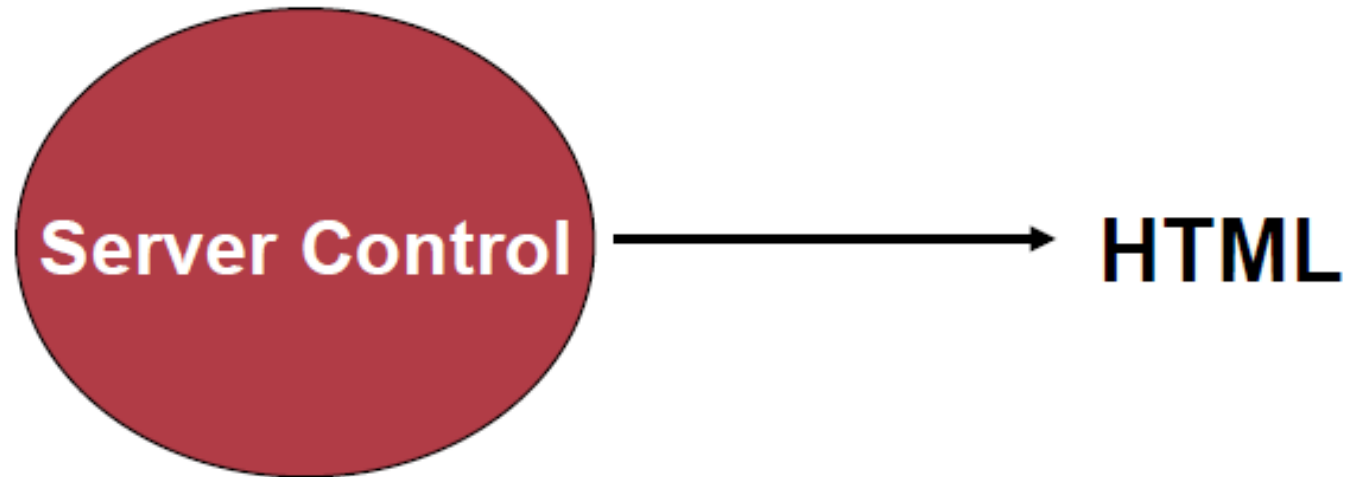


Html ve Web Kontrolleri

# Başlıklar

- Web Server Kontrolleri
- Kontrollerin Durumlarının Yönetimi (State Management)
- ViewState Yapısı
- ControlState Yapısı

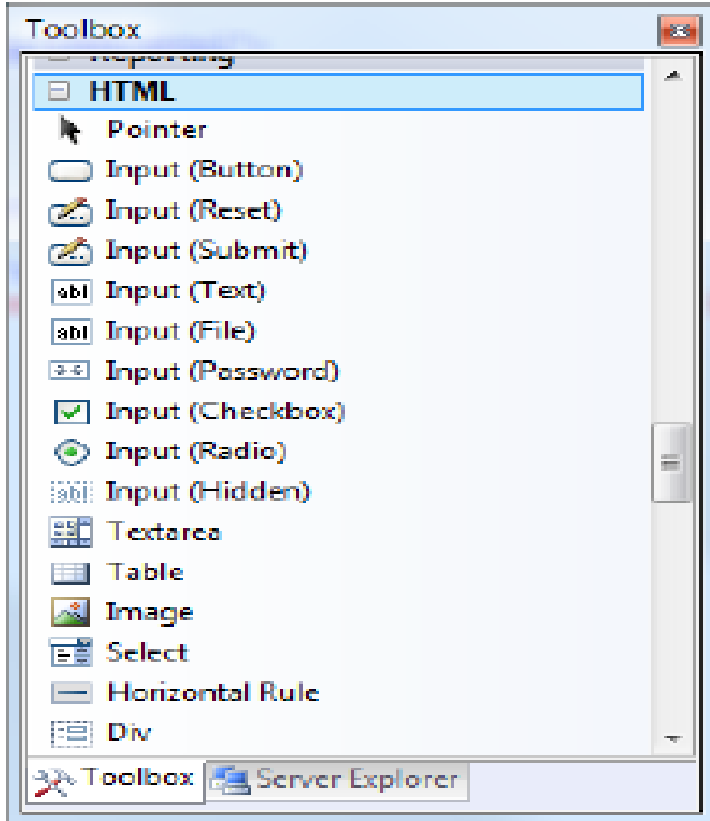
- ASP.NET'de, verinin toplanması, görüntülenmesi ve geçerli kılınması Web Server kontrollerine dayanır.
- Server kontrolleri özellikleri(properties), metotları ve olayları olan sınıflardır.
- Server kontrolleri dinamik olarak HTML kodlar üretirler.



