ASP.NET DERS NOTLARI

ASP.NET'İN KURULUMU

Asp.Net sayfalarını çalıştırabilmek için şunlara ihtiyaç vardır.

- a) .Net Framework (Microsoft işletim sistemini net uyumlu yapmak için kullanılır)
- b) En az IIS 5.X yüklü olmalı (İnternet Information Server)
- c) Minimum Pentium 133 Mhz., 128 MB RAM
- d) Windows 2000 ve üzeri.

.Net Framework'u şu adreslerden ücretsiz temin edebilirsiniz.

http://msdn.microsoft.com/netframework http://www.asp.net/download.aspx

.Net Framework'un iki dağıtım sürümü vardır. Gerekli bileşenlerin bulunduğu 21 MB lık versiyonu (Redistributable) ve hertürlü yardımcı dökümanların ve örnek uygulamaların bulunduğu SDK (Software Development Kit) sürümü bulunmaktadır.

ASP.NET SAYFALARININ YAPISI VE ÇALIŞTIRILMASI

Asp.Net sayfaları sunucuda çalışır. Html dosyalarında olduğu gibi bunları browser'ın adres çubuğundan dosyanın fiziksel yolunu kullanarak çalıştıramayız. Yazdığımız dosyayı sunucaya koymalıyız ve bize bu sayfayı sunu işleyip göndermelidir.

Sunucunun çalıştığı yer şöyledir (fiziksel yol).

C:\Inetpub\wwwroot

IIS kurulduğunda uygulamaların çalıştırılacağı bir dizin oluşturur. Bu dizine atılan sayfalar sunucuda çalıştırılır ve işlenir.

Ancak browser'dan bu yolu tanımlayarak çağırmak uygulamayı çalıştırmaz. Sadece text ve html kodları gözükür.

Bu nedenle önce sunucunun adını yazmalıyız ve böylece önce sunucu çalışır. Daha sonra bunun yanına sunucunun içinde dosyalarımızın bulunduğu yeri belirtirek yazmalıyız.

Peki sunucunun adı nedir? Sunucunun adı bilgisayarın adıyla aynıdır. Bilgisayarın adını windows'u kurarken belirtiriz. Bilgisayarın adını öğrenmek için "Masüstünde>Bilgisayarım>Özellikler>Bilgisayarın Adı" yolu takip edilebilir.

Eğer bilgisayarın adını bilmiyorsanız "Localhost" ifadesini yada bilgisayar tanımlı IP (127.0.0.1) numarasını kullanabilirsiniz.

Bundan sonra browserda adres çubuğuna şunları yazabiliriz.

http://bilgisayarım/deneme.aspx http://localhost/deneme.aspx http://127.0.0.1/deneme.aspx

Buradaki sistem aynı internette sayfa çağırır gibi olmaktadır. İnternette de bir sitenin ismini yazarız. Fakat o sitenin dosyalarının serverda hangi yolda bulunduğunu bilemeyiz. İnternette de bir sitenin adını yazmak yerine o sitenin IP adresini de yazabiliriz.

Bilgisayarımız sunucuyu kurarak Asp.Net sayfalarımızı tıpkı internet ortamında çalıştırıyor muşuz gibi denemiş oluyoruz.

Şimdi bir örnek Asp.Net sayfasını NotPad'de yazıp sunucunun içine (C:\Inetpub\wwwroot) Örnek.aspx

olarak kaydedelim.

Örnek: Şimdi bir örnek Asp.Net sayfasını NotPad'de yazıp sunucunun içine (C:\Inetpub\wwwroot) deneme.aspx olarak kaydedelim

```
→ Go
                                                 -
Address 😸 http://localhost/deneme.aspx
Bu yazıyı görebiliyorsanız, ASP.NET ile çalışmaya hazırsınız!
      <%@ Page Language="VB" codePage="28599"%>
      <script runat="server">
             sub Page Load(obj as object, e as eventargs)
             Mesaj.text="Bu yazıyı görebiliyorsanız, ASP.NET ile çalışmaya hazırsınız!"
             End sub
      </script>
      <html>
       <head>
       <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
       </head>
      <body>
             <asp:Label id="Mesaj" runat="server" />
      </bodv>
      </html>
```

ASP.NET'E GİRİŞ

ASP.NET Uygulama Yapısı

Bir web uygulaması internet üzerinden çalışır. Bu durum uygulama geliştiren kimselere ek maliyetler çıkarmaktadır. Bir Windows uygulaması geliştirmek ile web uygulaması geliştirmek aynı şey değildir. ASP.NET gibi teknolojiler ve bu teknoloji alt yapısına arabirimlik yapan Visual Developer 2005 Express veya Visual Studio 2005 gibi uygulama geliştirme editörleri bu engelleri ortadan kaldırmaya yöneliktir.

Hangi Programlar yüklü olacak ve hangi Editörü Kullanalım

Öncelikle Mikrosoft'un yeni araçları olan şu programları kullanılacaktır.

1- ASP.NET 2 (indirilen bir şey degildir)

2- .NET Framework 2 (Asp.net kodlarının çalışabilmesi için gereklidir. 23 MB lık temel çalışan kodlar yada 334 MB lık tüm help ve yardımcı dosyaların bulunduğu versiyonu indirebilirsiniz)

3- Visual Studio 2005 (Ücretlidir. Yerine "Visual Web Developer 2005 Express" indirilip kullanılabilir. . Bu editör ücretsiz, akıllı kod tamamlayabilen, bütün kontrolleri bir arada sunan, görsel işlem yapma ve sihirbazları ile mükemmel bir araçtır. Bunun için IIS'in (Internet Information Server) kurulu olmasına gerek yoktur. Ama IIS dünyaya açılan bir server da bulunması gerekir. Ayrıca böyle gelişmiş bir editör yerine not defteri gibi basit bir editör kullanırsanız sayfaları denemek için yine IIS gereklidir)

4- SQL Server 2005 (MS tun büyük uygulamalar için kullanılan veritabanıdır. Yerine küçük uygulamalar için MS Access (ofisin içinde) kullanabilirsiniz.)

Uygulamanın Bileşenleri

ASP.NET deki sayfa uzantısı .aspx'tir. Bu sayfalar sunucuda tutulur ve bağlantılı olduğu kodlar çerçevesinde sunucuda çalıştırılarak bir HTML belgesine (yalnız uzantısı aspx dir) çevrilerek, browser'a gönderilir. Browser'a gelen sayfaların içeriği tamamen HTML ve client (istemci) taraflı javascript kodlardır. Ancak browser'da görünen dosya_ismi.aspx şeklindedir.

FILE→New→project→ASP .NET Web Application seçip yeni bir uygulama başlatıyoruz.

Bir proje oluşturduğunuzda Solution Express penceresinden şu bilgileri görürüsünüz.



Burada App_Data klasörü veritabanın klasörü, Default.aspx ana aspx sayfası, Default.aspx.vb ise bu sayfaya ait programlama kodlarının yazıldığı sayfa, ve Web.config dosyası ise sitenin başlangıç ayarlarının yapıldığı dosyadır.

Sayfanın sol attan hem source/design kısımlarına geçebilirsiniz. Sürükle bırak ile ToolBox'dan kontrolleri alıp sayfamıza ekleyebiliriz. Örneğin sayfamıza eklediğimiz bir textbox nesnesi

Design bölümünde

```
şeklinde
```

Source bölümünde

```
<asp:textbox id="TextBox1"
style="Z-INDEX: 101; LEFT: 80px; POSITION: absolute; TOP: 48px"
runat="server" Width="120px">
</asp:textbox>
```

ifadesi ile konulmuştur.

Uygulamaların Kod Yapısı

ASP .NET 2'de kodları yazabileceğimiz iki farklı yer bulunur. Bunlardan biri sayfanın kendisidir.

Yani aspx uzantılı kontrollerin ve diğer tagların bulundu sayfamızın kendisidir.

Diğer bir seçeek ise sayfadan ayrı bir dosyadadır. bu dosya özel ifadelerle aspx sayfamıza bağlanır. (codebehind ile)

Uygulamaların kod yapısını daha iyi anlamak için bir örnek yapalım. Örneğimiz butona tıkladığımız da sayfada mesaj gösteren basit bir uygulama. Toolbox bölümünden sayfamıza birer tane label, textbox ve buton nesnelerini sürükleyip bırakalım. Kullanıcıya görünen arabirim tamamlanmış oldu.

Kod kısmında ise <script runat="server"></script>tagları arasına yazmak istediğimiz kodları yazarız. Sayfamızı F5 veya CTRL+F5 yardımıyla çalıştırırız.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
Label1.Text = "merhaba" + TextBox1.Text
End Sub
```

Çalıştırdığımızda ekran görünümü;

```
ASP 2 Merhaba ASP 2
Button
```

Şeklinde olacaktır. ASP .NET 2'de ilk sürümlerinden farklı olarak akıllı kod algılayıcıları bulunmaktadır. Böylelikle binlerce ifade, nesne, özellik ismini arama zorluğundan kurtulunur.



Kodları Ayrı bir Dosyada Yazmak

Sayfamızın kodları ayrı bir dosyada, özel bir uzantı ile saklanır. Bu uzantı.aspx.vb'dir.

Aslında bu dosya *Namespace* ve *class*'ları ihtiva eden geleneksel bir sınıf dosyasıdır. .aspx uzantılı dosyamızla bu sayfadaki CodeFile="WebForm1.aspx.vb" kodları ile sayfamıza bağlanır. WebForm1.aspx.vb sayfasında nesnelere olay ekleme için sol üstteki dropdownlistten olayları seçebiliriz. Buton seçiliyken buradan click olayını seçtiğimizde

Şu yordam kod sayfamıza otomatik olarak eklenir.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click()
End Sub
```

button1 in kontrolüne onclick ile HTML kod sayfasında olay tanımı yapabiliriz.

<asp:button id="Button1" style="Z-INDEX: 102; LEFT: 104px; POSITION: absolute; TOP:

104px" onclick="buton" runat="server" Text="Button"> </asp:button>

Aspx.vb sayfasında gerekli kodları aşağıdaki gibi değiştiririz.

```
Sub buton(ByVal sender As System.Object, ByVal e As System.EventArgs)
Label1.Text = "merhaba" + TextBox1.Text
End Sub
```

Handles ifadesi kullanılmamıştır buna rağmen olayımız çalışır.

ASP.NET

.NET 'e giriş, Neden ASP.NET?

ASP.NET kısaca Microsoft'un Haziran ayı itibari ile lanse edilen yeni sunucu taraflı web teknolojisidir. Microsoft'un yeni gözbebeği olan .NET alt yapısı çerçevesinde, bir çok açıdan yetersiz kalmaya başlamış ve tabiri caizse son kullanım tarihi geçmiş ASP 'nin yerini alacak olan ASP.NET, web programcılığı tarihinde çığır açacak yeniliklerle karşımıza çıkmakta. İlk olarak ASP+ olarak açıklanan bu yeni teknoloji, Microsoft'un sunucu taraflı internet teknolojilerini de .NET platformu altında toplama kararından sonra ASP.NET adını aldı.

Neden ASP den vazgeçip yeni bir dil öğrenmeliyim diyorsanız cevabı hazır. Öncelikle ASP nin neden terkedilmeye mahkum olduğunu anlayalım.

1. Bildiğiniz üzere ASP (Active Server Pages), uzantıları ".asp" olan dosyaların asp.dll adlı bir ISAPI yorumlayıcısı tarafından yorumlanması ve kodda yer alan işler yapıldıktan sonra sonucun istemciye gönderilmesi esasına dayanıyordu ki bu sunucunun performansını ciddi şekilde etkiliyordu. Her isteğin ardından tekrar yorumlanan kodlar aynı anda yüzlerce sitenin barındığı sunuculara ciddi yük getirmekteydi.

2. ASP ile yazdığınız kodlar her ne kadar site ziyaretçileri için görüntülenemez olsalarda, intranet uygulamaları geliştirenler aylarca uğraşıp geliştirdikleri programlarını herhangi bir ortama kurduklarında ciddi bir tedirginlik içine girerlerdi. Neden, kodlar ortalıkta da ondan.

3. Yarı nesne tabanlı bir programlama dili olan ASP, sadece temel öğeleri ve işlevleri içeren bir nesne ortamı sağlıyordu. Sunucuda kurulu olan COM objelerine erişerek veya kendi objelerinizi yazarak (sunucuya kaydetmek zorundaydınız) kendinize nispeten geniş bir hareket ortamı yaratabiliyordunuz. Ancak ASP'nin özgün nesne topluluğu bir çok sorunun çözülmesinde yetersiz kaldığından çoğunlukla üçüncü parti COM bileşenlerine ihtiyaç duyuluyordu. Bu çok ciddi bir sorundu çünkü her Hosting firması sizin ihtiyacınız olan bileşeni bulundurmayabiliyordu. Sizin yazdığınız bileşenlerin sisteme kurulması konusunda da güvenlik açısından çok hassas davranıyorlardı.

4. ASP, HTML kodlarının arasına serpiştirilerek yazılması gereken bir dil. Buna literatürde "Spagetti kodlama" deniliyor. HTML tagleri arasında kaybolan ASP kodları hiç hoş bir görünüm arzetmediği gibi, herşeyin birbirine girmesine de neden olabiliyordu.

5. ASP ile Nesne yönelimli çalışamadığınızdan tekrar kullanılabilir sınıflar (class) ve arabirimler yazılamıyordu.

6. Oluşturduğunuz formların yönetimi ise başlı başına bir dertti. Formun içindeki bilgilerin kontrolleri ASP ile yapılamadığından Javascript öğrenmek ve kullanmak zorunda kalmıştık. Formdaki değişiklikler neticesinde, formun bazı öğrelerinin içeriğinin değişmesinin gerektiği durumlarda formun mutlaka submit edilmesi gerekiyordu. Tabii Javascript'i iyi biliyorsanız durum farklı.

Peki ASP.Net cephesinde durum ne? Bize neler kazandırıyor ?

1. Yorumlayıcı

Uzantıları .aspx olan ASP.NET dosyalarının yorumlanmaları için yeni bir alt yapı kullanılıyor. Yeni çalışma ortamımızın adı IIS ile entegre çalışan .NET Framework. Bu yeni çalışma ortamının en büyük getirileri şüphesiz uyumluluk ve dayanıklılık. Bundan böyle yazacağınız ASP.NET kodları IIS ile sorunsuz çalışacak. Ayrıca ileriye dönük düşünüldüğünde bugün yazacağınız kodlar IIS'nin yeni sürümlerinde de problem çıkarmayacak.

2. Programlama dili, yazım kuralları

Temelde bir çok farklılık bulunmakta. Bunun en önemli sebebi artık hem Object Oriented Programming (Nesne Tabanlı Programlama) hemde Event Driven (Eylem Yönelimli) kullanılıyor olması. Bu konuyu gelecek yazılarımızda ayrıntıları ile inceleyecek ve artık hepimizin aşina olması gereken Nesne Yönelimli Programlama konusunu öğreneceğiz.

Peki ASP.Net programlarken hangi dili kullanacağız? Cevap, canınız hangisini isterse. Çünkü artık ASP ile en güzel günlerini görmüş VBScript adlı melez ve yarım yamalak dili kullanmak zorunda değilsiniz. Ne demiştik? ASP.NET, .Net platformunun bir parçası. Yani .Net için kullanabileceğiniz bütün dilleri kullanabilirsiniz. Bunlardan en önemlileri microsoftun yeni geliştirdiği ve "Java ya nispet olsun diye çıkardılar abi" şeklinde spekülasyonlar yapılan c#

(si şarp okunur) ve tüm özellikleri ile birlikte kullanabileceğiniz Visual Basic.NET . Ancak şunu akıldan çıkarmamak gerekir. Kullanacağınız diller gerçekten sınırsız. PHP.NET veya Perl.NET gibi üçüncü parti dillerle de programlama yapabilirsiniz.

3. Compile edilerek (derlenerek) çalıştırılan kodlar

Haydaaa! .exe ve .dll dosyaları derlenmiyormuydu? Evet! Kesinlikle öyle. Ancak aslında sizde bir .dll yazıyorsunuz. Yazdığınız kodlar, çalıştırılmadan önce bir defaya mahsus .NET derleyicisi aracılığı ile MSIL denen ara koda çevriliyor. Bunun ardından da makine kodu dediğimiz koda dönüştürülüyor, yani .dll haline getiriliyor. DLL lerin hızları göz önüne alındığında bunun ne çaplı bir performans artışı sağlayacağını düşünmek için matematik profesörü olmaya gerek yok. Derlenerek çalışan kodların daha bir çok getirisi var. Hatalarınızı anında görebiliyorsunuz. Düzeltmeleri yaptıktan sonra kodu tekrar çalıştırdığınızda, eğer gerçekten hata yoksa derlenen kod hemen ziyaretçilerinize hizmet vermeye başlıyor.

