

**T.C.  
MİLLÎ EĞİTİM BAKANLIĞI**

# **BİLİŞİM TEKNOLOJİLERİ**

**ETKİLEŞİMLİ WEB UYGULAMALARI  
İÇİN TEMEL İŞLEMLER  
482BK0141**

**Ankara, 2012**

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

# İÇİNDEKİLER

AÇIKLAMALAR .....	iii
GİRİŞ .....	1
ÖĞRENME FAALİYETİ-1 .....	3
1. WEB SAYFASI İŞLEMLERİ.....	3
1.1. Görünüm Modları .....	3
1.2. Kontrol Ekleme .....	5
1.2.1. Toolbox Paneli Kullanarak Kontrol Ekleme .....	5
1.2.2. Kod Yazarak Kontrol Ekleme .....	5
1.3. Tablo Ekleme .....	6
1.4. Web Form Ekleme .....	8
1.5. Kontrol Özellikleri .....	9
1.5.1. Properties Paneli .....	9
1.5.2. Format Menüsü.....	12
1.5.3. Araç Çubukları .....	18
1.6. Hataları Düzeltme .....	18
1.6.1. Breakpoint Kullanımı .....	19
1.6.2. Try – Catch – Finally Kullanımı.....	20
UYGULAMA FAALİYETİ .....	23
ÖLÇME VE DEĞERLENDİRME .....	26
ÖĞRENME FAALİYETİ-2 .....	27
2. DOĞRULAMA KONTROLLERİ .....	27
2.1. RequiredFieldValidator.....	27
2.2. RangeValidator .....	29
2.3. CompareValidator .....	31
2.4. RegularExpressionValidator .....	33
2.5. CustomValidator .....	35
2.6. ValidationSummary .....	35
2.7. ValidationGroup Özelliği.....	37
UYGULAMA FAALİYETİ .....	40
ÖLÇME VE DEĞERLENDİRME .....	43
ÖĞRENME FAALİYETİ-3 .....	44
3. WEB KULLANICI KONTROLLERİ .....	44
3.1. Web Kullanıcı Kontrolleri Oluşturma.....	45
3.2. Web Kullanıcı Kontrolleri Ekleme .....	45
3.3. Web Kullanıcı Kontrolleri Tasarlama .....	46
UYGULAMA FAALİYETİ .....	48
ÖLÇME VE DEĞERLENDİRME .....	50
MODÜL DEĞERLENDİRME .....	51
CEVAP ANAHTARLARI .....	54
KAYNAKÇA .....	56



# AÇIKLAMALAR

<b>KOD</b>	<b>482BK0141</b>
<b>ALAN</b>	<b>Bilişim Teknolojileri</b>
<b>DAL/MESLEK</b>	<b>Web Programcılığı</b>
<b>MODÜLÜN ADI</b>	<b>Etkileşimli Web Uygulamaları İçin Temel İşlemler</b>
<b>MODÜLÜN TANIMI</b>	Bu modül, programlama yazılımı kullanarak .NET ortamında etkileşimli web uygulamalarda temel işlemlerin kullanımı ile ilgili temel bilgi ve becerilerin kazandırıldığı bir öğrenme materyalidir.
<b>SÜRE</b>	40/32
<b>ÖNKOŞUL</b>	“Etkileşimli Web Uygulamalarına Giriş” modülünü tamamlamış olmak
<b>YETERLİK</b>	Web sitesi yapısını oluşturmak
<b>MODÜLÜN AMACI</b>	<b>Genel Amaç</b> Öğrenci bu modül ile gerekli ortam sağlandığında; Programlama dilinin temel komutları ile web sayfası hazırlayabileceksiniz. <b>Amaçlar</b> <ol style="list-style-type: none"><li>1. Bir web sayfası üzerinde işlemler yapabilecek ve hata denetimini kullanabileceksiniz.</li><li>2. Doğrulama kontrollerini kullanabileceksiniz.</li><li>3. Web kullanıcı kontrolleri oluşturabilecek ve bunları başka sayfalara ekleyebileceksiniz.</li></ol>
<b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b>	<b>Ortam:</b> Bilişim Teknolojileri laboratuvarı, işletme ortamı <b>Donanım:</b> Web programlama yazılımlarını çalıştırabilecek yeterlikte bilgisayar, internet bağlantısı
<b>ÖLÇME VE DEĞERLENDİRME</b>	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.



# GİRİŞ

**Sevgili Öğrenci,**

ASP.NET kullanıcılarla etkileşime giren web siteleri oluşturmanıza olanak sağlamaktadır. Kullanıcı etkileşiminin sağlıklı bir şekilde gerçekleşmesi için girilen verilerin doğruluğunun kontrol edilmesi büyük önem taşımaktadır. Bu, web sitesi içindeki formların kullanım amacının dışına çıkılmasını engelleyecektir.

Programlama yazılımı web sunucu kontrollerinin dışında girilen bilgilerin doğruluğunun kontrol edilmesi için doğrulama kontrollerini ve tekrar tekrar kullanım için kullanıcı kontrollerini sunmaktadır.

Modül içinde ASP.NET web projeleri içinde kullanılan kontrollerin yapılandırılması, doğrulama ve web kullanıcı kontrolleri anlatılmış, çeşitli örnek uygulamalarla konular desteklenmiştir.





# ÖĞRENME FAALİYETİ-1

## AMAÇ

Bir web sayfası üzerinde işlemler yapabilecek ve hata denetimini kullanabileceksiniz.

## ARAŞTIRMA

- Programlama yazılımın web sunucu kontrollerinin nasıl değiştirilebileceğini araştırınız.

## 1. WEB SAYFASI İŞLEMLERİ

### 1.1. Görünüm Modları

Programlama yazılımı ile bir web sitesi oluşturulduğunda, programlama yazılımı gerekli dosyaları ekleyecek ve yapılandırmayı yapacaktır.

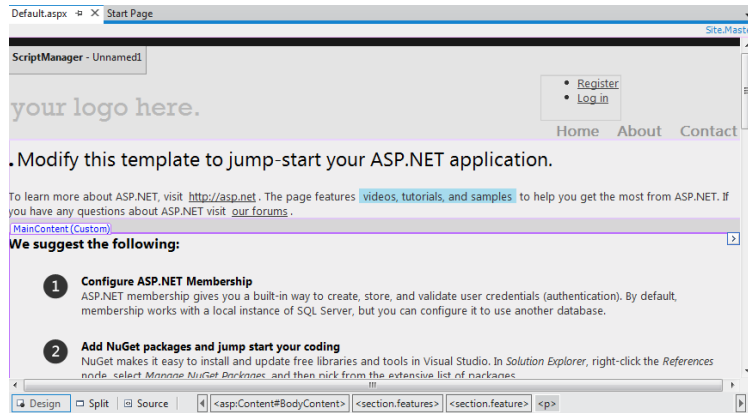
Web sitesi sayfaları *.aspx* ve *.aspx.cs* dosyalarının birleşimi ile oluşturulur. Bu dosyalardan *.aspx.cs* dosyası sunucu kodları oluştururken, *.aspx* dosyaları ise HTML ve kontrollerin oluşturulduğu sayfalardır.

*.aspx* dosyaları üç farklı şekilde görüntülenir. Bu üç mod da aynı işlevi görmektedir.



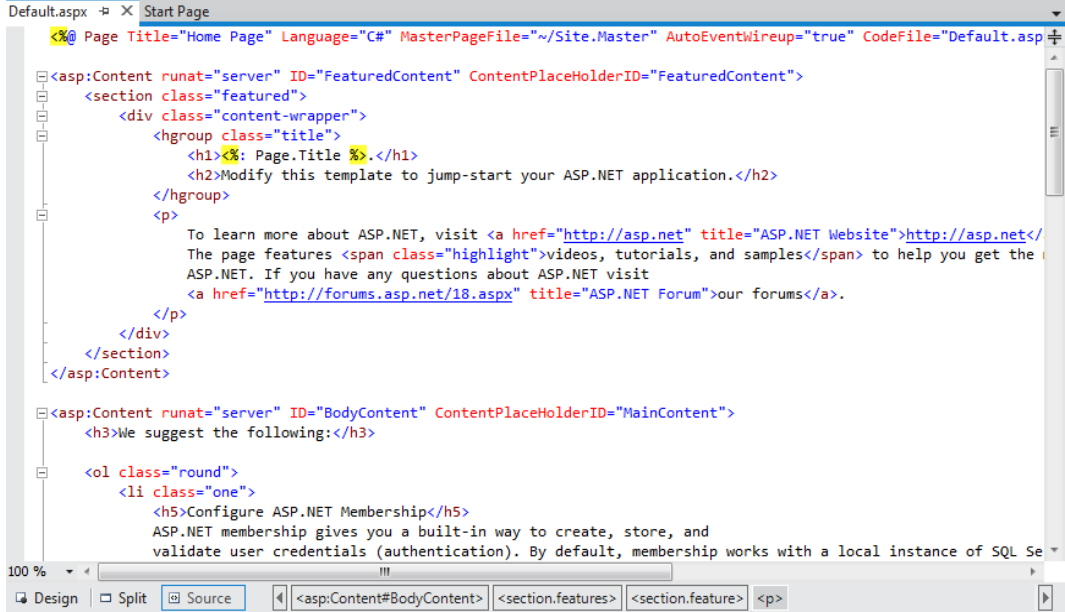
Resim 1.1: Görünüm modları

- **Design (Tasarım):** Bu modda web sayfası görsel öğeler kullanılarak hazırlanır. Bu modda bir görsel eklendiğinde programlama yazılımı kodları otomatik olarak ekler.



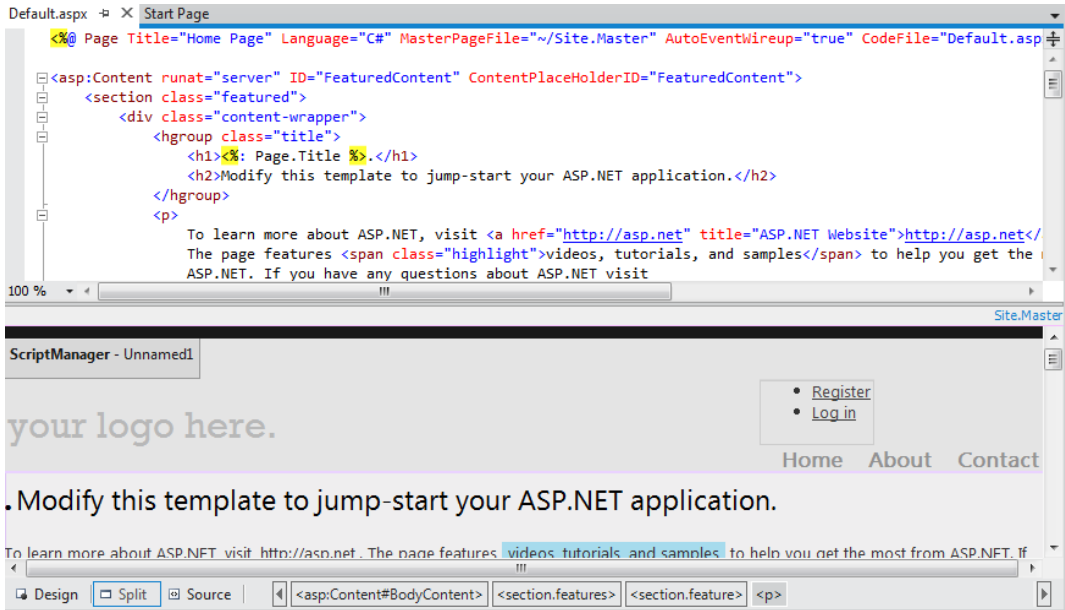
Resim 1.2: Design görünüm modu

- **Source (HTML kodları):** Bu modda HTML kodları ve kontrol kodları doğrudan yazılarak sayfa oluşturulur. Bu modda kodla bir öge eklendiğinde programlama yazılımı otomatik olarak görselini sayfaya ekler.



Resim 1.3: Source

- **Split (Tasarım ve HTML kodları):** Bu modda kodlar ve tasarım ekranı aynı anda gösterilir.



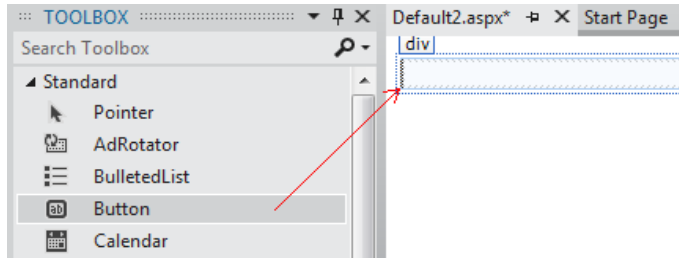
Resim 1.4: Split görünüm modu

## 1.2. Kontrol Ekleme

Programlama yazılımı, oluşturulan bir web sayfasına kontrol eklemek için birkaç farklı yol sunar.

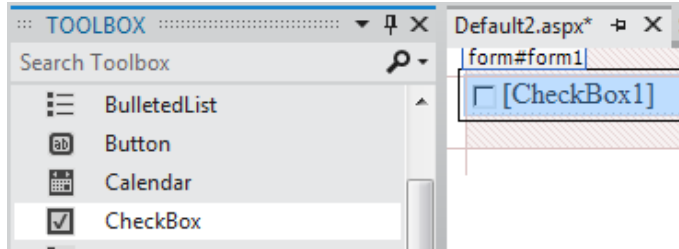
### 1.2.1. Toolbox Paneli Kullanarak Kontrol Ekleme

Kontrol eklemenin en kolay yöntemlerinden birisi, *Toolbox* panelinden istenilen kontrolün sürüklenerek sayfa içerisine bırakılmasıdır. Programlama yazılımı bu yöntemde kontrolün görselini sayfaya eklerken arka planda da HTML ve sunucu kodları sayfaya otomatik olarak kontrolün kodunu ekler.



