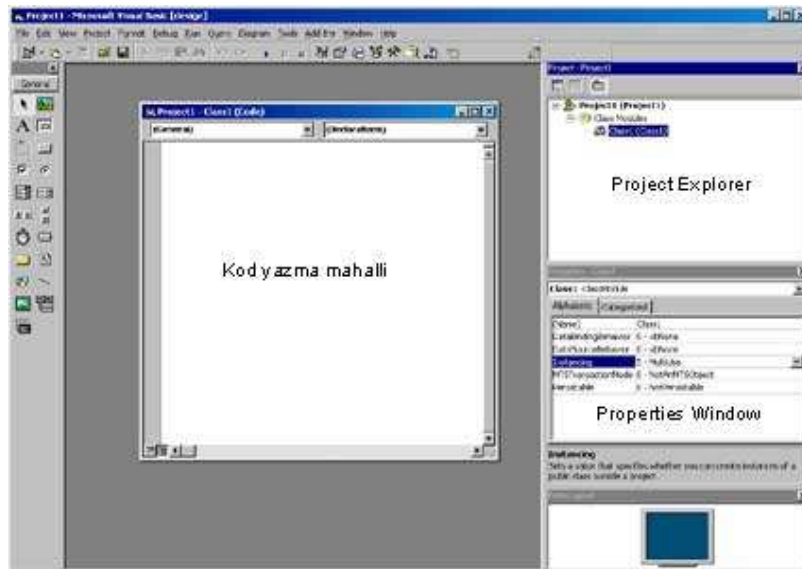
Peki DLL mizi oluşturduk diyelim. Bunu kendi bilgisayarımızda sinayabildik ama sistemizi sundurduğumuz hostes sunucumuzda nasıl çalıştıracamız? Valla orasını benimle değil de sunucunuzun serifiyle temasa geçerek halledersiniz. Biz burda bu DLL işini de evcilleştirmeye çalışacağız. Yani sırf zevk için. Bana eğlence değil, reel yaklaşımlar lazım dersiniz yazıyı okumanıza gerek yok. Ama eğer bir Activex DLL oluşturup, kahve yudumlayarak saf eylemeyi yeglerseniz buyurun. Fazladan bilgi göz çıkarmaz hem değil mi dostlar?

BASLIYORUZ...

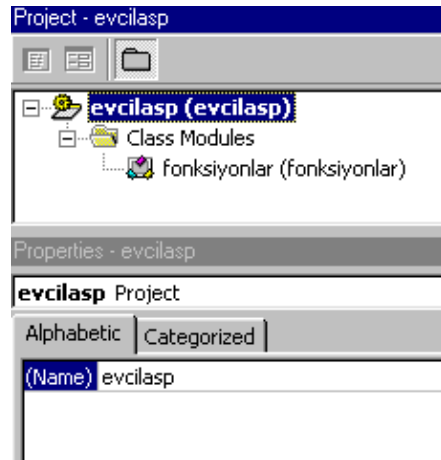
Evet, baslamak için Microsoft Visual Basic 5 veya yukarisindaki sürümü Baslat > Programlar > felan filan diyerek açin (saka yaw). Programi açınca artik her programin vazgeçilmezi olan çok seçenekli New/Open Project ekranı selamlayacak sizi. Sanki programi her zaman yeni bir sey yazmak için açiyoruz! Ya ben yeni bir program yapmak veya yaptigimi düzenlemek yerine programin about undaki yumurtalari seyredeceksem? Adami uyuz ediyorlar dostlar:) (Iyi ki altında 'Don't Show bu gibi muhabbetleri' seçenegi var)



Karsimiza INTEGRATED DEVELOPMENT ENVIRONMENT diye adlandırılan bir ortam çıkıcak (Burası ScottJr ve Stalker içindi..). Bu "alisilmadik" ortamda bize yardımcı olacak sagdan soldan boy gösteren edevati tanimaniz için size süre veriyorum :) Isiniz bitince müteakip sayfada devam edek.



PROJE...



Önce projemize bir ad verelim. Bunun için önce sağ tarafta üstte duran Project başlıklı pencerede Project1'i seçiyoruz. Ve bu anda altta properties penceresinde Project1'e ait özelliklerin listesi görüntüleniyor. Burada sadece tek özellik olarak gözümüze çarpan Name özelliğini hemen "evcilasp" diye değiştiriyoruz. Lütfen siz de "evcilasp" diye değiştirin. İlla kendi sitenizin adını yazacak değilsiniz ya. Yarın da sizin sitenizin adını yazarsınız :)

Project Explorer'e dönelim. Yani oraya odaklanalım. Vusal Basic bizim için hazır bir Class Module koymuş oraya. Adına da class1 demiş. Biz bu öğeye tıklayıp properties penceresinden name özelliğini (artık alistik buna değil mi?) 'fonksiyonlar' yapalım.

