

# HELPERS

Dr. Öğr. Üyesi Fatma AKALIN

ASP.net MVC'de ASP.NET web formlarındaki gibi hazır kontroller yoktur. Bunların yerine html kontrolleri kullanılmaktadır. Önceki örnek uygulamalarda html kontrollerini oluşturmustunuz Bu kontrolleri sunucu taraflı kodlarla oluşturma imkanımız vardır.

MVC' de kontrolleri oluşturmaya imkan sağlayan yardımcı metotlara Helper metodları denilmektedir. Helper metodları yazılım geliştiriciye kodlama açısından büyük kolaylık sağlayacaktır.

ASP.NET MVC'de hazır 4 çeşit helper методu vardır.Bunlar;

- 1-HTML Helper
- 2-Web Helper
- 3-Microsoft Web Helper
- 4-Rendering Helpers

# HTML Helpers (HTML Kontrolleri)

ASP.NET MVC View’larında kullanılan HTML yardımcı metotları HTML kontrollerinin elle oluşturulması yerine sunucu taraflı metodlarla hızlı bir şekilde oluşturulmasına imkan sağlar

Dr. Öğr. Üyesi FETTUH ALIN

# LINK OLUŞTURMA(ACTIONLINK)

HTML'de linkleri `<a>....</a>` etiketleriyle oluşturuyoruz. Link oluşturma işlemini ASP.NET MVC helper metodlarından ActionLink ile de gerçekleştirebiliriz.

ActionLink metodu 10 farklı şekilde yazılabilir. En sık kullanılan parametre değerleri:

Dr. Öğr. Üyesi Fatma KALIN

**linkText:** Linkte yazacak yazıdır Örneğin online satış sitesinde «satın al» linkini ActionLink ile tanımlamak isterseniz, linkText özelliğine kullanıcının göreceği yazı olan «satın al» tanımlamasını yapmanız gerekecektir.

**actionName:** Linke tıklandığında hangi action'un çalışacağını belirtir.

**controllerName:** Linke tıklandığında hangi controller'ın çalışacağını belirtir.

**routeValues:** linki ekleyeceğiniz route değerlerini belirtir. Örneğin 2 ID'li ders(Ders/Detay actionuna yönlendiğini düşünelim) için routeValue değerinde ID değerini 2 olarak tanımladığımızda linke tıklandığında ders/Detay/2 sayfasına yönlendirme yapacaktır.

**htmlAttributes:** Linke HTML özelliklerinin tanımlandığı parametredir. Linke CSS sınıfları ya da farklı özellikler tanımlanabileceği gibi JavaScript kütüphaneleri de çağrırlabilir

Şimdi bu özellikleri kullanarak farklı link örnekleri yapalım.

1-Aynı controller içindeki farklı bir actiona yönlenden link (Öğrenci bilgi sisteminde öğrencinin ilgili dönemde aldığı derslerin listesinin bulunduğu sayfanın linki)

```
@Html.ActionLink("Aldığım Dersler", "Dersler")
```

2-Farklı controller içindeki bir actiona yönlenden link (öğrenci bilgi sisteminde öğrencinin öğretim görevlisine mesaj gönderme işlemini yaptığı sayfanın linki)

```
@Html.ActionLink("Öğretim Görevlisine Mesaj Gönder", "OgrencidenOgretimGorevlisine", "Mesaj")
```

3-Id'si 2 olan ürünün (UrunController) detay (Detay) bilgilerine yönlenden link

```
@Html.ActionLink("Ürün Detayı", "Detay", "Urun", new { Id=2})
```

4-Farklı controller içindeki bir actiona yönlenen linkin renginin kırmızı  
ve fontunun kalın olması

```
@Html.ActionLink("Öğretim Görevlisine Mesaj Gönder", "OgrencidenOgretimGorevlisine", "Mesaj", new {@style="color:red; font-weight:bold;"})
```

5-Farklı controller içindeki bir actiona yönlenen linke bir css sınıfı  
tanımlama

```
@Html.ActionLink("Öğrenciye mesaj gönder", "OgretimGorevlisindenOgrenciye", "Mesaj", new {@class="link"})
```

# FORM OLUŞTURMA (BEGINFORM, ENDFORM)

HTML'de formları `<form>...</form>` etiketiyle oluşturuyoruz. Form oluşturma işlemini Asp.Net MVC HTML helper metodlarından BeginForm ile de gerçekleştirebiliriz.

Eğer BeginForm metodunu using ile kullanırsanız tek başına kullanılması yeterli olacaktır. using'sız kullanırsanız yani doğrudan `Html.BeginForm()` çağrımasını yaparsanız formun bitiş noktasını tanımlamak için EndForm metodunun kullanılması gerekmektedir.

BeginForm metodu 13 farklı şekilde yazılabilir. Hiçbir parametre tanımlamadan oluşturulabileceği gibi 12 farklı parametre dizilişiyle de oluşturulabilir. En sık kullanılan parametre değerleri:

**.actionName:** Formda bir işlem yapıldığında hangi action'un çalışacağını belirtir.

**.controllerName:** Formda bir işlem yapıldığında hangi controller'ın çalışacağını belirtir.

**.formMethod:** Formun hangi metotla çalışacağını belirtir. Parametrenin türü FormMethod enum'dır. Değer olarak Get veya Post'u alabilir.

**.routeValues:** Formda bir işlem yapıldığında yönlendirilecek action metoda gönderilecek route değerlerini belirtir.

**.routeValueDictionary:** Birden fazla route değerinin tanımlanması gereği durumlarda kullanılan parametredir.

**.htmlAttributes:** Forma HTML özelliklerinin tanımlandığı parametredir. CSS sınıfları ya da farklı özellikler tanımlanabileceği gibi javascript kütüphaneleri de çağrırlabilir.

Şimdi bu özellikleri kullanarak farklı form örnekleri yapalım..

1-Parametresiz ve using'li bir form tanımlaması yapma

```
@using (Html.BeginForm()) {}
```

2-Parametresiz ve usingsiz bir form tanımlaması yapma

```
@{Html.BeginForm();}
```

Deneme Formu

```
@{Html.EndForm();}
```

3-Form submit edildiğinde(buttona tıklanıp sunucuya bilgilerini gönderdiğinde) formdaki veriler, UrunController sınıfının Kaydet action'una gönderme

```
@using (Html.BeginForm("Kaydet", "Urun")){}
```

4-Form submit edildiğinde verileri UrunController sınıfının Kaydet action'una post metodu ile gönderme

```
@using (Html.BeginForm("Kaydet", "Urun", FormMethod.Post)) { }
```

5- Form submit edildiğinde formdaki veriler, UrunController sınıfının Kaydet Action'ına post metodu ile ve ID'si 2 olacak şekilde gönderme

```
@using (Html.BeginForm("Kaydet", "Urun", new { Id=2 }, FormMethod.Post)) { }
```

6-Formun kenarlarını 2 px ve siyah renkte yapma

```
@using (Html.BeginForm("Kaydet", "Urun", FormMethod.Post, new {@style="border-color:black; border-width:2px; border-style:solid;"})) { }
```

**<b>Form Yapısı</b>**

# YAZI GÖRÜNTÜLEME (LABEL)

HTML'de kullanıcının bilgi girişi yaptığı metin kutusu vb bir elemana bağlı bir başlığı label etiketiyle oluşturabilirsiniz. Bu başlığı görüntüleme işlemi için `<label>..</label>` etiketlerini kullanabilirsiniz. Label oluşturma işlemi Asp.Net MVC HTML metodlarından Label ile gerçekleştirilebilirsiniz.