**Resim1.5: Toolbox paneli ile sürkle – bırak yöntemi kullanarak kontrol ekleme**

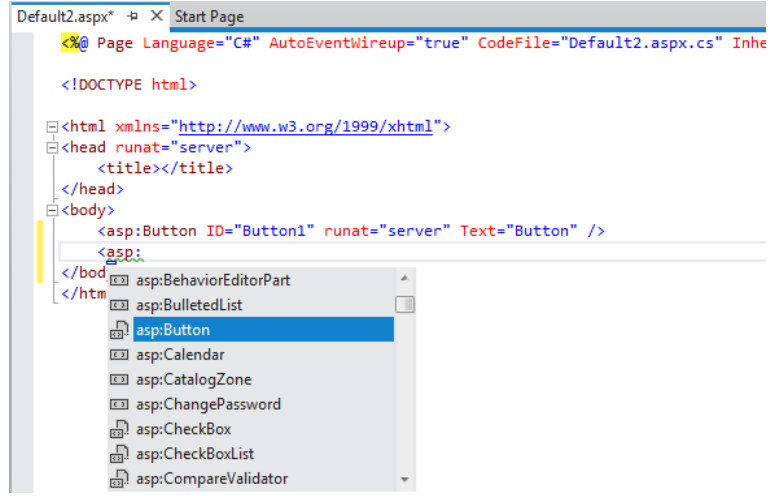
Diğer bir yöntem ise *Toolbox* panelinde kontrolün üzerine çift tıklayarak eklemektir. Bu durumda programlama yazılımı, aktif sayfaya kontrolün görselini ve kodunu otomatik olarak ekleyecektir.



**Resim 1.6: Toolbox panelinde kontrolü çift tıklayarak ekleme**

### 1.2.2. Kod Yazarak Kontrol Ekleme

Programlama yazılımında sayfaya kontrol eklemenin diğer bir yolu da HTML ve kontrol tanımlamaları kodları sayfaya geçip kod yazarak kontrol eklemektir. Kontrol kodları doğrudan yazılabileceği gibi *Toolbox* panelinden kontrol kodu sayfaya sürüklenerek de eklenebilir.



**Resim 1.7: HTML ve kontrol tanımlamaları kodları sayfasına kod yazarak kontrol ekleme**

*Toolbox* paneli kullanılarak bir kontrol eklendiğinde, programlama yazılımı kontrol adlandırmasını otomatik olarak yapar. Eğer bir kontrol kodla kullanıcı tarafından ekleniyorsa, projenin ilerleyen safhalarında karmaşıklığa neden olmaması için kontrol adlandırılmasına dikkat edilmelidir.

Kodla kontrol eklerken **<asp:** etiketi kullanılır. Programlama yazılımı otomatik tamamlama ile kodların yazımında kullanıcıya yardımcı olmaktadır.

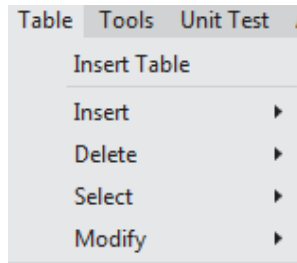
```
<asp:Button ID="Button1" runat="server" Text="Button" />
```

**Resim 1.8: Örnek kontrol kodu**

### 1.3. Tablo Ekleme

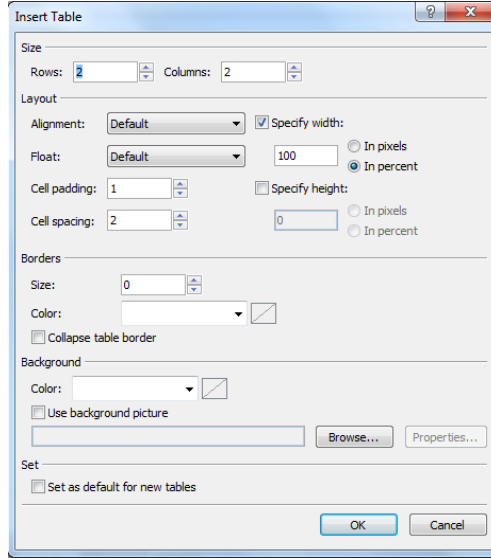
Tasarım aşamasında bir sayfada tablo kullanımı tasarımı kolaylaştırabilir. Nesnelerin yerleşimi tablolara aracılığıyla daha rahat yapılabilmektedir.

Programlama yazılımı sayfaya bir tablo eklemek için bir *Table* menüsü sunar. Bu menü üzerinden tablo eklenebilir, tabloyla ilgili işlemler yapılabilir.



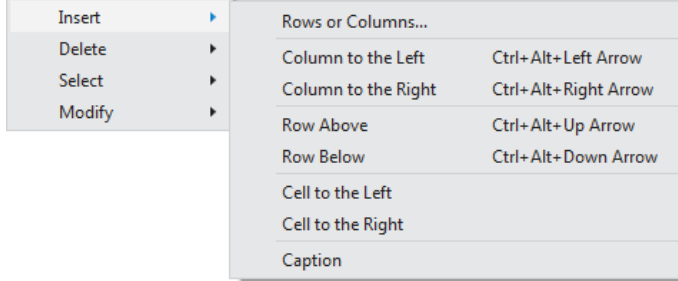
**Resim 1.9: Table menüsü**

- **Insert table:** Yeni tablo eklemek için kullanılır. Bu komut seçildiğinde *Insert Table* iletişim penceresi ekrana gelir. *Satır (rows)* ve *sütun (Columns)* değerleri ayarlanarak istenilen tablo oluşturulabilir.



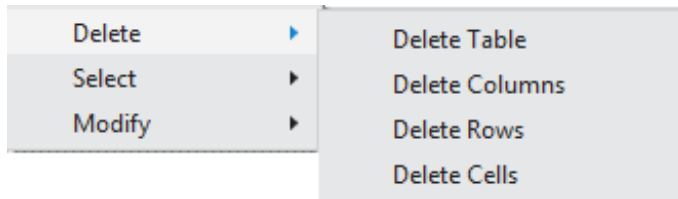
**Resim 1.10: Insert table iletişim penceresi**

- **Insert:** Tabloya satır, sütun veya hücre eklemek için kullanılır. Açılır menüden sağa (*Column to the Right*) ya da sol sütun (*Column to the Left*) sütun, üste (*Row Above*) ya da alta (*Row Below*) satır eklenebilir.



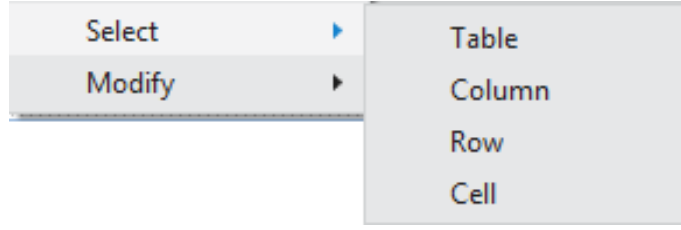
**Resim 1.11: Table > Insert açılır menüsü**

- **Delete:** Tabloyu (*Delete Table*), Sütun (*Delete Columns*), satır (*Delete Rows*) veya hücreleri (*Delete Cells*) silmek için kullanılır.



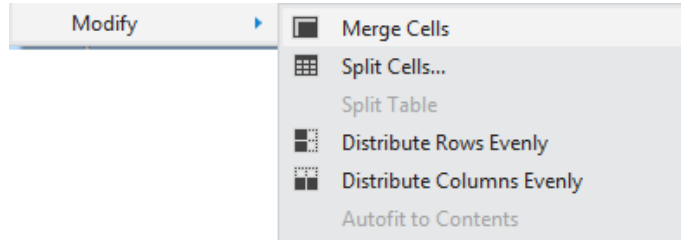
**Resim 1.12: Table > Delete açılır menüsü**

- **Select:** Tabloyu (*Table*), sütunu (*Column*), satırı (*Row*) veya hücreleri (*Cell*) seçmek için kullanılır.



**Resim 1.13: Table > Select açılır menüsü**

- **Modify:** Tablo üzerinde çeşitli işlemler yapmak için kullanılır. Bu işlemler, seçilen hücreleri birleştirme (*Merge Cells*), hücrelere bölme (*Split Cells*), tabloya bölme (*Split Table*), satır yüksekliklerini eşitleme (*Distribute Rows Evenly*) ve sütun genişliklerini eşitleme (*Distribute Columns Evenly*) gibi işlemlerdir.



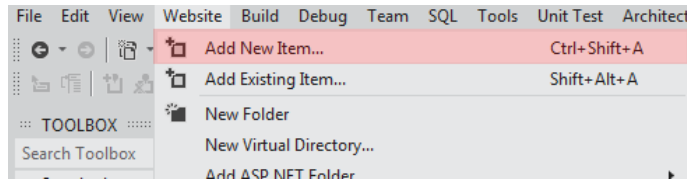
**Resim 1.14: Table > Modify açılır menüsü**

Table menüsü dışında sayfa kodlarından tablo üzerinde kodlarla işlem yapılabilir.

## 1.4. Web Form Ekleme

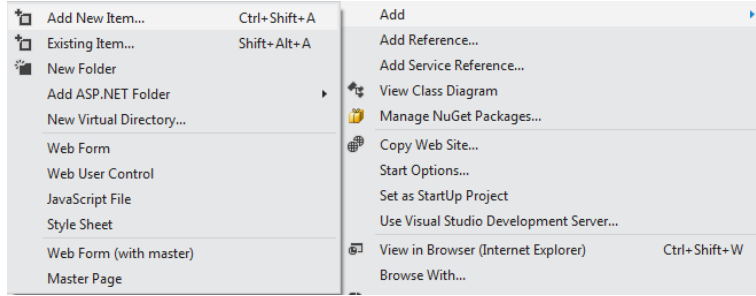
Bir web sitesi birden fazla sayfadan oluşmaktadır. Çalışma anında web sitesine yeni sayfa ekleme ihtiyacı doğabilir. Web sitesine yeni bir sayfa eklemenin farklı iki yolu vardır. Bunlar:

- *Website > Add New Item* komutu kullanarak web sitesine yeni bir sayfa eklenebilir.



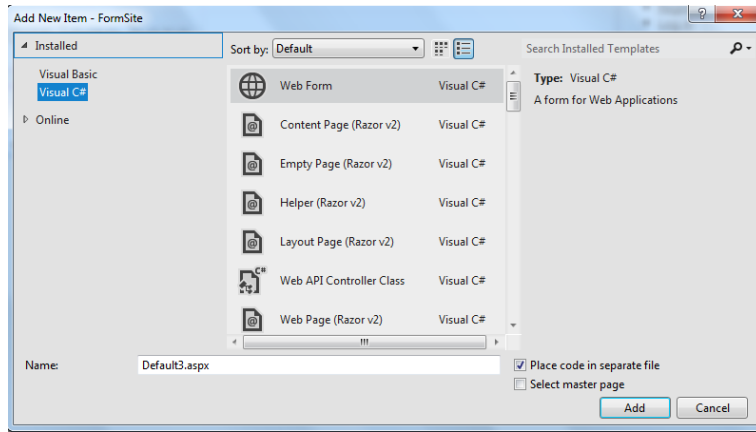
**Resim 1.15: Website > Add New Item komutu**

- Diğer bir yol ise *Solution Explorer* panelinin kullanılmasıdır. Proje adına sağ tıklayarak açılan sağ menüden *Add* açılır menüsü altında *Add New Item* komutu seçilebilir.



**Resim 1.16: Add > Add New Item komutu**

Her iki yöntemde *Add New Item* komutundan sonra ekrana *Add New Item – Form Site* iletişim penceresi gelecektir. Bu pencereden istenilen özellikte sayfa seçilip adlandırılabilir.



**Resim 1.17: Add New Item – For Site iletişim penceresi**

Ayrıca istenirse Add açılır menüsünden web sitesine, web sayfası ve diğer öğeler doğrudan eklenebilir.

## 1.5. Kontrol Özellikleri

### 1.5.1. Properties Paneli

Seçilen bir kontrole ait özellikler *Properties* panelinde görüntülenir. Panel içinde özellikler çeşitli gruplara ayrılarak gösterilir. *Properties* panelinde aşağıdaki gruplar yer almaktadır;

#### ➤ Accessibility

Kontrole ait erişilebilirlik özelliklerinin bulunduğu gruptur.



**Resim 1.18: Properties paneli – Accessibility grubu**

Bu grupta bulunana özellikler şunlardır:

- **AccessKey:** Daha önceden tanımlanmış alt tuşu ile birlikte kullanılan erişim tuşudur. Örneğin, *Yazdır* butonu için *AccessKey* “Y” atanacak olursa uygulama çalıştırıldığında “Alt + Y” tuş kombinasyonu bu düğmeyi tıklamayla aynı işlevi yapacaktır.

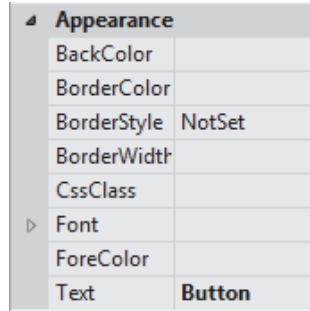
```
<asp:Button ID="Button1" runat="server" Text="Yazdır" AccessKey="Y"/>
```

- **TabIndex:** Sayfa içindeki denetimlerin tab tuşu kullanılarak atlama sırasını belirtir.

```
<asp:Button ID="Button1" runat="server" Text="Print" TabIndex="1"/>
```

#### ➤ Appearance

Görünüm ile ilgili özellikler yer aldığı gruptur. Web sunucu kontrolüne göre bu grupta yer alan özellikler değişiklik gösterir.