4. c# 'ın getirdikleri

Her ne kadar bu yeni ve müthiş yetenekli dili kullanmak zorunda olmasanızda, c# .NET ortamının resmi dili haline gelmiş sayılır. İlk defa .NET ile ortaya çıkmasının bunda etkisi büyük tabi. C# .NET Framework ile geliştirilmiş ve belkide .NET yapısı için tasarlanmış bir dil. Dolayısıyla .Net'in bütün özelliklerini barındırıyor ve sonuna kadar kullanabiliyor.Önemli artılarından biride tam manası ile Nesne yönelimli programlamaya destek vermesi. Yani Nesneye yönelik programlamanın getirdiği bütün avantajlardan faydalanabiliyorsunuz. Yazım kuralları açısından Java'ya çok benziyor ancak c++ da andırıyor. Kısacası iki dilin bütün iyi yönlerini almış. Öğrenmesi çok zor değil gibi gözükmesine rağmen, eski ASP cilerin Visual Basic.Net üzerinde yoğunlaşacağını tahmin ediyorum.

5. Devasa kütüphane desteği

Microsoft'un uzun uğraşlar neticesinde toparladığı ve yıllardır geliştirilerek Windows ortamında program yazanların beğenerek kullandığı sınıf kütüphaneleri (Base Class Library-BCL) bundan böyle ASP.Net programcılarının da hizmetinde. Bu kütüphanelerin yaptığı kısaca şu, işletim sisteminin alt yapısı ile gelen bir takım öğelerin kullanılabilir hale gelmesini sağlıyor. Yani bir mail göndermek veya dosya upload etmek için üçüncü şahısların componentlerine ihtiyacımız olmayak.

6. Sunucu kontrolleri (Server Controls)

ASP.NET, sunucu tarafında olay güdümlü olarak programlanabilen, sunum (presentation) ile mantığı (logic) başarıyla ayıran bir bileşen seti ve mimarisi sunuyor. Bu bileşenler, koddan bağımsız olarak sayfa üzerine yerleştirilebiliyor ve ayrı dosyalarda bu kontroller için programlama kodları yazılabiliyor. Biraz garip geldi değilmi, ilerleyen yazılarda detaylarına indikçe anlaşılır hale gelecek, meraklanmayın.

7. XML ile kolay konfigurasyon

Server konfigurasyonu için IIS yönetim arabirimi ile uğraşmanıza gerek kalmadı. Basit bir XML dosyası ile server konfigurasyonunu yapabiliyorsunuz.

Kısacası .NET platformunun saygıdeğer üyesi ASP.NET geldiği uzun ve tecrübelerle dolu yoldan eli boş gelmiyor. Bize bugüne kadar hayal bile edemeyeceğimiz bir çok yenilik ve imkan sunuyor. Umarım biz türk programcıları arasında hakettiği ilgiyi görür ve daha iyiye ulaşma çabamızda böyle güzel bir teknolojiden gerektiği gibi yararlanabiliriz.

Şunun altını özellikle çizmemiz lazım. .NET ve ASP.NET, ASP'nin devamı niteliğinde değil. O bambaşka bir teknoloji. Sanırım yüzyılın buluşlarından desek abartmış olmayız. Geleceğin trendini yakalamak adına en kısa sürede öğrenmek ve ciddi uygulamalar geliştirmekte ASP.NET 'i kullanmak zorundayız.

Bir dahaki yazımızda kafanızda oluşmaya başladığını düşündüğüm bir çok soruya cevap verecek açıklamalar bulacaksınız. Ayrıca .Net Framework ve uygulama geliştirme aracımız olan visual studio NET 'in kurulumunu ele alacağız...

Sevgilerle...

Madde I. ASP.Net ile Türkçe Karakter

Yazar: Bahadır ARSLAN [Bu yazarın tüm yazıları]

ASP.Net'e geçen kullanıcılarının er geç karşılaştıkları bir sorun Türkçe Karakter Problemi. Bu yazımızda bunun önlemenin yollarını anlatıyoruz. Okunma: **1029**

ASP.Net ile kod yazmaya başlamadan önce yaşamanız muhtemelen olan bir sorunu şimdiden çözelim. Kod yazmaya geçtiğinizde muhtemelen sayfanızda kullandığınız kontrollerde Türkçe karakterlerin yok olduğunu göreceksiniz. Bu yüzden önce bu durumdan nasıl kurtuluruz onu görelim ondan sonra kod yazmaya geçeceğiz...

Ancak isterseniz kısa biraz bilgi verelim.ASP.Net'in tanımlı karakter kodlama tekniği UTF-8'dir yani siz aşağıdaki ayarlamalardan hiç birini yapmazsanız ASP.Net varsayılan ayar olan UTF-8'i kullanacaktır. İşin güzel tarafı eğer UTF-8 kullanılırsa sayfanıza Türkçe karakterler görebilirsiniz ve hiçbir ayar yapmanıza gerek yoktur; fakat işin kötü tarafı kullandığınız metin editörünüzün UTF-8'i desteklemesi lazım. Eğer bu kodlamayı desteklemeyen bir editör kullanırsanız, editörünüz kodlarınızı ANSI formatında kaydeder ve sizin karakterleriniz yok olur gider...

Eğer siz UTF-8'i desteklemeyen bir editör kullanıyorsanız (örn:Homesite 4.5) ve Türkçe karakter sorunu yaşamak istemiyorsanız o zaman bütün web tasarımcılarının aşina olduğu bir karakter kodlama yöntemini kullanacağız ISO 8859-9 kodlaması. Bu kod Türkçe için özel bir koddur ve web tasarımı ile az çok uğraşan herkesin bildiği bir koddur. Ancak bu kodlamayı kullanmak için bazı ayarlamalar yapmak lazımdır..

ISO kodlamasını kullanarak ASP.Net'te Türkçe karakter sorunu yaşamamak için web.config dosyasında ve sayfamızda ayarlar yapmamız lazım.

Web.config dosyayı bildiğiniz gibi ASP.Net uygulamalarımız hakkında bazı ayarları yapmamızı sağlayan ve bu ayarları barındıran XML dosyasıdır. Aşağıda Türkçe karakter ayarı yapılmış bir web.config dosyasının kodları var. Ancak bu sadece bizi ilgilendiren kısmı, normalde bu sayfada daha çok şey var.

```
<?xml version="1.0" encoding = "iso-8859-9" ?>
<configuration>
<system.web>
<globalization requestEncoding="iso-8859-9"
responceEncoding="iso-8859-9" culture="tr-TR" uiCulture="tr"/>
</system.web>
</configuration>
```

Şimdi .aspx uzantılı sayfalarımızda yapacağımız ayara geçelim.ASP.Net sayfamızın "Page Directive"'inde yani ile başlayan satırında da Türkçe karakter kullanacağımızı belirtmemiz lazım. Bu satır aşağıdaki gibi olmalıdır.

<%@ Page Language = "C#" codePage="28599" %>

Ama bir dakika, bu da ne? 25899, hani biz ISO kullanıyorduk? Bu sayı ASP.Net'te bizim iso-8859-9 'a denk düşen bir kodlama.

Eğer yukarıdaki 2 ayarı yaptıysanız sayfalarınızda Türkçe karakter sorunu olmaması lazım. Ancak dikkat etmeniz gereken bir nokta web.config te böyle bir kodlamadan bahsederken, sayfamızda kod olarak başka bir dile işaret edecek bir kodlama kullanırsanız sayfalarınızın kafası karışır..

Sanıyorum artık Türkçe probleminiz kalmamıştır...

```
<body>
    <form id="form1" runat="server">
    <div>
         <br />
         <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label></asp:Label></asp:Label></asp:Label>
         <br />
         <br />
    </div>
    </form>
</body>
</html>
using System;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class Default : System.Web.UI.Page
{
    protected void Page Load (object sender, EventArgs e)
    {
        Label1.Text = "Merhaba Dünya!";
    }
}
  \leftarrow \rightarrow C
               ocalhost:54996/WebSite2/Default.aspx
  🗋 Bağlantıları Özelleştir 📄 Ücretsiz Hotmail 📄 Windows 郑 Windows M
```

Merhaba Dünya!

artık browser'ımızın adres çubuğunda yukarıdaki üç adresten birini tanımlayıp sayfamızı sunucuda çalıştıralım.

http://localhost/ornek.aspx

Dilersek wwwroot klasörü içinde yeni klasörler oluşturup sayfalarımızı buraya koyabiliriz. O zaman sanal adresimize bu alt klasörleride eklemeliyiz. Örnek olarak fiziki olarak sayfalarımızı şu adresin içine koyalım.

C:\Inetpub\wwwroot\deneme.aspx

Fiziki yol böyle olursa adres çubuğuna yazacağımız sanal yol şöyle olur.

http://bilgisayarım/deneme/deneme.aspx

Not: Buradaki kodları Word'den kopyalayıp notpad'a atarsanız çift tırnak (") işaretlerine dikkat edin. Bu işaretler farklı olarak gitmiş olabilir. Tırnaklar görünür ama asp nin anladığı tırnak kodunda değildir. Gerekirse bu işaretleri notpadde değiştirin.

ASP.NET çalışan kodlar

```
<html>
  <script language="VB" runat="server">
   Sub Page_Load(Sender As Object, E As EventArgs)
      Mesai.Text = "Merhaba Dünya! Su anda saat: " & DateTime.Now
   End Sub
  </script>
  <body>
    <h3><font face="Verdana">ASP.Net 'e
basladik!</font></h3>
     Bu örnekte, Server Kontrollerinden label
 kullanilarak, ASP.Net e basit bir giris
 yapiliyor. Yukaridaki tarihi ASP.Net e baslama
 tarihiniz olarak kaydedebilirsiniz.
     <hr>
     <asp:label id="Mesaj" font-size="24"
 font-bold="true" runat=server/>
   </body>
</html>
```

İnternetin İşleyiş Şekli: İnternet sunucu-istemci modeline göre çalışır. Bir istek olmadan sunucu yayın yapmaz (bir radyo gibi değildir).

En basit haliyle bir web sayfası HTML kodlarından oluşur. Bu kodlar sunucuda saklanır ve bir talep gelmesi durumunda bir kopyasını isteyiciye gönderir.

İstemcinin bilgisayarındaki browser (internet explorer) gelen HTML bilgisini yorumlayarak kullanıcıya gösterir.

Eğer daha aktif sayfalar yapmak isterseniz hem sunucuda hemde istemcide yapılabilecek kodlamalar vardır.

İstemci Taraflı Scripting (Client-Side): Script, konusu web olan program yada programcıklardır. Eğer sayfalarımıza (html kodlarının arasına) istemci taraflı script eklersek, istemcinin browseri bu scriptlerin gereklerini yerine getirerek istemciye gösterir. Bu kodlar tamamen istemcinin bilgisayarı tarafından çalıştırılır ve sunucunun bunlardan hiçbir haberi olmaz. Bu tip scripler bir çoğumuzun duyduğu JavaScript, VBScript, JScript dir.

Sunucu Taraflı Scripting (Server-Side): Sayfalarımızda sunucuda çalışmasını istediğimiz scripler varsa, sunucu bu durumda sadece Html kodlarını bize göndermekle kalmaz, bu scripleri çalıştırarak sonucunu html formatında bize gönderir.

Türkçe Karakterler:

Uluslararası standar kuruluşu olan ISO tarafından türkçe karakter kodlama standardı 8859-9 olarak belirlenmiştir. Eğer internet ortamında ziyaret ettiğimiz sitelerdeki türkçe karakterler görülemiyorsa bunları görmek için browser'ımızın karekter kodlamasını bu standarda ayarlamalıyız. Bunun için "Browser > Görünüm > Dil Kodlaması > UTF-8" olarak ayarlayın.

Bu çözüm istemci (cient) istemci taraflı bir çözümdür. Bazı durumlarda kodlama sunucu (server) taraflı olarak da problem çıkarmaktadır. Bu gibi durumları çözmek için birçok üreticinin üye olduğu "Unicode" kodlama sistemi geliştirilmiştir. Unicode bütün diller, bütün programlar ve ortamlar için geliştirilmiş bir kodlama organizasyonudur. Artık yeni çıkan bütün programlar Unicode sistemini destekleyerek çıkmaktadır. Bu kodlama sisteminin dönüşümünü yapan formata UTF (Unicode transformation format) denir. Anlaşılacağı üzere Unicode'un yaygınlaşmasıyla artık Türkçe karakter problemi kalmayacaktır.

Asp.net tanımlı kodlama sistemi UTF-8 dir. Eğer herhangi bir şey bildirilmezse türkçe karakterleri UTF-8 e göre değerlendirir.

Eğer kodlarımızı Notpad'de UTF-8 olarak kaydedersek Asp.net sayfaları içinde ek bildirimlere gerek yoktur. Türkçe karakterleri doğru biçimde görebilirsiniz.

Eğer UTF-8 dışında bir kodlama ile çalışacaksak iki işlem yapmalıyız. Wwwroot'un altında Web-Config dosyasını ayarlamalıyız ve kodlarımızın içinde de bildirimde bulunmayız. Ama UTF-8 kullanırsanız bunların hiçbirine gerek kalmayacaktır.

Örnek: Aşağıdaki kodları çalıştırdığınızda Asp.net sayfalarınızda türkçe karakter problemi olup olmadığını test etmiş olursunuz.

Adres 👩 http://localhost:82/NewFile.aspx

Ş,Ç,Ğ,I,Ö,Ü,ş,i,ç,ğ,ö,ül yandaki harfleri görebiliyorsanız sayfanızda Tükçe karakter problemi yoktur

<%@ Page Language="VB" codePage="28599"%> <script runat="server"> sub Page_Load(obj as object, e as eventargs) Mesaj.text="\$,C,Ğ,I,Ö,Ü,\$,i,C,Ğ,ö,ö! yandaki harfleri görebiliyorsanız sayfanızda Tükçe karakter problemi yoktur" End sub </script> <html> <head> <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9"> </head> <body> <asp:Label id="Mesaj" runat="server" /> </body> </html>

Not: Asp.NET uygulamalarında kullanılan codePage="28599" kodu "ISO-8859-9" koduna karşılık gelir.

Asp.Net Yazım Editörü: Asp.net kodlarınızı yazmak için notpad kullanabileceğiniz gibi aynı zamanda <u>www.asp.net</u> sitesinden indirebileceğiniz "Web Matrix" programınıda indirip kullanabilirsiniz. Bu program çok küçük boyutta (1.3 MB) olmasına rağmen çok gelişmiş özellikleri vardır. Eğer programı çalıştıramazsanız aşağıdaki yolu kullanarak

C:\Program Files\Microsoft ASP.NET Web Matrix\v0.6.812

Bu dizin altında WebMatrix.exe.config dosyasını notpad'de aç. 160 satırda bulunan Config yazan yerleri ConfIg olarak düzeltin. Baska satırlar varsa bunları da düzeltin. Sonra calısır.

Asp.Net Sayfasının Yapısı: Yukarıdaki örnek sayfalar baktığımızda temel iki bölüm göze çarpar. <script> </script> .. <html> </html>

<%@ Page Language="VB" codePage="28599" %>

<%@ Bu işaretler arasında sayfaya ait bildirimler verilir. Burada kullanılan programlama dili VB olacağı ve karakter kodlamasının ISO 8859-9 olacağını belirtir (iki kodda aynı anlamdadır)

```
<script runat="server">
Sub yazi_eklendi (Sender As Object, E As EventArgs)
Mesaj.text=yazi_kutusu.Text
End sub
</script>
```

Burası kod bildirim bloğudur. Yazacağımız VB program kodları html bloğunun dışında baş tarafta yazılır. runat="server" ifadesi scriptin sunucuda çalıştırılacağını belirtir. Bu bölümde yazılan kodlar sunucuda MSIL içinde derlenir. Diğer bir deyişle klasik bir script kodu gibi çalışmaz. Normal bir program gibi derlenerek çalışır. Halbuki scriptler yorumlanarak çalışır.

<html>

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>
```

Html sayfa bildirimlerinin bulunduğu kısımdır.

<body>

```
<font size="5">ASP.NET:sayfa yapısına örnek</font><hr> <font face="verdana">
```

<%Response.Write ("Temel sayfa Elemanlarını anlatmak için basit bir örnek<p>")%> Bu kullanım klasik Asp den gelen bir alışkanlıktır. Bu ifadeler sunucuda derlenmezler bu nedenle yavaştırlar. Pek tercih edilemezler.