Label çoğunlukla form içinde kullanılır Örneğin ürün kayıt sayfasında ürünün adı, fiyatı, stok miktarı, kategori ve benzeri bilgilerinin girişi için birer Metin kutusu ya da seçim kutusu bulunur. Her giriş kutusunun yanında hangi bilginin girileceği yazılıdır ürün adının yazıldığı Metin kutusunun yanına ürünün adı fiyatının yazıldığı giriş kutusunun yanına fiyatı yazılı bu yazılar için label kontrolleri kullanılır

Label metodu 6 farklı şekilde yazılabılır en sık kullanılan parametre değerleri:

**expression:** label kontrolünün hangi kontrole bağlı olduğu bilgisi tutulur. Proje çalıştırılıp sayfanın kaynak kodlarına bakıldığından bu parametreye tanımlanan değer Label kontrolünün for özelliğini atanır. for özelliğinde, label kontrolünün bağlı olduğu giriş kontrolünün id bilgisi tanımlanır.

**labelText:** Label kontrolünde görüntülenecek yazıyı tanımlar.

**htmlAttributes:** Label kontrolüne HTML özelliklerinin tanımlandığı parametredir CSS sınıfları ya da farklı özellikler tanımlanabileceği gibi kütüphaneleri de çağrırlabilir.

Şimdi bu özelliklerini kullanarak farklı label örnekleri yapalım.

1-Sadece expression tanımlı bir label oluşturma

```
@Html.Label("KullaniciAdi")
```

2-Label kontrolünde görünecek yazıyı tanımlama

```
@Html.Label("KullaniciAdi", "Kullanıcı Adı")
```

3-Label kontrolünün yazı rengini kırmızı ve fontunu eğik yapma

```
@Html.Label("KullaniciAdi", "Kullanıcı Adı", new  
{@style="color:red, font-style:italic"})
```

Bu kodlamayı yaptığınızda sayfada kırmızı ve kalın fontta «Kullanıcı adı» yazacaktır.

# METİN GİRİŞ KUTUSU (TEXTBOX)

HTML'de kullanıcının yazıyla bilgi girişi yaptığı kontrollere metin kutusu denir. Metin kutularını input etiketi ile oluşturabilirsiniz. Bu kontrolü görüntüleme işlemi için `<input type="text">` etiketini, inputun tipine text olarak ayarlayıp kullanabilirsiniz. Metin kutusu oluşturma işlemini ASP.NET MVC HTML Helper metodlarından TextBox ile gerçekleştirebilirsiniz.

Metin giriş kutusu, kullanıcının yazı ile veri girişini yapmasını sağlayan kontroldür. Örneğin Öğrenci Bilgi sisteminde öğrenci bilgilerini girerken bir kısım bilgileri kullanıcı eli ile girer, bir kısmını da sayfadaki listelerden seçer. Öğrencinin adı, soyadı, T.C kimlik numarası gibi elle gelen verileri kullanıcı TextBox kontrolünden girer.

TextBox metodu 7 farklı şekilde yazılabilir. En sık kullanılan parametre değerleri:

**.name:** TextBox kontrolünün ismini (name özelliğini) belirten parametredir.

**.value:** Textbox kontrolüne atanan değeri belirtir, object türündedir. Atanan değer Textbox kontrolünün içinde yazan yazıdır

**.htmlAttributes:** Textbox kontrolüne HTML özelliklerinin tanımlandığı parametredir. CSS sınıfları ya da farklı özellikler tanımlanabileceği gibi JavaScript kütüphaneleri de çağrırlabilir

**.format:** Textbox kontrolünün içinde yazan yazının hangi formatta görüneceğini tanımlayan parametredir.

Şimdi bu özellikleri kullanarak farklı TextBox örnekleri yapalım.

1-Sadece name tanımlı bir TextBox oluşturma

```
@Html.TextBox("UrunAdi")
```

2-TextBox'a bir değer atama

```
@Html.TextBox("UrunAdi", "Bilgisayar")
```

3-TextBox kontrolünün yazı rengini kırmızı ve fontunu kalın yapma

```
@Html.TextBox("UrunAdi", "Bilgisayar", new { @style = "color:red; font-weight:bold" });
```

4-TextBox kontrolüne girilen sayısal değeri parasal formatta gösterme(örneğin 1499.482->1.499,48)

```
@Html.TextBox("Fiyati", 1499.482, format: "{0:C2}")
```

# ÇOK SATIRLI METİN GİRİŞ KUTUSU (TEXTAREA)

HTML'de kullanıcının yazı ile birden fazla satırda bilgi girişi yaptığı kontrollere çok satırlı metin kutusu ya da TextArea denir. Çok satırlı metin kutularını <textarea .... /> etiketi ile oluşturabilirsiniz. Çok satırlı Metin kutusu oluşturma işlemini ASP.NET MVC helper metodlarından TextArea ile gerçekleştirebilirsiniz. TextArea, kullanıcının çok satırlı ve (istenirse) farklı fontlarda, farklı renklerde yazı ile veri girişi yapmasını sağlayan kontroltür.

Dr. Öğr.

Örneğin online satış sitesinde ürün açıklaması içinde başlıklar, yazılarından farklı renkte ve fonta olabilir, sayfa içinde ürün özelliklerini listelemek için maddelerleme kullanılabilir. Bu işlemler TextBox ile gerçekleştirilemez. TextArea kontrolü ise bu sorunlara çözüm olarak kullanılabilir.

TextArea metodu 8 farklı şekilde yazılabilen bir metottur. En sık kullanılan parametre değerleri:

Dr. Öğr.

**.name:** TextArea kontrolünün ismini belirten parametredir.

**.value:** TextArea kontrolüne atanan değeri belirtir, string türündedir. Atanan değer, TextArea kontrolünün içinde yazan yazıdır.

**.htmlAttributes:** TextArea kontrolüne HTML özelliklerinin tanımlandığı parametrede CSS sınıfları ya da farklı özellikler tanımlanabileceği gibi javascript kütüphaneleri de çağrırlabilir.

**.column:** TextArea kontrolünün genişliğinin tanımlandığı parametredir kontrolün bir satırında parametrenin değerinde aldığı sayı kadar harf gösterir.

**.row:** TextArea kontrolünün yüksekliğinin tanımlandığı parametredi. Kontrolün kaç satır yazı göstereceğini tanımlar

1-Sadece name özelliği tanımlı bir TextArea oluşturma

```
@Html.TextArea("Aciklama")
```

2-TextArea'nın yazı tipini değiştirme

```
@Html.TextArea("Aciklama", new { @style = "color:red; font-style:italic;" })
```

3-TextArea'ya bir değer atama

```
@Html.TextArea("Aciklama", "Ürün Açıklaması", new {@style = "color:red; font-style:italic;" })
```

4-TextArea kontrolünü 5 satır ve 50 sütundan oluşacak şekilde ayarlayalım

```
@Html.TextArea("Aciklama", "Ürün açıklaması buraya yazılacak", 5, 50, new { @style = "color:red; font-style:italic;" })
```

# ONAY KUTUSU (CHECKBOX)

HTML'de kullanıcının alt alta ya da yan yana listelenmiş olan seçeneklerden bir veya birden fazlasını seçmesine olanak sağlayan kontrole onay kutusu ya da CheckBox denir. Bu kontrolü görüntüleme işlemi için `<input type= "checkbox"/>` etiketini kullanabilirsiniz. Onay kutusu oluşturma işlemini ASP.NET MVC HTML helper metodlarından CheckBox ile gerçekleştirebilirisiniz.

Dr. Öğr. ÜYESİ  
ALİ İLKALIN

CheckBox kullanıcıya birden fazla seçim yapma imkanı sunar. Örneğin Microsoft sertifika sınavlarında bir sorunun birden fazla doğru yanıtı vardır. Tüm cevapların yanına onay kutusu koyulup doğru cevabı işaretlerken kutuya tik atma işlemi(seçili hale getirme) yapılabilir. Farklı bir örnek anket sistemi olabilir. Anket sisteminde soruların bir kısmının birden fazla cevabı olabilir. Bu tür çoktan seçmeli ve bu seçimlerden birkaçının seçilmesi istenen durumlarda CheckBox kontrolünün kullanılması kolaylık sağlayacaktır.