Appearance	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
CssClass	
Font	
ForeColor	
Text	Button

Resim 1.19: Properties paneli – Accessibility grubu

Bu grupta bulunan özellikler şunlardır:

- **BackColor:** Web sunucu kontrollerinin arkaplan rengini ayarlar.  
<asp:Button ID="Button1" runat="server" BackColor="Red"/>
- **BorderColor:** Web sunucu kontrollerinin kenarlık rengini ayarlar.  
<asp:Button ID="Button1" runat="server" BorderColor="#003366"/>
- **BorderStyle:** Web sunucu kontrollerinin kenarlık stilini ayarlar.  
<asp:Button ID="Button1" runat="server" BorderStyle="Solid"/>
- **BorderWidth:** Web sunucu kontrollerinin kenarlık genişliğini ayarlar.  
<asp:Button ID="Button1" runat="server" BorderWidth="5px"/>
- **CssClass:** Web sunucu kontrolünün istemci tarafından işlenen stil sayfası (css) sınıfını ayarlar.
- **Font:** Web sunucu kontrolünün yazı özelliklerini ayarlar.
  - **Bold:** Web sunucu kontrolünün yazısını kalın yapar.



```
<asp:Button ID="Button1" runat="server" Font-Bold="True"/>
```

- **Italic:** Web sunucu kontrolünün yazısını italik yapar.

```
<asp:Button ID="Button1" runat="server" Font-Italic="True"/>
```

- **Name:** Web sunucu kontrolünün yazı tipini belirler.

```
<asp:Button ID="Button1" runat="server" Font-Names="Arial Narrow"/>
```

- **Overline:** Web sunucu kontrolünün yazısının üstüne çizgi ekler.

```
<asp:Button ID="Button1" runat="server" Font-Overline="True"/>
```

- **Size:** Web sunucu kontrolünün yazı boyutunu belirler.

```
<asp:Button ID="Button1" runat="server" Font-Size="Small"/>
```

- **Strikeout:** Web sunucu kontrolünün yazısının üzerine çizgi ekler.

```
<asp:Button ID="Button1" runat="server" Font-Strikeout="True"/>
```

- **Underline:** Web sunucu kontrolünün yazısının altına çizgi ekler.

```
<asp:Button ID="Button1" runat="server" Font-Underline="True"/>
```

- **ForeColor:** Web sunucu kontrolünün yazı rengini ayarlar.

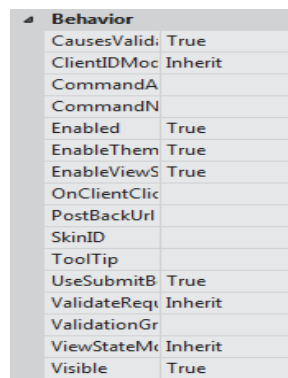
```
<asp:Button ID="Button1" runat="server" ForeColor="Red"/>
```

- **Text:** Web sunucu kontrolünün metnini ayarlar.

```
<asp:Button ID="Button1" runat="server" Text="Yazdır" />
```

## ➤ Behavior

Web sunucu kontrollerine ait davranış özelliklerinin yer aldığı gruptur. Web sunucu kontrolüne göre bu grupta yer alan özellikler değişiklik gösterir.



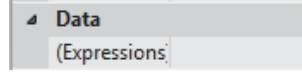
Behavior	
CausesValid	True
ClientIDMoc	Inherit
CommandA	
CommandN	
Enabled	True
EnableThem	True
EnableViewS	True
OnClientClic	
PostBackUrl	
SkinID	
ToolTip	
UseSubmitB	True
ValidateReq	Inherit
ValidationGr	
ViewStateMc	Inherit
Visible	True

Resim 1.20: Properties paneli – Behavior grubu

Bu grupta bulunan bazı özellikler şunlardır:

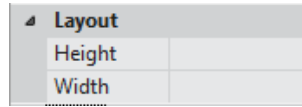
- **AutoPostBack:** Web sunucusunda değişiklik yapıldıktan sonra sayfanın geri döndürülmesi için kullanılır.

- **Enabled:** Web sunucu kontrolünün aktif ya da pasif olmasını ayarlar.
  - **OnClick:** *Button Click* olayı oluşturulduğunda, istemci taraflı komut setleri yürütür.
  - **PostBackUrl:** Tıklama işlemi sonrasında gidilecek sayfa URL'sini belirtir.
  - **ToolTip:** Kullanıcı kontrol üzerinde mouse bekletildiği zaman kontrolün amacını belirten kısa bir açıklamanın görüntülediği pop-up penceresini ifade eder.
  - **Visible:** Web sunucu kontrolünün görünür ya da gizli olmasını ayarlar.
- **Data:** Veri işlemlerinin yapıldığı gruptur.



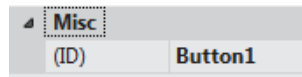
**Resim 1.21: Properties paneli – Data grubu**

- **Layout:** Web sunucu kontrolünün görünümü ile ilgili özelliklerin yer aldığı gruptur.



**Resim 1.22: Properties paneli – Layout grubu**

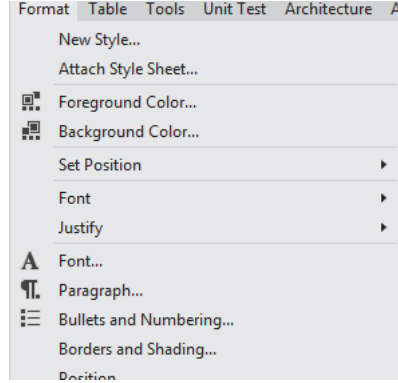
- **Height:** Kontrolün piksel türünden genişliği ayarlamak için kullanılır.
  - **Width:** Kontrolün piksel türünden yüksekliğini ayarlamak için kullanılır.
- **Misc:** Kontrolün adlandırılmasının yapıldığı gruptur. Programlama yazılımı kullanıcı müdahale etmediği sürece kontrolleri otomatik adlandırır.



**Resim 1.23: Properties paneli – Misc grubu**

## 1.5.2. Format Menüsü

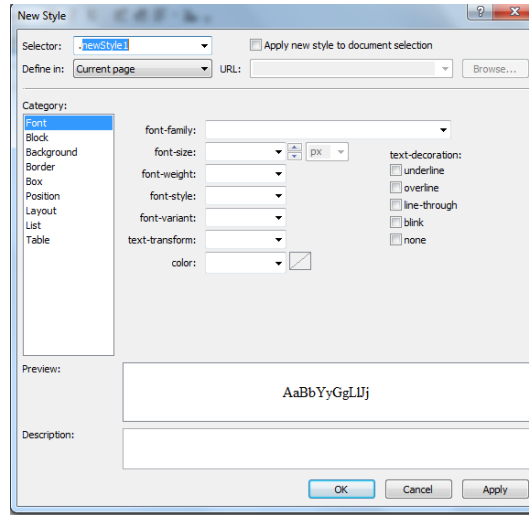
Programlama yazılımı kontrollerin özellikleri ayarlamak için *Format* menüsüne sahiptir.



**Resim 1.24: Format menüsü**

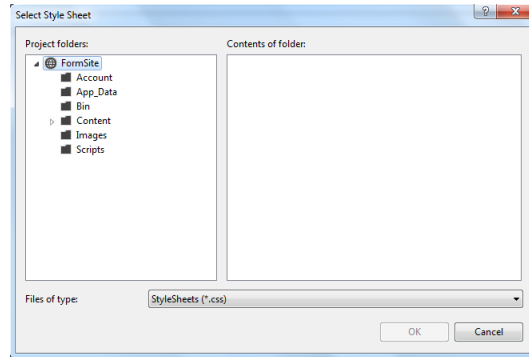
Bu menü aşağıdaki komutlara ve özellikler sahiptir:

- **New Style:** Yeni stil oluşturmak için kullanılır.



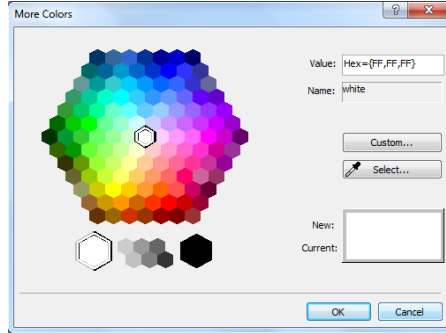
**Resim 1.25: New Style iletişim penceresi**

- **Attach Style Sheet:** Var olan stilleri seçmek için kullanılır.



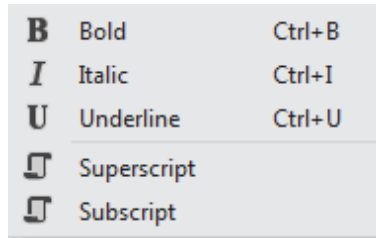
**Resim 1.26: Select Style Sheet iletişim penceresi**

- **Foreground Color:** Ön plan rengini ayarlamak için kullanılır.



**Resim 1.27: More Colors iletişim penceresi**

- **Background Color:** Arka plan rengini ayarlamak için kullanılır.
- **Font:** Metinlerin kalın, italik, altı çizili şekilde yazılması için kullanılır.



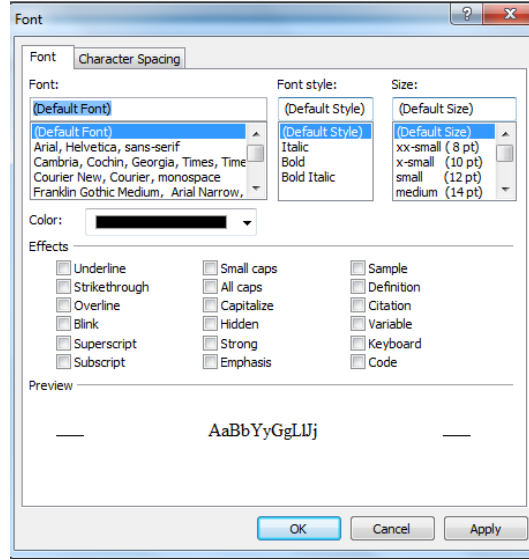
**Resim 1.28: Format > Font açılır menüsü**

- **Justify:** Metinleri hizalamak için kullanılır.



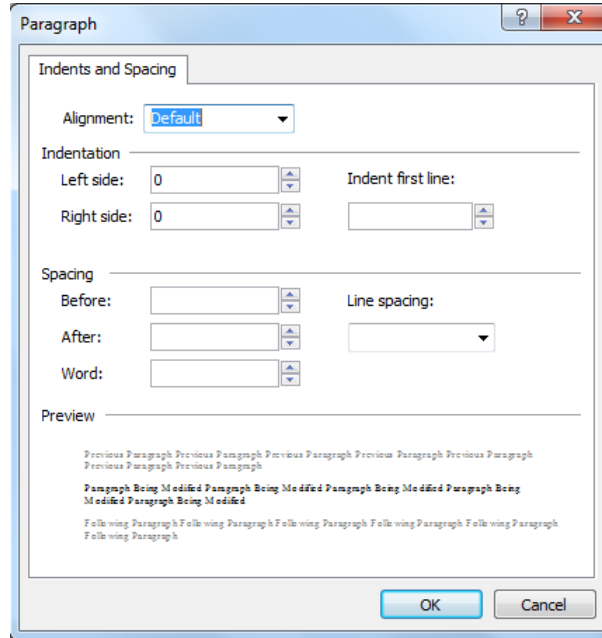
**Resim 1.29: Format > Justify açılır menüsü**

- **Font...:** Metin özelliklerini ayarlamak için kullanılır.



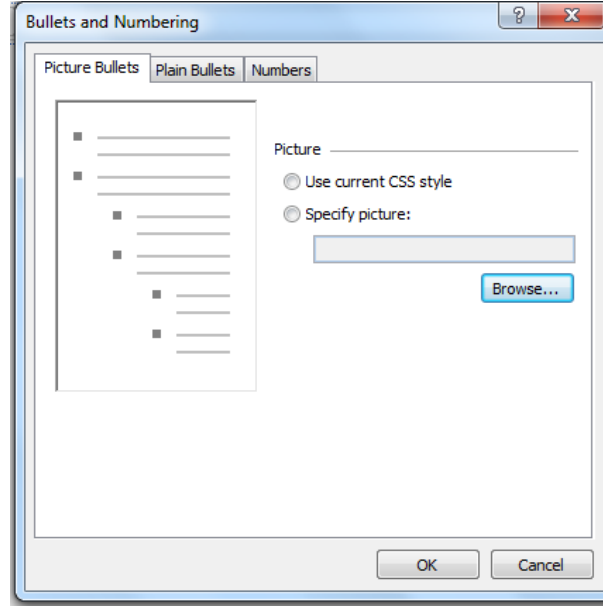
Resim 1.30: Font iletişim penceresi

- **Paragraph...:** Paragraf özelliklerini ayarlamak için kullanılır.



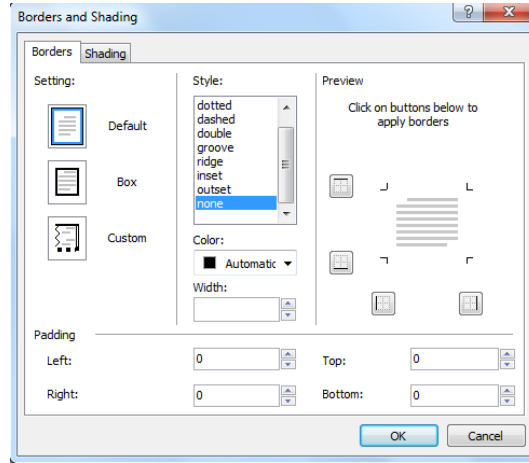
Resim 1.31: Paragraph iletişim penceresi

- **Bullets and Numbering:** Madde işaretleri ve numaralandırma için kullanılır.