```
<form runat="server">
```

```
Lütfen Kutuya herhangi bir şey yazıp, gönder butonuna tıklayın:
<asp:textbox id="yazi_kutusu" OnTextChanged="yazi_eklendi" runat="server" />
//Bu satırı sil çalışmadı <asp:buton id="buton" Text="gönder" runat="server" />
<asp:label id="Mesaj" font-size="15pt" runat="server" />
```

</form>

Bu satırlar sunucuda çalıştırılacak kodlardır. Form nesnesi oluşturur..

</body>

</html>

ÖRNEK: Formun üzerine bir text kutusu atalım. İsmimizi içine yazıp butona bastığımızda formun üzerine "Merhaba Ahmet Asp.Net'e hoş geldin" yazsın.

| Address 🐻 http://localhost/deneme.aspx 💌 🄁 Go |
|-------------------------------------------------|
| Lütfen kutuya isminizi yazıp butona tıklayınız. |
| isim: Ahmet |
| gönder |
| Merhaba Ahmet ASP.NET'e hoş geldin! |

<%@ Page Language="VB" codePage="28599"%> <script runat="server">

```
sub isim_girdisi(sender as object, e as eventargs)
              Mesaj.text="Merhaba " + isim_kutusu.Text + " ASP.NET'e hoş geldin!"
              End sub
       </script>
       <html>
       <head>
       <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
       </head>
       <body>
       Lütfen kutuya isminizi yazıp butona tıklayınız.
       <form runat="server">
              isim:
              <asp:textbox id="isim_kutusu" OnTextChanged="isim_girdisi" runat="server" />
              <asp:button id="buton" Text="gönder" runat="server" /> 
       </form>
       <asp:Label id="Mesaj" runat="server" />
       </body>
       </html>
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"</pre>
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
    <br />
    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
    <br />
    <br />
    <br />
  </div>
  </form>
</body>
</html>
using System;
using System.Configuration;
using System.Data;
using System.Ling;
```

```
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class _Default : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
  }
  protected void Button1_Click(object sender, EventArgs e)
  {
    string Ad = TextBox1.Text;
    Label1.Text = "Merhaba " + Ad + " Hos geldin!";
  }
}
           C
                localhost:54996/WebSite2/Default.aspx
  ←
      -
  🗅 Bağlantıları Özelleştir 📄 Ücretsiz Hotmail 📄 Windows 켇
  Ali
                             Button
 Merhaba Ali Hoş geldin!
```

Burada <form runat="server"> satırı ile server'da asp.net sistemine göre bir form oluşturulmaktadır.

Form: Kullanıcaya bir takım verileri girmeyi sağlayan bir arayüz (interface) oluşturur. İçerisindeki bilgilerin işlenmesi ise temel olarak bir butona (gönder) basmakla olur. Asp.Net ile birlikte form elemanları tamamen programlanıp yönetilebilir.

Web formu sunucuda nasıl işlenirse işlensin, geri gönderirken klasik anlamdaki HTML form kodları ile gösterilir. Aşağıda yukarıdaki sayfanın işlenmiş kodları vardır.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>
<body>
Lütfen kutuya isminizi yazıp butona tıklayınız.
```

```
<form name="_ctl0" method="post" action="dene2.aspx" id="_ctl0">
<input type="hidden" name="__VIEWSTATE"
value="dDwxMDgwMDE5NzY4O3Q8O2w8aTwxPjtpPDM+Oz47bDx0PDtsPGk8MT47PjtsPHQ8cD
xwPGw8VGV4dDs+O2w8QWhtZXQ7Pj47Pjs7Pjs+Pjt0PHA8cDxsPFRleHQ7PjtsPE1lcmhhYmFBa
G1ldCAgQVNQLk5FVOKAmWUgaG/FnyBnZWxkaW4hOz4+Oz47Oz47Pj47Pne9Jf1vok0+J5nTHr
RvxktN+osj" />
```

isim:

```
<input name="isim_kutusu" type="text" value="Ahmet" id="isim_kutusu" /> <input type="submit" name="buton" value="gönder" id="buton" />
```

</form>

MerhabaAhmet ASP.NET'e hoş geldin! </body> </html>

Klasik ASP sayfaları sunucu-istemci yapısına göre çalışır. Yani formun işlenebilmesi için her seferinde veriler sunucuya gönderilir. Sayfada herhangi bir değişiklik olsun veya olmasın farketmez.

Oysa ASP.NET'de durum oldukça farklıdır. Burada da yapı istemci-sunucu şeklindedir fakat sunucunun çalışması "olay" a bağlanmıştır. Yani sadece kontrollerde bir olay meydana gelmişse sunucu sayfayı tekrar işler. Diyelim text box kutusunda herhangi bir değişiklik yapılmışsa sunucu bunu işler. Eğer hiçbir değişiklik yoksa hiçbir işlem yapmaz.

ViewState Elemanı: Bu konuyu anlayabilmek için aşağıdaki uygulamayı yapalım.

Uygulama: Yukarıdaki örnekte önce textbox'a "Ahmet" yazın ve çalıştırın. Formun üzerinde "Merhaba Ahmet ASP.NET'e hoş geldin!" yazacaktır. Daha sonra Asp.Net kaynak kodlarını notpad den şu şekilde değiştirin ve kaydedin.

sub isim_girdisi(sender as object, e as eventargs) Mesaj.text="Merhaba " + isim_kutusu.Text + " işler nasıl gidiyor?" End sub

Browsera dönüp tekrar kodlarınızı çalıştırdığınızda hiçbir değişiklik olmadığını göreceksiniz. Oysa sizin kaynak kodlarınız değişmişti.. Yani sunucu hiçbir işlem yapmadı.

Şimdi textbox'ın içine "Mehmet" yazın. Denerseniz formun üzerinde "Merhaba Mehmet işler nasıl gidiyor?" yazısını göreceksiniz.

Peki bunu nasıl yapıyor? Aslında oldukça kurnazca bir biçimde.

Bunun için HTML kodu içine "viewstate" isminde gizli bir form elemanı oluşturur ve girilen değerleri burada kodlayarak saklar. Örneğin bir formda text box yazılmış mı? Check box işaretlenmiş mi, işaretlenmişse bunlar hangi kutulardır? Bu gibi forma ait bilgiler bu gizli eleman içine kodlanarak saklanır. Eğer yeni bir değişiklik yoksa sunucudan bilgiler istenmez ve sayfada bir değişiklik meydana gelmez.

Sayfanın Program Olarak Yorumlanması: Asp.Net de istemci sunucudan ilk kez bir sayfa isteğinde bulunduğunda kodlar önce derlenir (compile) daha sonra gerekli isteklere cevap verilir. Bu nedenle ilk çağrışta sayfa biraz gecikir. Daha sonra bu gecikme yaşanmaz (eğer kodlarda herhangi bir değişiklik olmadıysa).



Editörde Yazım Kuralları:

a) <u>Devamını alt satıra yazma:</u> HTML de kodların devamının bir alt satıra yazmayla aynı satıra yazmak aynı şeydir.

Bilgisayar (tek satırda yazma)

Bilgisayar (devamını alt satırda yazma)

Asp.Net kodlarını yazarken ise _ (alt çizgi) boşluklu olarak kullanılmalıdır. Örnek;

Mesaj.text="Merhaba " + isim_kutusu.Text + " işler nasıl gidiyor?" (tek satırda)

```
Mesaj.text="Merhaba " + isim_kutusu.Text + " işler _ (iki satırda yazma) nasıl gidiyor?"
```

b) Yorum satırları ekleme: VB de olduğu gibi (') tek tırnakla ifade edilen yazılar program içinde hiçbir anlam ifade etmez. Görmezden gelinir.

sub isim_girdisi(sender as object, e as eventargs) Mesaj.text="Merhaba " + isim_kutusu.Text + " işler nasıl gidiyor?" ' Bu satırı görmezden gelir... End sub

VBASIC KOMUTLARI VE UYGULAMALARI

Değişken bildirimleri:

Dim değişken_ismi As Tipi

Şeklinde yapılır. Örneğin

Dim mesafe As integer

Değişkenlere tanımlama satırında da değer atayabiliriz. Örneğin;

Dim mesafe As Integer = 5

Değişkenleri birbirine dönüştürmek için aşağıdaki fonksiyonlar kullanılır.

Cbool, Cbyte, Cchar, Cdate, Cdec, Cint, Clng, Cobj, Cshor, Cstr

Bu dönüşüm başka bir şekildede yapılabilir. Örneğin

mesafe_1.ToString

Diziler (Array):

Diziler aynı isimde farklı değerler barındıran değişkenlerdir. Bunların tanımlanmaları şu şekilde yapılır.

Dim isim(5) as String

Dersek isim değişkenine 5 tane ismi yükleyebiliriz. Dizi değişkenlerinin indis numarası 0 dan başlar. Dizilere tanımlama esnasında da değer yükleyebiliriz.

Dim isim() as String = {"ali", "oya", "can", "cem", "ata"}

gibi

Response.Write (isim(2-3))

Yazarsak ekrana Can ve Cem isimlerini getirir.

Döngü ve Bloklar:

If Deyimi: Yapısı

If (koşul) Then Else if (koşul) ... Else ... End If

Select Case: Kullanımı if deyimine benzer. Yapısı;

Select Case değişken Case değer1 ... Case deger2 Case Else

... End Select

For döngüsü:

For değişken=ilk To son [Step değer] Next

While döngüleri:

While koşul ... End While

Do

Loop While koşul

Alt Yordamlar Ve Fonksiyonlar

Subroutine yada Procedure Alt programlar yada fonksiyonlar üç aşamalı olarak kullanılır.

Örnek: 15 ve 28 sayılarını program içinden verip toplama işlemi subruoutine de yaptırın ve ekrana toplamını yazsın.



<%@ Page Language="VB" codePage="28599"%> <script runat="server">

Sub Page_Load(Sender As Object, E As EventArgs)

topla(5,23)

End Sub

Sub topla(a As integer, b As integer) response.write(a+b & "
") End Sub

</script>

<html> <head> </head> <body> </body> </html>

Sub routine kullanıldı. Sub routinler geri değer döndermeyen alt programlardır.

```
protected void Button1 Click(object sender, EventArgs e)
     {
          string[] Isimler = { "Ali", "oya", "Cem", "Ata", "Isa" };
          foreach (string Ad in Isimler)
          {
               Response.Write(Ad + "<br/>>");
          }
     }
******
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"</p>
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
    <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
  </div>
  </form>
</body>
</html>
using System;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class _Default : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
  }
  protected void Button1_Click(object sender, EventArgs e)
  {
    string[] Isimler = { "Ali", "oya", "Cem", "Ata", "Isa" };
    int i=8;
    foreach (string Ad in Isimler)
    {
      i = i +5;
       Response.Write("<font style='font-size:" + i + "px' > " + Ad + "</font><br/>>");
```

| } | | | |
|---------------------------|--------------------|---------------|--------|
| } | | | |
| 🗲 🔿 🤁 🗋 loca | alhost:54996/WebS | Site2/Default | t.aspx |
| 🗋 Bağlantıları Özelleştir | 🗋 Ücretsiz Hotmail | 🗋 Windows | 🔊 Wir |
| Ali oya Cem | | | |
| Ata | | | |
| Isa | | | |
| Button | | | |

</html>

Örnek: 5 ve 23 sayılarını bir fonksiyon içinde toplatıp ekranda yazdırın.

| Ornek: 5 ve 25 sayılarını bir lonksiyon içinde toplatip ekranda ya |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Address 🙋 http://localhost/deneme.aspx |
| 5+23=28 |
| <%@ Page Language="VB" codePage="28599"%> <script runat="server"></th></tr><tr><th>Sub Page_Load(Sender As Object, E As EventArgs) response.write("5+" & "23=" & topla(5,23)) End Sub</th></tr><tr><th>Function topla(a As integer, b As integer) as integer dim c as integer c=a+b return c End Function</th></tr><tr><th></script> |
| <html> <head> </head> <body> </body></html> |

Burada alt program olarak function kullanıldı. Function içerisine girdiler olan (burada a ve b) ve bir çıktısı olan alt programlardır. Burada çıktı (geri dönen değer) c dir.

Örnek: 35 ve 45 sayılarını function kullanarak hesaplatın. Ekranda gösterme işlemini function içinde yapın

Address 🙆 http://localhost/deneme.aspx

35+45=80

<%@ Page Language="VB" codePage="28599"%> <script runat="server">

Sub Page_Load(Sender As Object, E As EventArgs) topla(35,45) End Sub

```
Function topla(a As integer, b As integer)
response.write(a & "+" & b & "=" & a+b)
End Function
```

</script>

<html> <head> </head> <body> </body> </html>

Burada toplama işlemi ve ekranda gösterme işlemi function içinde yapılmıştır. Function geri bir değer göndermemiştir. Bu nedenle tıp kı subroutine gibi kullanılmıştır.

ASP.NET İÇİNDEKİ HAZIR NESNELER

```
************** Yazıları küçükten büyüğe doğru yazdırma-C#
     protected void Button1 Click(object sender, EventArgs e)
     {
           for (int i = 1; i <= 26; i++)</pre>
           {
                Response.Write("<font style='font-size:" + i + "px;'>" +
i.ToString() + "Merhaba" + "</font> </br>");
           }
     }
Açılış sayfasını ayarlama
   🕶 🗙 Solution Explorer - ... 🕶 📮 🗙
       - 🕒 🛃 🗉 🖬 🍹
        🧊 Solution 'WebSite1' (1 projec
       ...\WebSite1\
          App_Data
     ි Open
        Open With...
     🔄 View Code
     📃 View Designer
        View Markup
        View Component Designer
     Kiew in Browser
        Browse With...
        Set As Start Page
        Build Page
     🚱 Check Accessibility...
        Exclude From Project
        Cut
     *
     Сору
    Б
     ×
        Delete
- 1
        Rename
*****
```

Asp.net ile birlikte gelen birçok hazır nesne bulunmaktadır. Bu nesneleri kullanarak net ortamında birçok işlemimizi gerçekleştiririz. Her nesnenin içerisinde kullanılan alt işlevler (metodlar) bulunmaktadır. Şu anda sadece Response nesnesi ile bunun iki alt metodunu görelim.

Response Nesnesi: Bu nesne sayfadaki görüntüleme işlemlerinde kullanılır.

Response.Write: Bu metodla sayfada değişken veya sabitleri görüntüleriz.

Örnek: Sayfa yüklendiğinde gittikçe büyüyen 7 tane "merhaba" yazdırın.

| methibi | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| merhaba | |
| merhaba | |
| merhaba | |
| merhaba | |
| merhaba | |
| merhaba | |
| <%@ Page Language=' | 'VB" codePage="28599"%> |
| <script runat="server"> sub Page_Load(send dim k as integ for k=1 to 7 Response.W next End sub </script> | er as object, e as eventargs) er / /rite (" merhaba ") |
| <html> <head> <meta content="text/html; charset=iso-8859-9" http-equiv="Con </head> <body> </body></th><td>tent-Type"/></head></html> | |

</html>

Response.Redirect: Sayfamızdan başka bir sayfaya yada siteye yönlendirme yapmada kullanabiliriz.

Örnek: iki tane sayfa oluşturun (deneme1.aspx, deneme2.aspx). Birinci sayfadaki bir butona bastığımızda ikinci sayfaya yönlensin. İkinci butona bastığımızda ise <u>http://ktef.karaelmas.edu.tr</u> adresine yönlensin.



deneme1.aspx

<%@ Page Language="VB" codePage="28599"%>

```
<script runat="server">
sub sayfa_cagir(sender as object, e as eventargs)
Response.Redirect("deneme2.aspx")
End sub
</script>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-9">
</head>
```

```
<body>
<br/>
<form runat="server">
<br/>
<asp:button id="buton" text="deneme2" runat="server" onclick="sayfa_cagir" />
</form>
<br/>
</body>
</html>
<br/>
22.aspx
<%@ Page Language="VB" codePage="28599"%>
<br/>
<script runat="server">
<br/>
sub site_cagir(sender as object, e as eventargs)
<br/>
Response.Redirect("http://ktef.karaelmas.edu.tr")
End sub
</script>
<html>
```

deneme2.aspx

```
<script runat="server">
sub site_cagir(sender as object, e as eventargs)
Response.Redirect("http://ktef.karaelmas.edu.tr")
End sub
</script>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>
<body>
<form runat="server">
<asp:button id="buton" text="Site" runat="server" onclick="site_cagir" />
</form>
</body>
```

Web Formları

Web formları kullanıcının verilerini girmesini, tercihlerini belirtmesini sağlayan grafik bir arabirim oluşturan formlardır. Bu formlar klasik olarark html kodları içinde verilip browserda oluşturulabilmekteydi fakat bizim buradaki konumuz asp.net tarafından sunucuda oluşturulan formlar olacaktır.

Sunucuda oluşturulan formlar HTML form elemanlarına çevrilerek gönderilirler. Her türlü programatik işlem web formları üzerinde yapılabilir. Sonuçları html olarak gönderilir. Bu nedenle bütün kontrol tasarımcının elindedir.