CheckBox metodu 6 farklı şekilde yazılabilir. En sık kullanılan parametre değerleri:

Dr. Öğr. Üyesi  
Mehmet Yıldız

**name:** Checkbox kontrolünün ismini(name özelliğini) belirten parametredir.

**isChecked:** Checkbox kontrolünün sayfa açıldığında seçili olup olmayacağı tanımlar. Boolean türünden değer alır. Eğer değer true ise sayfa açıldığında kontrol seçili olacaktır, false ise seçili olmayacağı.

**htmlAttributes:** Checkbox kontrolünün HTML özelliklerinin tanımlandığı parametredir. CSS sınıfları ya da farklı özellikler tanımlanabileceği gibi Javascript kütüphaneleri de çağrılabılır.

Şimdi bu özellikleri kullanarak farklı CheckBox örnekleri yapalım.

1-Sadece name özelliği tanımlı bir CheckBox oluşturma

```
@Html.CheckBox("cevapA") A
```

2-CheckBox'ı seçili hale getirme

```
@Html.CheckBox("cevapB", true) B
```

# SEÇİM BUTONU (RADIOBUTTON)

HTML'de kullanıcının alt alta ya da yan yana listelenmiş olan seçeneklerden sadece birini seçmesine olanak sağlayan kontrole seçim butonu ya da RadioButton denir. Bu kontrolü görüntüleme işlemi için `<input type=radio .../>` etiketine kullanabilirsiniz. Seçim butonu oluşturma işlemini ASP.NET MVC HTML helper metodlarından RadioButton ile gerçekleştirebilirsiniz. RadioButton, kullanıcıya birden fazla seçenek arasından seçim yapma imkanı sunar.

Dr. Öğr.

Örneğin online bir sınavda her sorunun birden fazla yanıtı vardır. Soruların tek doğru cevap olması istenmektedir. Bu durumda kullanıcı yanıt şıklarından sadece birini seçebilmelidir. Farklı bir seçeneği tıkladığında eski seçimi kaldırılmalı ve yine tek seçim yapılmış olmalıdır. Bu tür çoktan seçmeli ve bu seçimlerden sadece bir tanesini seçilmesi istenen durumlarda RadioButton kontrolünün kullanılması kolaylık sağlayacaktır.

RadioButton metodu 6 farklı şekilde yazılabilir en sık kullanılan parametre değerleri:

Dr. Öğr.

**name:** RadioButton kontrolünün ismini belirten parametredir RadioButton kontrollerinde name, özelliği aynı zamanda gruplama işleminde kullanılır. Örneğin bir sayfada 5 tane soru var ve her sorunun 4 şıkkı varsa, sorunun şıklarının kendi arasında seçilebilirlik kontrollerinin olması gereklidir. Bu sebeple 1 sorunun şıklarının name özelliklerine cevap1, 2. soruya cevap 2 şeklinde grup isimleri verip soruların cevap seçimlerinin birbirlerine karışmalarını engelleyebilirsiniz.

**value:** RadioButton kontrolünü atanan değeri belirtir, string türündedir.

**isChecked:** RadioButton kontrolünün sayfa açıldığında seçili olup olmayacağı tanımlar. Boolean türünden değer alır. Eğer değer true ise sayfa açıldığında kontrol seçili olacaktır, false ise seçili olmayacağı.

**htmlAttributes:** RadioButton kontrolüne HTML özelliklerinin tanımlandığı parametredir. CSS sınıfları ya da farklı özellikler tanımlanabileceği gibi JavaScript kütüphaneleri de çağrılabılır.

Şimdi bu özelliklerini kullanarak farklı RadioButton örnekleri yapalım.

1-RadioButton kontrolü tanımlama

```
@Html.RadioButton("cevapA", "A") A
```

2-RadioButton'u seçili hale getirme

```
@Html.RadioButton("cevapB", "B", true) B
```

2 tane soru ve her sorunun 4 yanıtı olsun. İlk sorunun yanıtları «A1,B1,C1,D1», ikinci sorunun yanıtları ise «A2,B2,C2,D2» olsun. İlk sorunun cevapları arasında yapılan seçimler ikinci soruyu, ikinci sorunun cevapları arasında yapılan seçimler ise birinci soruyu etkilemeyecek (kendi içlerinde grüplanacak) şekilde soru-cevap sistemini tanımlama

Soru 1: .....

<br />

@Html.RadioButton("soru1", "A1") A1 @\*soru1 1. grubu ifade ediyor\*@

@Html.RadioButton("soru1", "B1") B1

@Html.RadioButton("soru1", "C1") C1

@Html.RadioButton("soru1", "D1") D1

<br />

<br />

Soru 2: .....

<br />

@Html.RadioButton("soru2", "A2") A2 @\*soru2 2. grubu ifade ediyor\*@

@Html.RadioButton("soru2", "B2") B2

@Html.RadioButton("soru2", "C2") C2

@Html.RadioButton("soru2", "D2") D2

<br />

<br />

# AÇILAN LİSTE KUTUSU (DROPDOWNLIST)

HTML'de kullanıcının bir listede bulunan seçeneklerden sadece birini seçmesine olanak sağlayan kontrole açılan liste kutusu ya da DropDownList denir. Bu kontrolü görüntüleme işlemi için `<select>...</select>` etiketlerini kullanmanız gereklidir. Listeye eleman eklerken select etiketleri arasında her eleman için bir tane option etiketi eklenmelidir. Açılan liste kutusu oluşturma işlemini ASP.NET MVC HTML helper metodlarından DropDownList ile gerçekleştirebilirsiniz

Dr. Öğr.

RadioButton, kullanıcıya birden fazla seçenek arasından seçim yapma imkanı sunar. Örneğin bir kariyer sitesinde üye kayıt işlemi sırasında kullanıcının cinsiyet bilgisi istenir. Bu bilgiyi kullanıcının açılır liste kutusundan kadın ya da erkek olacak şekilde seçim yapması istenebilir. Benzer şekilde askerlik durumu bilgisi de açılır liste kutusundan muaf, tamamlandı, yapmadı ve tecilli olacak şekilde seçilebilir. Bu tür çoktan seçmeli ve bu seçimlerden sadece de bir tanesinin seçilmesi istenen durumlarda DropDownList kontrolünün kullanılması kolaylık sağlayacaktır. DropDownList kontrolünü, içinde çok veri olan seçimlerde kullanmak mantıklıdır. Örneğin kariyer sitesinde işveren meslek bilgisini seçerken kullanıcıya onlarca mesleği RadioButton ile göstermek tüm sayfayı mesleklerle doldurmak anlamına gelir ve mantıklı bir işlem olmaz. Bunun yerine DropDownList'e meslekleri yerleştirip kullanıcının tek satırda bulunan bu kontrolden istediği mesleği seçmesini sağlayabiliriz.

DropDownList metodu 8 farklı şekilde yazılabilir. En sık kullanılan parametre değerleri:

**name:** Kontrolün ismini (name özelliğini) belirten parametredir

**selectList:** DropDownList kontrolündeki liste elemanlarının tanımlandığı parametredir. Değerine Ienumerable<SelectListItem> türünde bir liste alır.

**optionLabel:** DropDownList kontrolünde hiçbir seçim yapılmadığı durumda kontrolde görünecek yazıdır Örneğin bu yazı boş olabileceği gibi seçim yapınız şeklinde bir yazı da olabilir.

**htmlAttributes:** DropDownList kontrolüne HTML özelliklerinin tanımlandığı parametredir. CSS sınıfları ya da farklı özellikler tanımlanabileceği gibi javascript kütüphaneleri de çağrırlabilir.

Şimdi bu özellikleri kullanılarak farklı DropDownList örnekleri yapalım.

