



CSS'in Yapısı

Bu dersimizde CSS'in yapısını, terimlerini ve kavramlarını öğreneceğiz.

```
4 Seçici (selector)
5 body {
6     background-color: skyblue;
7 }
8 scope
```

Özellik (property) Değer (value)

Büyütmek için resme tıklayınız.

CSS'de *seçici (selector)* değiştirmek istediğimiz HTML etiketini belirtir. CSS ile üzerinde oynama yapmak istediğimiz HTML etiketini *seçmiş* oluruz. { ve } işaretlerinin arasında kalan alana *scope* [*skap* okunur] ya da *bildirim alanı* denir. Bu alana CSS'imizin özelliğini ve değerini yazarız.

Özellikten sonra mutlaka iki nokta üstüste (:) işareti gelir ve özelliğin değeri yazıldıktan sonra noktalı virgül (;) işareti ile kapatılır. Yazım kuralı daima böyledir.

Örneğin; yukarıdaki örneğimizde <body> etiketimizi seçtik ve arkaplan rengini (background-color) gök mavisi (skyblue) olarak belirledik.

```
4
5 body {
6     background-color: skyblue;
7     font-family: Verdana;
8 }
9
```

Büyütmek için resme tıklayınız.

İlk örneğimize ek olarak *body* seçicimize *font-family* adına yeni bir özellik ekledik ve değerini *Verdana* olarak belirledik. Bu şekilde <body> etiketi içine yazılan tüm metinler Verdana fontu olarak gösterilecektir.

CSS'in yapısı bu kadar basittir. Bir sonraki dersimizde temel CSS özelliklerini öğreneceğiz.

[Diğer derse geçmek için tıklayınız. >>>](#)

Çevrimiçi reklamcılıkla ilgileniyor musunuz?



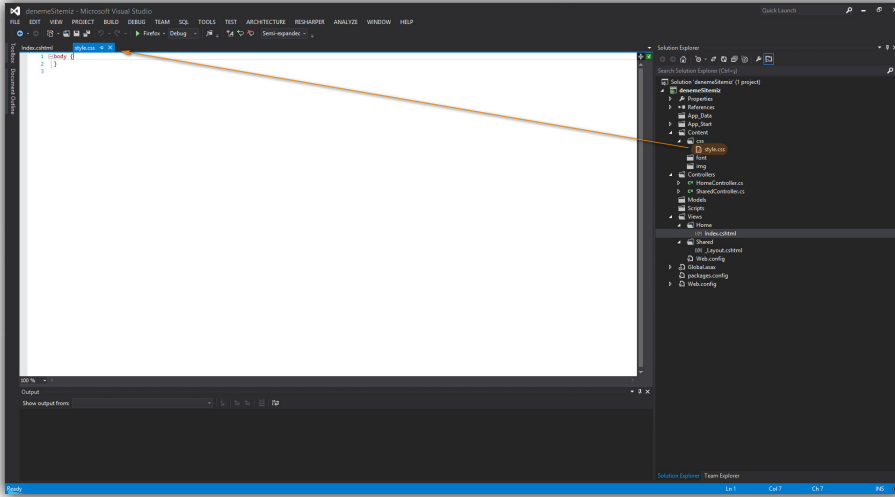
Temel CSS Özellikleri

Bu dersimizde temel birkaç CSS özelliği öğreneceğiz.

background-color

background-color dilimizde *arkaplan rengi* anlamına gelmektedir. İlgili seçicinin arkaplan rengini değiştirmek için kullanılır. Varsayılan değeri *transparent* yani *renksizdir*.

Şimdi bir önceki dersimizde yaptığımız *denemeSitemiz* adlı projeyi açalım ve Home klasörünün altındaki *Index.cshtml* sayfasına gelelim. Projeyi açabilmek için masaüstünüzdeki *denemeSitemiz* adlı klasöre girmeniz ve *denemeSitemiz.sln* adlı dosyaya çift tıklamanız yeterlidir.

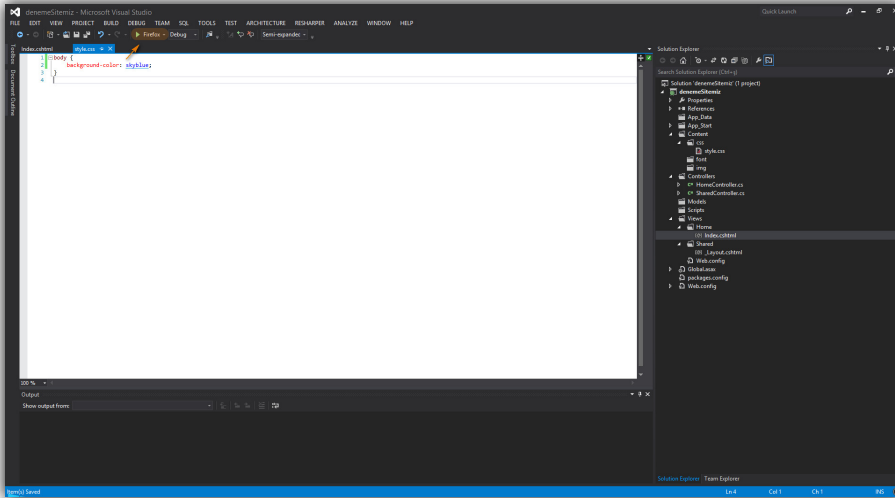


Büyütmek için resme tıklayınız.

Index.cshtml sayfasını açtıktan sonra *style.css* dosyasını açın.

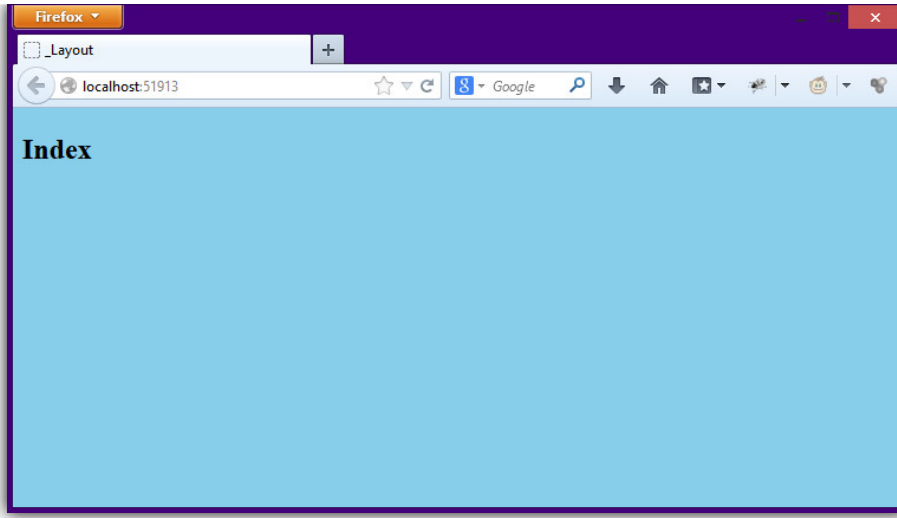
```
1 | body {  
2 |     background-color: skyblue;  
3 | }
```

Yukarıdaki kodları *body* bildirim alanına yazın.



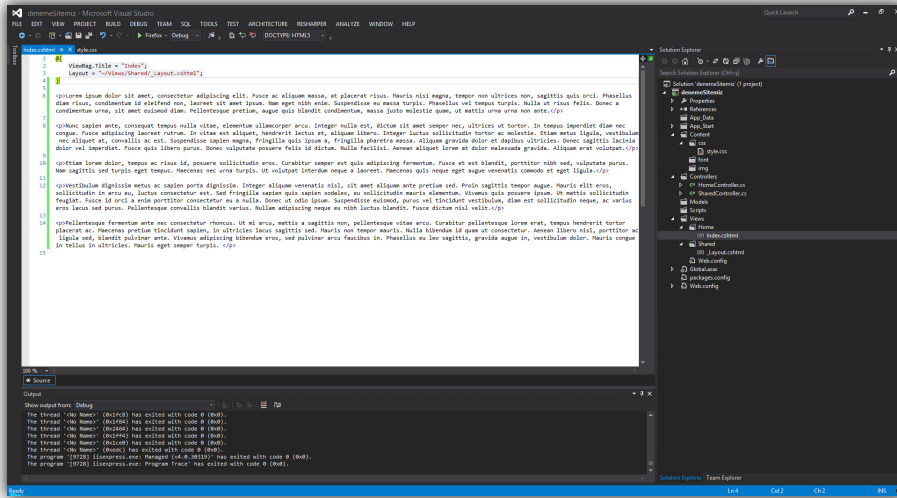
Büyütmek için resme tıklayınız.

Kodları yazarken Visual Studio'nun *intellisense* özelliğinden faydalanabilir ve daha hızlı yazabilirsiniz. Kodları yazdıktan sonra klavyenizden F5 tuşuna basın ya da görseldeki alandaki tarayıcı ismine tıklayın.



Büyütmek için resme tıklayınız.

Gördüğünüz gibi arkaplanın gök mavisi renginde olduğunu göreceksiniz. Şimdi lipsum.org sitesine girin ve 5 adet paragraf kopyalayın.



Büyütmek için resme tıklayınız.