Sayfamızda asp.net kodlarımız iki grup halinde toplanabilir. Bunlar sayfada gözükecek elemanlar ve bu elemanlar üzerinde bir olay gerçekleştiğinde çalıştırılacak kodlar. Bu iki kısım fiziksel olarak farklı yerde bulunsa bile birbirine bağlıdır.

<script> bir olay gerçekleştiğinde çalıştırılacak kodlar </script>

<html> <head></head> <body>

<form> sayfada gözükecek elemanların oluşturulduğu yerler. </form>

</body> </html>

Web formlarında kullanılan kontroller için üç tane temel bildirim mutlaka verilmelidir. Bunlar kontrolün ismi, kontrolü aynı tipteki başka konrollerden ayıran ismi, ve sunucuda çalışacağını gösteren ifade. Örneğin;

<asp:kontrol_tipi id="kontrol_ismi" runat=server /> <asp:buton id= "dugme1" runat= "server" />

Bu kontrolün yanında eğer bir olay tanımlanacaksa olayın adı ve olay gerçekleştiğinde çalıştırılacak procedurun ismi yazılır. Bunun yanında eğer konrolün üzerinde bir yazı yazması isteniyorsa Text ifadeside kullanılabilir. Örnek:

<asp:buton id= "dugme1" text="mesaj yazdir" runat= "server" OnClick= "yazdir" />

Örnek: Sayfaya iki adet Buton ve bir adet Label kontrolü koyun. Butonların birincisinde "mesaj yaz" ikincisinde ise "mesaj sil" yazsın. Birinci butona basınca Label nesnesinde "MERHABA" yazsın. İkinsine basıldığında ise bu yazıyı silsin. Sayfa ilk yüklendiğinde ise Label nesnesinde "burası label nesnesidir" yazsın.



<%@ Page Language="VB" codePage="28599"%>

```
<script runat="server">
sub yordam1(sender as object, e as eventargs)
mesaj.text="merhaba"
End sub
sub yordam2(sender as object, e as eventargs)
mesaj.text=" "
End sub
</script>
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>
<body>
<form runat="server">
<asp:label id="mesaj" text="burası label nesnesidir" runat="server" />
<br>
<asp:button id="buton1" text="mesaj yaz" runat="server" onclick="yordam1" />
<br>
<asp:button id="buton2" text="mesaj sil" runat="server" onclick="yordam2" />
</form>
</body>
</html>
```

Nesne, Nesne - Yordam İlişkisi: Bir yordama (procedure/subroutine/metod) bir çok nesne bağlanabilir. Dolayısı ile yazmış olduğumuz bir yordam bir çok nesne tarafından kullanılabilir. Yordam içinde hangi nesneyi bağlamışsak onun alt özelliklerini kullanabiliriz. Bu yapıyı anlayabilmek için şu örneği yapalım.

Örnek: Sayfaya iki tane buton koyun. Üzerlerinde "gonder" ve "iptal" yazsın. Birinci butona basınca sayfanın altında "gonder düğemsine bastınız", ikincisine basınca "iptal düğmesine bastınız" yazsın. Bu işlemler için tek bir yordam kullanın.

```
gönder düğmesine bastınız
gönder
iptal
```

<%@ Page Language="VB" codePage="28599"%>

```
<script runat="server">
        sub deneme(nesne as object, e as eventargs)
          Response.write(nesne.text & " düğmesine bastınız")
        End sub
      </script>
      <html>
        <head>
          <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
        </head>
        <body>
           <form runat="server">
             <asp:button id="buton1" text="gönder" runat="server" onclick="deneme" />
             <br>
              <asp:button id="buton2" text="iptal" runat="server" onclick="deneme" />
           </form>
        </body>
      </html>
*****
```

protected void btnOnay_Click(object sender, EventArgs e)

```
{
    Label1.Text = "Onay Butonuna tikladiniz!";
  }
  protected void btnlptal_Click(object sender, EventArgs e)
  {
    Label1.Text = "Iptal Butonuna tikladınız!";
  }
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class Default2 : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
  }
  public void HangiButon(object sender, EventArgs e)
  {
    Label1.Text = sender.Text
  }
}
```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
```

```
<asp:Button ID="btnOnay" runat="server" onclick="HangiButon"
Text="Onay" />
<asp:Button ID="btnIptal" runat="server"
onclick="btnIptal_Click"
Text="İptal" />
<asp:Label ID="Label1" runat="server"
Text="Label"></asp:Label>
```

```
</div>
</form>
</body>
</html>
```

DİKKAT BURADAKİ ÖRNEK TAMAMLANACAK. HANGİ BUTONA TIKLARSA AYNI OLAYA BAĞLI FONKSİYON ÇALIŞACAK VE HANGİ BUTONA TIKLANILDIĞI EKRANDA YAZACAK

Burada her iki butona da tıkayınca deneme yordamına (prosedure) ne gönderilmektedir. Burada geçen "nesne as object" ifadesi ile gönderen nesneyi bu procedurde "nesne" adıyla kullanacağımızı bildirmiş oluyoruz. Artık procedure içinde nerede nesne yazarsak bu bizi buraya gönderen nesneyi gösterecektir. Dolayısıyle "nesne.text" ifadesi gönderen butonun üzerindeki text yazısı olacaktır.

"e as eventargs" ifadesi olayı meydana getiren nesnenin argümanlarını kullanmayı sağlar. Burada buton nesnesi her hangi bir argüman göndermediği için içi boştur.

Web Formlarının Postalanması: Buraya kadarki yaptığımız örneklerde dikkat ettiyseniz formumuzda yaptığımız işlemler bir butona bastığımızda gerçekleşmektedir. Yani sayfamız sunucuya butona bastıktan sonra gönderilmektedir. Oysa bazı durumlarda butona basmadan da sayfamızın sunucuya gönderilip işlenmesini isteyebiliriz. Örneğin bir textbox'a yazdığımız her harfin anında başka bir textbox'a da yazılmasını isteyebiliriz. Her olay gerçekleştiğinde sayfanın sunucuya gönderilmesi için "AutoPostsBack=True" ifadesini olayın yanına yazmalıyız.

Örnek: Sayfaya bir textbox ve bir label nesnesi koyun. Textbox'a yazdığımız her karakterin Label nesnesinde anında yazıldığını gösterin. (ipucu textbox'ın OnTextChanged olayını kullanın)

bu işlemi Asp.net nasıl yapıyor. Bunun için asp.net sayfayı istemciye gönderirken istemci tarafında çalışan bir java script kodunu sayfanın içine ekliyor. Bu kod sayesinde her değişimi algılayarak sayfanın her karakterde sunucuya postalanmasını sağlıyor.

Dikkat: Söylendiği şekliyle bu sistem çalışmadı. Daha sonra bu konulara tekrar dönecegim.

Web Form Elemanları

Web form elemanları gerek kullanıcıya bir şeyler göstermek, gerekse veri girmek için olsun bütü kontrol sunucudadır. Geliştirici bütün dikkatini program geliştirme mantığına verebilir.

.NET herbiri farklı amaçlar için isimlendirilmiş onlarca NameSpace (isim uzayı) den oluşur. Herbir namespace ise içerisinde birçok Class (sınıf) barındırır. Class'lar kendisinden oluşturulan nesnelerin özelliklerini ve metodlarını (yordamlarını) barındırır.

| | Buton | Label | TextBox | ListBox | DropDownList | RadioButton | CheckBox | Hyperlink | Image | | | |
|------------------|-------|-------|---------|---------|--------------|-------------|----------|-----------|-------|--|--|--|
| AccessKey | Х | Х | Х | Х | Х | Х | Х | Х | Х | | | |
| AdvertismentFile | | | | | | | Х | | | | | |
| AltenateText | | | | | | | | | Х | | | |
| AutoPostBack | | | Х | Х | Х | Х | | | | | | |

Tablo 1: Her nesnenin özellik ve olayları vardır. Bunları toplu bir şekilde aşağıda bulabilirsiniz.

| BackColor | Х | Х | Х | Х | Х | Х | Х | Х | X | | | | |
|------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---|----------|------|------|
| BorderColor | Х | Х | Х | Х | Х | Х | Х | Х | X | | | | |
| BorderStyle | Х | Х | Х | Х | Х | Х | Х | Х | Х | | | | |
| BorderWidth | Х | Х | Х | Х | Х | Х | Х | Х | Х | | | | |
| CausesValidation | Х | | | | | | | | | | | | |
| Checked | | | | | | Х | | | | | | | |
| Colums | | | Х | | | | | | | | | | |
| CommandArgument | Х | | | | | | | | | | | | |
| CommandName | Х | | | | | | | | | | | | |
| CssClass | Х | Х | | Х | Х | Х | Х | Х | Х | | | | |
| DataMember | | | | Х | Х | | | | | | | | |
| DataSource | | | | Х | Х | | | | | | | | |
| DataTextField | | | | X | X | | | | | | | | |
| DataTextFormatString | | | | X | X | | | | | | | | |
| DataValueField | | | | x | x | | | | | | | | |
| Enabled | x | x | x | x | x | x | x | x | x | | | | |
| EnableViewState | x | x | x | x | X | x | x | X | x | | | | |
| Font Rold | x | X | x | x | x | x | x | x | x | | | | |
| Folit-Dold | N V | X V | N V | X V | X V | X V | X V | X V | N V | | | | |
| Font-Itanc | л v | л v | л v | л v | л v | л v | л v | л v | л v | | | | |
| Font-Name | A V | A V | A V | A V | A V | A V | A V | A V | A V | | | | |
| Font-Names | X | X | X | X | X | X | X | X | X | | | | |
| Font-Overline | X | X | X | X | X | X | X | X | X | | | | |
| Font-Size | Х | X | Х | X | X | X | X | Х | X | | | | |
| Font-Strikeout | Х | Х | Х | Х | X | Х | Х | Х | X | | | | |
| Font-Underline | Х | Х | Х | Х | Х | Х | Х | Х | Х | | | | |
| ForeColor | Х | Х | Х | Х | Х | Х | Х | Х | Х | | | | |
| GroupName | | | | | | Х | | | | | | | |
| Height | Х | Х | Х | Х | Х | Х | Х | Х | Х | | | | |
| ID | Х | Х | Х | Х | Х | Х | Х | Х | Х | | | | |
| ImageAlign | | | | | | | | | Х | | | | |
| ImageURL | | | | | | | | Х | Х | | | | |
| KeywordFilter | | | | | | | Х | | | | | | |
| MaxLength | | | Х | | | | | | | | | | |
| NavigateURL | | | | | | | | Х | | | | | |
| ReadOnly | | | Х | | | | | | | | | | |
| Rows | | | Х | Х | | | | | | | | | |
| Runat | Х | Х | Х | Х | Х | Х | Х | Х | Х | | | | |
| SelectedIndex | | | | Х | X | | | | | | | | |
| SelectionMode | | | | X | | | | | | | | | |
| TabIndex | x | x | x | x | x | x | x | x | x | | | | |
| Target | | | | | | | x | x | | | | | |
| Target | x | | | | | x | | | | | | | |
| Text | | | | | | x | | | | | | | |
| TextAligh | | | v | | | Λ | | | | | | | |
| | v | v | л v | v | v | v | v | v | v | | | | |
| VISIBLE | | л v | л v | л v | A V | л v | л v | л v | | | | | |
| Width | Λ | Λ | A V | Λ | Λ | Λ | Λ | Λ | Λ | | | | |
| Wrap | | | X | | | | | | | | | | |
| On AdCreated | | 1 | (| JLAY | LAK | 1 | x | | | 1 | 1 | | |
| OnChastadChanged | | | | | | x | | | | | | | |
| OnCheckedUnanged | v | | | | | л | | | | | | | |
| OnClick | л v | | | | | | | | | | | | |
| OnCommand | | v | v | v | v | v | v | v | v | | | | |
| OnDataBinding | A V | A V | A V | A V | A V | A V | A V | A V | A V | | | | |
| UnDisposed | X | X | X | X | X | X | X | X | X | | <u> </u> | | |
| OnInit | X | X | X | X | X | X | X | X | X | | | | |
| Onload | X | X | X | X | X | X | X | X | X | | | | |
| OnPreRender | Х | X | Х | X | X | Х | Х | X | X | | <u> </u> | | |
| OnSelectedIndexChanged | | | | Х | Х | | | | | | 1 | | |

| OnTextChanged | | | Х | | | | | | | | | |
|---------------|---|---|---|---|---|---|---|---|---|--|--|--|
| OnUnload | Х | Х | Х | Х | Х | Х | Х | Х | Х | | | |
| | | | | | | | | | | | | |

BUTTON: Daha önce birçok kez kullanıldı. Özellikleri ve olayları şöyledir.

Örnek: Formun üzerine Kırmızı görünümlü ve üzerinde Arial Fontla yazılmış Bold, İtalik, mavi renkle "Dikkat" yazan bir buton koyun. Butonun boyutları 200x100 pixel olsun. Tıklayınca "Merhaba" yazsın. Başka özelliklerde ekleyin.

DİKKAT <%@ Page Language="VB" codePage="28599"%> <script runat="server"> sub yordam(nesne as object, e as eventargs) Response.Write("merhaba") End sub </script> <html> <head> <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9"> </head> <body> <form runat="server"> <asp:button id="dugme" text="DİKKAT" BackColor="red" ForeColor="blue" Font-Bold="True" Font-Name="Arial" Font-Size="25" BorderStyle="dotted" width="200px" height="100px" runat="server" OnClick="yordam" /> </form> </body> </html>

LABEL: Label nesnesi sayfada herhangi bir değişkenin içini yada sabiti göstermede kullanılır.

TEXTBOX: Kullanıcının metin girmesini sağlayan kutudur. Özellikleri ve olayları aşağıdadır.

Bu özelliklerden bir tanesine değinelim. TextMode="MultiLine" yazılırsa birden çok satır girilebilir. TextMode="Password" yazılırsa karakterler sifre görünümü alır.

LISTBOX: Burada iki özelliğinden bahsedelim. SelectedItem özelliği listede seçilen elemanı varir. SelectedIndex özelliği ise listede seçilen elemanın sıra numarasını verir..

Örnek: Sayfaya birtane textbox ve birtane de listbox ekleyin. Listbox'ın içinde üç tane isim olsun ve herhangi bir ismi seçtiğimizde seçili olan elaman hemen textbox'da görülsün.