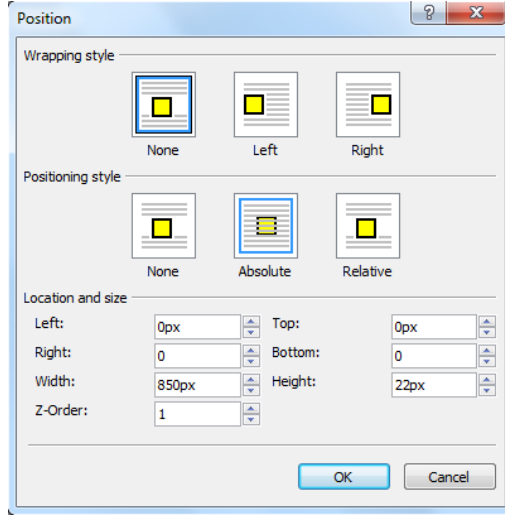
**Resim 1.32: Bullets and Numburing iletişim penceresi**

- **Borders and Shading:** Kenarlık ve gölgelendirmeleri ayarlamak için kullanılır.



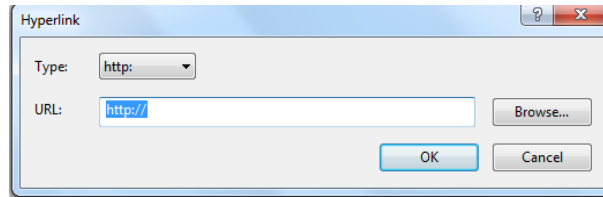
**Resim 1.33: Borders and Shading iletişim penceresi**

- **Position...:** Metnin pozisyonunu ayarlamak için kullanılır.



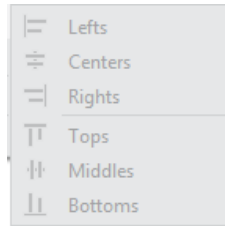
**Resim 1.34: Position iletişim penceresi**

- **Convert to Hyperlink:** Seçili metne link eklemek için kullanılır.



**Resim 1.35: Hyperlink iletişim penceresi**

- **Align:** Hizalama yapmak için kullanılır.



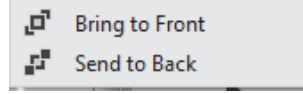
**Resim 1.36: Format > Align açılır menüsü**

- **Make Same Size:** Seçili nesnenin boyutlarını ayarlamak için kullanılır.



**Resim 1.37: Format > Make Same Size açılır menüsü**

- **Order:** Seçili nesnenin yerleşimdeki sırasını ayarlamak için kullanılır.

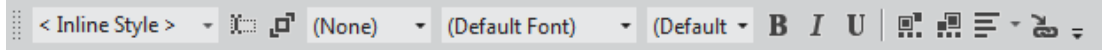


**Resim 1.38: Format > Order açılır menüsü**

- **Remove Formatting:** Seçili nesneye uygulanan biçimleri temizler.

### 1.5.3. Araç Çubukları

Araç çubukları sık kullanılan komutlara ait simgelerin bulunduğu alanlardır. Kullanılan birçok yazılımda bu tür çubuklar yer almaktadır. Programlama yazılımı da bu şekilde araç çubuklarına sahiptir. Kontrollerin özelliklerinin ayarlanabileceği, *Format* menüsünde yer alan komutlara ait kısa yollar simgeler halinde *Formating* araç çubuğunda yer almaktadır. Bu araç çubuğunda yer alan simgelerin işlevleri *Format* menüsündeki komutlarla aynıdır.

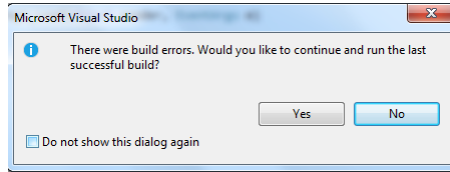


**Resim 1.39: Formating araç çubuğu**

## 1.6. Hataları Düzeltme

Bir uygulama geliştirilirken birçok hata ile karşılaşmak kaçınılmazdır. Programlama yazılımı kullanıcıya hatanın kaynağı hakkında çeşitli bilgiler verir.

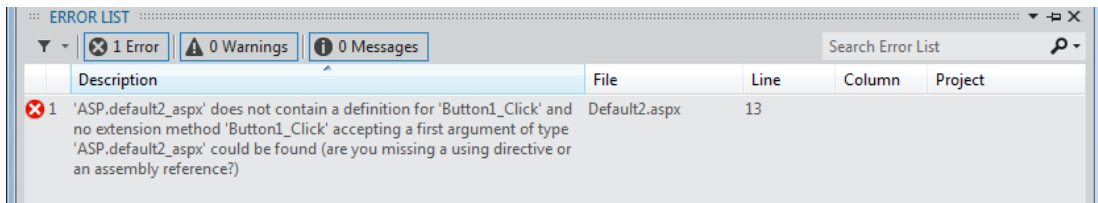
Uygulama içerisinde herhangi bir hata oluştuğunda bu hata öncelikle bir iletişim penceresi ile kullanıcıya gösterilir.



**Resim 1.40: Hata uyarısı iletişim penceresi**

Hata göz ardı edilerek en son çalışır durumdaki derlemenin çalıştırılması için “Yes” komutu seçilmelidir. Ancak son çalışır durumdaki derlemeden sonraki değişiklikler çalışmaya yansımayaacaktır.

Hatayı gidermek için “No” komutu tıklanmalıdır. Bu durumda programlama yazılımı derleme yapmadan çalışma sayfasına geri döner ve oluşan hatanın nedeni *Error List* panelinde gösterilir.



**Resim 1.41: Error List Panelinde hatanın gösterilmesi**



*Error List* panelinde hatanın nedeni, hatanın olduğu dosya ve satır numarası gösterilir. Bu satıra gidilerek hata giderilebilir.

Bazı durumlarda derleme yapılmasına rağmen çalışma zamanı ile ilgili hatalar meydana gelebilir. Bu durumda derleme sayfasında Resim 1.42'dekine benzer hata sayfası ekranı gelecektir. Bu sayfada hatanın nedeni olan durum sayfanın üst kısmında kullanıcıya bildirilir.

### Server Error in '/' Application.

*The ControlToValidate property of 'CompareValidator1' cannot be blank.*

**Description:** An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

**Exception Details:** System.Web.HttpException: The ControlToValidate property of 'CompareValidator1' cannot be blank.

**Source Error:**

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

**Stack Trace:**

```
[HttpException (0x80004005): The ControlToValidate property of 'CompareValidat
System.Web.UI.WebControls.BaseValidator.ControlPropertiesValid() +9672736
System.Web.UI.WebControls.CompareValidator.ControlPropertiesValid() +92
System.Web.UI.WebControls.BaseValidator.OnPreRender(EventArgs e) +36
System.Web.UI.Control.PreRenderRecursiveInternal() +83
System.Web.UI.Control.PreRenderRecursiveInternal() +155
System.Web.UI.Control.PreRenderRecursiveInternal() +155
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint
```

### Resim 1.42: Sunucu Hata Sayfası

Bazı durumlarda hataları gidermek yukarıda belirtildiği kadar kolay olmayabilir. Sayfa sayısının artması, buna bağlı olarak kodlamanın artması hata olasılığını artırır.

Hata denetimini yapmak için kullanılacak bazı yöntemler aşağıdaki gibidir:

- Değişiklik yaptıktan sonra uygulamayı derleyerek hata oluşup oluşmadığını kontrol edin.
- *Debug > Start Without Debugging* (Ctrl + F5) komutu ile derlemeyi adım adım yapın.
- *Breakpoint* ekleyerek çalışma adımlarını belirleyerek derlemeyi yapın.
- *Try – Catch – Finally* komutunu kullanarak hataların önüne geçin.

#### 1.6.1. Breakpoint Kullanımı

Breakpoint, uygulamanın çalışmasını istenen kod satırında durdurmak için kullanılır.

Breakpoint ile hata denetimi yapmak için:

- Breakpoint ekleyeceğiniz satıra gelin ve *Debug > Toggle Breakpoint (F9)* komutunu kullanın.

- Satırın başlangıcında kırmızı bir nokta oluşacaktır.

```
10 protected void Page_Load(object sender, EventArgs e)
11 {
12
13 }
14 protected void Button1_Click(object sender, EventArgs e)
15 {
16
17 }
```

**Resim 1.43: Breakpoint eklenmesi**

- Uygulamayı derleyin. Programlama yazılımı ilk breakpoint noktasına gelinceye kadar uygulamayı derler ve bekleme başlar.

```
10 protected void Page_Load(object sender, EventArgs e)
11 {
12
13 }
14 protected void Button1_Click(object sender, EventArgs e)
15 {
16
17 }
```

**Resim 1.44: Breakpoint çalışması**

- Bir sonraki adıma geçmek için *Debug > Step Into (F11)* veya *Debug > Step Over (F10)* komutlarını kullanın.
- *Step Over (F10)* komutu ile adım adım hata ayıklama yapılırken eğer kullanılan bir metot vb. varsa onların içerisine girilmez. *Step Into(F11)* komutu kullanıldığında her birinin içerisine girilir. Bu komutlar aynı zamanda *Debug* araç çubuğunda da yer almaktadır.



**Resim 1.45: Debug araç çubuğu**

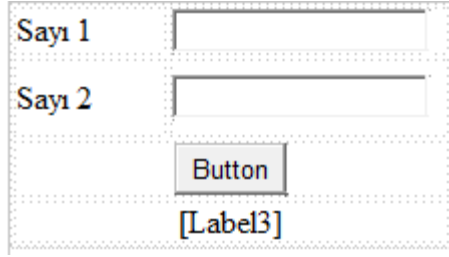
## 1.6.2. Try – Catch – Finally Kullanımı

*Try – Catch - Finally* kullanılarak bir blokta oluşabilecek hataların önüne geçilebilir.

*Try* bloğu içine hata oluşabilecek kodlar yazılır. *Catch* bloğunda, yakalanan hataya göre yapılacak işlemler yazılır. *Finally* bloğunda ise, *Try* ve *Catch* bloklarından hatanın üretildiği veya üretilmediği iki durumda da yapılması gereken işlemler yazılır.

*Try – Catch* ile örnek bir hata yakalama uygulaması gerçekleştirmek için;

- Boş bir sayfa açın ve aşağıdaki kontrolleri çalışma alanına ekleyin.

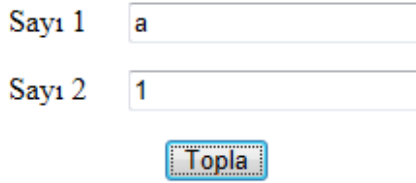


Resim 1.46: Try – Catch uygulaması

- *Button Click* olayına aşağıdaki kodları ekleyin.

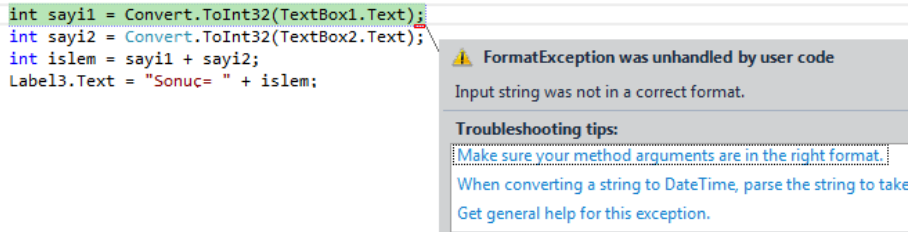
```
int sayi1 = Convert.ToInt32(TextBox1.Text);
int sayi2 = Convert.ToInt32(TextBox2.Text);
int islem = sayi1 + sayi2;
Label3.Text = "Sonuç= " + islem;
```

- Uygulamayı çalıştırın ve sayısal bir değer dışında bir değer girin.



Resim 1.47: Uygulama ekran çıktısı

- Hatalı değer girildiğinde uygulama çalışma zamanı hatası verecektir.



Resim 1.48: Çalışma zamanı hatası

- *Try – Catch* komutunu ekleyerek kodları yeniden düzenleyin.

```
try
{
    int sayi1 = Convert.ToInt32(TextBox1.Text);
    int sayi2 = Convert.ToInt32(TextBox2.Text);
    int islem = sayi1 + sayi2;
    Label3.Text = "Sonuç= " + islem;
}
catch
{
    Label3.Text = "Sayısal değer girin!";
}
```

}

- Tekrar uygulamayı çalıştırın ve sayısal bir değer dışında bir değer girin.

Sayı 1

Sayı 2

**Resim 1.49: Uygulama ekran çıktısı 2**

- Bu durumda uygulama hata vermeyecek ve *Catch* bloğunda yazılı kodlar çalışacaktır.

Sayı 1

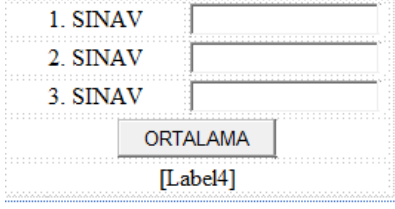
Sayı 2

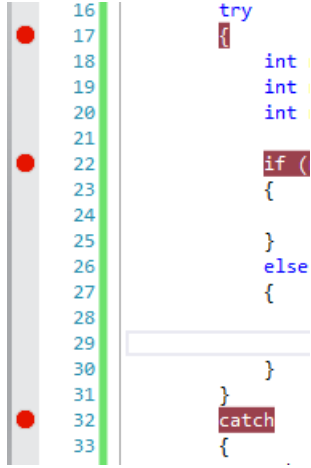
Sayısal değer girin!