Dört tür sunucu kontrolü(server control) bulunmaktadır.  
Bunlar:

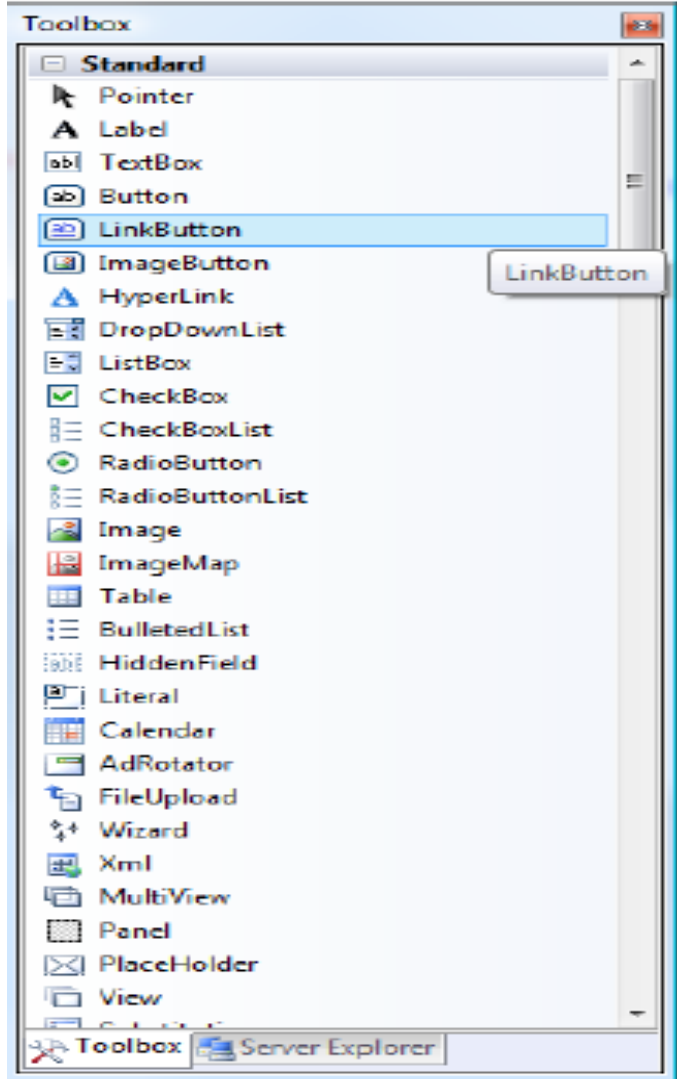
- Html Kontrolleri (**HTML Server Controls**)
- Web Sunucu Kontrolleri (**Web Server Controls**)
- Kullanıcı Kontrolleri (**User Controls**)
- Onaylama Kontrolleri (**Validation Controls**)