```
Ahmet
   Mehmet
   Hasan
   Mehmet
       <%@ Page Language="VB" codePage="28599"%>
       <script runat="server">
         sub yordam(nesne as object, e as eventargs)
              kutu1.text=liste1.selectedItem.text
         End sub
       </script>
       <html>
         <head>
           <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
         </head>
         <body>
            <form runat="server">
               <asp:listbox id="liste1" runat="server" OnSelectedIndexChanged="yordam"
       AutoPostBack="True" >
                <asp:listItem>Ahmet</asp:ListItem>
                <asp:listItem>Mehmet</asp:ListItem>
                <asp:listItem>Hasan</asp:ListItem>
               </asp:listBox>
               <br>
              <br>
              <asp:textbox id="kutu1" text="yazi" runat="server" />
           </form>
         </body>
       </html>
*****
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs" Inherits="Default2" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"</pre>
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
    <hr />
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <br />
    <asp:ListBox ID="ListBox1" runat="server" AutoPostBack="True" Height="111px"</pre>
      onselectedindexchanged="ListBox1_SelectedIndexChanged" Width="126px">
      <asp:ListItem>Ali</asp:ListItem>
      <asp:ListItem>Oya</asp:ListItem>
```

```
<asp:ListItem>Can</asp:ListItem>
    </asp:ListBox>
    <asp:Image ID="Image1" runat="server" BorderWidth="1px" Height="100px"
      Width="80px" />
    <hr />
    <asp:RadioButton ID="RadioButton1" runat="server" GroupName="Grup1"
      Text="İsimleri Göster" />
    <asp:RadioButton ID="RadioButton2" runat="server" GroupName="Grup1"
      Text="Indeks numarasını göster" />
    <hr />
    <br />
    <asp:Button ID="btnOnay" runat="server" onclick="Buton_Click" Text="Ekle"
      Width="121px" />
    <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
      Text="Temizle" />
    <br />
  </div>
  </form>
</body>
</html>
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Ling;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class Default2 : System.Web.UI.Page
  protected void Page_Load(object sender, EventArgs e)
  {
  3
  public void Buton_Click(object sender, EventArgs e)
  {
    string Ad = TextBox1.Text;
    ListBox1.Items.Add(Ad);
  }
  protected void Button1_Click(object sender, EventArgs e)
  {
    ListBox1.Items.Clear();
  }
  protected void ListBox1_SelectedIndexChanged(object sender, EventArgs e)
  {
    if (RadioButton1.Checked == true)
    {
      TextBox1.Text = ListBox1.SelectedItem.Text;
      Image1.ImageUrl = "Resimler/" + ListBox1.SelectedIndex + ".jpg";
    }
```

{

| else if (F { TextB | RadioButton2.Ch ox1.Text = ListB | ecked == lox1.Selec | true) ctedIndex. | ToString() |
|--------------------------|-------------------------------------|------------------------|---------------------|------------|
| } | | | | |
| | | | | |
| ← ⇒ C | 127.0.0.1 :5499 | 96/WebSit | e2/Default | 2.aspx |
| 🗋 Bağlantıları (| Özelleştir 🗋 Ücretsiz | z Hotmail | 🗋 Windows | 🝠 Wind |
| Ali Ali Oya Can | | | | |
| İsimleri Gö | ister O Indeks nun | narasını gös | ster | |

DROPDOWNLIST: Aynen listbox'a benzer. Tek farkı listedeki elemanların hepsini sayfada göstermez. Sadece seçili olanı gösterir.

Örnek: Yukarıdaki aynı örneği DropDownList kullanarak deneyin

RADIOBUTTON: tanımlanmış bir grup içinden yalnız birini işaretlememize izin veren kontroldür.

Örnek: Sayfaya üç tane takımın adını Radiobutton şeklinde yazdırın. İçlerinden birini seçip butona bastığımızda seçilen takımı sayfaya yazsın.

| seçtiğiniz takım:Galatasaray |
|---------------------------------------------|
| ⊙ Fenerbahçe ⊙ Galatasaray ⊙ Besiktas |
| gonder |

<%@ Page Language="VB" codePage="28599"%>

```
<script runat="server">
sub yordam(nesne as object, e as eventargs)
if Radyo1.checked="true" then
response.write("seçtiğiniz takım:Fenerbahçe")
else if radyo2.checked="true" then
response.write("seçtiğiniz takım:Galatasaray")
else
response.write("seçtiğiniz takım:Beşiktaş")
end if
End sub
```
```
</script>
<html>
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
  </head>
  <body>
    <form runat="server">
       <asp:Radiobutton id="Radyo1" Text=Fenerbahce Checked="True"
GroupName="takimlar" runat="server" /> <br>
       <asp:Radiobutton id="Radyo2" Text=Galatasaray GroupName="takimlar"
runat="server" /> <br>
       <asp:Radiobutton id="Radyo3" Text=Beşiktaş GroupName="takimlar" runat="server"
/> <br>
       <asp:button id="dugme1" Text=gonder runat="server" onclick="yordam" />
    </form>
  </body>
</html>
```

CHECKBOX: Birden fazla seçeneği seçmemize müsaade eden nesnelerdir.

Örnek: Formun Üzerine 5 tane hobiyi Checkbox kutusu kullanarak yerleştirin. Kullanıcı bunlardan kendi hobilerini seçtikten sonra butona basıldığında seçili olan hobileri sayfada tekrar göstersin.

| Seçtiğiniz hobiler: Dağcılık Yüzme Maketcilik |
|-----------------------------------------------|
| 🗹 Dağcılık |
| ✓ Yüzme |
| 📃 Yürüyüş |
| 🗹 Maketcilik |
| 🗖 Resim |
| gonder |

```
<script runat="server">
sub yordam(nesne as object, e as eventargs)
```

Dim SecilenMaddeler as string SecilenMaddeler=""

if secenek1.checked="true" then SecilenMaddeler=SecilenMaddeler + " Dağcılık" End If

```
if secenek2.checked="true" then
SecilenMaddeler=SecilenMaddeler + " Yüzme"
End If
```

```
if secenek3.checked="true" then
SecilenMaddeler=SecilenMaddeler + " Yürüyüş"
End If
```

if secenek4.checked="true" then SecilenMaddeler=SecilenMaddeler + " Maketcilik" End If

```
if secenek5.checked="true" then
       SecilenMaddeler=SecilenMaddeler + " Resim"
       End If
       Response.write("Sectiginiz hobiler: " + SecilenMaddeler)
 End sub
</script>
<html>
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
  </head>
 <body>
     <form runat="server">
        <asp:CheckBox id="secenek1" Text=Dağcılık runat="server" /> <br>
       <asp:CheckBox id="secenek2" Text=Yüzme runat="server" /> <br><asp:CheckBox id="secenek3" Text=Yürüyüş runat="server" /> <br>
        <asp:CheckBox id="secenek4" Text=Maketcilik runat="server" /> <br>
       <asp:CheckBox id="secenek5" Text=Resim runat="server" /> <br>
        <asp:button id="dugme1" Text=gonder runat="server" onclick="yordam" />
    </form>
  </body>
</html>
```

ÖRNEK: Bir öğrencinin Adını Soyadını Textbox'dan alıp, RadioButton'la bölümünü seçip, CheckBox'la ilgili dersleri seçsin.

```
← → C 🗋 localhost:54996/WebSite2/Default3.aspx
 🕒 Bağlantıları Özelleştir 🕒 Ücretsiz Hotmail 🕒 Windows 🎜 Windows Media 🌔
 Ad Soyad Ali Su
  Bilgisayar
  ○ Makine
  ○ Metal
  ✓ Matematik
  ✓ Fizik
  Bilgisayar
  ✓ Kimya
      Kaydet
 Sayın Ali SuBilgisayar de okumaktasınız. Matematik Fizik Kimya Seçtiniz
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Ling;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
```

```
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class Default3 : System.Web.UI.Page
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void btnKaydet Click(object sender, EventArgs e)
    {
        string AdSoyad = txtAdSoyad.Text;
        string Bolum = rblBolum.SelectedItem.Text;
        string Dersler = null;
        for (int i = 0; i < 4; i++)
        {
            if (cblDersler.Items[i].Selected == true)
            {
                Dersler = Dersler + cblDersler.Items[i].Text + " " ;
            }
        }
        Label2.Text = "Sayın " + AdSoyad + Bolum + " de okumaktasınız. " +
Dersler + " Seçtiniz";
   }
}
```

HYPERLINK: Bir sayfadan başka sayfaya geçiş yapmamızı sağlayan (yada aynı sayfada başka bir yere) geçiş yapmamızı sağlayan bir kontrol elemanıdır.

Örnek: Sayfaya Karabük Teknik Eğitim Fakültesi'nin yazısını yazdırıp üzerine tıklayınca fakültenin sitesine gitsin.



<%@ Page Language="VB" codePage="28599"%>

```
<script runat="server">
sub Page_Load(sender as object, e as eventargs)
link1.NavigateUrl="http://ktef.karaelmas.edu.tr"
End sub
</script>
<html>
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>
<body>
<form runat="server">
<asp:hyperlink id="link1" Target="top" runat="server">
Karabük Teknik Eğitim Fakültesi
</asp:hyperlink>
</form>
</body>
</html>
```

Burada fakültenin yazısına değilde herhangi bir resme tıklayınca gitmesini istersek "ImageUrl" kullanmamız yeterlidir.

```
<asp:hyperlink ImageUrl="c:\resim1.jpg" id="link1" Target="top" runat="server">
Karabük Teknik Eğitim Fakültesi
</asp:hyperlink>
```

LINKBUTTON: Link verilen metine basınca kendisine bağlanmış olan yordamı işletir. Bu yordamda ise istenirse bir başka siteye yönlendirme yada istenen bazı komutların çalıştırılması sağlanır.

Örnek: Sayfaya bir metin yazdırın. Bu metinede bir link verin ve bu metne tıkladığınızda sayfaya birden 5 kadar sayıları yazsın.

```
1,2,3,4,5,
Bu yazıya tıkladığınızda 1 den 5 kadar sayıları yazacaktır
     <%@ Page Language="VB" codePage="28599"%>
     <script runat="server">
        sub yordam(sender as object, e as eventargs)
             dim i as integer
             for i=1 to 5
             response.write(i & "," )
             next i
        End sub
     </script>
     <html>
     <head>
     <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
      </head>
     <body>
      <form runat="server">
             <asp:LinkButton id="linkdugmesi" onclick="yordam" runat="server" >
             Bu yazıya tıkladığınızda 1 den 5 kadar sayıları yazacaktır
             </asp:LinkButton>
      </form>
     </body>
      </html>
```

ÖRNEK: Link verilebilecek tüm nesneler bir uygulama yapınız.



3 Git (Hyperlink Uygulaması)</asp:HyperLink> Ayrı bir yaprakda sayfa açılabilir.

**
**

4. Etiketi kullanarak Link Verme Ayrı bir yaprakda sayfa açılabilir.


```
<br />
    <br />
    <br />
  </div>
  </form>
</bodv>
</html>
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class Default4 : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
  }
  protected void Button1_Click(object sender, EventArgs e)
  {
    Response.Redirect("Default1.aspx"); //Burada ayrı bir sayfada
açtırma yapamayız.
  }
```

protected void LinkButton1_Click(object sender, EventArgs e)

```
{
    Response.Write("LinkButtona tıklandı");
}
```

IMAGE: Sayfaya konulacak resimleri çıkarmak ve kontrol etmek için kullanılır.

Örnek: 3 tane öğrencinin resimlerini C:/ altına koyun. Daha sonra sayfa koyacağınız bir DropDownList nesnesi içinden bu öğrencilerin resimlerini seçip Image nesnesi içinde öğrencinin resmini çıkmasını sağlayın. Herhangi bir şekilde öğrencinin resmi bulunamazsa en azından Image nesnesi içerisinde öğrencinin adı gözüksün.



```
<script runat="server">
Sub degistir(Sender As Object, E As EventArgs)
       Image1.ImageUrl="c:/"&DropDown1.SelectedItem.Value
       Image1.AlternateText =DropDown1.SelectedItem.Value '
DropDown1.SelectedItem.Text
End Sub
</script>
<html>
  <head>
  </head>
 <body>
  <form runat="server">
       <asp:Image id="Image1" ImageUrl="c:/varsayilan.jpg" AlternateText="Öğrenciler"
runat="server" />
       Öğrencilerin Resimleri:
       <asp:DropDownList id="DropDown1" OnSelectedIndexChanged="degistir"
AutoPostBack="True" runat="server">
       <asp:ListItem value="ahmet.jpg"> Ahmet </asp:ListItem>
       <asp:ListItem value="mehmet.jpg"> Mehmet </asp:ListItem>
       <asp:ListItem value="hasan.jpg"> Hasan</asp:ListItem>
       </asp:DropDownList>
 </form>
 </body>
</html>
```

TABLE, TABLEROW, TABLECELL: Programatik olarak tablo oluşturmamızı sağlar. Bu kontrol sayesinde tek bir html kodu yazmadan yordam içinde istediğimiz koşullarda Tablolar oluşturabiliriz.

Tablolar Row(satır) ve Cell (hücre=sütün manasında kullanacağız) lerden oluşur. Bunun için TableRow ve TableCell kontrolleri kullanılır. Öncelikle Row oluşturulur ve sonra bu row içinde Cell ler oluşturulur.

Örnek: Sayfaya iki tane metin kutusu koyun. Bunlardan birine satır sayısnı yazın diğerine ise sütun sayısınız yazın. Butona bastığınızda belirtilen sayıda tabloyu oluştursun.

```
Adres 🙆 http://localhost:82/NewFile.aspx
 satır1,hücre1 satır1,hücre2 satır1,hücre3 satır1,hücre4 satır1,hücre5
 satır2,hücre1 satır2,hücre2 satır2,hücre3 satır2,hücre4 satır2,hücre5
 satır3,hücre1 satır3,hücre2 satır3,hücre3 satır3,hücre4 satır3,hücre5
 satır4,hücre1 satır4,hücre2 satır4,hücre3 satır4,hücre4 satır4,hücre5
 satır5,hücre1 satır5,hücre2 satır5,hücre3 satır5,hücre4 satır5,hücre5
           Tablo Oluştur
5
    3
      <%@ Page Language="VB" codePage="28599"%>
      <script runat="server">
      Sub tabloOlustur(Sender As Object, E As EventArgs)
              Dim satirNo, hucreNo, satirSayisi, hucreSayisi as integer
              Dim satir as TableRow
              Dim hucre as TableCell
              satirNo=CInt(metinKutusu1.Text)
              hucreNo=CInt(metinKutusu1.Text)
              For satirSayisi=1 To satirNo
              satir= new TableRow()
                  For hucreSayisi=1 To hucreNo
                    hucre= new TableCell()
                    hucre.text="satır" & satirSayisi & ",hücre" & hucreSayisi
                    satir.Cells.Add(hucre)
                  Next hucreSayisi
              tablo1.Rows.Add(satir)
              Next satirSayisi
      End Sub
      </script>
      <html>
        <head>
        </head>
        <body>
        <form runat="server">
              <asp:Table id="tablo1" BorderWidth="2" BorderColor="black" Gridlines="Both"
      runat="server" />
              <asp:TextBox id="metinKutusu1" width="20px" runat="server" />
              <asp:TextBox id="metinKutusu2" width="20px" runat="server" />
              <asp:button Text="Tablo Oluştur" runat="server" OnClick="tabloOlustur" />
        </form>
       </body>
      </html>
```

Burada

Dim satir as TableRow Dim hucre as TableCell Satırları ile satır ve hücrelerimizi temsil edecek değişkenleri tanımlıyoruz. satir= new TableRow() hucre= new TableCell() ifadeleri ile önce yeni bir satır ve yeni bir hücre oluşturuyoruz. satir.Cells.Add(hucre) ifadesiyle oluşturduğumuz hücreyi satır'ın içine ekliyoruz. tablo1.Rows.Add(satir) ifadesiyle de oluşan satırı tablonun içine ekliyoruz.

Eğer hücreler arasındaki boşluğu ayarlamak istersek CellSpacing="3" satırını, hücre içinde metinle kenar arasındaki boşlukları ayarlamak isterseniz CellSpacing="2" şeklinde kullanarak tablo nesnesini oluşturduğumuz satılara yazınız.

Tablonun gözleri içerisine metin kutusu eklemek için aşağıdakine benzer satırlar araya eklenmelidir.

```
TextBox MetinKutusu = new TextBox();
MetinKutusu.ID = "TextBoxBilgi";
MetinKutusu.Width = 200;
hucre.Controls.Add(MetinKutusu);
satir.Cells.Add(hucre);
```

Örnek: Yukarıdaki Örneği C# kullanarak bir daha oluşturun

```
← → C Dicalhost:54996/WebSite2/Default5.aspx
 🗋 Bağlantıları Özelleştir 📋 Ücretsiz Hotmail 📋 Windows 🎤 Windows Media 🧰 IE'den Al
 DİNAMİK TABLO OLUŞTURMA (Runtime-Çalışma Esnasında Tablo Oluşturma)
 Satir Savisi 2
 Sutun Sayısı 3
     Asp.Net Tablosu Oluştur
                                   HTML Tablosu Oluştur
 00 01 02
 10 11 12
 STATİK TABLO OLUŞTURMA (Tasarım Esnasında Oluşturulur)
 Html Tablosu
 11
                 12
                              13
 21
                 22
                              23
 Asp.Net Tablosu
 11
               12
                            13
               22
 21
                            23
<html xmlns="http://www.w3.org/1999/xhtml">
   <title>Untitled Page</title>
   <style type="text/css">
      .style1
     {
```

<mark><%</mark>@ Page Language="C#" AutoEventWireup="true" CodeFile="Default5.aspx.cs" Inherits="Default5" <mark>%></mark>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"</p> "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```
<head runat="server">
       width: 112px;
    }
    .style2
    {
       width: 94px;
```

```
}
   .style4
   {
     width: 112px;
     height: 23px;
   }
   .style5
   {
      width: 94px;
     height: 23px;
   }
   .style6
   {
     height: 23px;
   }
   .style7
   {
     color: #0000FF;
   }
   .style8
   {
     color: #0066FF;
   }
 </style>
</head>
<body>
 <form id="form1" runat="server">
 <div>
   <span class="style8">DİNAMİK TABLO OLUŞTURMA (Runtime-Çalışma Esnasında Tablo
   Oluşturma)</span><br />
   <br />
   Satır Sayısı  
   <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
   <br />
   Sutun Sayısı
   <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
   <br />
   <asp:Button ID="Button1" runat="server" Height="38px" Text="Asp.Net Tablosu Oluştur"
     Width="203px" onclick="Button1_Click" />
   <asp:Button ID="Button2" runat="server" Height="36px" onclick="Button2_Click"
     Text="HTML Tablosu Oluştur" Width="225px" />
   <br />
   <br />
   <asp:Panel ID="Panel1" runat="server">
   </asp:Panel>
   <br class="style7" />
   <span class="style7">STATİK TABLO OLUŞTURMA (Tasarım Esnasında Oluşturulur)</span><br />
   Html Tablosu
   >
       11
       12
       13
     >
       21
```

```
22
         23
       <hr />
    Asp.Net Tablosu
    <asp:Table ID="Table1" runat="server" BorderStyle="Solid" BorderWidth="1px"
      Width="300px">
      <asp:TableRow runat="server" BorderStyle="Solid" BorderWidth="1px">
         <asp:TableCell runat="server" BorderStyle="Solid" BorderWidth="1px">11</asp:TableCell>
         <asp:TableCell runat="server" BorderStyle="Solid" BorderWidth="1px">12</asp:TableCell>
         <asp:TableCell runat="server">13</asp:TableCell>
       </asp:TableRow>
       <asp:TableRow runat="server">
         <asp:TableCell runat="server">21</asp:TableCell>
         <asp:TableCell runat="server">22</asp:TableCell>
         <asp:TableCell runat="server">23</asp:TableCell>
       </asp:TableRow>
    </asp:Table>
    <hr />
    <div id="DivDeneme" runat="server" >
    </div>
  </div>
  </form>
</body>
</html>
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Ling;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class Default5 : System.Web.UI.Page
  protected void Page_Load(object sender, EventArgs e)
  {
  }
  protected void Button1_Click(object sender, EventArgs e)
  Ł
    //ASP.NET TABLOSU OLUSTURMA
    Table Tablo; //Tablo nesnesini Runtime (çalışma esnasında) oluşturuyor.
    TableRow Satir; //İleri de oluşturacağımız satırı burada tanımladık.
    TableCell Hucre; //Aşağıda oluşturacağımız hucre nesnesini tanımladık.
```

int SatirSayisi=Convert.ToInt32(TextBox1.Text);