**Resim 1.50: Try – Catch kullanarak hatanın yakalanması ekran çıktısı**

## UYGULAMA FAALİYETİ

Programlama yazılımında web sayfası işlemleri ile ilgili aşağıdaki uygulamayı yapınız.

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"><li>➤ Programlama yazılımını başlatınız.</li></ul>	<ul style="list-style-type: none"><li>➤ Yazılım kısayolunu kullanabilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Sayfaya 5x3 bir tablo ekleyiniz.</li></ul>	<ul style="list-style-type: none"><li>➤ <i>Table &gt; Insert Table</i> komutunu kullanabilirsiniz.</li><li>➤ Tabloyu tasarıma uygun boyutlandırabilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Tablonun içine aşağıdaki web kullanıcı kontrollerini ekleyiniz.</li></ul> 	<ul style="list-style-type: none"><li>➤ Toolbox panelinden kontrolleri çalışma alanına sürükleyerek ekleyebilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ TextBox bileşenlerini sırası ile tab tuşu ile atlayacak şekilde ayarlayınız.</li></ul>	<ul style="list-style-type: none"><li>➤ Properties panelinden <i>TabIndex</i> özelliğini kullanabilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Label 4 özelliklerinden yazı rengini kırmızı ve koyu yapınız.</li></ul>	<ul style="list-style-type: none"><li>➤ Yazı rengi için <i>Properties</i> panelinden <i>ForeColor</i> özelliğini kullanabilirsiniz.</li><li>➤ Yazısını koyu yapmak için <i>Properties</i> panelinden <i>Font</i> özelliği altındaki <i>Bold</i> özelliğini kullanabilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Sınav notlarının 0-100 arasında ve sayısal değer olup olmadığını kontrol ettikten sonra ortalamayı hesaplayan ekleyiniz.</li></ul> <pre>try {     int not1 = Convert.ToInt32 (TextBox1.Text);     int not2 = Convert.ToInt32 (TextBox2.Text);     int not3 = Convert.ToInt32 (TextBox3.Text);      if (not1 &lt; 0    not1 &gt; 100    not2 &lt; 0    not2 &gt; 100    not3 &lt; 0    not3 &gt; 100)     {         Label14.Text = "Not 0-100 değerleri arasında olmalıdır!";     }     else     {         Label14.Text = "Ortalama= " + (not1 + not2 + not3) / 3;     } }</pre>	<ul style="list-style-type: none"><li>➤ Bu kodları <i>Button Click</i> olayına yazabilirsiniz.</li><li>➤ TextBox kontrollerine girilen değerleri <i>integer</i> veri tipini dönüştürmelisiniz.</li><li>➤ Hata oluşturabilecek kodları <i>try</i> bloğu içine yazmalısınız.</li><li>➤ Hata durumundaki mesajı <i>catch</i> bloğu içine yazmalısınız.</li><li>➤ Notun 0-100 arasında olmasını <i>if - else</i> komutu ile kontrol edebilirsiniz.</li></ul>

<pre> } catch {     Label4.Text = "Not sayısal değer olmalıdır!"; } } </pre>	
<p>➤ Uygulamayı çalıştırınız.</p>	<p>➤ <i>Debug &gt; Start Debugging (F5)</i> komutunu kullanabilirsiniz.</p>
<p>➤ try, if , catch komutlarına breakpoint ekleyiniz.</p> 	<p>➤ Komut satırı üzerinde sağ tıklayın <i>Breakpoint &gt; Insert Breakpoint</i> komutunu seçin veya satırın sol kısmında bulunan boşluğu tıklayınız.</p>
<p>➤ Uygulamayı tekrar çalıştırınız.</p>	<p>➤ Adımları atlamak için <i>F10</i> veya <i>F11</i> komutlarını kullanabilirsiniz.</p> <p>➤ Adımları farklı değerler girerek tekrar çalıştırabilirsiniz.</p>

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Tablo ekleyebildiniz mi?		
2. Web sunucu kontrollerinin özelliklerini ayarlayabildiniz mi?		
3. <i>Try – Catch</i> bloklarını yazabildiniz mi?		
4. <i>Breakpoint</i> ile uygulamaya adımlar ekleyebildiniz mi?		
5. <i>Breakpoint</i> adımlarını sırayla çalıştırabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) Programlama yazılımında bir görünüm modunda yapılan değişiklik diğer görünüm modlarında gösterilmez.
2. ( ) Toolbox panelinde sunucu kontrolleri üzerinde çift tıklandığında, ilgili kontrol, otomatik olarak çalışma alanına eklenir.
3. ( ) Programlama yazılımında bir tablo eklendikten sonra istenirse satır ve/veya sütun eklenebilir.
4. ( ) Bir web sitesi çalışmasına web sayfası *Solution Explorer* panelinden veya *Web Site* menüsünden eklenebilir.
5. ( ) *AccessKey* özelliği CTRL tuşu ile birlikte diğer tuşların kombinasyonu kullanılarak kontrollere kısayol atamak için kullanılır.
6. ( ) *Font* özelliği sunucu kontrollerinin yazı özelliklerini ayarlamak için kullanılır.
7. ( ) *Formatting* araç çubuğunda *Format* menüsü komutlarından bazılarının kısayolları yer almaktadır.
8. ( ) *Try – Catch* komutu kullanılarak program içinde oluşabilecek hataların önüne geçilebilir.

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.



# ÖĞRENME FAALİYETİ-2

## AMAÇ

Doğrulama kontrollerini kullanabileceksiniz.

## ARAŞTIRMA

- Doğrulama kontrollerinin kullanım amacını araştırınız.

## 2. DOĞRULAMA KONTROLLERİ

Doğrulama Kontrolleri, kullanıcılardan veri alınması gerektiği durumlarda, alınan verilerin doğruluğunu denetlemek için kullanılır. Bu kontroller, istemcide çalışan script tabanlı kontrollerdir.

### 2.1. RequiredFieldValidator

Kullanıcının belirtilen alanı boş geçmesini engelleyerek mutlaka doldurmasını sağlayan doğrulama kontrolüdür. Bu kontrol, belirtilen alan boş geçildiğinde istenen hata mesajını ekrana getirerek sayfanın sunucuya gitmesini engeller. *RequiredFieldValidator* kontrolü *Toolbox* panelinde *Validation* grubunda yer almaktadır.

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="TextBox1" ErrorMessage="Bu alan boş geçilemez..."
```

*RequiredFieldValidator* kontrolüne ait özellikler şunlardır:

- **ID:** Kontrol adını belirlemek için kullanılır.
- **ControlToValidate:** Kontrol edilecek olan sunucu kontrolünü belirlemek için kullanılır. Kontrol edilecek sunucu kontrolünün ID'si burada belirtilir.
- **ErrorMessage:** Kullanıcıya gönderilecek hata mesajı bu alana girilir.

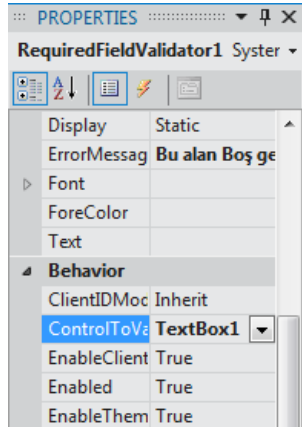
**Örnek:** Kullanıcı adı ve şifre alanlarının boş geçilmesini engellemek için oluşturulan uygulama yapınız.

- Web projesine boş bir sayfa ekleyin.
- Sayfaya, ekran görüntüsüne uygun kontrolleri ekleyin (Resim 1.40).

Kullanıcı Adı  
 RequiredFieldValidator  
 Şifre  
 RequiredFieldValidator  
 Giriş

**Resim 2.1: RequiredFieldValidator uygulaması**

- *RequiredFieldValidator* tarafından kontrol edilecek sunucu kontrollerini *ControlToValidate* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.
- *RequiredFieldValidator* hata mesajını *ErrorMessage* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.



**Resim 2.2: RequiredFieldValidator kontrolü özellikleri**

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="TextBox1" ErrorMessage="Bu alan boş geçilemez">
</asp:RequiredFieldValidator>
```

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
ControlToValidate="TextBox2" ErrorMessage="Bu alan boş geçilemez">
</asp:RequiredFieldValidator>
```

- İstenirse hata mesajı font özellikleri ve rengi *Properties* panelinden *Font* ve *ForeColor* özelliğinden ayarlanabilir.
- Uygulamayı çalıştırın ve alanları boş bırakarak butona tıklayın. Bu durumda ekran çıktısı aşağıdaki gibi olmalıdır.

Kullanıcı Adı  Bu alan boş geçilemez

Şifre  Bu alan boş geçilemez

Resim 2.3: RequiredFieldValidator uygulaması ekran çıktısı

## 2.2. RangeValidator

*RangeValidator* kontrolü, kullanıcı tarafından girilen bir değerin belirli bir aralıkta tutulmasını sağlamak için kullanılır. Bunun için *RangeValidator* kontrolüne maksimum ve minimum değerler girilir. *RangeValidator* kontrolü *Toolbox* panelinde *Validation* grubunda yer almaktadır.

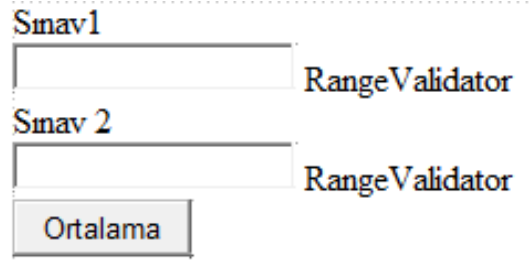
```
<asp:RangeValidator ID="RangeValidator1" runat="server"
ControlToValidate="TextBox1" ErrorMessage="RangeValidator"
MaximumValue="100" MinimumValue="0" Type="Double"></asp:RangeValidator>
```

Bu kontrole ait özellikler şunlardır;

- **ID:** Kontrol adını belirlemek için kullanılır.
- **ControlToValidate:** Kontrol edilecek olan sunucu kontrolü belirlemek için kullanılır. Kontrol edilecek sunucu kontrolünün ID'si burada belirtilir.
- **ErrorMessage:** Kullanıcıya gönderilecek hata mesajı bu alana girilir.
- **MaximumValue:** Sınırlamada üst sınırı belirtir.
- **MinumumValue:** Sınırlamada alt sınırı belirtir.
- **Type:** Verinin tipini belirlemek için kullanılır.

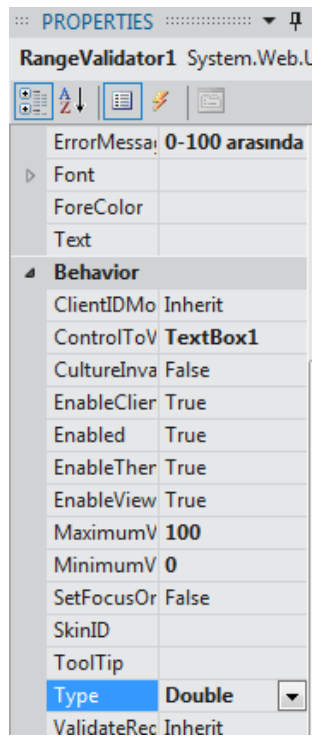
**Örnek:** Bir öğrenciye ait 1. Ve 2. Sınav notlarının 0-100 arasında girilmesini sağlayan uygulamayı oluşturun.

- Web projesine boş bir sayfa ekleyin.
- Sayfaya ekran görüntüsüne uygun kontrolleri ekleyin.



#### 2.4: RangeValidator uygulaması

- *RangeValidator* tarafından kontrol edilecek sunucu kontrollerini *ControlToValidate* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.
- *RangeValidator* hata mesajını *ErrorMessage* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.
- *RangeValidator* maksimum sınırı *MaximumValue* ve minimum sınırı *MinimumValue* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.



Resim 2.5: RangeValidator kontrolü özellikleri

```
<asp:RangeValidator ID="RangeValidator1" runat="server"
ControlToValidate="TextBox1" ErrorMessage="0-100 arasında bir deęer
giriniz" MaximumValue="100" MinimumValue="0"
Type="Double"></asp:RangeValidator>
```

```
<asp:RangeValidator ID="RangeValidator2" runat="server"
ControlToValidate="TextBox2" ErrorMessage="0-100 arasında bir deęer
giriniz" MaximumValue="100" MinimumValue="0"
Type="Double"></asp:RangeValidator>
```

- İstenirse hata mesajı font özellikleri ve rengi *Properties* panelinden *Font* ve *ForeColor* özelliğinden ayarlanabilir.
- Uygulamayı çalıştırın ve alanlara 0-100 aralığı dışında bir deęer girin. Ekran çıktısı aşağıdaki gibi olmalıdır.

Sınav1  
1000 0-100 arasında bir deęer giriniz

Sınav 2  
-20 0-100 arasında bir deęer giriniz

Ortalama

Resim 2.6: RangeValidator uygulaması ekran çıktısı

## 2.3. CompareValidator

*CompareValidator* kontrolü, kullanıcılar tarafından girilen deęerlerin başka bir deęerle karşılaştırmasını yapar. Girilen deęer, başka bir alandaki bir deęerle ya da belirtilen sabit bir deęerle karşılaştırılabilir.