## 1-Cinsiyet seçimi yapılan bir DropDownList kontrolü tanımlama

```
@Html.DropDownList("cinsiyet",
    new []
    {
        new SelectListItem
        {
            Text="Kadın",
            Value="1"
        },
        new SelectListItem
        {
            Text="Erkek",
            Value="2"
        }
    }
)
```

2-Cinsiyet seçimi yapılan bir DropDownList kontrolünde sayfa açıldığında seçili olan elemanı, seçimlerin dışında bir eleman(Seçim yapınız) yapma

```
@Html.DropDownList("meslek",
    new []
    {
        new SelectListItem
        {
            Text="Kadın",
            Value="1"
        },
        new SelectListItem
        {
            Text="Erkek",
            Value="2"
        }
    }, "Seçim yapınız")
```

# LİSTE KUTUSU (LISTBOX)

HTML'de kullanıcının bir listede bulunan seçeneklerden bir ya da birden fazlasını seçmesine olanak sağlayan kontrole liste kutusu ya da listbox denir. Bu kontrolü görüntüleme işlemi için `<select size="4">...</select>` etiketlerini kullanmanız gereklidir. Select kontrolünün size özelliğine kaç değeri verirseniz o kadar satırlık bir liste kutusu oluşturursunuz. Listeye eleman eklemek için Select etiketleri arasında her eleman için bir tane option etiketi kullanmanız gereklidir. Liste kutusu oluşturma işlemini ASP.NET MVC HTML helper metodlarından ListBox ile gerçekleştirebilirsiniz

Dr. Oğuzhan AKALIN

Listbox kullanıcıya birden fazla seçenek arasından bir ya da birden çok seçim yapma imkanı sunar. En sık kullanılan örneği sahibinden.com sitesinde bulunmaktadır. Örneğin site içinde ev ararken sol tarafta yer alan filtre bölümünde evin oda sayısı banyo sayısı binanın yaşı gibi kriterleri liste kutusundan seçebilirsiniz. birden fazla seçenekte arama yapmak isteyebileceğinizi düşünülerek çoklu seçimde izin verilmiştir.

ListBox metodu 4 farklı şekilde yazılabilir. En sık kullanılan parametre değerleri:

Dr. Öğr. Üyesi  
ALİ İLKALIN

**name:** Listbox kontrolünün ismini (name özelliğini) belirten parametredir

**selectList:** ListBox kontrolündeki liste elemanlarının tanımlandığı parametredir. Değerine `IEnumerable<SelectListItem>` türünde bir liste alır.

**htmlAttributes:** ListBox kontrolüne HTML özelliklerinin tanımlandığı parametredir. CSS sınıfları ya da farklı özellikler tanımlanabileceği gibi javascript kütüphaneleri de çağrırlabilir.

Şimdi bu özellikleri kullanarak farklı listbox örnekleri yapalım.

Dr. Öğr. Üyesi KÜLTÜRMA AKALIN

```
@Html.ListBox("odasayıları",
    new[]
    {
        new SelectListItem
        {
            Text="1+1",
            Value="1"
        },
        new SelectListItem
        {
            Text="2+1",
            Value="2"
        },
        new SelectListItem
        {
            Text="3+1",
            Value="3"
        },
        new SelectListItem
        {
            Text="4+1",
            Value="2"
        }
    })
}
```

# ŞİFRE GİRİŞ KUTUSU (PASSWORD)

HTML'de kullanıcının şifre ya da gizli bilgilerinin girişini yaptığı kontrollere şifre giriş kutusu ya da Password denir. Şifre giriş kutularını input etiketi ile oluşturabilirsiniz. Bu kontrolü görüntüleme işlemi için `<input type="password" ...>` etiketini inputun tipine password (şifre) olarak ayarlayıp kullanabilirsiniz. Şifre kutusu oluşturma işlemini ASP.NET MVC Html helper metodlarından password ile gerçekleştirebilirsiniz.

Dr. Öğr.

Örneğin online satış sitelerinde kullanıcı girişleri sırasında kullanıcı adı ve şifresi istenmektedir. Kullanıcı adını TextBox kontrolüne yazarken şifre bilgisini yazdığınız süre içinde metin kutusunda sadece noktalar görülmektedir. Benzer işlem şifre değiştirme için de geçerlidir. Eski şifreyi, yeni şifreyi ve yeni şifreyi tekrar yazarken yazılan şifre Metin kutusunda görülmemektedir. Sadece yazdığımız karakter sayısı kadar nokta simgeleri görülmektedir.

Dr. Öğr. Üyesi FETHİYE YILMAZ

Password metodu 4 farklı şekilde yazılabilir. En sık kullanılan parametre değerleri:

**name:** Password kontrolünün ismini (name özelliğini) belirten parametredir.

**value:** Password kontrolünü atanan değeri belirtir, object türündedir. Atanan değer Password kontrolünün içinde yazan yazıdır.

**htmlAttributes:** Password kontrolüne HTML özelliklerinin tanımlandığı parametredir. CSS sınıfları ya da farklı özellikler tanımlanabileceği gibi javascript kütüphaneleri de çağrırlabilir.

Şimdi bu özellikleri kullanarak farklı password örnekleri yapalım

1-Sadece name tanımlı bir Password oluşturma

```
@Html.Password("sifre")
```

2-Password kontrolüne bir değer atama

```
@Html.Password("sifre",123)
```

# GİZLİ VERİ KONTROLÜ (HIDDEN)

HTML'de kullanıcıların göremeyeceği ama sayfa içinde kullanılabilen bilgilerin yer aldığı kontrollere gizli veri kontrol üyeleri ya da Hidden Control denir. Gizli veri kontrollerini input etiketi ile oluşturabilirsiniz. Bu kontrolü sayfaya ekleme işlemi için `<input type= "hidden" />` etiketini, inputun tipini (type) hidden olarak ayarlayıp kullanabilirsiniz. Gizli veri kontrolü kutusu oluşturma işlemini ASP.NET MVC HTML helper metodlarından Hidden ile gerçekleştirebilirsınız.

Dr. Öğr.

Gizli veri kontrolünü, kullanıcılar sayfanın html (kaynak) kodlarını görüntüleme işlemini yapmadan göremezler. Bu kontrolün en sık kullanıldığı örnek silme işlemleridir. Örneğin, online satış sitesinin yönetim panelinde aynı anda birden fazla ürünü silmek istediğiniz zaman, ürün listesinde ürün bilgilerinin yanına onay kutusu koyup seçilen ürünleri sayfanın sonunda bulunan sil butonuna tıklayarak silebilirsiniz. Ürün listesine tüm ürünlerin ID bilgilerini gizli olarak yazarsanız ürünleri sistemden silmeniz kolaylaşacaktır. Bu gibi durumlarda Hidden kontrolü kullanılmalıdır.

Hidden metodu 4 farklı şekilde yazılabılır. En sık kullanılan parametre değerleri:

Dr.

**name:** kontrolünün ismini (name özelliğini) belirten parametredir.

**value:** Hidden kontrolüne atanan değeri belirtir. Object türündedir. Atanan değer, hidden kontrolünün içinde yazan yazıdır.

**htmlAttributes:** Hidden kontrolüne HTML özelliklerinin tanımlandığı parametredir.

Şimdi bu özellikleri kullanarak farklı Hidden örnekleri yapalım.

Dr. Öğr.

1-Sadece name tanımlı bir Hidden oluşturma

```
@Html.Hidden("Id")
```

2-Hidden'a bir değer atama

```
@Html.Hidden("Id",5)
```

## STRONGLY TYPED HELPERS

Strongly Typed Helper metodlarını öğrenmeden önce Strongly Typed'ın ne olduğunu öğrenmemiz gerek. Strongly Typed, türü belirli olan anlamına gelir. Yani bir değeri tanımlarken hangi türden olduğunu da doğru belirtmek (tanımlamak) gereklidir. Örneğin bir tam sayıyı String olarak tanımlayıp sayı gibi işleme alırsanız bu weakly typed (zayıf tipli) bir tanımlama olur. Çünkü değer tam sayı olmasına ve kod içinde tam sayı gibi kullanılmasına rağmen daha geniş kapsamlı bir türde (string) şeklinde tanımlanmıştır.

Dr. Oğuzhan KALIN

Strongly typed tanımlamanın avantajı, değişkenlere sadece türlerine uygun veriler gönderilebilir. Weakly typeed bir değişken tanımlamasında ise örneğin tam sayı olması gereken değişkene Metin verisi gönderilip hataya sebep olunabilir ve bu durum uygulamada açık oluşmasına sebebiyet verebilir.