{

int HucreSayisi=Convert.ToInt32(TextBox2.Text);

```
Tablo=new Table();

for(int i=0;i<SatirSayisi;i++)

{

Satir =new TableRow();

for(int j=0;j<HucreSayisi;j++)

{

Hucre =new TableCell();

Hucre.BorderWidth = 1;

Hucre.Text =i.ToString() + j.ToString();

Satir.Cells.Add(Hucre);

}

Tablo.Rows.Add(Satir);

}
```

Panel1.Controls.Add(Tablo); //DivDeneme.Controls.Add(Tablo);

```
}
  protected void Button2_Click(object sender, EventArgs e)
  {
    //HTML TABLOSU OLUŞTURMA
    int SatirSayisi = Convert.ToInt32(TextBox1.Text);
    int HucreSayisi = Convert.ToInt32(TextBox2.Text);
    Response.Write("");
    for (int i = 0; i < SatirSayisi; i++)</pre>
    {
       Response.Write("");
       for (int j = 0; j < HucreSayisi; j++)</pre>
       {
         Response.Write("" + i.ToString() + j.ToString() + "");
       3
       Response.Write("");
    }
    Response.Write("");
  }
}
```

CALENDAR (TAKVİM): Bu nesneyi kullanarak sayfalarınızda istediğiniz şekilde bütün yılların ve ay ve günlerini gösterecek şekilde takvim oluşturabilirsiniz. Takvimden istediğiniz yerleri seçip ilgili kodları çalıştırabilirsiniz.

Örnek: En basit haliyle sayfada bir tane takvim oluşturun.

| < | ≤ Kasım 2004 ≥ | | | | | |
|-----------|----------------|-----------|----------------------|-----------|-----------|-----------|
| Pzt | Sal | Çar | Per | Cum | Cmt | Paz |
| <u>25</u> | <u>26</u> | <u>27</u> | <u>28</u> | <u>29</u> | <u>30</u> | <u>31</u> |
| <u>1</u> | <u>2</u> | <u>3</u> | <u>4</u> | <u>5</u> | <u>6</u> | 2 |
| <u>8</u> | <u>9</u> | <u>10</u> | <u>11</u> | <u>12</u> | <u>13</u> | <u>14</u> |
| <u>15</u> | <u>16</u> | <u>17</u> | <u>18</u> | <u>19</u> | <u>20</u> | <u>21</u> |
| <u>22</u> | <u>23</u> | <u>24</u> | <u>25</u> | <u>26</u> | <u>27</u> | <u>28</u> |
| <u>29</u> | <u>30</u> | <u>1</u> | <u>2</u> | <u>3</u> | <u>4</u> | <u>5</u> |

<%@ Page Language="VB" codePage="28599"%>

En basit haliyle bu satırları kullanarak bir takvim oluşturabilirsiniz. Aşağıdaki özellikleride deneyerek takvime yeni görünümler kazandırabilirsiniz.

DayNameFormat="full" SelectionMode="DayWeekMonth" selectWeekText="hafta" SelectMonthText="ay" NextPrevFormat="FullMonth" TitleStyle-BackColor="red"

Buradaki özellikleri biraz açıklayalım.

DayNameFormat= "Full", "FirsLetter", "FirstTwoLetters", "Short" olmak üzere toplam 4 değer alabilmektedir. Bunlar takvimimizdeki günlerin karakter olarak ne uzunlukta görüneceğini belirler.

SelectionMode="DayWeekMonth", "Day", "DayWeek", "None" olmak üzere 4 değer alabilir. Bu seçenekler kullanılarak kullanıcının hangi alanları seçebileceği kontrol edilmiş olur. "None" seçersek ne gün ne hafta nede ay seçmemize izin verir. Day sadece günlerin seçilmesine izin verir.

NextPrevFormat="FullMonth", "ShortMonth", "CustomText" değerlerinden birini alarak önceki ve sonraki ay isimlerinin hangi uzunlukta görüleceğini belirler.

Örnek: Sayfada bir takvim oluşturun. Takvimde istenen bir güne tıklayınca seçilen tarihi altta yazsın.



Burada takvimde bir seçim meydana gelirse "tarihsec" yordamı çalıştırılıyor. Bu yordamın içinde ise "takvim" nesnesinin seçilen tarihi kullanıcıya gösteriliyor.

OnDayRender olayı ve **DayRenderEventArgs** argümanı: Bu olay ve argümanı kullanarak takvimdeki bütün gün alanlarını programatik olarak düzenleyebiliriz. OnDayRender olayı takvimin her hücresi için belirtilen yordamı bir kez işletir. Bunu bir örnekle açıklayalım.

Örnek: Randevularımızı takvimin içerisinde göstermek için sayfaya bir tane DropDownList bir tane TextBox, bir tane Buton ve bir tane de Takvim nesnesi koyalım. DropDownList'in içerisinden o ayın günlerini seçelim. TextBox'ın içerisine ise Randevumuzun adını yazalım. Butona bastığımızda takvimin içinde belirtilen günün hucresine randevuyu yazsın.

```
Aylık Randevularınızı Giriniz
                                Randevu Ekle
15 🔹 Maaş
              Kasım 2004
<
                                      >
 Pzt
       Sal
                 Per
                       Cum
                             Cmt
                                   Paz
            Çar
  1
       2
             3
                   4
                         5
                              6
                                    2
  8
       9
             10
                        12
                              13
                  11
                                    14
 15
                        19
       16
            17
                  18
                              20
                                    21
Maaş
 22
       23
             24
                  25
                        26
                              27
                                    28
 29
       30
```

```
<%@ Page Language="VB" %>
<script runat="server">
```

```
Sub Hucreleritara(sender As Object, e As DayRenderEventArgs)
dim gun as CalendarDay
dim hucre as TableCell
```

```
gun = e.Day
hucre = e.Cell
If gun.Date.Day=dropdownlist1.SelectedItem.Value Then
hucre.Controls.Add(new LiteralControl("<br>" & textbox1.text))
End If
```

```
If gun.IsOtherMonth Then
hucre.Controls.Clear
End if
```

```
End Sub
```

```
</script>
<html>
<head>
</head>
<body>
  <form runat="server">
     <asp:Label id="Label1" runat="server">Aylık Randevularınızı Giriniz</asp:Label>
     <asp:DropDownList id="DropDownList1" runat="server">
          <asp:ListItem Value="1">1</asp:ListItem>
          <asp:ListItem Value="2">2</asp:ListItem>
          <asp:ListItem Value="3">3</asp:ListItem>
          <asp:ListItem Value="15">15</asp:ListItem>
          <asp:ListItem Value="29">29</asp:ListItem>
          <asp:ListItem Value="30">30</asp:ListItem>
          <asp:ListItem Value="31">31</asp:ListItem>
          <asp:ListItem></asp:ListItem>
       </asp:DropDownList>
```

```
<asp:TextBox id="TextBox1" runat="server"></asp:TextBox>
<asp:Button id="Button1" runat="server" Text="Randevu Ekle"></asp:Button>
<asp:Calendar id="Calendar1" runat="server"
OnDayRender="Hucreleritara"></asp:Calendar1" runat="server"
OnDayRender="Hucreleritara"></asp:Calendar1" runat="server"
OnDayRender="Hucreleritara"></asp:Calendar1" runat="server"
```

Butona basıldığında sayfa servera yenilenmek üzere gider. Bu esnada takvimin içindeki OnDayRender olayını gören server bu olayın gönderdiği yordamı takvimin bütün hucreleri için tarar. Yani her hucre için bir kez işletilir. Burada gun degişkeni takvim gününü, hucre degişkeni ise tablo hücresini temsil etmektedir.

Çalışma esnasında takvimin hücrelerinde mevcut ayın günlerinin öncesinde ve sonrasında diğer ayın günlerinin gözükmesini engellemek için aşağıdaki üç satır yordamın içine eklenmiştir. Bu satırlar kaldırılırsa diğer ayın günleri gözükmeye başlar.

If gun.IsOtherMonth Then hucre.Controls.Clear End if

ÖRNEK: YUKARIDAKİ ÖRNEĞİ ÇALIŞTIRIN.

| : | | Ma | ayıs 2 | 013 | | > |
|-----|-----|-----|--------|-----|-----|-----|
| Pzt | Sal | Çar | Per | Cum | Cmt | Paz |
| 29 | 30 | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default6.aspx.cs" Inherits="Default6" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
  <div>
    <br />
    <asp:Calendar ID="Calendar1" runat="server" ondayrender="Calendar1_DayRender"
      onselectionchanged="Calendar1_SelectionChanged"></asp:Calendar>
    <br />
    <asp:DropDownList ID="DropDownList1" runat="server">
      <asp:ListItem>1</asp:ListItem>
      <asp:ListItem>2</asp:ListItem>
      <asp:ListItem>3</asp:ListItem>
       <asp:ListItem></asp:ListItem>
```

```
</asp:DropDownList>
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button"</pre>
       Width="123px" />
    <br />
    <hr />
    <br />
  </div>
  </form>
</body>
</html>
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class Default6 : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
  }
  protected void Calendar1_SelectionChanged(object sender, EventArgs e)
  ł
    TextBox1.Text = Calendar1.SelectedDate.ToShortDateString();
  }
  protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
  ł
    CalendarDay gun=new CalendarDay("01.01.2013");
    TableCell hucre =new TableCell();
       gun = e.Day;
       hucre = e.Cell;
          if (gun.Date.Day==DropDownList1.SelectedItem.Value )
          {
          hucre.Controls.Add(new LiteralControl("<br>>" & TextBox1.text));
         }
          if (gun.IsOtherMonth)
          {
            hucre.Controls.Clear;
          }
  }
  protected void Button1_Click(object sender, EventArgs e)
  {
  }
}
```

ASP.NET'DE RANDOM DOSYA KULLANIMI

Bilgilerimizi daha önceden başka programlar tarafından hazırlanmış database programlarında saklayabileceğimiz gibi kendimizin hazırlayacağı dosyalarda da saklayabiliriz. Bunun için istediğimiz kayıda ulaşmamızı sağlayan ve kayıtların bir yapısınız oluşturmamıza imkan veren RANDOM dosyaları kullanabiliriz. Tıpkı bir database kullanıyormuş gibi.

ÖRNEK: Sayfaya üç tane textbox ve iki tane de buton koyun. Textboxlara Ad, Soyad, ve Yaş bilgilerini girdikten sonra birinci butona bastığınız bunları bir random dosyaya kaydetsin. İkinci butona bastığınızda ise Random dosyadaki tüm bilgileri tablo içinde listelesin.

```
<% @ Page Language="VB" %>
<script runat="server">
  'Insert page code here
   Structure Kayit
       <VBFixedString(15)> Public ad As String
       <VBFixedString(30)> Public soyad As String
       <VBFixedString(10)> Public numara As String
       <VBFixedString(30)> Public email As String
       <VBFixedString(20)> Public sifre As String
     End Structure
     dim alanlar as kavit
'kayıt iÅŸlemi yapılıyor...
  Sub Button1_Click(sender As Object, e As EventArgs)
   FileOpen(1, "c:\bilgiler1.dat", OpenMode.Random, OpenAccess.ReadWrite, OpenShare.Default,
Len(alanlar))
      alanlar.ad=textbox1.text
      alanlar.soyad=textbox2.text
      alanlar.numara=textbox3.text
      alanlar.email=textbox5.text
      alanlar.sifre=textbox6.text
  FilePut(1, alanlar, cint(textbox4.text))
  textbox4.text=cstr(cdbl(textbox4.text)+1)
  me.textbox1.text=""
  me.textbox2.text=""
  me.textbox3.text=""
  me.textbox5.text=""
  me.textbox6.text=""
  me.textbox7.text=""
   FileClose(1)
  End Sub
'okuma işlemi yapılıyor...
  Sub Button2 Click(sender As Object, e As EventArgs)
   FileOpen(1, "c:\bilgiler1.dat", OpenMode.Random, OpenAccess.ReadWrite, OpenShare.Default,
Len(alanlar))
      FileGet(1, alanlar, cint(textbox4.text))
      textbox1.text=trim(ucase(alanlar.ad))
      textbox2.text=trim(ucase(alanlar.soyad))
      textbox3.text=trim(ucase(alanlar.numara))
      textbox5.text=trim(ucase(alanlar.email))
```

textbox6.text=trim(ucase(alanlar.sifre)) FileClose(1)

End Sub

Sub Page_Load (sender as object, e as eventargs)

FileOpen(1, "c:\bilgiler1.dat", OpenMode.Random, OpenAccess.ReadWrite, OpenShare.Default, Len(alanlar))

label1.text=lof(1)/len(alanlar) FileClose(1)

end sub

Sub Button4_Click(sender As Object, e As EventArgs)

End Sub

Sub Button3_Click(sender As Object, e As EventArgs)

End Sub

```
Sub Button4_Click_1(sender As Object, e As EventArgs)
if not eof(1) then
textbox4.text=val(textbox4.text) + 1
fileget(1,alanlar,cint(textbox4.text))
me.textbox1.text=trim(ucase(alanlar.ad))
me.textbox2.text=trim(ucase(alanlar.soyad))
me.textbox3.text=trim(ucase(alanlar.numara))
me.textbox5.text=trim(ucase(alanlar.email))
me.textbox6.text=trim(ucase(alanlar.sifre))
end if
End Sub
```

```
Sub Button5_Click(sender As Object, e As EventArgs)
if val(textbox4.text)>1 then
textbox4.text=val(textbox4.text) - 1
fileget(1, alanlar,cint(textbox4.text))
me.textbox1.text=trim(ucase(alanlar.ad))
me.textbox2.text=trim(ucase(alanlar.soyad))
me.textbox3.text=trim(ucase(alanlar.numara))
me.textbox5.text=trim(ucase(alanlar.email))
me.textbox6.text=trim(ucase(alanlar.sifre))
end if
```

End Sub

```
</script>
<html>
<head>
</head>
<body>
<form runat="server">
```

```
<table style="WIDTH: 435px; HEIGHT: 252px" height="252" width="435" align="center"
bgcolor="#ffe0c0" border="1">
     Ad:           
        <asp:TextBox id="TextBox1" runat="server" Width="155px"
BorderStyle="Groove"></asp:TextBox>
        \langle tr \rangle
        Soyad:
        <asp:TextBox id="TextBox2" runat="server" Width="156px"
BorderStyle="Groove"></asp:TextBox>
        Numara:
        <asp:TextBox id="TextBox3" runat="server" Width="156px"
BorderStyle="Groove"></asp:TextBox>
        E mail:
        <asp:TextBox id="TextBox5" runat="server" Width="158px"></asp:TextBox>
        Å□ifre:
        <asp:TextBox id="TextBox6" runat="server" Width="158px"></asp:TextBox>
        \langle tr \rangle
        Å□ifre Tekrar:
        <asp:TextBox id="TextBox7" runat="server" Width="158px"></asp:TextBox>
        Kayıt No:
           <asp:TextBox id="TextBox4" runat="server" Width="68px"
BorderStyle="Groove">1</asp:TextBox>
```