```
<asp:CompareValidator ID="CompareValidator1" runat="server"
ErrorMessage="CompareValidator"></asp:CompareValidator>
```

Bu kontrole ait özellikler şunlardır:

- **ID:** Kontrol adını belirlemek için kullanılır.
- **ControlToValidate:** Karşılaştırılacak deęerin girildiği alandır.
- **ControlToCompare:** Karşılaştırma başka bir alan ile yapılacaksa dięer alanı belirtir.
- **ValueToCompare:** Eđer karşılaştırma sabit bir deęerle yapılacaksa sabit deęer burada girilir.
- **ErrorMessage:** Kullanıcıya gönderilecek hata mesajı bu alana girilir.

**Örnek:** Kullanıcı kayıt ekranında şifre onayında şifrenin aynı olması istenir. CompareValidator kullanarak deęer karşılaştırmasını yapın.

- Web projesine boş bir sayfa ekleyin.

- Sayfaya ekran görüntüsüne uygun kontrolleri ekleyin.

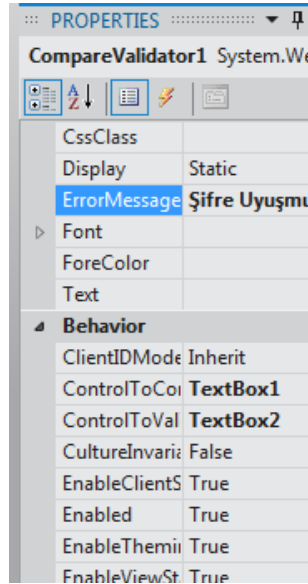
Şifre  
 CompareValidator

Şifre Onay

Kayıt

**Resim 2.7: CompareValidator uygulaması**

- *CompareValidator* tarafından kontrol edilecek sunucu kontrolünü *ControlToValidate* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.
- *CompareValidator* tarafından karşılaştırma yapılacak sunucu kontrolünü *ControlToCompare* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.
- *CompareValidator* hata mesajını *ErrorMessage* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.

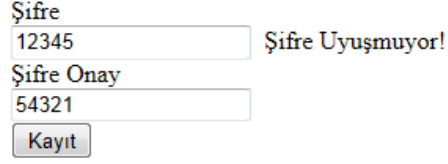


**Resim 2.8: CompareValidator kontrolü özellikleri**

```
<asp:CompareValidator ID="CompareValidator1" runat="server"
ControlToCompare="TextBox1" ControlToValidate="TextBox2"
ErrorMessage="Şifre Uyuşmuyor!"></asp:CompareValidator>
```

- İstenirse hata mesajı font özellikleri ve rengi *Properties* panelinden *Font* ve *ForeColor* özelliğinden ayarlanabilir.

- Uygulamayı çalıştırın ve alanlara şifre alanına farklı şifre onay alanına farklı bir değer girin. Ekran çıktısı aşağıdaki gibi olmalıdır.



Şifre  
12345 Şifre Uyuşmuyor!  
Şifre Onay  
54321  
Kayıt

Resim 2.9: CompareValidator uygulaması ekran çıktısı

## 2.4. RegularExpressionValidator

*RegularExpressionValidator* kontrolü, kullanıcı tarafından girilen değer belirlenilen bir söz dizimi şeklinde girilmesini sağlamak için kullanılır. Girilen değer istenilen söz dizilimine uygun değilse sayfanın sunucuya gitmesini engelleyerek hata mesajı üretir.

Geniş bir kullanım alanı vardır. Örneğin, e-mail adreslerinin, telefon numaralarının, web site adreslerinin vb. girilmesinde bu kontrol ile söz dizilimi oluşturulabilir.

```
<asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server" ErrorMessage="RegularExpressionValidator">
</asp:RegularExpressionValidator>
```

Bu kontrole ait özellikler şunlardır:

- **ID:** Kontrol adını belirlemek için kullanılır.
- **ErrorMessage:** Kullanıcıya gönderilecek hata mesajının girileceği alandır.
- **ValidationExpression:** Girilecek söz diziliminin seçileceği alandır. Söz dizilimi için programlama yazılımı belirli söz dizimleri sunar. Ancak bu söz dizimleri oldukça dar kapsamlıdır.

**Örnek:** Bir web adresinin söz dizilimini kontrol eden bir uygulama yapın.

- Web projesine boş bir sayfa ekleyin.
- Sayfaya ekran görüntüsüne uygun kontrolleri ekleyin.

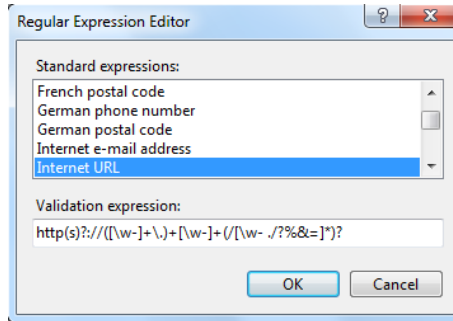


Web Adresiniz  
RegularExpressionValidator  
Kayıt

Resim 2.10: RegularExpressionValidator uygulaması

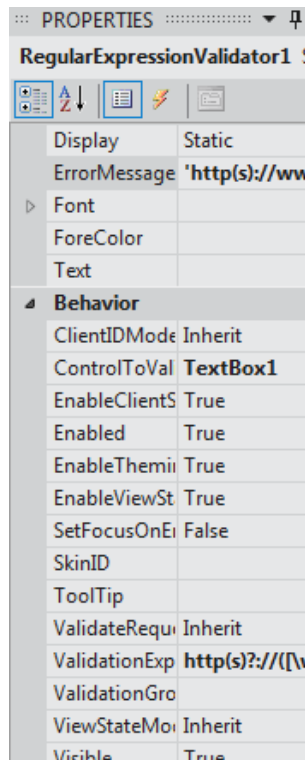
- *RegularExpressionValidator* tarafından kontrol edilecek sunucu kontrolünü *ControlToValidate* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.

- *RegularExpressionValidator* söz dizilimini *ValidationExpression* özelliğinden Internet URL olarak seçin. HTML ve sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.



**Resim 2.11: RegularExpressionValidator kontrolü söz dizilimini seçilmesi**

- *RegularExpressionValidator* hata mesajını *ErrorMessage* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.



**Resim 2.12: RegularExpressionValidator kontrolü özellikleri**

```
<asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server" ControlToValidate="TextBox1"
ErrorMessage="'http(s)://www.siteadı.uzantı' şeklinde girin!"
ValidationExpression="http(s)?://([\w-]+\.)+[\w-]+(/[\w- ./?%&=]*)?">
</asp:RegularExpressionValidator>
```



- İstenirse hata mesajı font özellikleri ve rengi *Properties* panelinden *Font* ve *ForeColor* özelliğinden ayarlanabilir.
- Uygulamayı çalıştırın ve web sitesini istenilen formatın dışında bir şekilde girin. Ekran çıktısı aşağıdaki gibi olmalıdır.

Web Adresiniz

 'http(s)://www.siteadı.uzantı' şeklinde girin!

Resim 2.13: RegularExpressionValidator uygulaması ekran çıktısı

## 2.5. CustomValidator

*CustomValidator* kontrolü, kullanıcı tarafından girilen değerleri kontrol etmek için doğrulama kontrolleri ihtiyaca cevap vermediği zaman, geliştiricilerin kendi yazdıkları doğrulama fonksiyonlarının kullanımı içindir.

```
<asp:CustomValidator ID="CustomValidator1" runat="server">
</asp:CustomValidator>
```

## 2.6. ValidationSummary

*ValidationSummary* kontrolü, doğrulama kontrollerinin ürettiği hata mesajlarını bir yerde toplamak için kullanılır. Bu kontrol doğrudan sayfa içine kopyalanarak kullanılabilir. Bu durumda hata mesajı *ValidationSummary* ile gösterilecektir. Ancak doğrulama kontrolünün kendi hata mesajı da aktif olduğu için o kontrol de hata mesajı gösterecektir. Bu durumda iki hata mesajı üretilir ki bu, tasarımda istenmeyen bir durum yaratır.

```
<asp:ValidationSummary ID="ValidationSummary1" runat="server" />
```

*ValidationSummary* kontrolü kullanarak hata mesajı görüntülenecekse, hata mesajı bu kontrole yazılır, diğer kontrollerin *Text* özelliğine hatanın nerede oluştuğunu göstermek için \* vb. tek karakterlik mesaj yazılır.

**Örnek:** Kullanıcı Adı ve Şifre ile giriş yapılan bir login panelinde gerekli kontrolleri yapan bir uygulama hazırlayın. (Bu alanlar boş geçilemez ve şifre 6 haneli sayısal bir ifade olmalıdır.)

- Web projesine boş bir sayfa ekleyin.
- Sayfaya ekran görüntüsüne uygun kontrolleri ekleyin.

Kullanıcı Adı  
 RequiredFieldValidator

Şifre  
 RequiredFieldValidatorRegularExpressionValidator

Button

- Error message 1.
- Error message 2.

**Resim 2.14: RegularExpressionValidator Uygulaması**

- *RequiredFieldValidator* tarafından kontrol edilecek sunucu kontrolünü *ControlToValidate* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.
- *RequiredFieldValidator* hata mesajını *ErrorMessage* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.
- *RegularExpressionValidator* tarafından kontrol edilecek sunucu kontrolünü *ControlToValidate* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.
- *RegularExpressionValidator* hata mesajını *ErrorMessage* özelliğine ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.
- *RegularExpressionValidator* söz dizilimini *ValidationExpression* özelliğine `\d{6}` kodunu yazın. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.
- *RegularExpressionValidator* ve *RegularExpressionValidator* kontrollerinin *Text* özelliğine `*` karakterini ekleyin. HTML ve Sunucu kontrolleri kodları sayfasından ya da *Properties* panelinden bu özellik ayarlanabilir.

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
ControlToValidate="TextBox1" ErrorMessage="Boş Geçilemez!"> *
</asp:RequiredFieldValidator>
```

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
ControlToValidate="TextBox2" ErrorMessage="Boş Geçilemez!"> *
</asp:RequiredFieldValidator>
```

```
<asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server" ControlToValidate="TextBox2" ErrorMessage="Şifre 6 haneli
sayısal ifade olmalıdır!" ValidationExpression="\d{6}"> *
</asp:RegularExpressionValidator>
```

```
<asp:ValidationSummary ID="ValidationSummary1" runat="server" />
```

- İstenirse hata mesajı font özellikleri ve rengi *Properties* panelinden *Font* ve *ForeColor* özelliğinden ayarlanabilir.
- Uygulamayı çalıştırın ve web sitesini istenilen formatın dışında bir şekilde girin. Ekran çıktısı aşağıdaki gibi olmalıdır.

Kullanıcı Adı <input type="text"/> *	Kullanıcı Adı <input type="text"/> *
Şifre <input type="text"/> *	Şifre <input type="text" value="1234"/> *
<input type="button" value="Giriş"/>	<input type="button" value="Giris"/>
<ul style="list-style-type: none"> <li>• Boş Geçilemez!</li> <li>• Boş Geçilemez!</li> </ul>	<ul style="list-style-type: none"> <li>• Boş Geçilemez!</li> <li>• Şifre 6 haneli sayısal ifade olmalıdır!</li> </ul>

Resim 2.15: ValidatonSummary uygulaması ekran çıktısı

## 2.7. ValidationGroup Özelliği

Bir sayfada birden fazla doğrulama kontrolü kullanıldığı zaman programın çalışmasıyla ilgili problemler yaşanabilmektedir. Programlama yazılımı, bu duruma çözüm olarak *ValidationGroup* özelliğini sunmaktadır. *ValidationGroup* özelliği ile aynı sayfa içinde farklı doğrulama kontrolleri kullanılarak bu kontrollerin diğer kontrollerden etkilenmesi engellenmiştir.

*ValidationGroup* özelliği için:

- Yeni bir sayfa açın ve aşağıdaki kontrolleri sayfaya ekleyin.

Kullanıcı Adı	<input type="text"/>	Boş geçilemez!
E Mail	<input type="text"/>	Boş geçilemez!
Şifre	<input type="text"/>	Boş geçilemez!
Şifre Doğrulama	<input type="text"/>	Şifre aynı olmalıdır!
<input type="button" value="Üye Ol"/>		
Doğrulama Kodu	[Label7]	
Doğrulama Kodu	<input type="text"/>	Boş geçilemez!
<input type="button" value="Gönder"/>		
[Label6]		

Resim 2.16: ValidationGroup özelliği uygulaması

- Hazırlanan üyelik paneli doğrulama kontrolleri ve *Üye Ol* butonu *ValidationGroup* özelliğini *Properties* panelini kullanarak *uye* olarak atayın.

ValidationGroup	uye
-----------------	-----

Resim 2.17: Üyelik paneli ValidationGroup özelliğinin ayarlanması

- Hazırlanan doğrulama kodu panelinin *Gönder* butonu *ValidationGroup* özelliğini *Properties* panelini kullanarak **dogrulama** olarak atayın.

ValidationGroup    dogrulama

**Resim 2.18: Doğrulama kodu paneli ValidationGroup özelliğinin ayarlanması**

- *Üye Ol* butonuna aşağıdaki kodları ekleyin. Bu kodlar ile üyelik panelindeki kontrol değerleri temizlenirken *Label7* değerine rastgele bir doğrulama kodu oluşturulacaktır.

```
TextBox1.Text = "";  
TextBox2.Text = "";  
TextBox3.Text = "";  
TextBox4.Text = "";  
Random kod = new Random();  
Label7.Text = Convert.ToString(kod.Next(1, 10000));
```

- *Gönder* butonuna aşağıdaki kodu yazın. Bu kodlar ile doğrulama kodu ile yazılan kod karşılaştırılacak ve doğru ise *Doğrulama kodu doğru! Kayıt başarılı*, değilse *Doğrulama kodu Yanlış! Tekrar deneyin* yazarak yeni bir kod üretilecektir.