# Html Kontrolleri

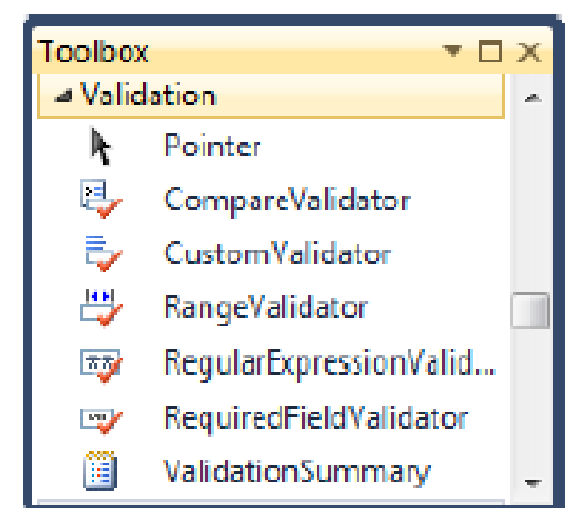
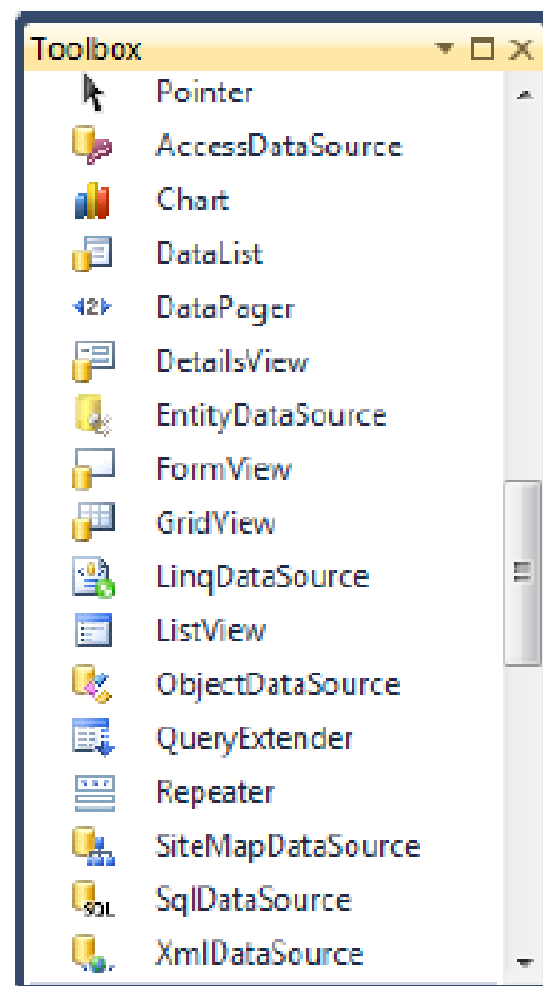
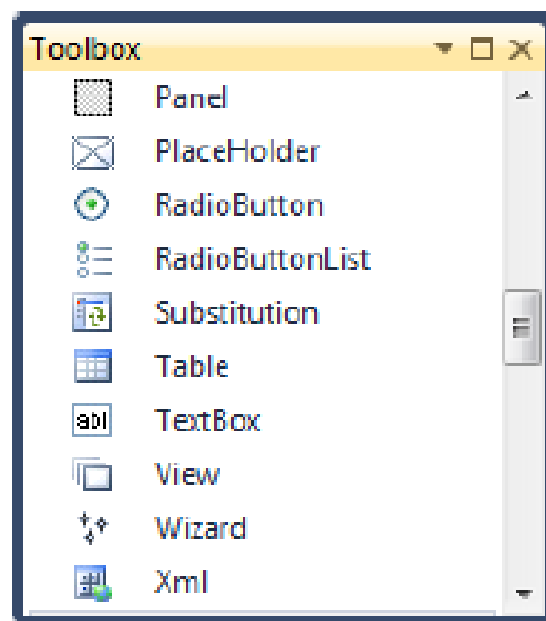
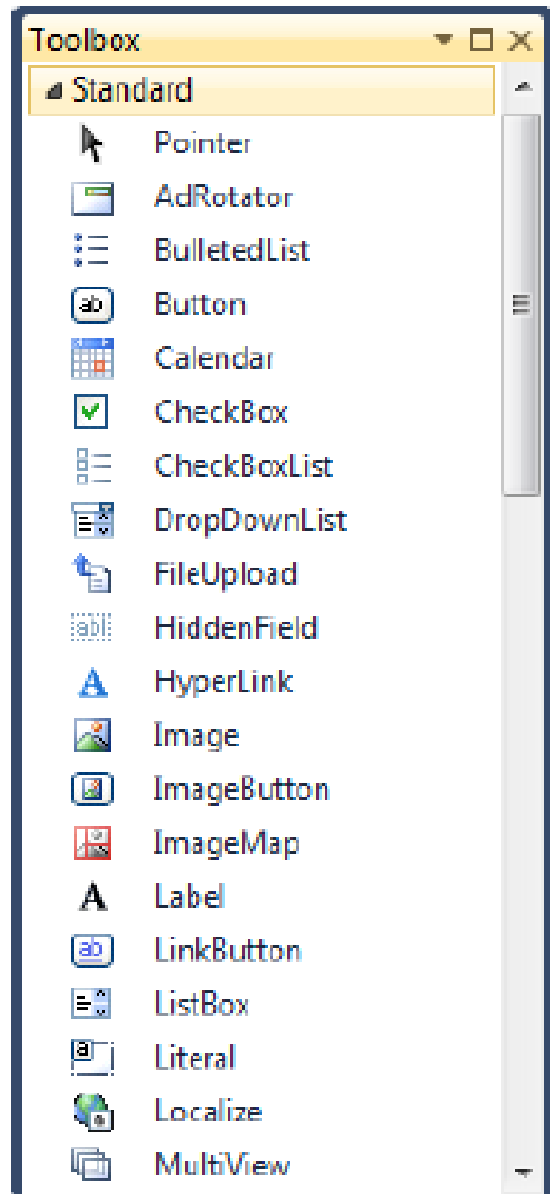