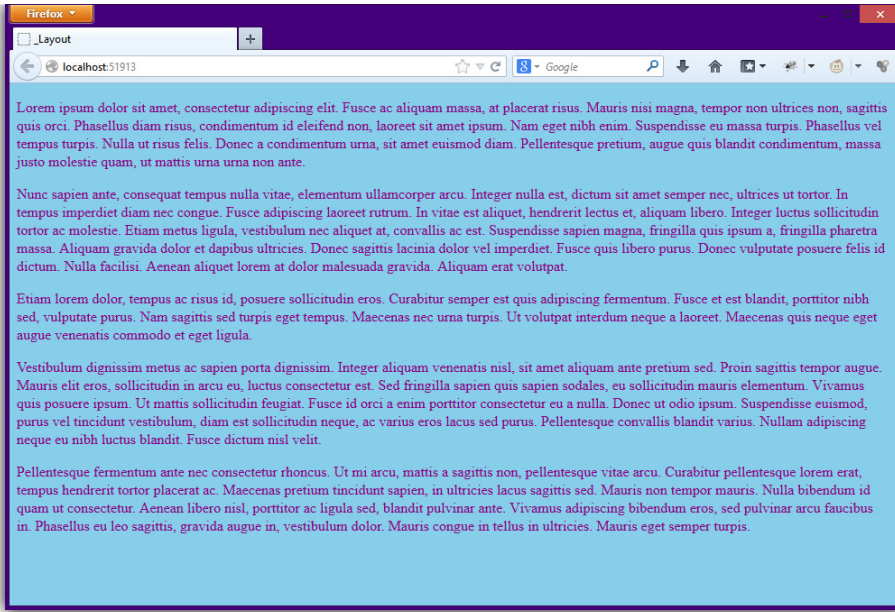
Paragraflarınızı şekilde görüldüğü gibi index.cshtml sayfasının içine yapıştırın. Başlarına ve sonlarına paragraf açma ve kapatma etiketlerini yazmayı unutmayın. Şimdi tekrar style.css dosyasına dönün.

color

color özelliği metin renklerini değiştirmek için kullanılır. Varsayılan olarak *black* yani siyahtır.

```
1 body {
2   background-color: skyblue;
3   color: darkmagenta;
4 }
```

Yukarıdaki yeni özelliğimizi ve değerini body seçicisine ekleyin ve tekrar F5'e basın.



Büyütmek için resme tıklayınız.

Yazı renginin *koyu magenta* olduğunu göreceksiniz. Şimdi tekrar style.css dosyasına dönelim ve aşağıdaki özellikleri ekleyelim.

font-family, font-size ve font-weight

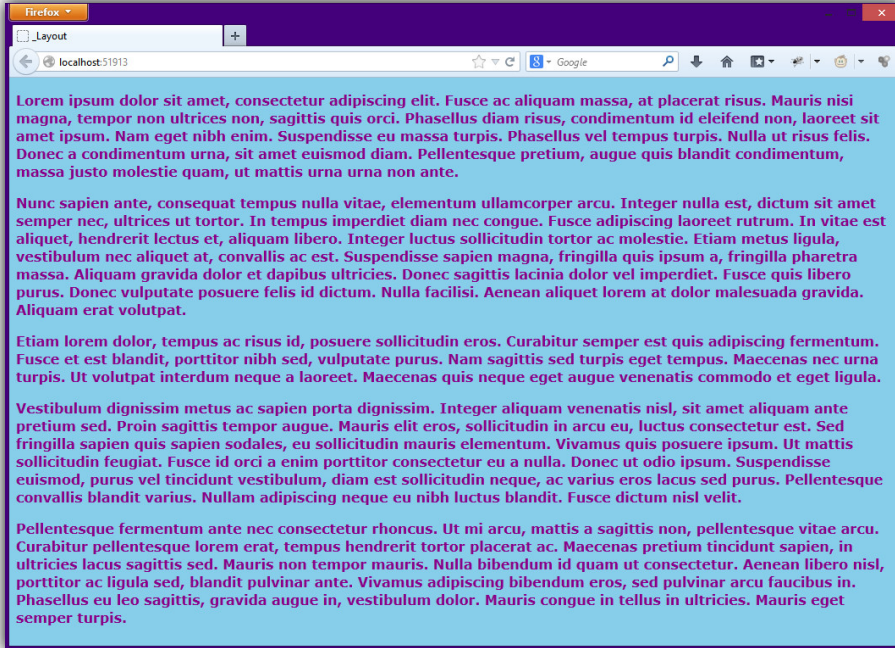
```

1 body {
2   background-color: skyblue;
3   color: darkmagenta;
4   font-family: Verdana;
5   font-size: 16px;
6   font-weight: bold;
7 }

```

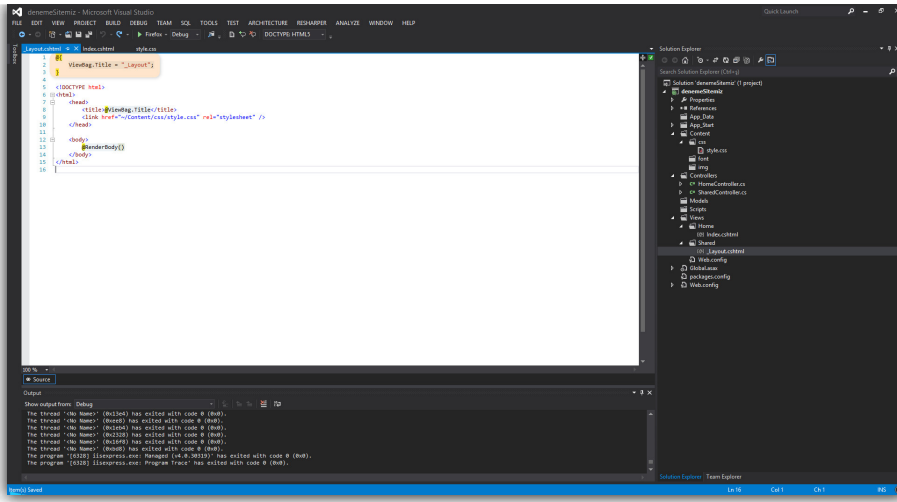
font-family dilimizde *font ailesi* anlamındadır. Kullanıcıya göstermek için kullanacağımız font'ları buraya yazarız. Varsayılan olarak tarayıcı tarafından belirlenir. *font-size*, *font büyüklüğü* anlamındadır ve fontların küçük ya da büyük olmasını belirler. *font-weight* ise *font ağırlığı* anlamındadır ve kullanılan fontların kalın ya da normal yazılmasını belirler. *bold*, *kalın* anlamındadır.

Şimdi tekrar F5'e basalım ve sonu görelim.



Büyütmek için resme tıklayınız.

Görüldüğü gibi font ailemiz Verdana oldu. Font büyüklüğümüzü 16px olarak belirledik ve font ağırlığını kalın olarak seçtik.



Büyütmek için resme tıklayınız.

Gözüme bir şey çarptı. Bizim <title> etiketimiz sürekli layout'taki ile aynı. _Layout.cshtml dosyasına dönün ve görselde işaretlenen alanı silin. Böylece artık yeni title'ımız Home klasörünün altındaki Index.cshtml sayfasına göre belirlenecektir.

Şimdi tekrar style.css dosyamıza dönebiliriz.

width ve height

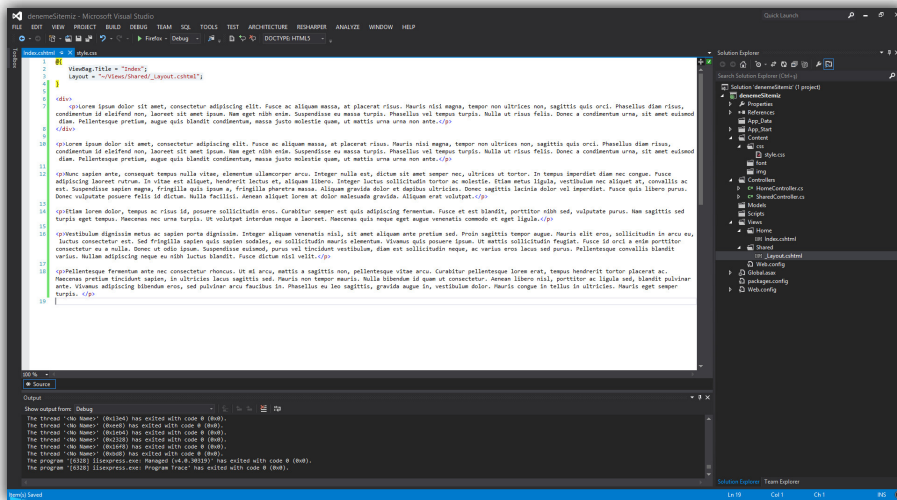
width genişlik, height ise yükseklik anlamındadır. Varsayılanları auto'dur.

```

1 body {
2   background-color: skyblue;
3   color: darkmagenta;
4   font-family: Verdana;
5   font-size: 16px;
6   font-weight: bold;
7 }
8
9 div {
10  width: 600px;
11  height: 300px;
12  background-color: goldenrod;
13  color: darkslategray;
14  font-size: 18px;
15 }

```

div adında yeni bir seçici açın ve özelliklerini yukarıdaki gibi belirleyin ve daha sonra Index.cshtml dosyasına dönün.



Büyütmek için resme tıklayınız.

Bu dosyanın içine <div> etiketi açın ve içine daha önce kullandığınız ilk paragrafı yapıştırın. Eğer hizalamada sorun yaşarsanız CTRL + A'ya yaptıktan sonra CTRL + K + D yapın.



Büyütmek için resme tıklayınız.

Görüldüğü gibi <div> etiketi sayfamıza eklendi. Arkaplan rengini *goldenrod*, metin rengini *darkslategray*, genişliğini 600px ve yüksekliğini 300px olarak belirledik.

Bu dersin videosunu 1080p HD kalitesinde izleyebilirsiniz.

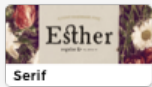
CSS3 - Temel CSS Özellikleri



Bu dersimizde bu kadar. Bir sonraki dersimizde sayfada CSS kullanma yöntemlerini inceleyeceğiz.

Diğer derse geçmek için tıklayınız. >>>

2018's Best Fonts



2019 | Creative Commons | zinzinzibidi.com

Bu site en iyi Google Chrome web tarayıcısında görüntülenmektedir.



CSS Kullanma Yöntemleri

Bir önceki derslerimizde oluşturduğumuz *oldumuz denemeSitemiz.sln* dosyasını çalıştırın ve projeyi açın.

<link> yöntemi ile CSS kullanma

Aslına bakarsanız bu yöntemi çoktan yaptık ve kullanıyoruz.



Büyütmek için resme tıklayınız.

Görselde de gözüktüğü gibi sayfamızın <head> etiketi içindeki alana <link> yöntemi ile CSS dosyamızı ekledik.