Evet... Bu kadar merasimden sonra artık kod yazmaya başlayalım ha? Eger bir kod penceresi görünür değilse, Project Explorer'deki fonksiyonlar modülü üzerine sağ tıklayıp View Code deyiniz. Başlığında 'evcilasp-fonksiyonlar (code)' yazan bir pencere peyda olacak. Kod mahalline aşağıdaki kodları yazınız dostlar.

```
Public Sub PanoyaYaz(str As String)
Clipboard.SetText (str)
End Sub

Public Sub PanodanYaz(response)
response.write Clipboard.GetText
End Sub
```

Evet, haklısınız, tek satırlık çok basit prosedürler yazdık. Ama burada görmek istediğimiz olayı örneklemesi açısından böyle oldu. Şimdi dikkat ederseniz (ben VBASIC bilmem ama anlıyorum) VBASIC bilmeyen ASP cillerin anlayacağı basitlikte bir iş var burada. PanoyaYaz isimli ilk sub imiz dışarıdan bir stringle çağrılıyor, ve sub'imiz sunucu sistemin Clipboard'ına (pano) gönderilen stringi koyuyor. İkinci sub'imiz ise dışarıdan bir nesne istiyor. Bu da ASP object modelinin iyi tanıdığımız delikanlılarından "Response". Burada response yerine başka bir şey de yazabiliriz. Ama boşverin. Güllü dalında güzel. Sub'imizde ise panodan aldığımız metni response.write metoduna gönderiyoruz. Ve panodaki metnin ASP sayfamıza yazılmasını sağlıyoruz.

Şimdi bu projemizi kaydedelim de bir yerde dursun. File menüsünden her zaman yaptığınız kaydetme işlemlerine bezner bir işlemle bu kaynak dosyalarımızı saklayalım. Proje ve sınıf isimleriyle sakladığımız kaynak dosyalar aynı ismi taşımayabilir. Ama ben uğrasmadan aynı isimlerle kaydettim. Size bu konuda özgürlük tanıyorum. İstedığınız isimle kayıtlı olanlarınız var.

Sonunda can alıcı noktaya geldik. Hazırladığımız bu projenin neticesi bir DLL olmalı değil mi? Sıra bu DLL'i oluşturmaya geldi. Hemen vakit kaybedip de bu eğlenceyi sekteye ugratmadan File menüsünde "Make evcilasp.dll?" yazını bulalım. Ne güzel değil mi? Adam size DLL ni yapayım mı diye soruyor.. Çok

yardımsız insanlar bu programcılar :) Biz de uzatılan bu dali tutuyoruz. Karsımıza bir pencere çıkıyor. Dll mizi ne adla kaydedeceğimizi sorduğu her halinden belli. Kendisi de evcilasp.dll diye tercihini sunmuş. Biz ona uymadan adını ilkdeneme.dll yapalım orayı. Hani projemize evcilasp ismini vermiştik. İkisinin de farklı olabilirliğini göstermiş olalım böylece. Sonra OK tuşuyla bir devrim yaşamaya başlıyoruz. Visual Basic sizin için projenizi derlemeye başlıyor.. Bir DLL dogacak.. Sabir.. Taslar yerine oturdu.. Sonra bu DLL yi gidip sizin yerinize Windows Registry veritabanına da kaydediyor.

Artık bir sunucu bileseniniz var.. Nasıl bir duygu?

Peki her şey buraya kadar mıydı diye soracak olursanız Hayir! Sıra geldi bu bilesenimizi (yani bize ait olan bileseni) ve sağladığı metodları bir asp scriptinde keyifle kullanmaya. Hemen sevdiğimiz veya hayran olduğumuz bir text editorü açıyoruz. Ve aşağıdaki kodları kopyeliyoruz:

DENEME.ASP KOD MANZARASI

```
<%  
Set BenimNesne = server.Createobject("evcilasp.fonksiyonlar")  
BenimNesne.PanoyaYaz("Bu metin sunucu bilgisayarın  
panosundan gelmektedir.")  
BenimNesne.PanodanYaz(response)  
>%
```

Ve her server-side scripting teknolojisi bağımlısı gibi hemen kaydettiğimiz bu scripti localhost tan çağırmaya koyuyoruz. Ve manzarayı temasa ediyoruz.

Adres	http://mucit/deneme.asp
-------	-------------------------

Bu metin sunucu bilgisayarın panosundan gelmektedir.