```
if (Label7.Text == TextBox5.Text)  
{  
    Label6.Text = "Doğrulama kodu doğru! Kayıt başarılı";  
}  
else  
{  
    Label6.Text="Doğrulama kodu Yanlış! Tekrar deneyin";  
    Random kod = new Random();  
    Label7.Text = Convert.ToString(kod.Next(1, 10000));  
}
```

- Uygulamayı çalıştırın.
- Üyelik panelinde değerleri boş geçerek uygulamanın çalışmasını gözlemleyin.

Kullanıcı Adı	<input type="text"/>	Boş geçilemez!
E Mail	<input type="text"/>	Boş geçilemez!
Şifre	<input type="text"/>	Boş geçilemez!
Şifre Doğrulama	<input type="text"/>	
<input type="button" value="Üye Ol"/>		

**Resim 2.19: Üyelik paneli RequiredFieldValidator kontrollerinin çalışması**

- Üyelik panelinde şfre değerlerini farklı girerek uygulamanın çalışmasını gözlemleyin.

Kullanıcı Adı	<input type="text" value="mtegm"/>
E Mail	<input type="text" value="mtegm@meb.gov.tr"/>
Şifre	<input type="text" value="1234"/>
Şifre Doğrulama	<input type="text" value="4321"/> Şifre aynı olmalıdır!
<input type="button" value="Uye Ol"/>	

**Resim 2.20: Üyelik paneli CompareValidator kontrollerinin çalışması**

- Doğrulama kodu panelinde, doğrulama kodunu boş geçerek uygulamanın çalışmasını gözlemleyin.

Doğrulama Kodu	3733
Doğrulama Kodu	<input type="text"/>
<input type="button" value="Gönder"/>	

**Resim 2.21: Doğrulama Kodunun oluşturulması**

Doğrulama Kodu	3733
Doğrulama Kodu	<input type="text"/> Boş geçilemez!
<input type="button" value="Gönder"/>	

**Resim 2.22: Doğrulama kodu paneli RequiredFieldValidator kontrolünün çalışması**

- Doğrulama kodu panelinde, doğrulama kodunu yanlış girerek uygulamanın çalışmasını gözlemleyin.

Doğrulama Kodu	6463
Doğrulama Kodu	<input type="text" value="373333"/>
<input type="button" value="Gönder"/>	
Doğrulama kodu Yanlış! Tekrar deneyin	

**Resim 2.23: Hatalı doğrulama kodunun girilmesi sonucu oluşan ekran çıktısı**

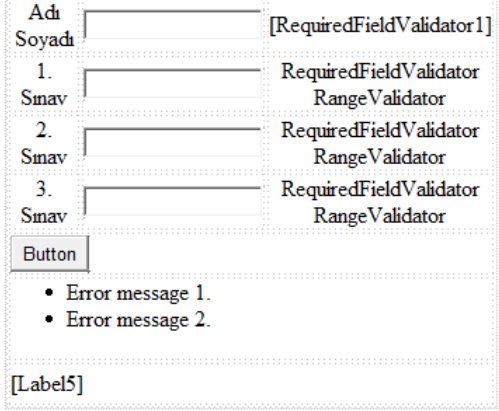
- Doğrulama kodu panelinde, doğrulama kodunu doğru girerek uygulamanın çalışmasını gözlemleyin.

Doğrulama Kodu	6463
Doğrulama Kodu	<input type="text" value="6463"/>
<input type="button" value="Gönder"/>	
Doğrulama kodu doğru! Kayıt başarılı	

**Resim 2.24: Doğru doğrulama kodunun girilmesi sonucu oluşan ekran çıktısı**

## UYGULAMA FAALİYETİ

Doğrulama kontrollerini kullanarak aşağıdaki uygulamayı yapınız.

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"> <li>➤ Programlama yazılımını başlatınız.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Yazılım kısayolunu kullanabilirsiniz.</li> </ul>
<ul style="list-style-type: none"> <li>➤ Sayfaya 7x3 bir tablo ekleyiniz.</li> </ul>	<ul style="list-style-type: none"> <li>➤ <i>Table &gt; Insert Table</i> komutunu kullanabilirsiniz.</li> <li>➤ Tabloyu tasarıma uygun boyutlandırabilirsiniz.</li> </ul>
<ul style="list-style-type: none"> <li>➤ Tablonun içine aşağıdaki web kullanıcı kontrollerini ve doğrulama kontrollerini ekleyiniz.</li> </ul> 	<ul style="list-style-type: none"> <li>➤ <i>Toolbox</i> panelinden kontrolleri çalışma alanına sürükleyerek ekleyebilirsiniz.</li> <li>➤ Alanların boş geçilmemesi için <i>RequiredFieldValidator</i> kontrolünü kullanabilirsiniz.</li> <li>➤ Sınavlara 0-100 arasında değer girilmesi için <i>RangeValidator</i> kontrolünü kullanabilirsiniz.</li> <li>➤ Hata mesajlarını bir arada göstermek için <i>ValidationSummary</i> kontrolünü kullanabilirsiniz.</li> </ul>
<ul style="list-style-type: none"> <li>➤ Doğrulama kontrollerinin kontrol edileceği sunucu kontrollerini ayarlayınız.</li> </ul>	<ul style="list-style-type: none"> <li>➤ <i>Properties</i> panelinden <i>ControlToValidate</i> özelliğini kullanabilirsiniz.</li> </ul>
<ul style="list-style-type: none"> <li>➤ Doğrulama kontrollerinin hata mesajlarını ekleyiniz.</li> </ul>	<ul style="list-style-type: none"> <li>➤ <i>Properties</i> panelinden <i>ErrorMessage</i> özelliğini kullanabilirsiniz.</li> <li>➤ <i>ValidationSummary</i> kontrolü kullanıldığı için diğer doğrulama kontrolü <i>Text</i> özelliğine * karakteri girebilirsiniz.</li> </ul>
<ul style="list-style-type: none"> <li>➤ <i>RangeValidator</i> özelliğine girilecek değerlerin sınırlarını ayarlayınız.</li> </ul>	<ul style="list-style-type: none"> <li>➤ <i>Properties</i> panelinden <i>Maximum Value</i> ve <i>Minimum Value</i> özelliklerini kullanabilirsiniz.</li> <li>➤ Değerler sayısal değer olduğundan <i>Type</i> özelliğini <i>integer</i> olarak ayarlamayı unutmayınız.</li> </ul>

<p>➤ Button <i>Click</i> olayına aşağıdaki kodları ekleyiniz.</p> <pre>int not1=Convert.ToInt32(TextBox2.Text); int not2=Convert.ToInt32(TextBox3.Text); int not3=Convert.ToInt32(TextBox4.Text); double ort=(not1 + not2 + not3) / 3; Label5.Text = TextBox1.Text + " Ortalaması= " + ort;</pre>	<p>➤ Kod sayfasını kullanabilirsiniz.</p>
<p>➤ Uygulamayı çalıştırınız.</p>	<p>➤ <i>Debug &gt; Start Debugging (F5)</i> komutunu kullanabilirsiniz.</p>
<p>➤ Hatalı değerler girerek çalışmayı izleyin.</p> <p>Adı Soyadı <input type="text"/> *</p> <p>1. Sınav <input type="text"/> *</p> <p>2. Sınav <input type="text" value="121"/> *</p> <p>3. Sınav <input type="text"/> *</p> <p><input type="button" value="Button"/></p> <p>Boş Geçilemez! Boş geçilemez! 0 -100 arasında bir değer girin! Boş geçilemez!</p>	<p>➤ Değerleri boş geçebilirsiniz. ➤ Sınav notlarını 0-100 aralığı dışında girebilirsiniz.</p>
<p>➤ Doğru değerler girerek ortalamayı hesaplatınız.</p> <p>Adı Soyadı <input type="text" value="Ali Veli"/></p> <p>1. Sınav <input type="text" value="75"/></p> <p>2. Sınav <input type="text" value="80"/></p> <p>3. Sınav <input type="text" value="70"/></p> <p><input type="button" value="Button"/></p> <p>Ali Veli Ortalaması= 75</p>	<p>➤ Adı ve soyadını tam girebilirsiniz. ➤ Notları 0-100 arasında bir değer girebilirsiniz.</p>

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Tablo ekleyebildiniz mi?		
2. Sunucu kontrollerini ekleyebildiniz mi?		
3. Doğrulama kontrollerini ekleyebildiniz mi?		
4. Doğrulama kontrollerinin denetleyeceği sunucu kontrollerini ayarlayabildiniz mi?		
5. Doğrulama kontrolleri hata mesajlarını ayarlayabildiniz mi?		
6. Doğrulama kontrolleri Text özelliğini ayarlayabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.



## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) Kullanıcılar tarafından girilen değerlerin doğruluğunu kontrol etmek için doğrulama kontrolleri kullanılır.
2. ( ) *RequiredFieldValidator* kontrolü alanların boş geçilmesini engellemek için kullanılır.
3. ( ) *RangeValidator* kontrolünde girilecek aralık belirtilerek sınırlama getirilebilir.
4. ( ) *CompareValidator* kontrolü sadece iki kontrolün değerini karşılaştırır.
5. ( ) *RegularExpressionValidator* kontrolünde söz dizimi isteğe uygun ayarlanabilir.
6. ( ) *CustomValidator* kontrolü doğrulama kontrolleri isteğe cevap vermediğinde kullanılabilir.
7. ( ) *ValidationSummary* birden çok doğrulama kontrolü kullanıldığında hata mesajlarını bir yerde göstermek için kullanılır.
8. ( ) *ValidationSummary* kontrolü kullanıldığında diğer doğrulama kontrollerinin hata mesajları gösterilmez.

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-3

## AMAÇ

Web kullanıcı kontrollerini oluşturabilecek ve başka sayfalarda kullanabileceksiniz.

## ARAŞTIRMA

- Web kullanıcı kontrollerinin diğer kontrollerden farkını araştırınız.

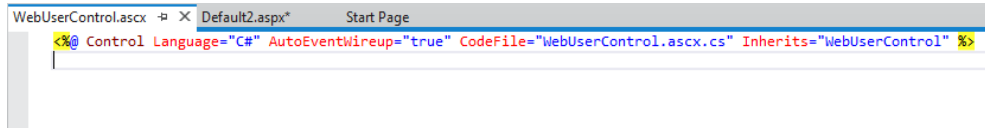
## 3. WEB KULLANICI KONTROLLERİ

Kullanıcı kontrolleri bir içeriğin birden fazla sayfada kullanılmasına olanak sağlar. Bir web sayfasına eklenerek içerik sağlar ve içeriğin yönetiminin tek bir yerden yapılmasını sağlar.

Web kullanıcı kontrolleri sunucu tarafı çalışan bir form içermezler. Bu nedenle çalışmak için mutlaka sunucu tarafı çalışan bir web formun içine eklenmelidir.

ASP.NET yeni sürümlerinde *MasterPage* kullanımı ile kullanıcı kontrollerinin kullanımı azalmıştır (MasterPage konusuna ilerleyen modüllerde yer verilmektedir).

Web kullanıcı kontrollerinin dosya uzantısı **ascx**'dir. Web kullanıcı kontrolleri normal ASP.NET sayfaları gibi programlanabilir ve kodları farklı bir dosyada tutulabilir. Kod sayfasının uzantısı da ASP.NET sayfalarına benzer şekilde **.ascx.cs**'dir.



```
WebUserControl.ascx  x Default2.aspx*  Start Page
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="WebUserControl.ascx.cs" Inherits="WebUserControl" %>
```

Resim 3.1: WebUserControl Sayfası Kodları

Web kullanıcı kontrolü kodları, klasik ASP.NET sayfalarının kodlarına benzerlik göstermektedir. *.aspx* dosyaları *@Page* direktifi ile başlarken *.ascx* dosyaları *@Control* direktifi ile başlamakta ve kontrol ayarlarının bir kısmı bu alanda gerçekleştirilebilmektedir.

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="WebUserControl.ascx.cs" Inherits="WebUserControl" %>
```

Yukarıdaki kod incelenecek olursa:

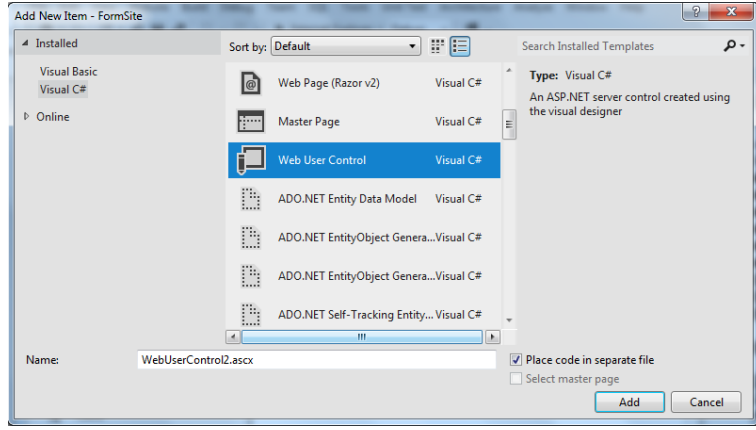
- Arka planda kullanılacak olan programlama dili *Language* özelliği ile belirlenmektedir.
- *PageLoad* gibi olayların otomatik olarak tetiklenmesini sağlamak amacı ile *AutoEvenWireup* özelliği *true* olarak atanmıştır.

- Arka plandaki kodları tutacak olan dosya *CodeFile* özelliği ile belirlenmektedir.
- *UserControl*'ün sınıf adı *Inherits* özeliğiyle belirtilmektedir.

### 3.1. Web Kullanıcı Kontrolleri Oluşturma

Bir web sitesine yeni bir kullanıcı kontrolü eklemek için:

- *Web Site > Add New Item* komutunu seçin ya da *Solution Explorer* panelinde proje adını sağ tıklayın ve *Add* açılır menüsünü kullanın.
- Ekranı gelen *Add New Item – Form Site* iletişim penceresinden *Web User Control* komutunu seçin.