<style> yöntemi ile CSS kullanma

Şimdi <style> etiketi içinde CSS kullanmayı göreceğiz. Bunun ilk önce <link> yöntemi ile eklediğimiz CSS dosyasını siliyoruz ve <style> adında yeni bir etiket açıyoruz. <style> etiketi daima <head> etiketinin içine yazılır. <style> etiketinin içine style.css dosyasında yazdığımız kodları kopyalıyoruz. Şu şekilde bir kod yapısı elde etmiş olmalıyız.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>@ViewBag.Title</title>
5   <style>
6
7   body {
8     background-color: skyblue;
9     color: darkmagenta;
10    font-family: Verdana;
11    font-size: 16px;
12    font-weight: bold;
13  }
14  div {
15    width: 600px;
16    height: 300px;
17    background-color: goldenrod;
18    color: darkslategray;
19    font-size: 18px;
20  }
21 </style>
22 </head>
23 <body>
24   @RenderBody()
25 </body>
26 </html>
```

Bu kod yapısını aşağıdaki görselden de görebilirsiniz:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>ViewBag.Title</title>
5   <style>
6     body {
7       background-color: skyblue;
8       color: darkslategray;
9       font-family: Verdana;
10      font-size: 16px;
11      font-weight: bold;
12    }
13
14    div {
15      width: 600px;
16      height: 100px;
17      background-color: gold;
18      color: darkslategray;
19      font-size: 18px;
20    }
21  </style>
22 </head>
23 <body>
24   @RenderBody()
25 </body>
26 </html>
27

```

Büyütmek için resme tıklayınız.

İşte <style> etiketi ile bu şekilde CSS kodları yazabiliriz. Fakat ileride yazacağımız kodlar o kadar çok olacaktır bu yöntem tavsiye edilmez. Sadece belirli sayfalarda yapacağımız küçük düzeltmeler için bu yöntemi kullanacağız.

@import yöntemi ile CSS kullanma

Bu yöntem en az kullanılan yöntemdir. Daima CSS dosyalarının içinde kullanılır. Genellikle normalize.css ya da reset.css dosyaları için kullanılır.

```
1 | @import url('~/Areas/web_tasarim/Content/css/normalize.css');
```

Eğer sayfamızda bir normalize.css dosyası kullanmış olsaydık style.css dosyamızın içine bu kodları yazacaktır. normalize.css ne işe yarar soruna şimdilik takılmayın. İleride hepsini anlatacağım.

inline yöntemi ile CSS kullanma

Son olarak *inline* yani *satır içi* ile CSS kullanmayı göstereceğim. Bunun için ilk önce Index.cshtml dosyanıza dönün ve <div> etiketinin içine aşağıdaki kodları yazın.

```

1 <div style="color: white; font-size: 10px;">
2   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce ac aliquam massa,
3     at placerat risus. Mauris nisi magna, tempor non ultrices non, sagittis quis orci.
4   </p>
5 </div>

```

Görselde ise şu şekilde görebilirsiniz:

Büyütmek için resme tıklayınız.

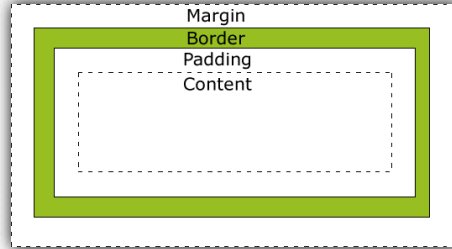
<div> etiketinin içine *style* adlı yeni bir atribut açıyoruz ve bu atributun içine CSS kodlarımızı yazıyoruz. Bu yöntem *inline kod yazma tekniği* denir. Küçük düzenlemeler yapmak için idealdir.

Bu dersin videosunu 1080p HD kalitesinde izleyebilirsiniz.



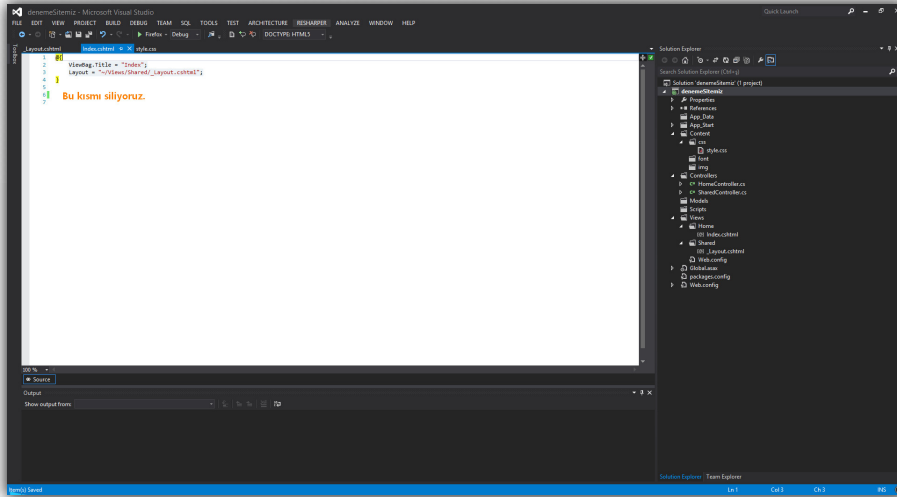
CSS'te Padding ve Margin

padding CSS'te hücre içi boşluk, *margin* ise hücre dışı boşluk anlamındadır. *border* ise sınır anlamındadır. Metinleri ve diğer içerikleri hizalamak için sıklıkla *padding*'i, *div*'leri sayfaya ve birbirleri arasında hizalamak içinse *margin*'i kullanacağız.



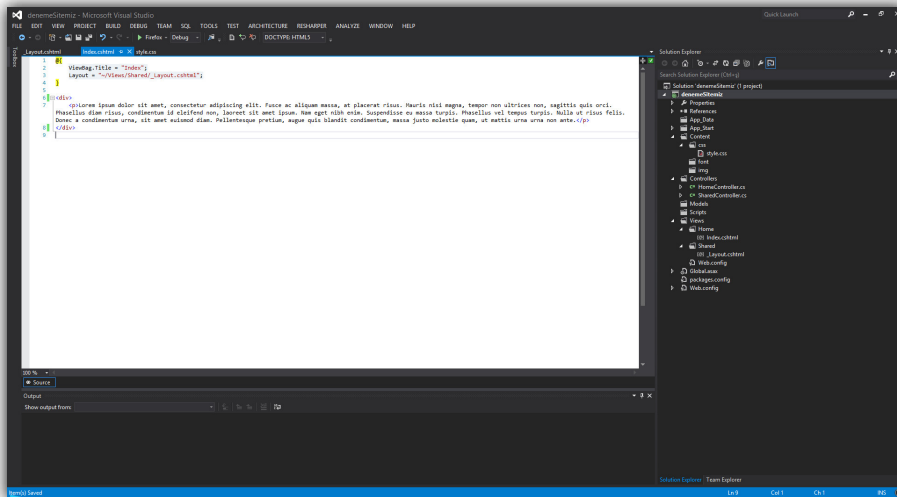
Büyütmek için resme tıklayınız.

Şimdi daha önceki derslerimizde oluşturduğumuz denemeSitemiz.sln dosyasını çalıştırıyoruz ve projemizi açıyoruz.



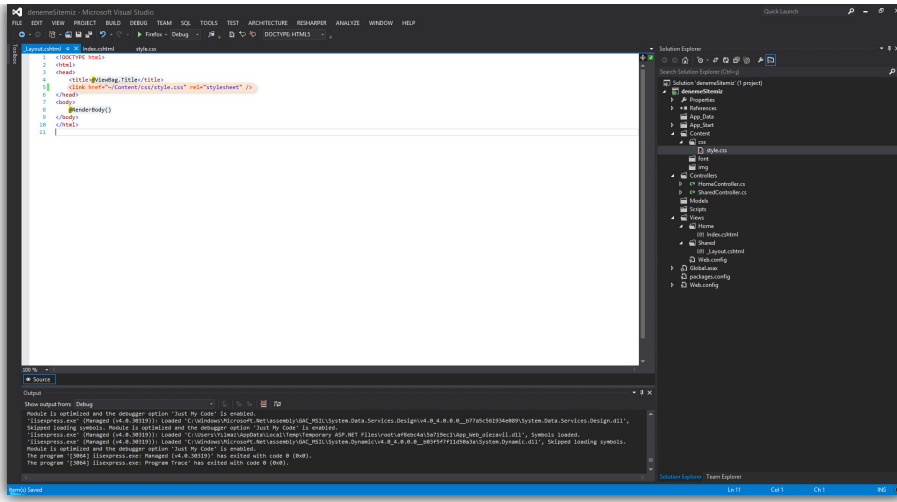
Büyütmek için resme tıklayınız.

Index.cshtml dosyasının içindeki daha önce yazdığımız içeriği siliyoruz.



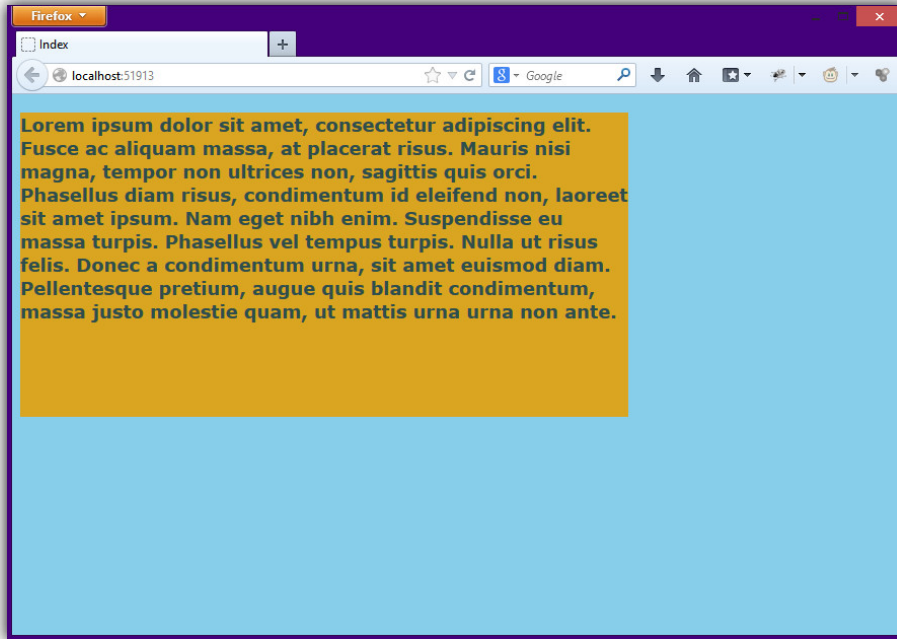
Büyütmek için resme tıklayınız.