ASP.NET MVC'de Strongly Typed HTML helper metotları bulunmaktadır. Örneğin insan kaynakları projesinde personel modeline kaydedilecek personel bilgileri girilirken strongly type helper metotları kullanılırsa uygulama, modelin özelliklerine uygun veri girişine izin verecektir. Modelde personelin ad ve soyad bilgilerini String olarak tanımlarken yaşı belgesini integer tanımlayıp veri giriş kontrollerini Strongle Type helper metotları ile oluşturursanız yaşı bilgisinin geldiği kontrolde sadece integer veri girilmesine izin verilecektir. Strongly typed yardımcı metodlar, lambda expression ile tanımlanır.

Bir proje üzerinden Strongly Typed Helper Metotlarını irdeleyeceğiz.

Strongly Typed Helper Metotları:

1-LABELFOR

2-TEXTBOXFOR

3-TEXTAREAFOR

4-CHECKBOXFOR

5-RADIOBUTTONFOR

6-DROPODOWNLISTFOR

7-PASSWORDFOR

8-VALIDATIONMESSAGEFOR

Dr.

ogr. Uyesi Fatma AKALIN

# ÖRNEK PROJE –ADIM ADIM (Personele ilişkin bilgilerin girileceği bir form yapısı)

1- Helper isimli bir bos fakat MVC seçeneğinin seçili olduğu bir ASP.NET MVC projesi oluşturalım. (Layoutsuz)

2- Models klasörünün altında 3 ayrı sınıf oluşturalım. Bu sınıflar;

Cinsiyet.cs

MedeniDurum.cs

Personel.cs

Şimdi bu sınıflara sırasıyla özellikleri tanımlayacağız.

# Personel.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Helper.Models
{
    public class Personel
    {
        [Key]
        public int Id { get; set; }

        [Required]
        [DisplayName("Adı")]
        public string Ad { get; set; }

        [Required]
        [DisplayName("Soyadı")]
        public string Soyad { get; set; }

        [Required]
        [DisplayName("Doğum Tarihi")]
        public DateTime DogumTarih { get; set; }

        [Required]
        [DisplayName("TC Kimlik NO")]
        public string TCKimlikNo { get; set; }

        [Required]
        [DisplayName("Çocuğu Var MI")]
        public bool CocukVarMi { get; set; }

        [Required]
        [DisplayName("Sakatlık Durumu")]
        public int SakatlikDurumu { get; set; }

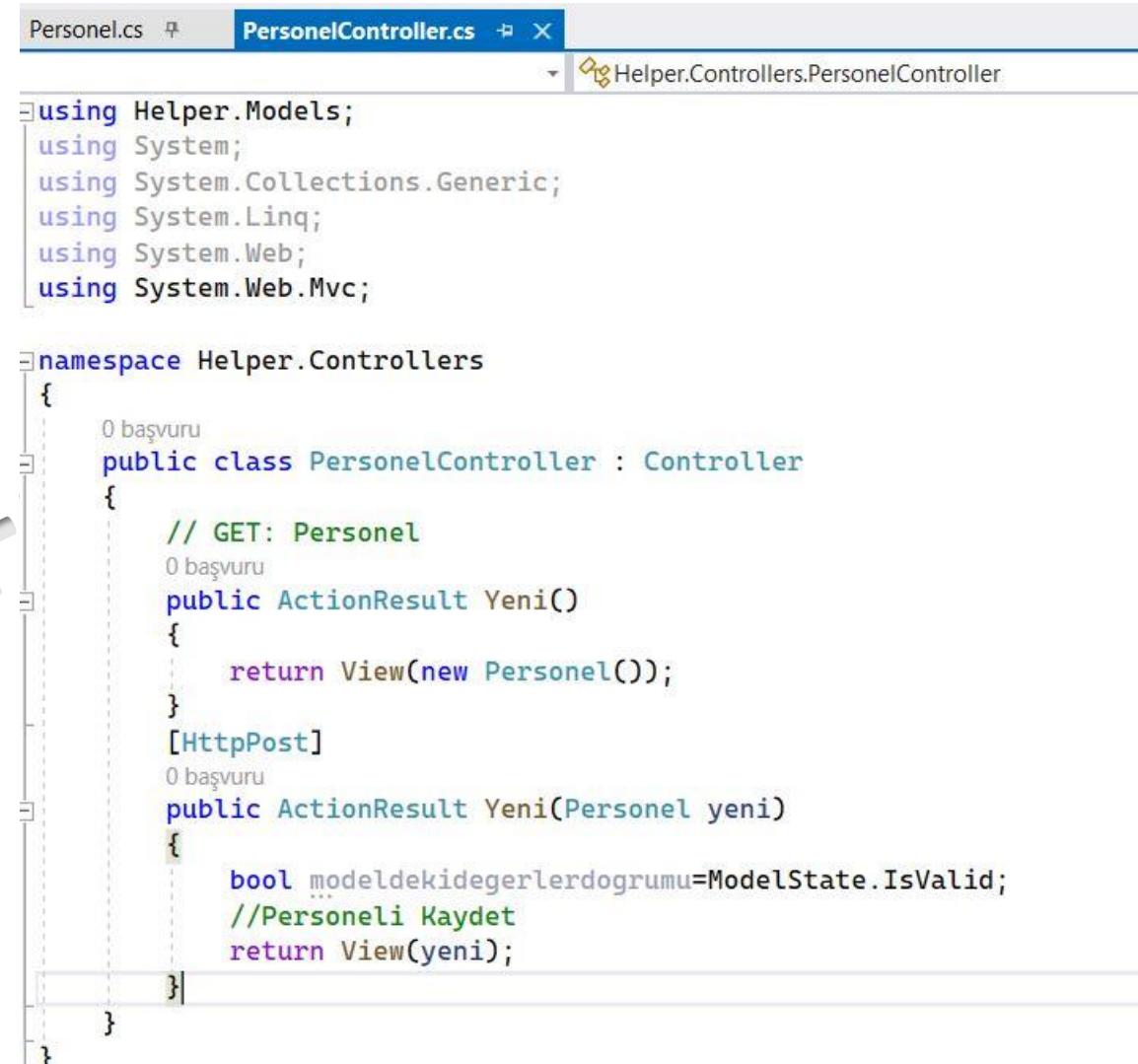
        [Required]
        [DisplayName("Şifre")]
        public string Sifre { get; set; }

        [DisplayName("Diğer Bilgiler")]
        public string DigerBilgiler { get; set; }
    }
}
```

### 3-Controller sınıfını oluşturalım.

Personel Controller sınıfında personel bilgilerinin kaydedileceği Yeni isminde ActionResult türünde veri döndüren bir action oluşturup action'un geri dönüş değerine parametre olarak boş bir Personel nesnesi gönderiniz.

Kayıt işlemlerini gerçekleştirmek için Personel parametrelıHttpPost attribute'sine sahip bir action metot daha oluşturup doğruluk kontrollerinin sağlanması için View'dan gelen verilerin doğru olup olmadığı IsValid özelliği ile kontrol edilecektir.



```
Personel.cs PersonelController.cs
PersonelController.cs
using Helper.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Helper.Controllers
{
    public class PersonelController : Controller
    {
        // GET: Personel
        public ActionResult Yeni()
        {
            return View(new Personel());
        }

        [HttpPost]
        public ActionResult Yeni(Personel yeni)
        {
            bool modeldekidegerlerdogrumu=ModelState.IsValid;
            //Personeli Kaydet
            return View(yeni);
        }
    }
}
```

Son aşamada Controller kapsamında oluşturduğumuz Yeni actionu için bir View ekleyelim ve form kontrollerimizi yapalım.