Evet, arkanıza yaslanın şimdi. "Elin adamı yapmış" değil mi dostlar? Bu demek oluyor ki ASP bize sunucuyla sınırsız etkileşim alanı sunuyor. Yukarıdaki çok basit uygulama bunun özeti değil mi? Evet, ASP nin sınırlı öz yeteneğini sınırsıza dönüştüren onun sunucu tarafındaki bileşenlerle iş yapabilmesidir dostlar. Ve bu kâpi sonuna kadar açıktır. Ve o yüzden dünyada bir sürü üçüncü parti bileşen üretilip satılan yazılım ekipleri türemiş. Tabii Microsoft da geri duymuyor. Kendi amacına yönelik komponentler çıkarıyor. Eğer ASP ile bir şeyi yapamadığınızdan şikayetçiyseniz sizin işinizi halledebilecek bir component elbette bir yerde bekliyordur sizi. (Müşteri bekleyen bir çok sunucu componentlerine ulaşmak için [Aspin](#) sitesi idealdir.)

"Zeki Müren de bizi görecektir mi?" diye kaygılanmayın. Samimi duygularla iyi işler yapanı herkes görür, herkes duyar..

DERLERKEN...

Bu isin gıcık tarafından da bahsedelim biraz. Hani dll dosyalarının bir kez çağrıldıktan sonra hafızada kaldığını, geri gitmediğini soyledik. Az önce deneme.asp dosyamızı çalıştırdığımızda farkında olmadan ilkdeneme.dll kütüphanesini de sistem belleğine yüklemiş olduk. Bu dll yi daha sonraki çağrılarınızda işlemler daha hızlı yürüyecektir haliyle.

Ancaaaak! Siz bir bileşen geliştiriyorsunuz ve dll yi tekrar derlemeniz gerekiyor. Örneğin bir önceki sayfadaki örneği biraz kurcalayıp tekrar derlemek istediniz. Ve acı manzara.. "Permission denied" denen yürek uçuklatıcı hata mesajını alıyorsunuz.. Niye? Hani bir vecize vardı hatırladınız mı? "Windows kullandığı dll yi elletmeee...". Yani ilkdeneme.dll dosyası hafızada yüklü ise üstüne yazmanız mümkün değildir. E.. peki bunun çaresi var mı? İlkdenem.dll yi hafızadan kovmanın bir yolu var mı?

Var.. Hem de çok radikal!: Windows'un reboot edilmesi... Bu Win95/98-PWS sistemiyle çalışanlar için tek çözüm dostlar. Yani bunun dışında bu dll yi hafızadan kovamıyorlar.

Ama NT gençliği sansli. Zira onların bir IIS i var. Internet Information Server size PWS deki baslat-durdur seçeneğinden çok daha geniş servis yönetimi sunuyor. Hemen IIS yönetim (internet services manager) programınızı açıyorsunuz. Ve ağaçta gördüğünüz bilgisayar ismine sağ tıklayıyorsunuz. Çıkan menüden "Restart IIS..." gibisinden bir yazı var, ona tıklayınca karşınıza çeşitli seçenekler içeren bir ekran çıkıyor. Burada Restart Internet Information Services demeniz yeterlidir. IIS yeniden başladığında dll nizi derleyip tekrar deneyebilirsiniz.

Bunun yerine komut satırı da işinizi görebilir. Start > Run diyip "cmd" yazın. Açılan kara ekrandaki komut satırına

```
net stop iisadmin
```

yazın. Yoneltile soruyu Y (yes) diye yanıtlayın. Size orayı burayı durdurdum diye verilen bilgilerden sonra komut satırına şimdi de sunu yazın:

```
net start w3svc
```

Şimdi dll nizi yeniden derleyebilirsiniz.

KAYIT...

Bir yerden edindiğiniz derlenmiş bir dll yi sisteminize kayıt etmek için (register) regsvr32.exe programını kullanabilirsiniz. Bu program Windows'un system dizininde bulunmakta. Komut satırından bu dizine gelip, şöyle bir komut verebilirsiniz:

```
regsvr32.exe C:\belgelerim\ilkdeneme.dll
```

Bu komutla derlediğiniz bir dll yi, başka bir bilgisayara kayıt edebilirsiniz.

Sayet kayıt olunmamış bir nesne örneği yaratmaya çalışırsanız şu meşhur ASP hatasıyla bulursunuz:

```
Sunucu nesnesi error 'ASP 0177 : 800401f3'
```

```
Server.CreateObject Başarısız
```

```
/xxx.asp, line 2
```

```
800401f3
```

Evet, bu mevzuya burada ara verelim. Bir gün bu makalenin devamını da yazıp veritabanı işlerini de bileşenlere yükleyip performans kazanmayı işleriz. 0555 hepinize hayırlı olsun :)

**** Bu makale ile ilgili dosyaları [buradan](#) alabilirsiniz.