Görselde görünen kodları yazıyoruz ve _Layout.cshtml dosyamıza dönüyoruz.



Büyütmek için resme tıklayınız.

Görseldeki gibi tekrar style.css dosyamızı sayfamıza ekliyoruz ve F5'e basıyoruz.



Büyütmek için resme tıklayınız.

Bu şekilde bir ekran çıktısı elde etmiş olmalıyız.

padding

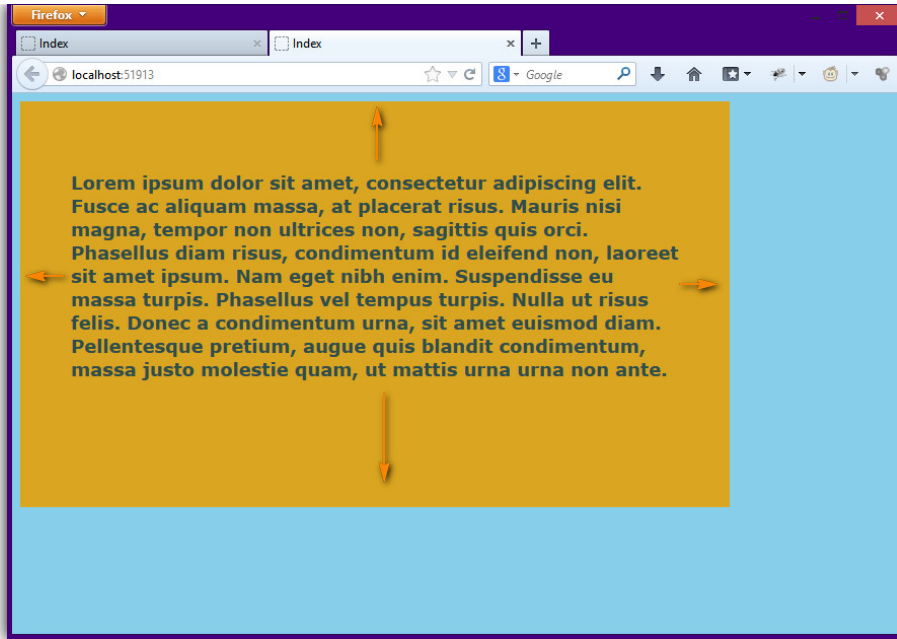
padding özelliği *hücre içi boşluk* vermek için kullanılır. Buradaki hücre tanımı <div>'i temsil etmektedir. style.css dosyasına dönüyoruz ve aşağıdaki kodları yazıyoruz.

```

1 | div {
2 |   width: 600px;
3 |   height: 300px;
4 |   background-color: goldenrod;
5 |   color: darkslategray;
6 |   font-size: 18px;
7 |   padding: 50px;
8 | }

```

div elementimize padding özelliğini ve 50px değerini ekledik. Şimdi tekrar F5'e basıyor ev ekran çıktımıza bakıyoruz.



Büyütmek için resme tıklayınız.

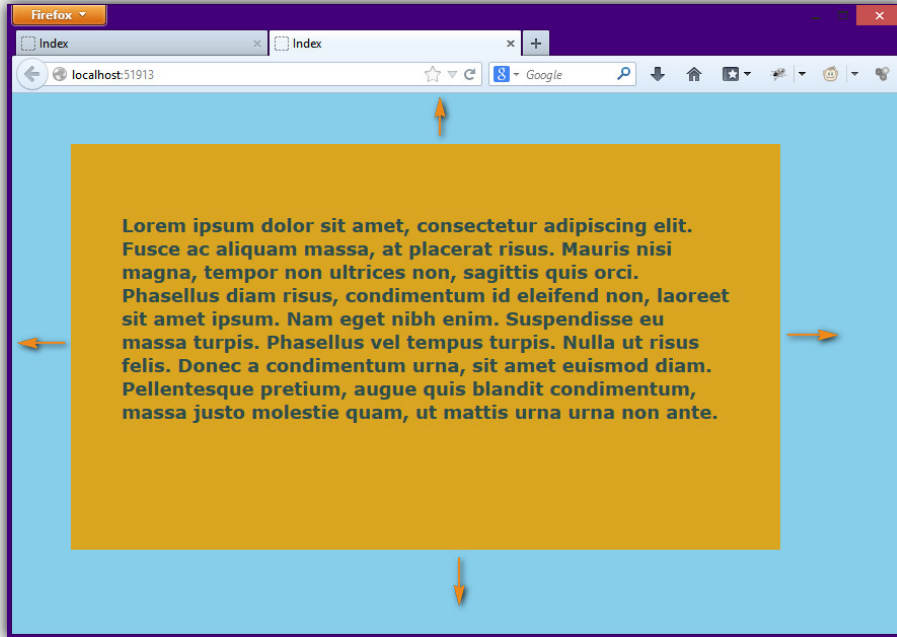
Gördüğünüz gibi dört bir yana 50px'lik bir hücre içi boşluk vermiş olduk.

margin

margin özelliği *hücre dışı boşluk* vermek için kullanılır. *style.css* dosyamıza dönüyoruz ve aşağıdaki kodları yazıyoruz.

```
1  div {
2    width: 600px;
3    height: 300px;
4    background-color: goldenrod;
5    color: darkslategray;
6    font-size: 18px;
7    padding: 50px;
8    margin: 50px;
9  }
```

Çalışan penceremizi F5 ile refresh ediyoruz ve ekran görüntüsünün nasıl değiştiğini inceliyoruz.



Büyütmek için resme tıklayınız.

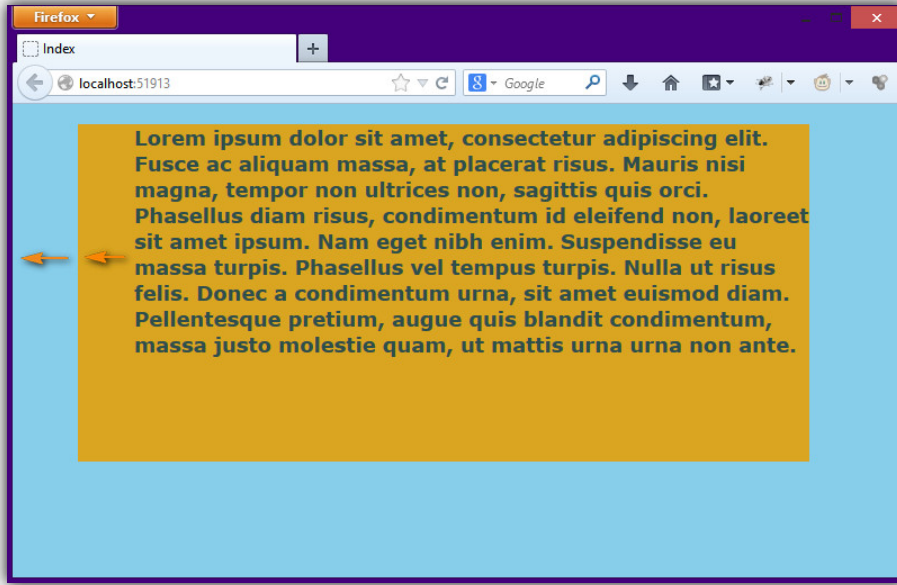
Görüldüğü gibi 50px'lik bir hücre dışı boşluk vermiş olduk.

```
1  div {
2    width: 600px;
3    height: 300px;
4    background-color: goldenrod;
5    color: darkslategray;
6    font-size: 18px;
7    padding-left: 50px;
```



```
8 | margin-left: 50px;  
9 | }
```

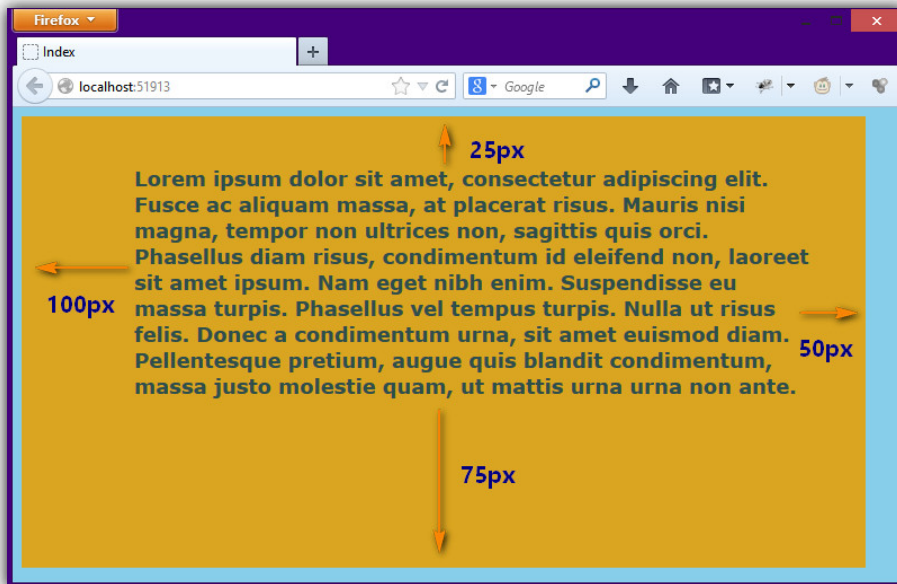
İsterse bu boşlukları sadece sol tarafa da verebiliriz. Yukarıdaki örnekte sola 50px hücre içi ve dışı boşluk verdik. Ekran görüntümüz ise şu şekilde olacaktır.



Büyütmek için resme tıklayınız.

Dört bir yana ayrı ayrı hücre içi boşluklar vermek için aşağıdaki kodları kullanabiliriz.

```
1 | div {  
2 |   width: 600px;  
3 |   height: 300px;  
4 |   background-color: goldenrod;  
5 |   color: darkslategray;  
6 |   font-size: 18px;  
7 |   padding: 25px 50px 75px 100px;  
8 | }
```



Büyütmek için resme tıklayınız.

Bu dört sayı saat yönüne göre hesaplanır. 25px padding-top, 50px padding-right, 75px padding-bottom ve 100px padding-left anlamındadır. Sıklıkla kullanılan bir yöntemdir. Aynı şekilde margin içinde bu şekilde hücre dışı boşluklar verebiliriz.