```
<asp:Button id="Button1" onclick="Button1_Click" runat="server" Width="77px"
Text="Kaydet"></asp:Button>
                                       <asp:Button id="Button2" onclick="Button2_Click" runat="server" Width="78px"
Text="Oku"></asp:Button>
                                 Toplam Kayıt Sayısı:
                                  <asp:Label id="Label1" runat="server"></asp:Label>
                            <table style="Z-INDEX: 102; LEFT: 92px; WIDTH: 441px; POSITION: absolute; TOP: 362px;
HEIGHT: 29px" height="29" width="441" align="center">
                      <asp:CompareValidator id="CompareValidator1" runat="server"
ControlToValidate="TextBox7" ErrorMessage="şifreler aynı olacak"
ControlToCompare="TextBox6"></asp:CompareValidator>
                                       <asp:RegularExpressionValidator id="RegularExpressionValidator1" runat="server"
ValidationExpression="\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"ControlToValidate="TextBox5" and a started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started by the started 
ErrorMessage="e mail yanlıÅŸ"></asp:RegularExpressionValidator>
                                 <!-- Insert content here -->
      </form>
</body>
</html>
```

KONTROLLERİN (NESNELERİN) ve PROGRAMATİK OLARAK OLUŞTURULMASI

Bilindiği gibi sayfalarımızda kullandığımız kontrolleri (nesneleri) Html kodları arasındaki uygun yerlere asp.net den dolayı gelen özel etiketler(tag) kullanarak kontrollerimizi oluşturuyorduk. Tabii bu işlemi henüz sayfamız runtime olarak çalışmadan önce kendimiz kod yazarak oluşturuyoruz.

Örnek: Şu satırlar ekranda butonun oluşmasını sağlıyordu.

<html>

... <body>

```
<asp:Buton id="Button1" Text="Gönder" onclick="yordam" runat="server" /> </body>
```

</html>

Peki biz herhangi bir kontrolü program çalışırken oluşturmak istersek ne yapacağız?

Bunun için sayfamızda **Panel, PlaceHolder** kontrollerinin bulunması gerekmektedir. Yada bu kontroller olmadan **Page**' e yani direk sayfaya ekleme yapabilirsiniz.

Herhangi bir kontrolü sayfada programatik olarak oluşturmak demek, kontrol oluşturulmadan önce kod yazım esnasında <script>... </script> etiketleri içinde kontrolü tanımlamak ve daha sonra program çalışırken de Panel nesnesi içinde o nesneyi oluşturmak demektir.

(Hatırlatma: Panel nesnesi aslında diğer kontrolleri barındırmak ve bunları istendiğinde gizlemek yada göstermek için kullanılır. Yani bu kontrol kullanılarak üzerine yerleştirilen nesneleri görünür veya görünmez yapabiliriz.)

A) Panel Kontrolü kullanarak Runtime nesne oluşturma

Örnek: Sayfaya bir tane Panel kontrolü koyarak, sayfa yükleme esnasında bu panelin içerisinde bir tane textbox, bir tane de buton oluşmasını sağlayın.

| Adres | littp://localhost:8 | 8080/NewFile.aspx |
|-------|---------------------|-------------------|
| - | | |
| Gö | önder | |

<%@ Page Language="VB" Debug="true" CodePage="28599" %>

<script runat="server">

Sub Page_Load(obj as object, e as eventargs)

Dim metinkutusu as textbox=new textbox

'TextBox oluşturuyoruz... Panel1.Controls.Add(metinkutusu) Panel1.Controls.Add(new LiteralControl("
")) 'Burası panel1 içinde alt satıra inmeyi sağlıyor

'Buton oluşturuyoruz...

Dim Denemebuton as button=new button Panel1.Controls.add(denemebuton)

```
Denemebuton.Text="Gönder"
Denemebuton.id="postala"
```

Panel1.Controls.Add(new LiteralControl("
")) 'Burası panel birin içinde alt satıra inmeyi sağlıyor

End Sub

```
</script>
<html>
<head>
</head>
<body>
<form runat="server">
<asp:Panel id="Panel1" runat="server" Height="332px" Width="529px"></asp:Panel>
</form>
</body>
</html>
```

B) PlaceHolder Kontrolü kullanarak Runtime nesne oluşturma

PlaceHolder kontrolü runtime esnasında oluşturulan nesneleri barındırmak için kullanılır. Bu nedenle panel benzer. Fakat görünür bir arayüzü yoktur. İçindeki nesneleri gösterme ve gizleme gibi bir uygulama yapma imkanı yoktur.

Örnek: PlaceHolder kullanarak Runtime esnasında sayfada bir sabit yazı, bir textbox ve bir buton oluşturun. Butonun üzerinde ise "Gönder" ibaresi yazsın.

| Adres | http://localhost:8080/NewFile.aspx | | | |
|-------|------------------------------------|--|--|--|
| Isim | ı: Gönder | | | |

<% @ Page Language="VB" Debug="true" CodePage="28599" %> <script runat="server">

Sub Page_Load(obj as object, e as eventargs)

'Metin kutusu oluşturuyoruz..

PlaceHolder1.Controls.Add(new LiteralControl("Isim: "))

Dim metinkutusu as textbox=new textbox metinkutusu.id="met" PlaceHolder1.Controls.Add(metinkutusu)

'Buton oluşturuyoruz... Dim Denemebuton as button=new button Denemebuton.Text="Gönder" Denemebuton.id="gon"

PlaceHolder1.Controls.Add(Denemebuton)

End Sub

```
</script>
<html>
<head>
</head>
<body>
<form runat="server">
 <asp:PlaceHolder id="PlaceHolder1" runat="server"></asp:PlaceHolder>
</form>
</body>
</html>
```

C) Sayfada (Page) Runtime esnasında Kontrollerin Oluşturulması

Bu uygulama da aynı diğer iki uygulama gibi yapılır. Kontrol eklenirken şu şekilde yazılır.

Page.Controls.Add(eklenecek_kontrol)

Oluşturulan kontrollere Olay Tanımının Yapılması

ÖRNEK: TAMAMLANACAK.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default9.aspx.cs"</p>
Inherits="Default9" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div id="DivDeneme" runat="server" >
        <br />
        <asp:Button ID="Button1" runat="server" Height="22px"
onclick="Button1 Click"
            Text="Button" Width="145px" />
        <br />
        <br />
    </div>
    </form>
</body>
</html>
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
```

```
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class Default9 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        TextBox MetinKutusu = new TextBox();
        MetinKutusu.ID = "Metin";
         DivDeneme.Controls.Add(MetinKutusu);
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Response.Write(((TextBox)DivDeneme.FindControl("Metin")).Text);
    }
}
← → C 🗋 localhost:54996/WebSite2/Default9.aspx
🕒 Bağlantıları Özelleştir 🕒 Ücretsiz Hotmail 🕒 Windows ಶ Window
 ALİ
 Button
 ALİ
```

ASP.NET VE NESNEYE YÖNELİK PROGRAM KULLANIMI

Nesneye yönelik programcılıkta kullanılan "sınıf" ve "nesne" kavramlarının tanıtımı ve kendimize ait nesnelerin nasıl oluşturulduğunu ve bunların Asp.Net sayfalarında nasıl kullanıldığı bu bölümde anlatılacaktır.

Büyük bir projeyi bir yazılımcının tek başına geliştirmesi mümkün değildir. Böyle durumlarda uygulamayı bölümlere ayırmak gerekmektedir. Başkasının yaptığı uygulamayı kendi uygulamamız içinde kullanmamız bunun çözümüdür. Fakat bu işlem o kadar kolay olmamıştır. Diğer uygulamacılar farklı dillerde yazdığında bunları nasıl bağdaştıracağız. Her ikisi de aynı dili kullansa bile bu iki program birbirini nasıl tanıyacak.

İşte bu noktada nesneye yönelik programlama fikri ortaya çıktı. Buna göre her uygulama programatik nesnelerden oluşacaktır.

Her nesnenin özellikleri ve işlevleri vardır. Gerçek hayatta da böyledir. Örneğin bir kol saatinin (ki nesnedir) hem özelliği (rengi, biçimi, ağırlığı gibi) hemde işlevleri (zamanı göstermesi, tarihi göstermesi vs) vardır.

Bu konuda geliştirilen teknolojiler OLE (Object Linking and Embedding-1990), COM (Compenent object model-1993) ve daha sonralarıda COM+, DCOM versiyonları oluşturulmuştur. En sonunda da NET oluşturulmuştur.

Uygulamaları parçalara bölmek olağan üstü bir esneklik sağlamış ve yazılım fiyatlarını ucuzlatmıştır.

SINIF (CLASS) KAVRAMI

Her nesne özellik ve işlevleri bağlı bulundukları Sınıflardan alırlar. Gerçek hayattada böyledir. Televizyon bir nesnedir ama elektronik eşyaların grublandırıldığı bir sınıfa aittir. Papatya bir nesnedir ama çiçeklerin toplandığı bir sınıfa aittir. Ve tüm çiçeklerde bulunan ortak özelliklere aittir.

Nesneler sınıflar sayesinde oluşturulurlar. Sınıflar nesnelerin program alt yapısını oluştururlar. Bir nesneyi oluşturmak için buna ait sınıfla irtibata geçip nesneyi oluşturabiliriz. Bir nesneyi oluşturduktan sonra, onun bağlı bulunduğu sınıfa ait özellikleri kullanabiliriz.

Bir sınıfı VB veya C# ile yada başka programlarla oluşturabiliriz. Hangisiyle oluşturursak oluşturalım sayfadaki kullanımı açısından bir fark yoktur.

Sınıfların (Class) ların toplandığı daha üst gruba ise "Namespace" isim uzayı denir.

Örnek: Dört işlemi yapan bir sınıf oluşturun ve bunları sayfada kullanın.

1. Notdefterini açın aşağıdaki ifadeleri yazın.

Imports System Namespace matematik *'kendimiz bir tane isim uzayı oluşturduk* Public class dortislem 'sınıf oluşturduk Public Function topla (ByVal A As integer, ByVal B as integer) As integer topla=A+B End Function End class End Namespace

Açıklamaları:

Imports System: NET de kullanılan bir bildirimdir. Anlamı NET'in "system" adlı namespace'inden faydalanabilir hale getirmektir.

<u>Namespace Matematik</u>: Bu satırla kendimize ait bir Namespace oluşturmuş oluyoruz. Bu namespace'in içerisinde daha birçok class oluşturabiliriz.

<u>Public Class Dortislem</u> : Bu satırla Dortislem adında bir sınıf (class) oluşturuyoruz. Puclic olması dışarıya açık manasında bu sınıfa herhangi bir yordam tarafından ulaşılabilir.

<u>Public Function toplama(...)</u> : satırları ile genele açık bir yordam (fonksiyon) oluşturulmuş oluyor. (yordam ile function benzer fakat çok küçük bir fark vardır.)

2.Bu dosyayı bilgisayarda D nin altına matematiksınıfı.vb olarak kaydedin.

Not::Public olması herkese açık demektir. Başka kişiler dışardan bunu kullanabilir.

Not: Subrutine dışarıdan değerler alabilir sonunda bir işlem yapar ama geri değer göndermez. Fonksiyon değer alır ve sonuçta çıkış değeri üretir. Private de kullanabilirdik o zaman herkes kullanamazdı.

Not: com, com+, ole, drom, net nesneye yönelik programları kullanan teknolojiler. ama bunların en sonuncusudur.

Not: .exe ile .dll derlenmiş program parçacıkları. yani program 1 ve 0 lara dönüştürür. derleme (compile) işlemi yazılan programı bilgisayarın anlayacağı 1 ve 0 lara çevirmektir.

Exe direk çalışır. .dll ise başka programlar vasıtasıyla çağrılarak çalıştırılır.

3. kodların .dll ye dönüştürülmesi.(1.yöntem)

- Bunun için .NET framework ile gelen vbc.exe programı kulanılır.bu programı kullanmak için cmd.exe çalıştırılır.
- Start(başlat)>run(çalıştır)>cmd.exe enter
- Açılan pencereden matematiksinifi.vb uzantılı dosyanın olduğu yere gidin ve şunları yazın. Vbc/tilibrary/out:toplam.dll matematiksinifi.vb

Bu işlem sonunda toplam dll dosyası çalıştığınız dizinde oluşacaktır. * Bu dosyanın mutlaka bin altında bulunması gerekir. (bu konuya kitaptan bak. 109 dan sonra)

4.kodların dll ye dönüştürülmesi(**2.yöntem**)

- Kodları visual studio da yaz.
- Açılan ilk pencerede create Project seçildikten sonra tekrar açılan pencereden "class library" uygulaması seçilir.
- Class library dosyasına programı yazıyoruz .yani not defterine yazdığımız kodları buraya yapıştırıyoruz.
- Class library üzerinde solution penceresinden sağ tuşla build seçilir. Bu işlemden sonra oluşturulan proje altında "bin" klasörü içerisinde "dll" dosyası oluşur.
- Yeni bir web projesi oluşturulur. Web sayfasında kod bölümünde en üst alana imports "class_adı" tanımlanır.

Kesinlikle oluşturduğunuz dll dosyası studyonun altında bin dosyasının altında olacaktır.yoksa kendimiz oraya atıyoruz

5. VS(visual studio) de hazırlanan .dll dosyası bin dizininin altına otomatik olarak atılır. Eğer 1. yöntemle kendimiz oluşturduysak yine bu dosya bin in altına atılmalıdır. VS bu dosyayı bin in altına sinif.dll olarak atar.
6..dll dosyalarının (ki bunlara asembly denir) asp.net sayfasında kullanılması. Bunun için webmatrix kullanarak şu satırları deneyin.

dim nesne as new dortislem response.write("sonuç=" & nesne.topla(textbox1.text,textbox2.text))

Ödev Bir tane hesaplama nesnesi, bir tane zaman nesnesi ve bir tanede Öğrenci nesnesi oluşturun. Öğrenci nesnesi kullanıldığında okul nosunu gönderin ve o da bize öğrencinin bütün özelliklerini listelesin. Hesaplama nesnesinde a ,b sayısı birde işlemi gönderin ve bize sonucu versin. Zamanla ilgili olarak şu anki zamanın saatini dakikasını saniyesini, tarihi bize versin.

Sınıf (Class) Oluşturmak

Project→Add Class ile aspx.vb sayfamıza class ekleyebiliriz. Class dosyalarının

derlenebilmesi için Projemize "App_Code" isminde bir klasör eklemeliyiz. Bu klasörü

eklemek için bir üst dizine sağ tıklayıp "Add New Item" seçilir. Açılan pencereden Class seçeneği seçildiğinde App_Code klasörü oluşturulur.