**Resim 3.2: Add New Item – form site iletişim penceresi**

- *Name* alanına web kullanıcı kontrolüne uygun bir isim verilebilir. Değiştirilmezse programlama yazılımı sıralı olarak isim atayacaktır.
- Web kullanıcı kontrolü programlama yazılımı tarafından otomatik olarak proje dosyalarına eklenecek ve ekrana getirilecektir.

### 3.2. Web Kullanıcı Kontrolleri Ekleme

Web kullanıcı kontrolleri hazırlandıktan sonra uygulama çalıştırılacak olursa ekrana kullanıcı kontrolü yerine uygulama açılış sayfası gelir. Çünkü web kullanıcı kontrolleri tek başlarına çalışmazlar, mutlaka bir sunucu tarafı çalışan form içine eklenmelidirler.

Programlama yazılımı hazırlanan web kullanıcı kontrollerinin doğrudan *Solution Explorer* panelinden sayfaya sürüklenerek eklenmesine izin verir. Web kullanıcı kontrolü sürüklendikten sonra sayfada doğrudan görüntülenecektir.

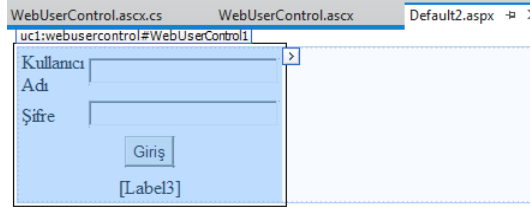
Web kullanıcı kontrolü sayfaya sürüklenerek eklendikten sonra, sayfa kodlarına *@Page* direktifinden hemen sonra *@Register* direktifi eklenir. Bu direktif ile sayfa içerisinde kullanılan kullanıcı kontrolünün konumu ve sayfa içerisindeki etiket adı tanımlamaları yapılır.

```
<%@ Register src="WebUserController.ascx" tagname="WebUserController"
tagprefix="uc1" %>
```

Eklenen web kullanıcı kontrolünün kodları, HTML ve sunucu kontrolleri kodları sayfasına eklenir.

```
<uc1:WebUserController ID="WebUserController1" runat="server" />
```

Tasarım görünümünde ise, web kullanıcı kontrolünün tasarlanan görünümü eklenir.



**Resim 3.3: Web kullanıcı kontrolünün sayfaya eklenmesi**

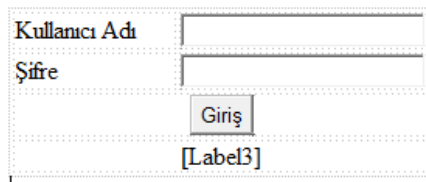
Kullanıcı tarafından gerçekleştirilen sürükleyerek web kullanıcı kontrolünü ekleme işleminden sonra gerçekleşen işlemler programlama yazılımı tarafından otomatik olarak yapılır. Kullanıcı kontrolünde değişiklik yapıldığı zaman, web kullanıcı kontrolünün eklendiği sayfalar yine programlama yazılımı tarafından otomatik olarak güncellenir.

### 3.3. Web Kullanıcı Kontrolleri Tasarlama

Web kullanıcı kontrollerinin tasarım sayfası, klasik ASP.NET sayfalarına benzerlik göstermektedir. Bu nedenle web kullanıcı kontrolleri tasarlanmanın klasik ASP.NET sayfalarını tasarlamaktan bir farkı yoktur.

**Örnek:** Bir Login panelini *WebUserController* ile tasarlayın.

- Projeye yeni bir *WebUserController* ekleyin.
- *WebUserController* tasarım alanına aşağıdaki kontrolleri ekleyin.



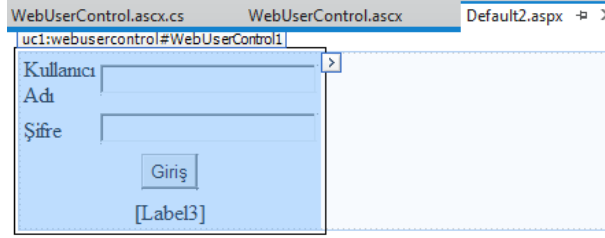
**Resim 3.4: Web kullanıcı kontrolünün tasarlanması**

- Buton *Click* olayına aşağıdaki kodları ekleyin. Kullanıcı adı = "MEB", şifre= "MTEGM" olarak belirlenmiştir.

```
if (TextBox1.Text == "MEB" && TextBox2.Text == "MTEGM")
{
    Label3.Text = "Giriş Başarılı!";
}
```

```
}  
else  
{  
    Label3.Text = "Kullanıcı Adı ya da Şifre yanlış!";  
}
```

- Yeni bir web form ekleyin.
- Web forma oluşturduğunuz Web kullanıcı kontrolünü ekleyin.



**Resim 3.5: Web kullanıcı kontrollerinin tasarlandıktan sonra sayfaya eklenmesi**

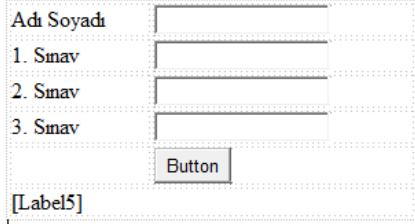
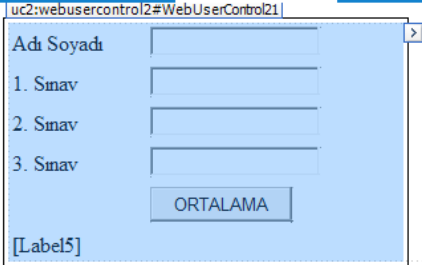
- Uygulamayı çalıştırın ve form alanlarına istenilen formatın dışında değerler girin. Ekran çıktısı aşağıdaki gibi olmalıdır.

Kullanıcı Adı	<input type="text" value="mtegm"/>	Kullanıcı Adı	<input type="text" value="MEB"/>
Şifre	<input type="text" value="mtegm"/>	Şifre	<input type="text" value="MTEGM"/>
	<input type="button" value="Giriş"/>		<input type="button" value="Giriş"/>
	Kullanıcı Adı ya da Şifre yanlış!		Giriş Başarılı

**Resim 3.5: Web kullanıcı kontrolleri uygulaması ekran çıktısı**

## UYGULAMA FAALİYETİ

Web kullanıcı kontrolleri ile ilgili aşağıdaki uygulamayı yapınız.

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"><li>➤ Programlama yazılımını başlatınız.</li><li>➤ Yeni bir kullanıcı kontrol sayfası ekleyiniz.</li></ul>	<ul style="list-style-type: none"><li>➤ Yazılım kısayolunu kullanabilirsiniz.</li><li>➤ <i>Website</i> menüsü veya <i>Solution Explorer</i> panelini kullanabilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Sayfaya 5x2 bir tablo ekleyiniz.</li></ul>	<ul style="list-style-type: none"><li>➤ <i>Table &gt; Insert Table</i> komutunu kullanabilirsiniz.</li><li>➤ Tabloyu tasarıma uygun boyutlandırabilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Tablonun içine aşağıdaki web sunucu kontrollerini ekleyiniz.</li></ul> 	<ul style="list-style-type: none"><li>➤ <i>Toolbox</i> panelinden kontrolleri çalışma alanına sürükleyerek ekleyebilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Oluşturduğunuz kullanıcı formunu web formuna ekleyiniz.</li></ul> 	<ul style="list-style-type: none"><li>➤ Web formuna <i>Solution Explorer</i> panelinden sürükleyerek kullanıcı kontrolünü ekleyebilirsiniz.</li></ul>

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

<b>Değerlendirme Ölçütleri</b>	<b>Evet</b>	<b>Hayır</b>
1. Yeni bir kullanıcı kontrolü sayfası ekleyebildiniz mi?		
2. Tablo ekleyebildiniz mi?		
3. Kullanıcı kontrolünü tasarlayabildiniz mi?		
4. Kullanıcı kontrolünü web formuna ekleyebildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise bir sonraki “Uygulama Faaliyeti”ne geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) Kullanıcı kontrolleri oluşturulan içeriğin birden fazla sayfada kullanılmasına olanak sağlar.
2. ( ) Web kullanıcı kontrolleri sunucu taraflı çalışan bir form sayfasından oluşur.
3. ( ) MasterPage kullanımı ile web kullanıcı kontrollerinin kullanımı da azalmıştır.
4. ( ) Kullanıcı kontrolleri sayfası normal ASP.NET sayfaları gibi programlanabilir.
5. ( ) Web kullanıcı kontrolleri sayfalarının uzantısı ASP.NET sayfaları gibi .aspx'dir.
6. ( ) Web kullanıcı kontrollerinin sayfaları @Page direktifi ile başlar.
7. ( ) Yeni bir web kullanıcı kontrolü Add New Item – FormSite iletişim penceresinden eklenir.
8. ( ) Web kullanıcı kontrolü bir web formuna eklendikten sonra konumu @Register direktifi ile sayfa kodlarında gösterilir.

### DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.



# MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden hangisi programlama yazılımı görünüm modlarından biri değildir?  
A) Design  
B) Split  
C) Code  
D) Source
2. Aşağıdaki komutlardan hangisi bir tabloda seçili satırın altına satır eklemek için kullanılır?  
A) Table > Insert > Column to the Left  
B) Table > Insert > Column to the Right  
C) Table > Insert > Row Above  
D) Table > Insert > Row Below
3. AccessKey özelliği hangi tuş ile kombinasyon oluşturarak kontrole kısayol sağlar?  
A) Alt  
B) Ctrl  
C) Shift  
D) Enter
4. Aşağıdaki sunucu kontrolü özelliklerinden hangisi kenarlık rengini belirler?  
A) BackColor  
B) BorderColor  
C) BorderStyle  
D) BorderWidth
5. Format menüsünde numaralandırma işlemleri için hangi komut kullanılır?  
A) New Style  
B) Font...  
C) Paragraph...  
D) Bullets and Numbering
6. Breakpoint eklemek için aşağıdaki fonksiyon tuşlarından hangisi kullanılır?  
A) F9  
B) F10  
C) F11  
D) F12
7. Try – Catch – Finally kullanırken hata üretebilecek kodlar hangi blok içine yazılmalıdır?  
A) try  
B) catch  
C) finally  
D) try - catch

8. Kullanıcının belirtilen alanı başka bir alandaki değer ile karşılaştırmak için aşağıdaki doğrulama kontrollerinden hangisi kullanılır?
- A) RequiredFieldValidator
  - B) RangeValidator
  - C) CompareValidator
  - D) RegularExpressionValidator
9. Kullanıcının belirtilen alana belirli bir söz dizimi ile değer girmesini sağlamak için aşağıdaki doğrulama kontrollerinden hangisi kullanılır?
- A) ValidationSummary
  - B) RangeValidator
  - C) CompareValidator
  - D) RegularExpressionValidator
10. Doğrulama kontrollerinde hata mesajları hangi özelliğe yazılmalıdır?
- A) ID
  - B) Type
  - C) ControlToValidate
  - D) ErrorMessage
11. ValidationSummary kontrolü kullanıldığında hata mesajının farklı yerlerde ekrana gelmesini engellemek için kontrollerin hangi özelliğine \* vb. bir değer atanmalıdır?
- A) Text
  - B) ErrorMessage
  - C) ControlToValidate
  - D) ID
12. Web kullanıcı kontrollerinin dosya uzantısı aşağıdakilerden hangisidir?
- A) .aspx
  - B) .ascx
  - C) .config
  - D) .asax
13. Web kullanıcı kontrolleri kodları hangi direktif ile başlar?
- A) @Page
  - B) @Control
  - C) @Register
  - D) @uc
14. Aşağıdakilerden hangi web kullanıcı kontrolleri için söylenemez?
- A) Sunucu tarafı çalışan bir form içermezler.
  - B) MasterPage sayfalarının geliştirilmesi ile kullanımı azalmıştır.
  - C) Tek başına bir sayfa gibi çalışabilir.
  - D) Birden fazla web form içinde kullanılabilir.

15. Web kullanıcı kontrolleri bir web formuna eklendikten sonra hangi direktif otomatik olarak sayfa kodlarına eklenir?
- A) @Page
  - B) @Control
  - C) @Register
  - D) @uc

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

# CEVAP ANAHTARLARI

## ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	Doğru
4	Doğru
5	Yanlış
6	Doğru
7	Doğru
8	Doğru

## ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	Doğru
2	Doğru
3	Doğru
4	Yanlış
5	Doğru
6	Doğru
7	Doğru
8	Yanlış

## ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

1	Doğru
2	Yanlış
3	Doğru
4	Doğru
5	Yanlış
6	Yanlış
7	Doğru
8	Doğru

## MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

1	C
2	D
3	A
4	B
5	D
6	A
7	A
8	C
9	D
10	D
11	A
12	B
13	B
14	C
15	C

## KAYNAKÇA

- SHEPHERD George, **Microsoft ASP.NET 4.0 Step by Step**, Microsoft Press, Washington, 2010.
- SHARP John, **Microsoft C# 2008 Step by Step**, Microsoft Press, Washington, 2008.
- ÖZDEMİR Selçuk, **ASP.NET 2.0**, NET Soft.
- [www.acikakademi.com](http://www.acikakademi.com) (14-18 Mayıs 2012)
- [msdn.microsoft.com/en-us/library](http://msdn.microsoft.com/en-us/library) (14-18 Mayıs 2012)
- [msdn.microsoft.com/tr-tr/library](http://msdn.microsoft.com/tr-tr/library) (14-18 Mayıs 2012)