Bu dersin videosunu 1080p HD kalitesinde izleyebilirsiniz.

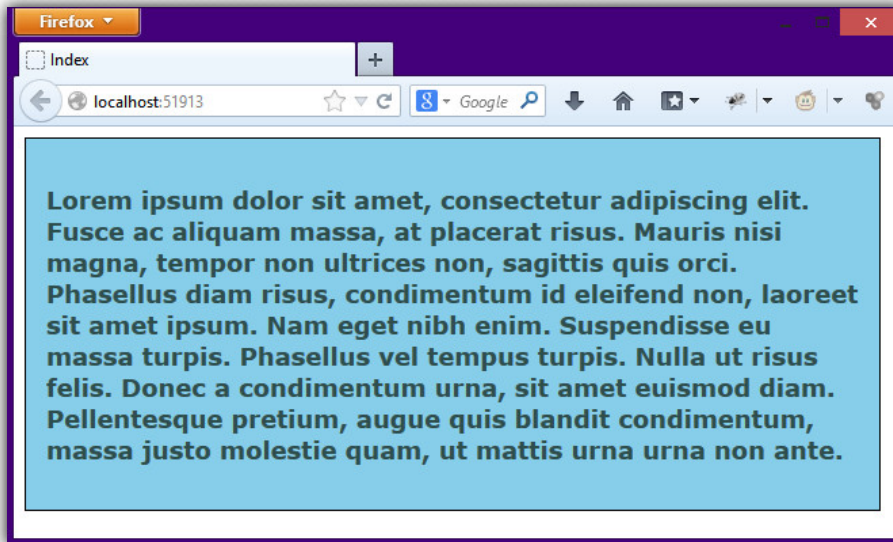


CSS'te Border

CSS'te *border*, elementlerimize *sınır* vermek için kullanılır. Şimdi her zaman yaptığımız gibi denemeSitemiz.sln dosyasını çalıştırıyoruz ve style.css dosyasını açıyoruz.

```
1 body {
2   background-color: inherit;
3   color: black;
4   font-family: Verdana;
5   font-size: 16px;
6   font-weight: bold;
7 }
8 div {
9   width: 600px;
10  background-color: skyblue;
11  color: darkslategray;
12  font-size: 18px;
13  padding: 15px;
14  border: 1px solid black;
15 }
```

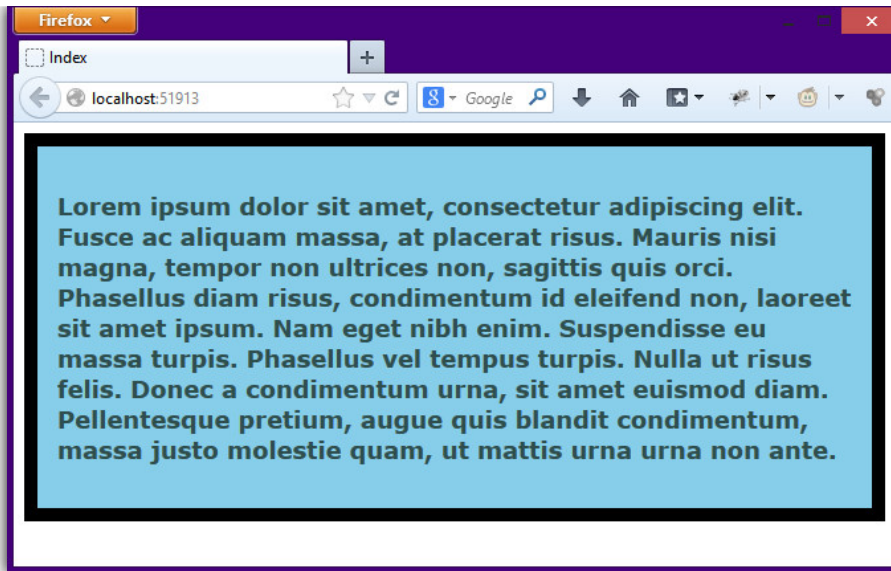
Dikkatinizi çekerim <body> etiketinin özelliklerini değiştirdim. Siz de aynı şekilde yapmaya özen gösterin. <div> etiketinin değerlerini de değiştirdim ve border özelliği ekledim. Şimdi çıktımıza bakalım.



Büyütmek için resme tıklayınız.

Daha önce oluşturduğumuz div'e 1px genişliğinde tipi solid (katı) olan ve siyah renkte bir sınır ekledim. Şimdi sınır kalınlığını biraz arttıralım tekrar çıktımıza bakalım.

```
1 body {
2   background-color: inherit;
3   color: black;
4   font-family: Verdana;
5   font-size: 16px;
6   font-weight: bold;
7 }
8 div {
9   width: 600px;
10  background-color: skyblue;
11  color: darkslategray;
12  font-size: 18px;
13  padding: 15px;
14  border: 10px solid black;
15 }
```



Büyütmek için resme tıklayınız.

Sınıf kalınlığını 10px yaptım ve sınır daha belirgin bir hâle geldi. İstersem tüm bu değerleri şu şekilde de yazabilirdim.

```
1  div {  
2    width: 600px;  
3    background-color: skyblue;  
4    color: darkslategray;  
5    font-size: 18px;  
6    padding: 15px;  
7    border-width: 10px;  
8    border-style: solid;  
9    border-color: black;  
10 }
```

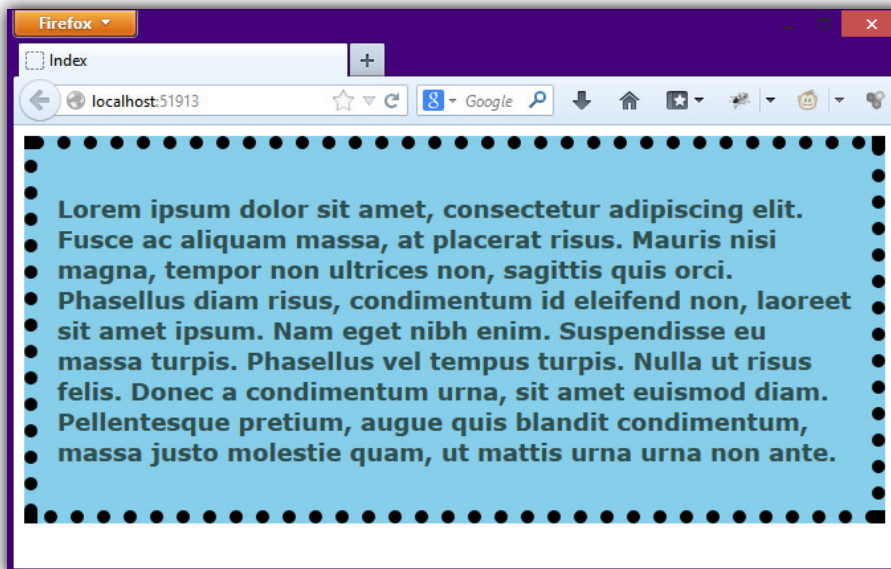
Ayrı ayrı böyle tanımlamaktansa tek bir özellik içinde üç değeri de tanımlamak daha mantıklı olacaktır ve çoğunlukla da böyle kullanılır.

border-style

border-style'lar yani border stilleri birkaç çeşittir ve solid'ten sonra en çok kullanılanı *dotted* yani noktalıdır. Kodlarımı tekrar değiştiriyorum.

```
1  div {  
2    width: 600px;  
3    background-color: skyblue;  
4    color: darkslategray;  
5    font-size: 18px;  
6    padding: 15px;  
7    border: 10px dotted black;  
8  }
```

Şimdi tekrar çıktımıza bakalım:



Büyütmek için resme tıklayınız.

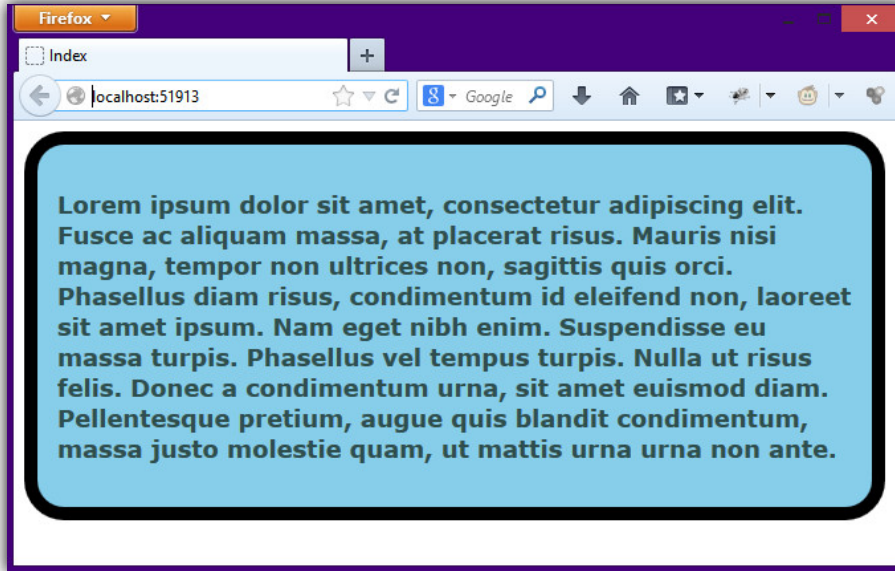
Bu şekilde border stilinin noktalı bir şekilde deđiřtiđini grebilirsiniz. *dotted* dışında *double*, *dashed*, *groove* gibi birkaç stil daha vardır. Bunları Visual Studio'nun intellisense yardımı ile grebilirsiniz. Tek yapmanız gereken CSS dosyanızda *border-style*: yazmak ve Visual Studio'nun intellisense'nin aılmasını beklemek.

border-radius

Eđer border'larımızı oval yapmak istersek *border-radius*, yani *border apı* zelliđini kullanıyoruz.

```
1 div {
2   width: 600px;
3   background-color: skyblue;
4   color: darkslategray;
5   font-size: 18px;
6   padding: 15px;
7   border: 10px solid black;
8   border-radius: 30px;
9 }
```

Bu kodun ıktısı ařađıdaki ekran grnts olacaktır:

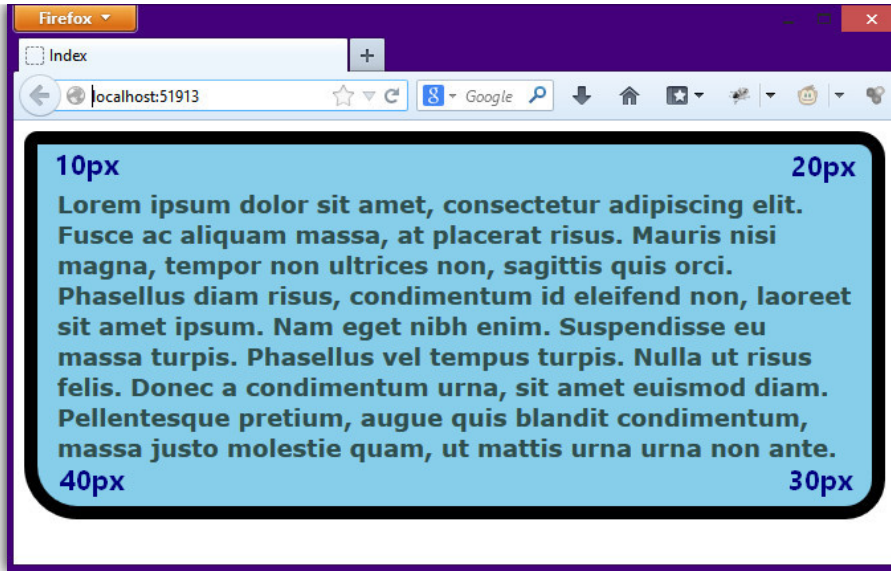


Bytmek iin resme tıklayınız.

Grldđ gibi div'imize 30px'lik bir sınır apı verdik ve bylece oval bir div elde etmiř olduk.

```
1 div {
2   width: 600px;
3   background-color: skyblue;
4   color: darkslategray;
5   font-size: 18px;
6   padding: 15px;
7   border: 10px solid black;
8   border-radius: 10px 20px 30px 40px;
9 }
```

Bu şekilde bir border-radius kullanımı padding ve margin'de olduđu gibi aynıdır. Sol st kředen bařlamak zere saat ynnde her bir kře iin ayrı ayrı border-radius deđerleri verebiliriz. ıktımız ise řu şekilde olacaktır:

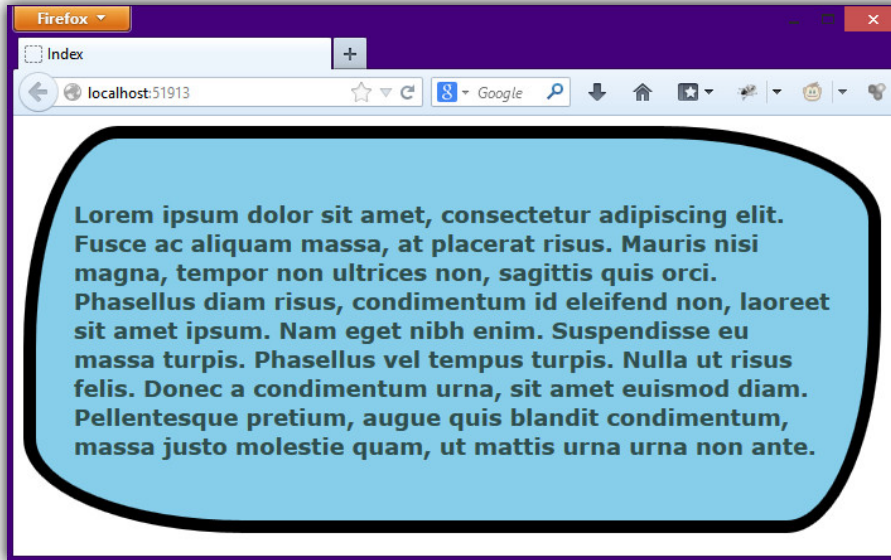


Büyütmek için resme tıklayınız.

İstersek aşağıdaki kodlar ile bayrak görünümü bir div elde edebiliriz.

```
1 div {  
2   width: 600px;  
3   background-color: skyblue;  
4   color: darkslategray;  
5   font-size: 18px;  
6   padding: 30px;  
7   border: 10px solid black;  
8   border-radius: 75px 175px / 175px 75px;  
9 }
```

Sayılarla oynamak size kalmış. Ekran görüntümüz şu şekilde olacaktır:



Büyütmek için resme tıklayınız.

Bu dersin videosunu 1080p HD kalitesinde izleyebilirsiniz.

CSS BORDER(KENARLIK)



Css border(kenar) çizgisi özellikleri ile kenar çizgisinin biçimini, rengini ve kalınlığını ayarlanabilir. Tüm kenarlara veya ayrı ayrı her kenara bu ayarları uygulayabiliriz. Bu özellikler şunlardır.

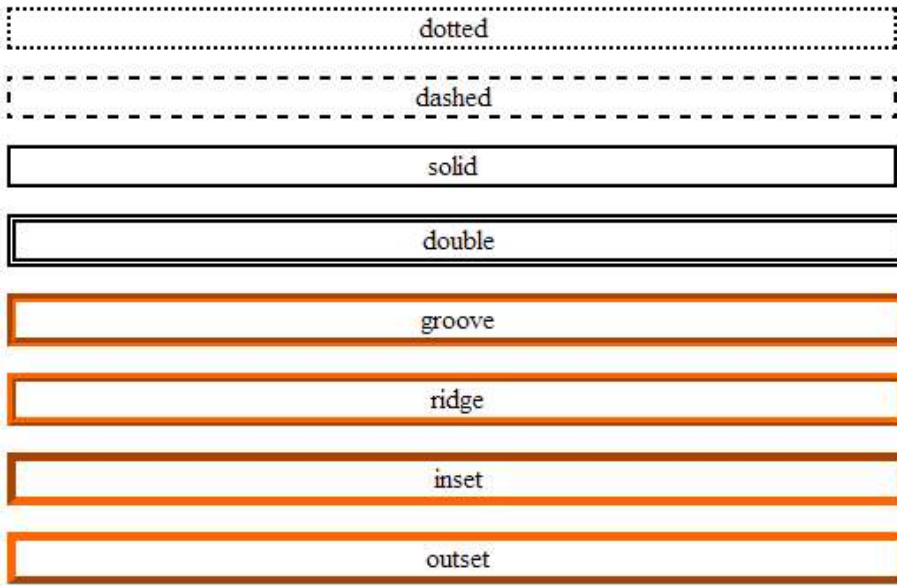
- **border-style**(Kenar çizgi biçemi)
 - **border-top-style**(üst kenar çizgi biçemi)
 - **border-right-style**(sağ kenar biçemi)
 - **border-bottom-style**(alt kenar çizgi biçemi)
 - **border-left-style**(sol kenar biçemi)
- **border-width**(Kenar çizgi genişliği-kalınlığı)
 - **border-top-width**(üst kenar çizgi genişliği)
 - **border-right-width**(sağ kenar çizgi genişliği)
 - **border-bottom-width**(alt kenar çizgi genişliği)
 - **border-left-width**(sol kenar çizgi genişliği)
- **border-color**(Kenar çizgi rengi)
 - **border-top-color**(üst kenar çizgi rengi)

- > **border-right-color**(sağ kenar çizgi rengi)
- > **border-bottom-color**(alt kenar çizgi rengi))
- > **border-left-color**(sol kenar çizgi rengi)

> **border**(Kısaltma işlemleri için kullanılır)

CSS KENAR ÇİZGİ BİÇEMİ

Kenar çizginin nasıl olacağını (noktalı,kesik kesik,düz,çift, vb.) belirler Herhangi bir değer yazılmadığı zaman **none** varsayılan olarak geçerlidir. Aşağıdaki değerleri alabilir.



Örnek olarak üst ve alt kenar için düz; sağ sol kenarlar için çift çizgi biçimini uygulayalım.

CSS:

```
.kutu{width:500px;
  height:30px;
  border-top-style:solid;
  border-bottom-style:solid;
  border-left-style:double;
  border-right-style:double;
}
```

HTML:

```
<div class="kutu"></div>
```

En Zarif Çiçekli

Birbirinden Güzel Çiçek
Buketleri, Hemen Sizi
Verin!



www.ciceksepeti.cc

KATEGORİ

 HTML

 CSS

 PHP

 JQUERY

 PHOTOSHOP

 JavaScript

ÇIKTI:



yaptığımız örneğin kısaltılmış halini yazalım. Dört değer olduğu için (üst,sağ,alt ve sol) şu şekilde olmalıdır.

```
.kutu {border-style:solid double solid double}
```

CSS BORDER (KENAR) ÇİZGİ GENİŞLİĞİ

"border-width" özelliği kenar çizgisinin genişliğini(kalınlığını) ayarlamak için kullanılır. Genişliği ayarlamak için piksel(px) olarak verilebilir veya aşağıda değerleri kullanabiliriz.

- > Thin(ince)
- > medium(orta)
- > thick(kalın)

Örnek:

CSS:

```
.kutu{width:500px;
height:30px;
border-style:solid double solid double;
border-top-width:5px;
border-bottom-width:5px;
border-left-width:5px;
border-right-width:5px;
}
```

HTML:

```
<div class="kutu"></div>
```

ÇIKTI:



Border-width özelliğini kısaltılmış yazalım.



Web Dünyası



Linux

POPÜLER MAKALE

> HTML LİNK(Bağlantı)EKLEME - LİNK VERME

> HTML RESME LİNK VERME

> HTML YAZI VE RESİM ORTALAMA

> HTML FORM OLUŞTURMA

> HTML DİV ve SPAN NEDİR?

> HTML KAYAN RESİM VE Marguee KULLANIMI

> HTML ARKAPLANA MÜZİK EKLEME

> CSS (BOYUT) GENİŞLİK VE YÜKSEKLİK (Width-Height)

> HTML RESİM EKLEME

> PHP ceil()-floor()-round() YUVARLAMA FONKSİYONU

ETİKET

• html dersleri

• HTML

• css dersleri

• CSS

• php dersleri

• PHP

• javaScript dersleri

- **border-width:2px 4px 5px 8px;**(üst 2px, sağ 4px, alt 5px, sol 8px olur.)
- **border-width:2px 5px 7px;**(üst 2px, sağ ve sol 5px, alt 7px olur.)
- **border-width:2px 5px;**(üst ve alt 2px, sağ ve sol 5px olur.)
- **border width:2px**(Tüm kenar çizgisi genişlikleri 2px olur.)