- HTML kontrolleri, standart HTML elemanlarını temsil edebilmek için sunucu tarafında sınıflar sağlarlar. Herhangi bir HTML elemanının özelliğine "runat=server" eklemesi yapılarak sunucu tarafında temsil edilir.

# Web Kontrolleri



- HTML kontrollerinin alternatifidir. İsimlendirmeleri daha tutarlıdır.
- Bazı Web kontrolleri HTML elemanlarının rollerini üstlenebilirler. Örn:
  - Button = `<input type="button"...>`
  - Label = `<span . . ./>`
- Bazı karmaşık Web kontrolleri ise birden fazla HTML elemanı şeklinde temsil edilebilirler.
  - Calendar
  - AdRotator
  - Xml



# Properties

- Tüm Web kontrolleri yaygın olarak kullandıkları property'lerin tümünü WebControl temel sınıfından miras alırlar.
  - Attributes, BackColor, BorderColor, BorderWidth, ClientID, CssClass, Enabled.....
- Yukarıda belirtilen propertylere ek olarak kendilerine özel bazı yeni property'ler ve olayları da içerebilirler.
  - Örn Calendar için
    - DayHeaderStyle, DayNameFormat, DayStyle, FirstDayOfWeek, NextMonthText, NextPrevFormat, NextPrevStyle, ....
    - DayRender, SelectionChanged, VisibleMonthChanged vb ...



## Web Server Controls:

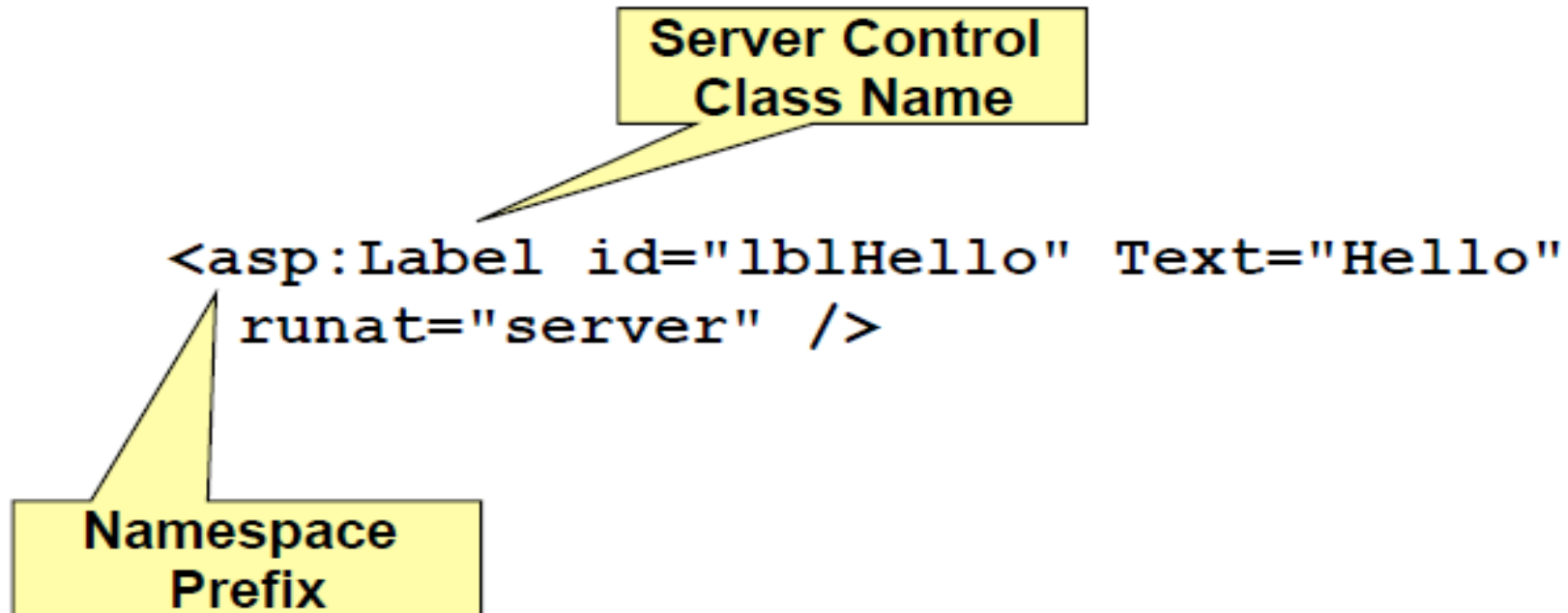
```
<asp:TextBox id="txtName" runat="server" />
```