```
@model Helper.Models.Personel
```

Tanımını yapmayı kesinlikle unutmayalım..

Dr

Yeni.cshtml Personel.cs PersonnelController.cs

```
1 @model Helper.Models.Personel
2 @{
3     Layout = null;
4 }
5
6 <!DOCTYPE html>
7
8 <html>
9 <head>
10    <meta name="viewport" content="width=device-width" />
11    <title>Yeni</title>
12    <link href="~/Content/bootstrap.min.css" rel="stylesheet" />
13
14    <script src="~/Scripts/jquery-3.7.1.min.js"></script>
15    <script src="~/Scripts/bootstrap.min.js"></script>
16
17
18 </head>
19 <body style="margin:10px">
20    <h2>Yeni</h2>
21    @using (Html.BeginForm())
22    {
23        <div class="form-horizontal">
24            <div class="form-group">
25                <label class="control-label"> BASLIK</label>
26                <div>
27                    <input type="text" />
28                    Validation Mesaji
29                </div>
30            </div>
31        </div>
32    }
```

AKALIN

DV

Yeni.cshtml Personel.cs PersonelController.cs

```
33     <br />
34     @using (Html.BeginForm())
35 {
36         <div class="form-horizontal">
37             <div class="form-group">
38                 @Html.LabelFor(model => model.Ad, new { @class = "control-label col-md-2" })
39                 <div class="col-md-10">
40                     @Html.TextBoxFor(model=>model.Ad)
41                     @Html.ValidationMessageFor(model=>model.Ad)
42                 </div>
43
44             </div>
45             <br />
46             <div class="form-group">
47                 @Html.LabelFor(model => model.Soyad, new { @class = "control-label col-md-2" })
48                 <div class="col-md-10">
49                     @Html.TextBoxFor(model => model.Soyad)
50                     @Html.ValidationMessageFor(model=>model.Soyad)
51                 </div>
52             </div>
53             <br />
54
55             <div class="form-group">
56                 @Html.LabelFor(model => model.DogumTarih, new { @class = "control-label col-md-2" })
57                 <div class="col-md-10">
58                     @Html.TextBoxFor(model => model.DogumTarih)
59                     @Html.ValidationMessageFor(model => model.DogumTarih)
60                 </div>
61             </div>
62             <br />
```

ALIN

```
Yeni.cshtml  Personel.cs  PersonelController.cs
62      <br />
63      <div class="form-group">
64          @Html.LabelFor(model => model.TCKimlikNo, new { @class = "control-label col-md-2" })
65          <div class="col-md-10">
66              @Html.TextBoxFor(model => model.TCKimlikNo)
67              @Html.ValidationMessageFor(model => model.TCKimlikNo)
68          </div>
69      </div>
70      <br />
71
72      <div class="form-group">
73          @Html.LabelFor(model => model.CocukVarMi, new { @class = "control-label col-md-2" })
74          <div class="col-md-10">@Html.CheckBoxFor(model => model.CocukVarMi)</div>
75      </div>
76      <br />
77      <div class="form-group">
78          @Html.LabelFor(model => model.SakatlikDurumu, new { @class = "control-label col-md-2" })
79          <div class="col-md-10">
80              @Html.RadioButtonFor(model => model.SakatlikDurumu, 0)Sakatlığı Yok
81              @Html.RadioButtonFor(model => model.SakatlikDurumu, 1)1. Derece
82              @Html.RadioButtonFor(model => model.SakatlikDurumu, 2)2. Derece
83              @Html.RadioButtonFor(model => model.SakatlikDurumu, 3)3. Derece
84          </div>
85      </div>
86      <br />
```

LIN

Dr.

```
Yeni.cshtml  Personel.cs  PersonnelController.cs
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
```

```
<br />

<div class="form-group">
    @Html.LabelFor(model => model.Sifre, new { @class = "control-label col-md-2" })
    <div class="col-md-10">@Html.PasswordFor(model => model.Sifre)</div>
</div>
<br />
<div class="form-group">
    @Html.LabelFor(model => model.DigerBilgiler, new { @class = "control-label col-md-2" })
    <div class="col-md-10">@Html.TextAreaFor(model => model.DigerBilgiler)</div>
    <br />
    <div class="col-md-10">@Html.TextAreaFor(model => model.DigerBilgiler, new { @cols=30, @rows=5 })</div>
</div>
<br />
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" value="Kaydet" class="btn btn-default" />
    </div>
</div>
}