Şimdi en son yaptığımız örneği kısaltılmış halde yazalım.

CSS:

```
.kutu{width:500px;
      height:30px;
      border-style:solid double solid double;
      border-width:5px;
}
```

CSS BORDER (KENAR) ÇİZGİSİ RENGİ

"border-color" özelliği, kenar çizgisine renk vermek için kullanılır. Kenarların hepsine birden renk verebileceğimiz gibi tek tek de renk verebiliriz.

```
.kutu{border-style:solid;
      border-color:blue;
}
```

Her kenarı ayrı ayrı ele alalım; üst ve alt kenar rengi #ff600; sağ ve sol kenar rengi #00cc00, biçemi düz çizgi ve genişliği 5 piksel olsun.

CSS:

```
.kutu{width:500px;
      height:30px;
      border-style:solid;
      border-width:5px;
      border-top-color:#ff6600;
```

• JAVASCRIPT

• jquery dersleri

• JQUERY

```
border-left-color:#00cc00;  
border-right-color:#00cc00;  
border-bottom-color:#ff6600;  
}
```

ÇIKTI:



Şimdi border renk özelliğinin kısa yazılış şekline bakalım.

- **border-color:#ff6600 #00cc00 #ff6600 #00cc00;** (üst #ff6600, sağ #00cc00, alt #ff6600 sol #ff6600)
- **border-color:#ff6600 #00cc #ccc;** (üst #ff6600, sağ ve sol #00cc00, alt #ccc;)
- **border-color:#ff6600 #00cc00;** (üst ve alt #ff6600, sağ ve sol #00cc00)
- **border-color:#ff6600;** (Tüm kenar çizgi renkleri #ff6600 olur.)

Şimdi de en son yaptığımız örneğin kısaltılmış halini yazalım.

```
.kutu{width:500px;  
height:30px;  
border-style:solid;  
border-width:5px;  
border-color:#ff6600 #00cc00;  
}
```

CSS BORDER(KENAR) ÇİZGİSİ TÜM ÖZELLİKLERİ İÇİN KISALTMA

Kenar çizgisinin biçemi, rengi ve kalınlığını tek seferde biçimlendirmek için "border" özelliğini kullanmamız yeterlidir.

ÖRNEK:

Kenar çizgi genişliği 1px, düz ve rengi de #ff6600 olsun.

```
.kutu{  
border:1px solid #ff6600;  
}
```



CSS'te Class'lar ve ID'ler

Bu konu CSS'in en temel konularından biridir. Bu yüzden iyi bir şekilde öğrenmeniz önemlidir.

Class'lar

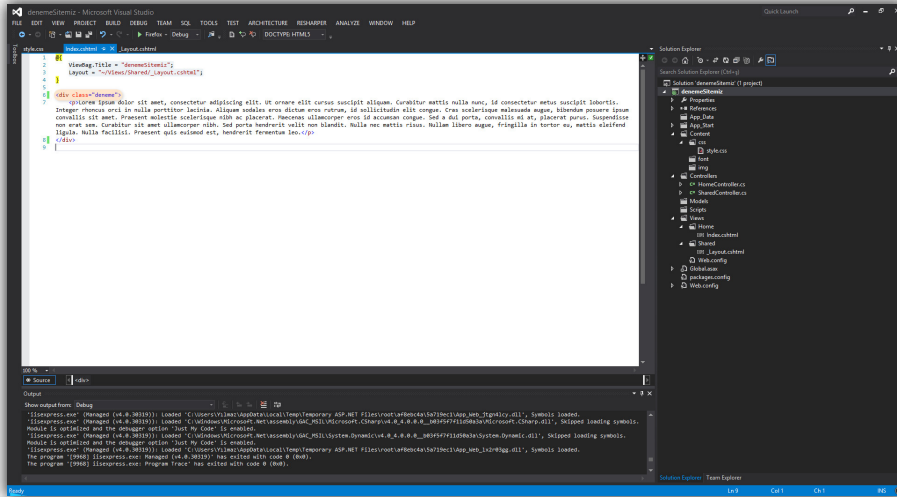
Class dilimizde *sınıf* anlamına gelmektedir. Aynı sayfada birden fazla HTML etiketine aynı CSS özelliklerini vermek için kullanılır. Class verilmeden önce seçicinin başına nokta (.) yazılır. Aşağıdaki kodları inceleyelim:

```

1  body {
2      background-color: white;
3      color: darkmagenta;
4      font-family: Verdana, monospace, sans-serif;
5      font-size: 12px;
6      font-weight: bold;
7  }
8
9  .deneme {
10     width: 600px;
11     background-color: skyblue;
12     color: darkslategray;
13     font-size: 14px;
14     padding: 50px;
15 }

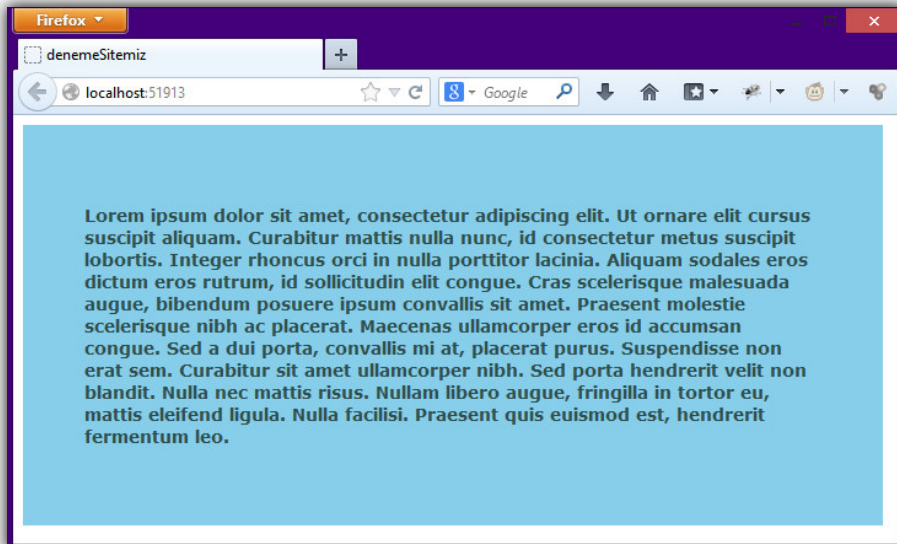
```

Şimdi bunu projemizde kullanalım.



Büyütmek için resme tıklayınız.

Index.cshtml dosyamın içindeki <div> etiketinde *class* isimli bir atfı kullandım ve atfın özelliğini ise *deneme* olarak belirledim. Şimdi ekran görüntümüze bakalım.



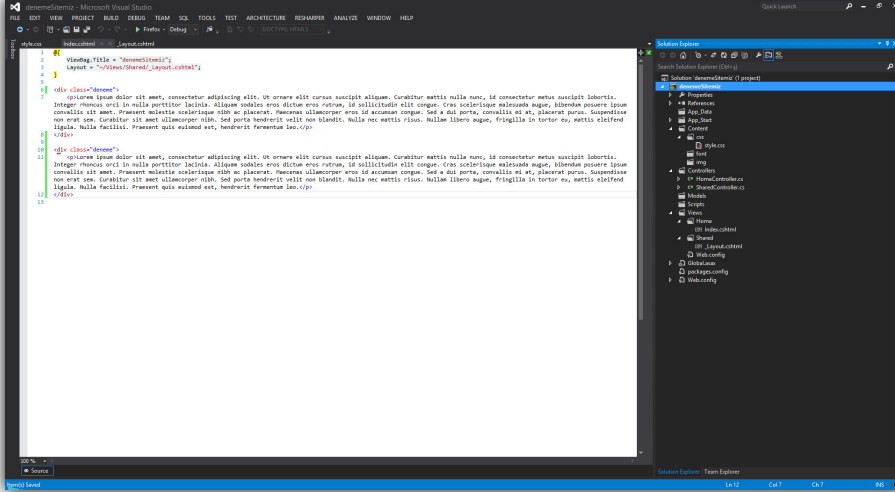
Büyütmek için resme tıklayınız.



Gördüğünüz gibi <div> elementimiz class'ta belirttiğimiz özelliklere göre şekillendi. Şimdi tekrar aynı div'i aynı class ile kullanalım. Ama bundan önce class'ıma bir margin-top vermeliyim, ki iki div birbirine yapışmasın. Kodları aşağıdaki gibi değiştirdim.

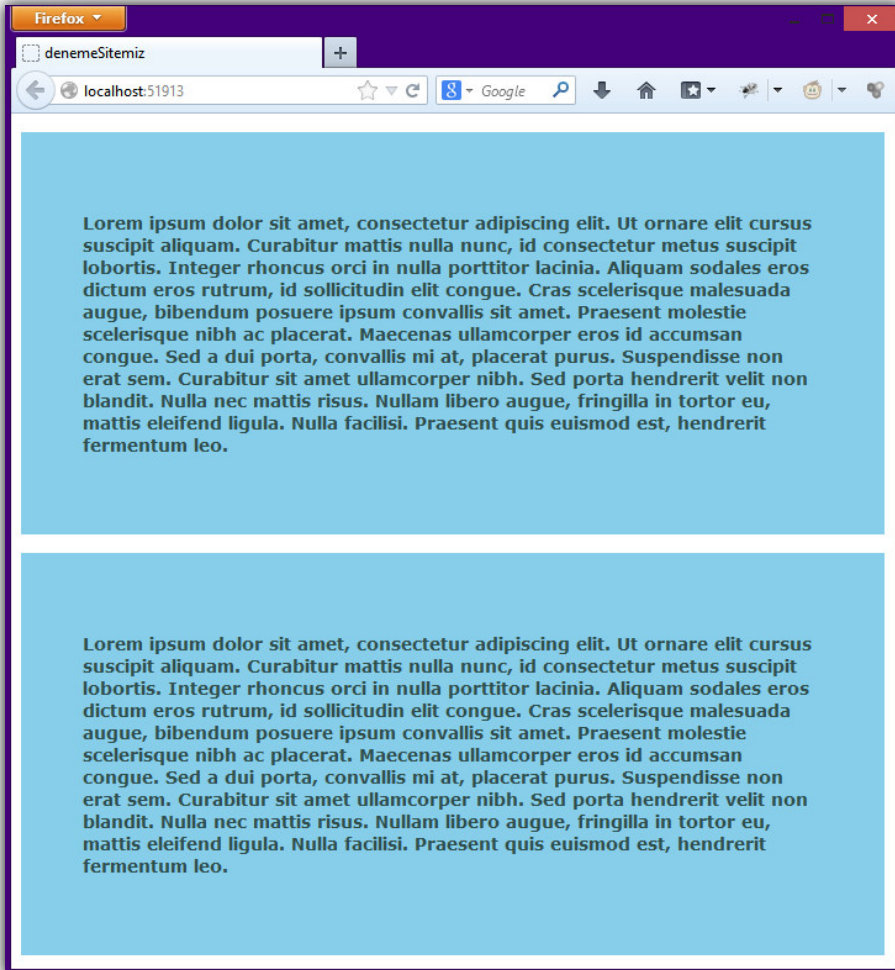
```
1 .deneme {
2   width: 600px;
3   background-color: skyblue;
4   color: darkslategray;
5   font-size: 14px;
6   padding: 50px;
7   margin-top: 15px;
8 }
```

Sadece margin-top ekledim. Şimdi index.cshtml dosyamızdaki div'i klonlayalım.