## HTML Server Controls:

```
<input type="hidden" id="hidVal"  
      name="hidVal" runat="server" />
```

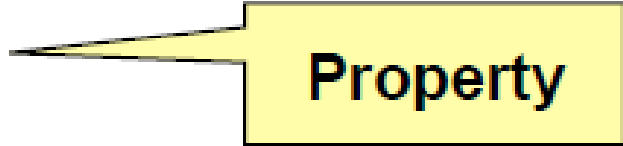
# Tanımlama Şekli

Web Formlarında kullanılan Server kontrollerinin başlangıcında "asp" öneki yer alır.



# Properties

```
<asp:GridView id="gvRecords" runat="server"  
    BorderColor="black"  
    BorderWidth="1"  
    GridLines="Both"  
    CellPadding="3"  
    CellSpacing="0"  
    Font-Name="Verdana"  
    Font-Size="8pt"  
/>
```



**Property**

# Web Server Kontrol Olayları (Events)



```
public void btnSubmit_Click(object sender, EventArgs e)
{
    lblMessage.Text = "You clicked btnSubmit!";
}
```

```
<asp:Button id="btnSubmit" OnClick="btnSubmit_Click"
    runat="server" Text="Submit" />
```

# Kullanıcı Kontrolleri (User Controls)

- Kullanıcı kontrolleri sık kullanılan UI fonksiyonlarını birleştirerek, web sitesi içerisinde yeniden kullanılmasını sağlar.
- Kullanıcı kontrolleri Web Form sayfalarına "register" edilerek kullanılır.
- Kullanım yerleri :
  - Site içerisinde bir çok yerde kullanılan Login sayfaları
  - Web Form'larının header ve footer'ları
  - Tekrar eden menüler

- Kullanıcı kontrolü özel bir direktif olan "Control" direktifi kullanılarak oluşturulur.

`<%@ Control Language="C#" %>`

Oluşturulan kullanıcı kontrollerinin uzantısı ".ascx" dir.

# Onaylama Kontrolleri (Validation Controls)

- Bu controller sayesinde Web formlarına girilen verilerin geçerlilik kontrolü yapıldıktan sonra sunucuya gönderilir.
- Hem istemci hem de sunucu tarafında geçerlilik kontrolü yapılabilir.

Bazı geçerlilik türleri :

- Gerekli Giriş (**Required entry**)
- Özel bir kriterin geçerlik kontrolü  
(**Validating specific criteria**)
- Kontrol değerlerinin karşılaştırılması  
(**Comparing control values**)
- Aralık kontrolü (**Range Checking**)
- Desen eşleşmesi (**Pattern Matching**)
- Özel geçerlilik kontrolü (**Custom Validation**)



Bahsi geçen geçerlilik türlerine ilişkin olarak ASP.NET'de bulunan Validation kontrolcülere :

<asp:RequiredFieldValidator>

<asp:RangeValidator>

<asp:CompareValidator>

<asp:RegularExpressionValidator>

<asp:CustomValidator>

<asp:ValidationSummary>

# Varsayılan Düğme

Sayfa içerisinde birden fazla düğme bulunması durumunda kullanıcının "enter" a basması durumunda hangi düğmenin aktif olacağının belirlenmesi bazı durumlarda önem arz edebilir.

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

---

```
<asp:Panel runat="server" defaultButton="btnOK">
```

```
...
```

```
</asp:Panel>
```

# Odaklanma (Focus)

Sayfa açıldığında hangi kontrolün aktif olacağının çoğu zaman belirlenmesi gerekmektedir.

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

Odaklanma deklaratif olarak belirlenebileceği gibi imperative olarak da belirlenebilir.

- Page.SetFocus(control)
- Page.SetFocus("ClientID")
- txtName.Focus()