ÖRNEK: MesajSinifi adından yeni bir sınıf oluşturarak bu sınıfın içerisine MesajOlustur isminde bir fonksiyon yazın. (bu fonksiyon sınıftan oluşturulan nesnenin metodu olacak).Daha sonra bu sınıfı kullanarak mesaj adında bir nesne oluşturun. Bu nesneyi deneyin.

| App_Code/Class1.vb* Default.aspx.vb* Default.aspx* Start Page | | | | |
|--------------------------------------------------------------------------------|---------------|--|--|--|
| 🎬 (General) | 💌 🎬 (Declarat | | | |
| Imports Microsoft.VisualBasic | | | | |
| Public Class MesajSinifi | | | | |
| Public Function MesajOlustur (ByVal isim As String Return "Merhaba " & isim | g) As String | | | |
| - End Function | | | | |
| - End Class | | | | |

Burada bir fonksiyon oluşturup sayfadan gelen değişkenin önüne "merhaba" ifadesini ekleyip sayfaya geri gönderiyor. Bu class'ı sayfamızda aşağıdaki gibi kullanıyoruz.

| / | Start Pa | age / App_Code/Class1.vb / Default.aspx.vb / Default.aspx | | |
|---|----------|------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|-----------------------|
| Ŭ | (Gene | ral) | (Declarations) | |
| | ⊡ Par | tial Class _Default Inherits System.Web.UI.Page | | |
| | 8 | <pre>Protected Sub Button1_Click1(ByVal sender As Object, Dim mesaj As New MesajSinifi Label1.Text = mesaj.MesajOlustur(TextBox1.Text)</pre> | ByVal e As System.EventArgs) | Handles Button1.Click |
| | - End | End Sub Class | | |

Burada MesajSinifi dan yeni bir "nesne" oluşturuluyor ve bu nesnenin MesajOlustur metodu ile sayfamıza mesaj yazdırılıyor. Aynı Sınıf oluşturma işlemleri direk .aspx.vb sayfasında da yapılabilirdi. İleride bu konulara bakacağız.

| ttp://localhost:49158/WebSite1/Default.aspx | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| mywebsearch • | D |
| 🔆 🍄 🄏 Untitled Page | |
| | _ |
| Button Ali | |
| ÖRNEK | |
| ← → C □ localhost:54996/WebSite2/Default8.aspx | Solution Explorer - S Solution Explorer - S Solution 'WebSite2' (1 project) C App_Code Class1.cs App_Data |
| 🗋 Bağlantıları Özelleştir 📋 Ücretsiz Hotmail 🗋 Windows 郑 Win | |
| Sayı 1 : 6 | |
| Sayı 2 : 2 | ian 📴 Default4.aspx ian 🛅 Default5.aspx ian 🖬 Default5.aspx |
| Topla Cikar | |
| 7 | < > |
| <pre><%@ Page Language="C#" AutoEventWireup="true" C Inherits="Default8" %></pre> | <pre>odeFile="Default8.aspx.cs"</pre> |
| <pre><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Tr "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transit</pre> | ansitional//EN" ional.dtd"> |
| <html xmlns="http://www.w3.org/1999/xhtml"> <head runat="server"> <title>Untitled Page</title> </head> <body> <form id="form1" runat="server"></form></body></html> | |
| <div></div> | |
| Say1 1 : <asp:textbox id="TextBox1" runat="serve </td><td>r"></asp:textbox> | |
| Say1 2 : <asp:textbox id="TextBox2" runat="serve </td><td>r"></asp:textbox> | |
| <pre></pre> | <pre>onclick="Button1_Click"</pre> |
| <pre>desc= lopid" /></pre> | <pre>onclick="Button2_Click"</pre> |
| <pre>'Text="CIKar" /></pre> | ext="Label"> |

```
<br />
    </div>
    </form>
</body>
</html>
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public partial class Default8 : System.Web.UI.Page
{
    MatematikIslemleri Islem = new MatematikIslemleri();
    int A = 0;
    int B = 0;
    int C;
    protected void Page Load(object sender, EventArgs e)
    {
    }
    protected void Button1 Click(object sender, EventArgs e)
        A =Convert.ToInt32(TextBox1.Text);
        B = Convert.ToInt32(TextBox2.Text);
        //C= A + B; //1.. En ilkel yöntem. İşlemin direk giriş çıkış
satırları arasında gerçekleştirilmesi
        //C = Topla(A, B); //2...İkinci daha gelişmiş yöntem Fonksiyon
kullanmaktır.
        C = Islem.Topla(A, B); //3. Yöntem... Nesne (Class) kullanarak
kodları ayrı bir yerde oluşturup tüm projede herkesin kullanabileceği bir
yapıya dönüştürme.
        Label1.Text = C.ToString();
    }
    //public int Topla(int a, int b)
    //{
    11
         int c = a + b;
    11
         return c;
    //}
```

```
protected void Button2 Click(object sender, EventArgs e)
    {
        A = Convert.ToInt32(TextBox1.Text);
        B = Convert.ToInt32(TextBox2.Text);
        C = Islem.Cikar(A, B);
        Label1.Text = C.ToString();
    }
}
using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
public class MatematikIslemleri
{
    public int Topla(int a, int b)
    {
        int c = a + b;
       return c;
    }
    public int Cikar(int a, int b)
    {
        int c = a - b;
       return c;
    }
}
```

ASP.NET'DE SAYFALARIN DERLENMESİ

Asp.net sayfalarını derleyerek çalıştırır. Yani kullanıcının yazdığı kodlar sunucuda 1,0 kodlarına dönüştürülür ve her çağrıldığında hızlı bir şekilde sayfa işletilir. Hazırladığımız bir sayfayı Studio Net de F5 basarak yada play tuşuna basarak derleriz ve çalıştırırız. Bu konuyla ilgili olarak 3 türlü derleme işleminden bahsedebiliriz.

a) Ön Derleme (Precompiling): Herhangi bir siteyi server a attığımızda yada yerel bilgisayarımızda bir proje geliştirdiğimizde, projemizin bulunduğu dizin içindeki tüm sayfalar ilk çalıştırıldığı esnada derlenirler. Bu esnada sitenin çalışmasında bir miktar gecikme gözlenebilir fakat daha sonra sayfalar çok hızlı açılır. Peki kodlarda bir değişiklik olursa ne olacak? İşte cevabı.

b) Dinamik Derleme: Asp.Net de kodlar dinamik olarak da derlenir. Herhangi bir kodda değişiklik olduğu anda sadece o sayfa yada ilgili kısım ilk çalıştırıldığında tekrar derlenir. Asp.Net bu optimizasyonu başarı ile yapar.

Yukarıdaki örneğin Function olduğu kısmı şöyle değiştirip kaydedin. Derleme (F5 basmadan) yapmadan browser daki butona tekrar tıklayın. Yeniden Derleme yapmadan sayfanın değiştiğini göreceksiniz.

```
Public Function MesajOlustur(ByVal isim As String) As String
Return "Selam " & isim
```

Button Ali Selam Ali

Dağıtım İçin Derleme (Precompiling For Deployment)

Önceki sürümlerde bütün bir projenin kodları tek bir .dll(assembly) de derlenip daha sonra çalıştırılırdı. Bu dll de özel bir dizin olan bin dizininde saklanırdı. Ancak ASP .NET 2 de bu model işlemiyor. Çünkü ASP .NET 2 dinamik derlemeyi kullanıyor. En küçük değişiklikleri algılayıp otomatik olarak derliyor.

Proje tamamlanıp dağıtım aşamasına gelindiğinde kodlara ulaşılmasını istemiyorsak projeyi ASP .NET 2 de dll(assembly)'e çevirip kullanıcıya verebiliriz. Bunun için "aspnet compiler" isimli bir araç kullanılır.

Komut satırında çalışan bu araç, projenin tamamını dağıtıma hazır hale getirir. Bu aracı kullanarak daha önce oluşturulan programları dağıtıma hazır hale getirecek olursak;

Başlat→Çalıştır→cmd.exe yazıp komut satırı açılır. Framework'un bulunduğu dizine kadar inilir.



Aynen ekran görüntüsündeki gibidir. aspnet compiler in komut ifadesi şöyledir:

Aspnet_compiler –v /<WebSiteİsmi> -p <KaynakDosyalarınYeri> <KopyalanacakYer>

<u>< WebSiteİsmi ></u> istenen isim verilebilir.

<u>< KaynakDosyalarınYeri ></u> derleme yapacağınız uygulamanın adresi.

< KopyalanacakYer > derleme yapılacak çıktılar bu klasöre atılır. Aralarda boşluk kullanılır büyük küçük işaretleri konmaz. Sonuç olarak şu şekilde olacaktır.

Aspnet_compiler –v /karabuk –p C:\Documents and Settings\Admin\Belgelerim\denemelerim c:\derleme_deneme

bu ifadeyi komut satırına yazıp enter tuşuna basılıp çalıştırdığımızda kodlar **derleme_deneme** klasöründe derlenecektir.

Not: Denedim Vistada yollarla ilgili problem oldu sanırım çalıştıramadım.....

DATABINDING YÖNTEMİ İLE NESNELERE VERİLERİN BAĞLANMASI

Veri bağlama işlemi, verilerin (bilgilerin) sayfamızda bulunan bir kontrole (nesneye) bağlanması işlemidir. Bu işlem sayesinde sayfada bulunan nesnenin görünüm özellikleri kullanılarak veri istendiği şekilde görüntülenebilir. Bu nesneneler aynı zamanda görüntülenen bilgilerde değişiklikler yapıldığında bu değişikliklerin verikaynağına (veritabanına) yansıtılmasını da sağlarlar.

Verileri kontrollere bağlamak için iki yöntem kullanılır.

- a) Kontrolün DataSource özelliği kullanılarak: Bu yöntem biraz daha karmaşık olup özel sunucu kontrollerinde kullanılır.
- b) Data_Binding (veri bağlama) ifadesi kullanlarak: Bu yöntem her yerde ve her kontrol için kullanılabilir.

Veri kaynağından alınan bilgiler Kontrollerde görüntülenirken aşağıdaki format kullanılır ve bu format html kodları arasında bulunmalıdır. Bu kod sayesinde bağlama işlemi yapılır.

<%# Degisken %>

Bu ifadenin çalışması için .DataBind() metodunun çalıştırılması lazımdır.

Dikkat edilmesi gereken bir konuda verinin yapısı ve veriyi görüntüleyecek olan kontrolün yapısıdır. Bu ikisi uyumlu olmalıdır.

Örnek: Veri kaynağı olarak iki tane değişken tanımlayın. Bu değişkenlere bilgileri yükleyin. Bu bilgileri sayfada görüntüleyin. (Not: bu işlemi daha önceden Response.Write(degisken) olarak yapıyorduk. Önce bu yöntemi deneyin.)

a) 1. Yöntem: Response. Write kullanarak görüntüleme.



<% @ Page Language="VB" %> <script runat="server">

Dim ad as string= "Ali" Dim soyad as string="Su"

Sub Page_Load(obj as object, e as eventargs)

Response.Write(ad & " " & soyad)

End Sub

```
</script>
<html>
<head>
</head>
<body>
<form runat="server">
```

</form> </body> </html>

b) 2. Yöntem: Databind() metodu kullanarak görüntüleme.



<% @ Page Language="VB" %> <script runat="server">

Dim ad as string= "Ali" Dim soyad as string="Su"

Sub Page_Load(obj as object, e as eventargs)

Page.DataBind()

End Sub

```
</script>
<html>
<head>
</head>
<body>
<form runat="server">
<%#ad%>
<%#soyad%>
</form>
</body>
</html>
```

Burada ad ve soyad değişkenleri doğrudan sayfanın görünen kısmına bağlanmıştır. Page.DataBind() ifadesi ile sayfadaki tüm kontroller için bağlama işlemini gerçekleştirecektir. Bu metod sayfanın tamamı için degil sadece bir nesne içinde kullanılabilirdi (bknz: aşağıdaki diğer örnek).

Örnek: Veri kaynağı olarak bir dizi tanımlayın. Bu dizideki bilgileri bir DropDownList 'in içinde görüntüleyin.



a) 1. Yöntem: Daha önceden kullandığımız yöntem.

```
Sub Button1_Click(sender As Object, e As EventArgs)

Dim x(3) as string

Dim a,b,c as string

x(0)="ali"

x(1)="yasa"

x(2)="kanun"

dim i as integer

For i=0 to 2

DropDownList1.items.add(x(i))

Next

End Sub
```

b) 2. Yöntem: DataBinding Kullanarak verileri görüntüleme

<% @ Page Language="VB" %> <script runat="server">

Dim isimler() as string={"Oya","Can","Cem"}

Sub Page_Load(obj as object, e as eventargs)

Page.DataBind()

End Sub

```
</script>
<html>
<head>
</head>
<body>
<form runat="server">
```

<asp:DropDownList id="DropDownList1" **DataSource='<%# isimler %>'** runat="server"> </asp:DropDownList>

</form> </body> </html>

Burada isimler() değişkenimiz içinde bir çok bilgiyi tutan bir dizi olduğundan DropDownList 'in yapısına uygundur. Eğer 'ad' değişkenini görüntüleseydik harfleri alt alta sıralardı.

Örnek: Sayfanın geneline değilde bir Label nesnesine veri bağlantısı yapınız. Bu işlem sayfa yüklendiğinde değil bir butona tıklandığında gerçekleşsin.

| Adres | 🕘 http://localhost:8080 | | |
|-------|-------------------------|--|--|
| ali | Button | | |

```
<% @ Page Language="VB" %>
<script runat="server">
  Sub Button1_Click(sender As Object, e As EventArgs)
  Label1.DataBind()
  End Sub
</script>
<html>
<head>
</head>
<body>
  <form runat="server">
    <asp:TextBox id="TextBox1" runat="server" Width="60px"></asp:TextBox>
    <asp:Button id="Button1" onclick="Button1_Click" runat="server" Text="Button"></asp:Button>
    <asp:Label id="Label1" runat="server" text='<%#TextBox1.Text%>' borderstyle="Dotted"
bordercolor="Red" borderwidth="1px"></asp:Label>
  </form>
</body>
```

</html>
Örnek: iki tane değişkenin içindeki bilgiyi sayfada görüntüleyin

<%page...> <script runat="server">

Dim ad as string="ali" Dim soyad as string="su"

Sub page load... Page.DataBind() End sub </script>

<html> <head> </head> <body> <form runat="server"> <%#ad%> <%#soyad%> </form> </body> </html>

Örnek2:Veri kaynağı olarak bir dizi oluşturun.Bu dizideki bilgileri DropDownList de görüntüleyin

Herkes ali, oya ,veli diye bir dizi tanımlayın ve eski usulle dropdownlist de görüntüleyin

<u>1.yöntem</u>

Sub Button1_Click(sender As Object, e As EventArgs) dim x(3) as string dim a,b,c as string x(0)="ali" x(1)="yasa" x(2)="kanun"

dim i as integer for i=0 to 2 DropDownList1.items.add(x(i))

next End Sub

2.yöntem Html kısmında değişken kullanmadan verileri gösterme. <% @ Page Language="VB" %> <script runat="server">

'Insert page code here

</script> <html> <head> </head> <body>

```
<form runat="server">
<asp:DropDownList id="DropDownList1" runat="server">
<asp:Listitem value="1"> ahmet</asp:ListItem>
<asp:Listitem value="2"> mehmet</asp:ListItem>
</asp:DropDownList>
<!-- Insert content here -->
</form>
</body>
</html>
```

Örnek: Label a textbox1 deki bilgileri butona basınca databind yardımı ile yazdırma

```
<% @ Page Language="VB" %>
<script runat="server">
  'Insert page code here
  dim a as string
  Sub Button1_Click(sender As Object, e As EventArgs)
  "html kısmında text bölümüne birşeyler yazdık
  a=textbox1.text
  Page.DataBind()
  End Sub
</script>
<html>
<head>
</head>
<body>
  <form runat="server">
    <asp:TextBox id="TextBox1" runat="server"></asp:TextBox>
    <asp:Button id="Button1" onclick="Button1_Click" runat="server" Text="Button"></asp:Button>
    <asp:Label id="Label1" runat="server" Text="<%#a%>"></asp:Label>
    <!-- Insert content here -->
  </form>
</body>
</html>
```

ÖRNEK: ÇALIŞTIRILACAK....

| Start Page Default7.aspx.cs Default7.aspx* |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ad Soyad 11. Bilgiler Asp.Net nesneleri içinde görüntelenir. Button [Label1] |
| 2. Bilgileri Sayfa alanında görüntüleme |
| Button |
| 3. Bilgileri .aspx sayfası içerisinde değişkene bilgi bağlayarak görüntüleme Button |
| <pre>using System; using System.Collections; using System.Configuration; using System.Data; using System.Linq; using System.Web; using System.Web.Security; using System.Web.Security; using System.Web.UI; using System.Web.UI.HtmlControls; using System.Web.UI.HtmlControls; using System.Web.UI.WebControls; using System.Web.UI.WebControls; using System.Web.UI.WebControls.WebParts; using System.Xml.Linq;</pre> |
| <pre>public partial class Default7 : System.Web.UI.Page {</pre> |
| <pre>string Ad; string Soyad;</pre> |
| <pre>protected void Page_Load(object sender, EventArgs e) { Page.DataBind(); }</pre> |
| <pre>} protected void Button1_Click(object sender, EventArgs e) </pre> |
| Ad = TextBox1.Text; Soyad = TextBox2.Text; |
| <pre>Label1.DataBind();</pre> |
| <pre>//Label1.Text=Ad + " " + Soyad; }</pre> |
| <pre>protected void Button2_Click(object sender, EventArgs e) {</pre> |
| <pre>Ad = TextBox1.Text; Soyad = TextBox2.Text;</pre> |
| <pre>Response.Write(Ad + " " + Soyad); }</pre> |

}

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default7.aspx.cs"</pre>
Inherits="Default7" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        Ad      
        <asp:TextBox ID="TextBox1" runat="server"</pre>
Width="128px"></asp:TextBox>
        <br />
        Soyad   
                                   <asp:TextBox ID="TextBox2"</pre>
runat="server"></asp:TextBox>
        <br />
        <br />
        11. Bilgiler Asp.Net nesneleri içinde görüntelenir.<br />
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
Text="Button" />
        <br />
        <asp:Label ID="Label1" runat="server" Text="<%#Ad<mark>%></mark>"></asp:Label>
        <br />
        <br />
        <br />
        2. Bilgileri Sayfa alanında görüntüleme<br />
        <br />
        <asp:Button ID="Button2" runat="server" onclick="Button2 Click"</pre>
Text="Button" />
        <br />
        <br />
        <br />
        3. Bilgileri .aspx sayfası içerisinde değişkene bilgi bağlayarak
görüntüleme<br />
        <asp:Button ID="Button3" runat="server" Text="Button" />
        <br />
        <br />
        <mark><%</mark>#Ad<mark>%></mark> <mark><%</mark>#Soyad<mark>%></mark>
       <br />
        <br />
    </div>
    </form>
</body>
</html>
```