</body>
</html>
```

Dr. Ç

## Yeni

BASLIK

Validation Mesaji

Adı

Soyadi

Dogum tarihi

1.01.0001 00:00:00

TC Kimlik NO

Çocuğu Var MI

Sakatlık Durumu

- Sakatlığı Yok  1. Derece  2. Derece  3. Derece

Sifre

Diger Bilgiler

Kaydet

## Yeni

BASLIK

Validation Mesaji

Adı

Adı alanı gereklidir.

Soyadi

Soyadı alanı gereklidir.

Dogum tarihi

1.01.0001 00:00:00

TC Kimlik NO

TC Kimlik NO alanı gereklidir.



Çocuğu Var MI

Sakatlık Durumu

- Sakatlığı Yok  1. Derece  2. Derece  3. Derece

Sifre

Diger Bilgiler

  
,  
,

Kaydet

ALIN

Şimdiye kadar derslerimizin temel mantığını kavramak için görselliğe hiç önem vermedik. Şimdi Bootstrap yapısını kabataslak anlatacağım fakat yine görselliğe(frontend tarafına) çok degeinmeyeceğim(1 dönem anlatılan dersi 1-2 saatte sıkıştırmam mümkün değil).

Ancak ilerleyen haftalarda zamanımız kalırsa frontend kapsamında(html-css-javascript-flexbox-bootstrap) projeler yaparak projelerinizin görsel açıdan da mükemmelliği yakalaması için destek olmak hedefindeyim

Dr. Öğr. Üyesi  
ALİ İLKALIN

# Bootstrap Framework

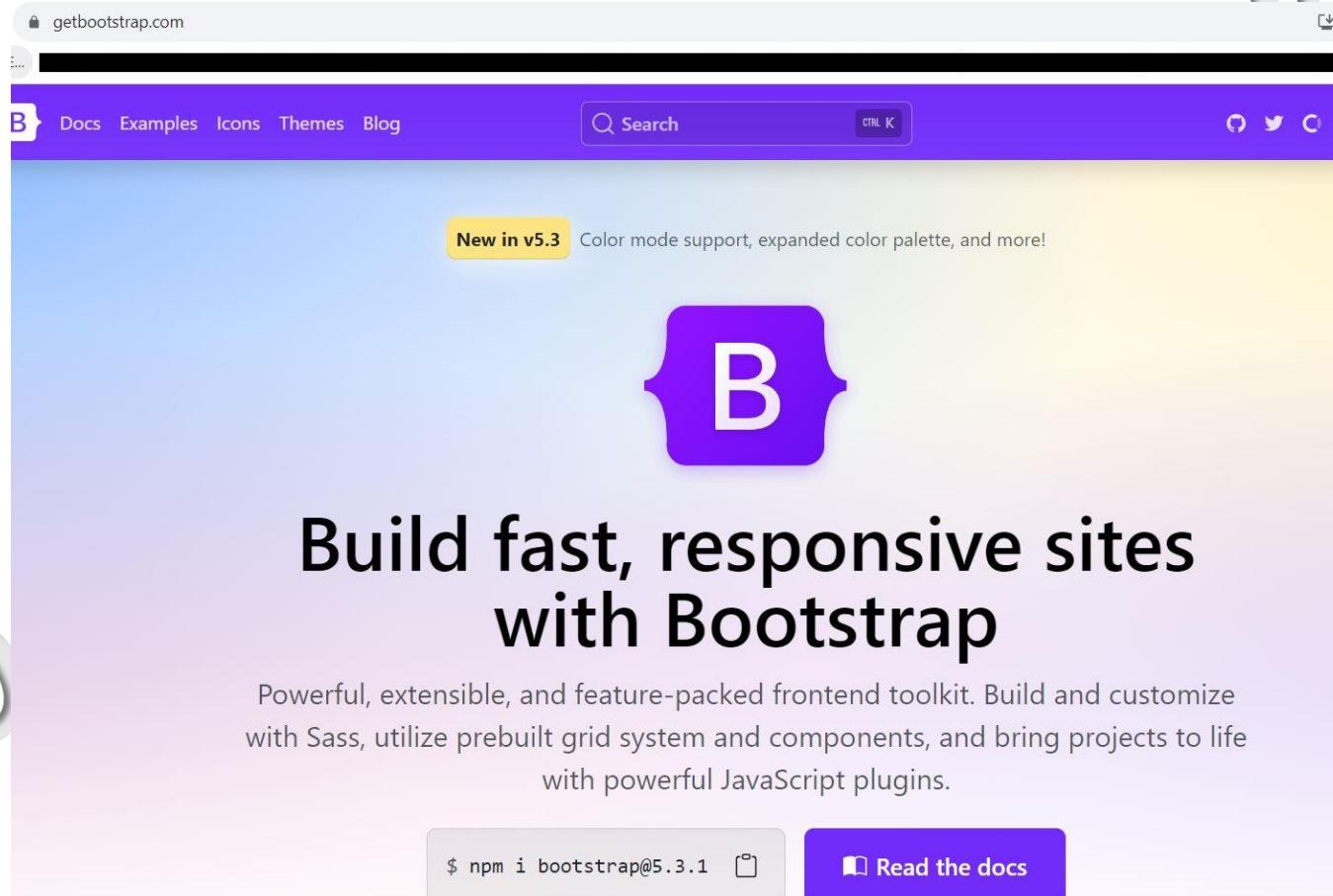
## Bootstrap Yapısı ve Grid Sistem

Bootstrap bir css frameworktür. Kullanımı kolay ve hızlıdır. Aynı zamanda sürekli geliştirilmekte ve ücretsiz olarak hizmete sunulmaktadır.

Bootstrap hazır tasarım ve yapılar içermektedir. Böylece yazdığınız uygulamaların responsive(web sayfasını küçültüğünüzde ya da büyütüğünzde ekran çözünürlük ayarlaması) olarak bozulmadan görülmesi kolaylaşmaktadır. Tüm cihazlar içerisinde okunabilir bir görünüm sağlama kullanışlılık özelliğini optimize etmiştir.

Dr. Şimdi bootstrap css frameworkten faydalanalım..

# <https://getbootstrap.com/>



Bootstrap'in sitesinde her türlü bilginin mevcut olması ile birlikte Web Teknolojileri dersinde ayrıntılı olarak anlatım sağlandı. Dersler boyunca karşılaşıldıkça hatırlatma yapılacak. Bu doğrultuda ilk hatırlatmamız grid sistem üzerinde yoğunlaşacak.

The screenshot shows the Bootstrap website's navigation bar at the top, followed by a sidebar with categories like Sass, Options, Color, etc., and a main content area. The main content area features a large heading 'Grid system' and a descriptive paragraph about its features. Below this is an advertisement for Adobe Stock. The sidebar also includes a section for 'Content' with 'Reboot' listed under it.

**Grid system**

Use our powerful mobile-first flexbox grid to build layouts of all shapes and sizes thanks to a twelve column system, six default responsive tiers, Sass variables and mixins, and dozens of predefined classes.

Get 10 Free Images From Adobe Stock.  
Start Now.

ads via Carbon

## Example

Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. It's built with [flexbox](#) and is fully responsive. Below is an example and an in-depth explanation for how the grid system comes together.

### On this page

#### Example

- [How it works](#)
- [Grid options](#)
- [Auto-layout columns](#)
- [Equal-width](#)
- [Setting one column width](#)
- [Variable width content](#)
- [Responsive classes](#)
- [All breakpoints](#)
- [Stacked to horizontal](#)
- [Mix and match](#)
- [Row columns](#)
- [Nesting](#)
- [CSS](#)
- [Sass variables](#)
- [Sass mixins](#)
- [Example usage](#)
- [Customizing the grid](#)
- [Columns and gutters](#)
- [Grid tiers](#)

Grid sistem şöyle bir mekanizma getiriyor...  
768 px'den küçük cihazlarda (telefon kategorisi), tabler desktop ve large cihazla için css tasarlamak istersen ilgili sütunlardaki bilgileri kullanmalısın.

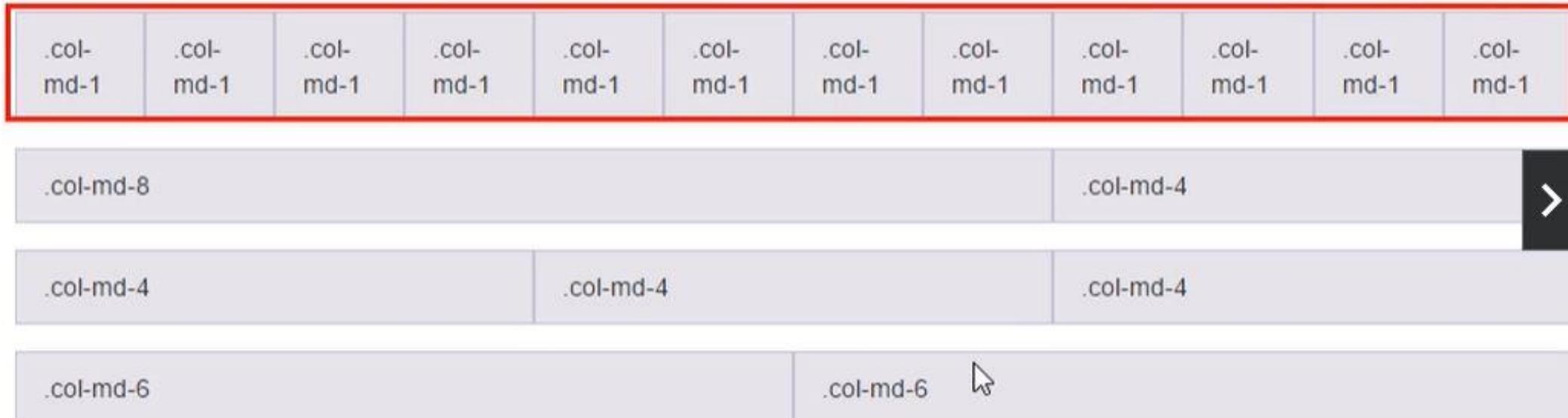
Dr. Ogr. i

## Grid options

See how aspects of the Bootstrap grid system work across multiple devices with a handy table.

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-
# of columns	12			
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)			
Nestable	Yes			
Offsets	Yes			
Column ordering	Yes			

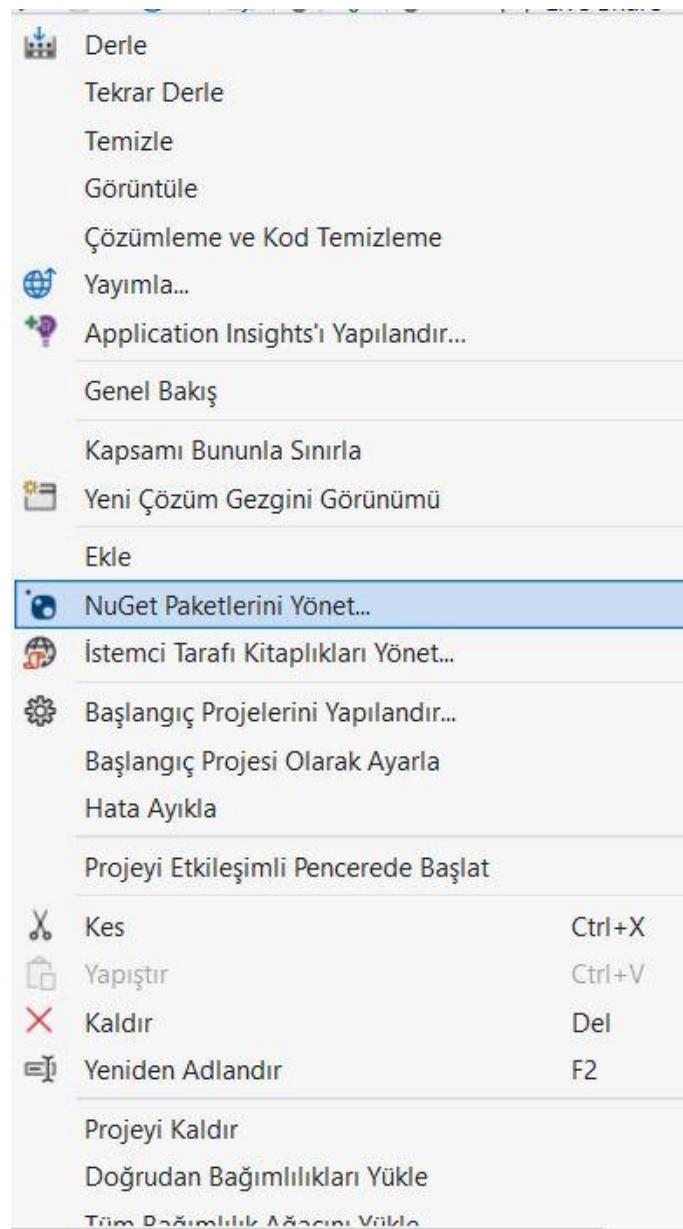
Bootstrap grid sistemi, yapıyı 12 eşit parçaya bölüyor ve bir satırı 12 birim olarak kabul ediyor. Siz bu 12 birimi istediğiniz kombinasyon ile bölebiliyorsunuz. Ek olarak örneğin 8'e 4 olarak böldüğünüz sekizlik sütunun içerisine de 12'lik başka sütun koyabilirsin.



# !!! UYARI !!!

ALIN

- PROJELYE BAŞLAMADAN ÖNCE BOOTSTRAP DE DAHİL OLMAK ÜZERE  
ÖNEMLİ FRAMEWORKLERİ YÜKLEYELİM...



The NuGet Package Manager interface shows the search results for "bootstrap". The results pane displays the following information:

Sıra	İsim	Yazar	Sürüm	Detaylar
1	bootstrap	The Bootstrap Team	5.3.1	Bootstrap framework in CSS. Includes JavaScript.

Below the search results, there are two pinned packages displayed:

- Modernizr** Faruk Ateş, P 2.8.3 (Download button)
- jQuery** jQuery Foundation, Inc. tarafından, 162M indirme 3.7.0 (Download button)
- jQuery.Validation** Jörn Zaefferer tarafından, 82,4M indirme 1.19.5 (Download button)

Yüklediğimiz frameworklerin sayfalarımıza eklediğimzde aktif olarak kullanma imkanımız oluşacaktır..

```
<link href("~/Content/bootstrap.min.css" rel="stylesheet" />
<link href("~/Content/bootstrap-theme.min.css" rel="stylesheet"
/>
<script src "~/Scripts/jquery-3.7.0.min.js"></script>
<script src "~/Scripts/bootstrap.min.js"></script>
```

Görüldüğü üzere min yani sıkıştırılmış versiyonlar eklenmiştir. Bu işlem sayfanın daha hızlı açılmasını sağlayacaktır.

YA DA

- `<meta charset="utf-8">`
- `<meta name="viewport" content="width=device-width, initial-scale=1">`
- `<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">`
- `<link href="https://fonts.googleapis.com/css?family=Montserrat" rel="stylesheet">`
- `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>`
- `<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>`

Kodlarını kullanabiliriz. Fakat bu kod satırları bootstrap gibi çeşitli teknolojilerin sağladığı işlevselliklerin son kullanıcıya gösterilmesi için internet bağlantısına ihtiyaç duymaktadır.

# Ders kapsamında gerçeklediğimiz projede responsive yapısı sunan bir çıktı ürettik.

AKALIN

## Anasayfa

### **Baslik1**

Bootstrap bir css frameworktür. Kullanımı kolay ve hızlıdır. Aynı zamanda sürekli geliştirilmekte ve ücretsiz olarak hizmete sunulmaktadır. Bootstrap hazır tasarım ve yapılar içermektedir. Böylece yazdığınız uygulamaların responsive(web sayfasını küçültüğünüzde ya da büyütüğünzde ekran çözünürlük ayarlaması) olarak bozulmadan görülmesi kolaylaşmaktadır. Tüm cihazlar içerisinde okunabilir bir görünüm sağlamaası kullanışlılık özelliğini optimize etmiştir.

### **Baslik2**

Bootstrap bir css frameworktür. Kullanımı kolay ve hızlıdır. Aynı zamanda sürekli geliştirilmekte ve ücretsiz olarak hizmete sunulmaktadır. Bootstrap hazır tasarım ve yapılar içermektedir. Böylece yazdığınız uygulamaların responsive(web sayfasını küçültüğünüzde ya da büyütüğünzde ekran çözünürlük ayarlaması) olarak bozulmadan görülmesi kolaylaşmaktadır. Tüm cihazlar içerisinde okunabilir bir görünüm sağlamaası kullanışlılık özelliğini optimize etmiştir.

### **Baslik3**

Bootstrap bir css frameworktür. Kullanımı kolay ve hızlıdır. Aynı zamanda sürekli geliştirilmekte ve ücretsiz olarak hizmete sunulmaktadır. Bootstrap hazır tasarım ve yapılar içermektedir. Böylece yazdığınız uygulamaların responsive(web sayfasını küçültüğünüzde ya da büyütüğünzde ekran çözünürlük ayarlaması) olarak bozulmadan görülmesi kolaylaşmaktadır. Tüm cihazlar içerisinde okunabilir bir görünüm sağlamaası kullanışlılık özelliğini optimize etmiştir.

## Anasayfa

### **Baslik1**

Bootstrap bir css frameworktür. Kullanımı kolay ve hızlıdır. Aynı zamanda sürekli geliştirilmekte ve ücretsiz olarak hizmete sunulmaktadır. Bootstrap hazır tasarım ve yapılar içermektedir. Böylece yazdığınız uygulamaların responsive(web sayfasını küçültüğünüzde ya da büyütüğünzde ekran çözünürlük ayarlaması) olarak bozulmadan görülmesi kolaylaşmaktadır. Tüm cihazlar içerisinde okunabilir bir görünüm sağlamaası kullanışlılık özelliğini optimize etmiştir.

### **Baslik2**

Bootstrap bir css frameworktür. Kullanımı kolay ve hızlıdır. Aynı zamanda sürekli geliştirilmekte ve ücretsiz olarak hizmete sunulmaktadır. Bootstrap hazır tasarım ve yapılar içermektedir. Böylece yazdığınız uygulamaların responsive(web sayfasını küçültüğünüzde ya da büyütüğünzde ekran çözünürlük ayarlaması) olarak bozulmadan görülmesi kolaylaşmaktadır. Tüm cihazlar içerisinde okunabilir bir görünüm sağlamaası kullanışlılık özelliğini optimize etmiştir.

### **Baslik3**

Bootstrap bir css frameworktür. Kullanımı kolay ve hızlıdır. Aynı zamanda sürekli geliştirilmekte ve ücretsiz olarak hizmete sunulmaktadır. Bootstrap hazır tasarım ve yapılar içermektedir. Böylece yazdığınız uygulamaların responsive(web sayfasını küçültüğünüzde ya da büyütüğünzde ekran çözünürlük ayarlaması) olarak bozulmadan görülmesi kolaylaşmaktadır. Tüm cihazlar içerisinde okunabilir bir görünüm sağlamaası kullanışlılık özelliğini optimize etmiştir.

Dr. Öğr.

# CUSTOM HELPER METODU OLUŞTURMA

ASP.NET MVC projenizde kendi oluşturacağınız yardımcı metotları da kullanabilirsiniz.

Ders akışında gerçekleşen proje üzerinden custom helper metodlarının çeşitli örnekler vasıtasıyla nasıl kullanıldığını öğreneceğiz..

# KAYNAKÇA

<https://learn.microsoft.com/tr-tr/aspnet/mvc/>

Veysel Uğur Kızmaz, ASP.NET MVC5, Kodlab Yayınları

<https://github.com/muratbaseren/udemy-yazilimcilarin-yukselisi-aspnet-my-evernote-sample>