Büyütmek için resme tıklayınız.

Şekilde görüldüğü gibi aynı div'den iki adet oluşturduğum. Şimdi de ekran çıktısına bakalım:



Büyütmek için resme tıklayınız.

Ne demiştik? Bir class'ı aynı sayfa içinde birden fazla kullanabilirsiniz. Böyle aynı div'i iki kez kullanmış olduk.

ID'ler



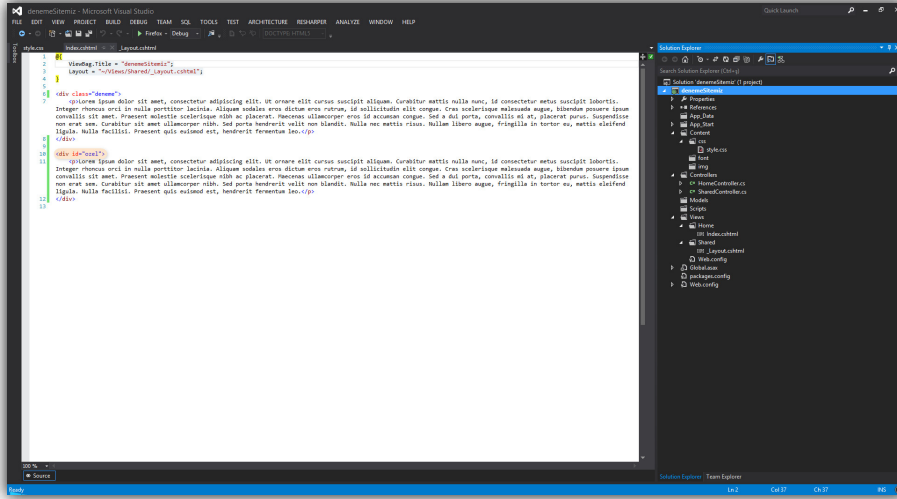
ID, *identity* kelimesinin kısaltmasıdır ve dilimizde *kimlik* anlamına gelmektedir. CSS'te ID'ler bir sayfada yalnızca bir kez kullanılırlar. ID kullanılırken seçiciden önce *diez* (#) işareti yazılır.

```

1  #ozel {
2      width: 600px;
3      background-color: violet;
4      color: darkslategray;
5      font-size: 14px;
6      padding: 50px;
7      margin-top: 15px;
8  }

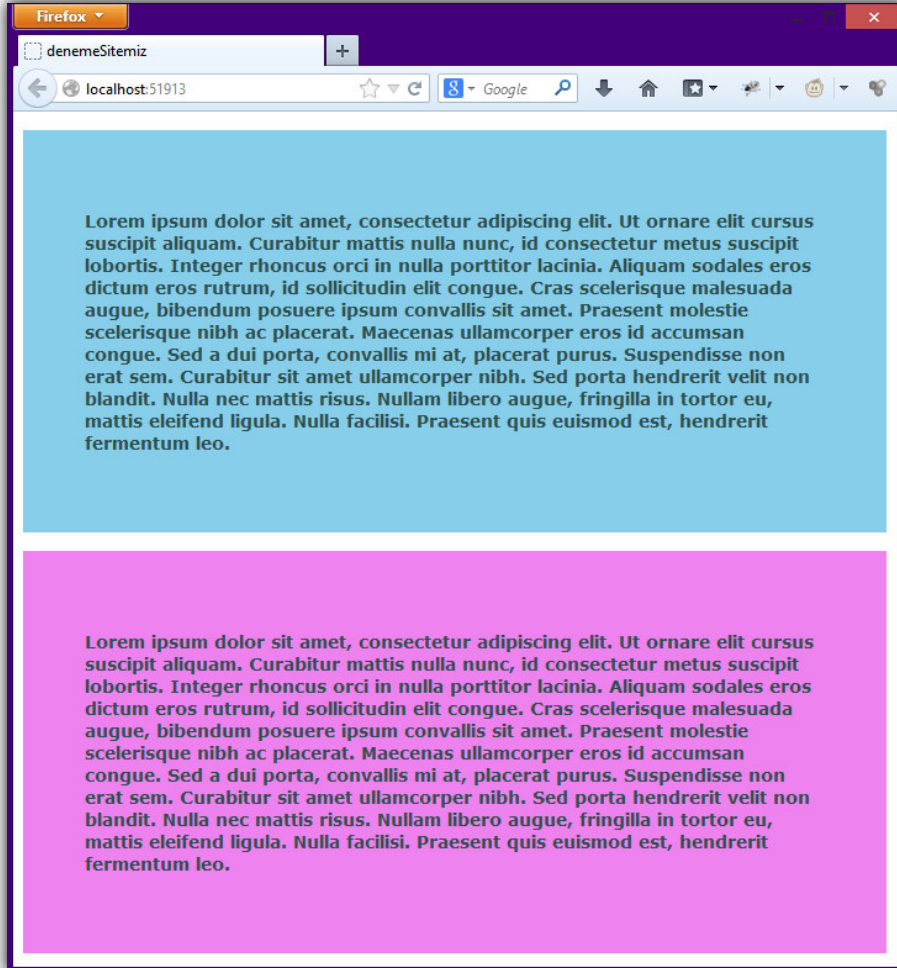
```

CSS kodlarına *ozel* adında bir ID seçicisi ekledim. Şimdi bunu sayfamızda kullanalım.



Büyütmek için resme tıklayınız.

Klonladığım deneme class'ına sahip div'i sildim ve yerine özel ID'sine sahip yeni bir div oluşturdum. HTML'de bir etikete ID verilirken etiketten sonra *id* atfı yazılır ve değerine CSS'te belirttiğiniz seçici adını yazarsınız. Şimdi bunun ekran görüntüsüne bakalım:



Büyütmek için resme tıklayınız.



CSS'te Metin Biçimlendirme

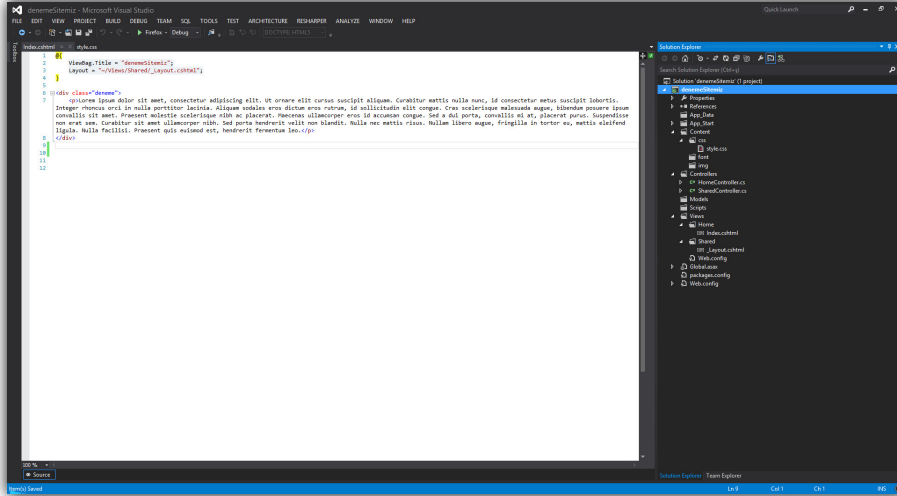
Bu dersimizde CSS'te metin (text) biçimlendirmek için kullandığımız özellikleri inceleyeceğiz. İlk olarak denemeSitemiz.sln adlı projemizi açıyoruz.

```

1  body {
2    background-color: white;
3    color: darkmagenta;
4    font-family: Verdana, monospace, sans-serif;
5    font-size: 12px;
6    font-weight: bold;
7  }
8  .deneme {
9    width: 600px;
10   background-color: skyblue;
11   color: darkslategray;
12   font-size: 14px;
13   padding: 50px;
14   margin-top: 15px;
15 }
16 #ozel {
17   width: 600px;
18   background-color: violet;
19   color: darkslategray;
20   font-size: 14px;
21   padding: 50px;
22   margin-top: 15px;
23 }

```

CSS kodlarım bu şekilde.



Büyütmek için resme tıklayınız.

Index.cshtml dosyamın içeriği ise bu şekilde.

color

color özelliğini daha önce görmüştük. Yine de tekrar edelim. *color* dilimizde *renk* anlamındadır ve metinlerin rengini değiştirmek için kullanılır. Varsayılan olarak *black* yani *siyah* değerine sahiptir.

Genel olarak dört farklı şekilde renk verme değeri kodu kullanılır. İlk olarak rengin adı ile, HEX kodları ile, rgb ve rgba kodları ile...

```

1  .deneme {
2    width: 600px;
3    background-color: skyblue;
4    font-size: 14px;
5    padding: 50px;
6    margin-top: 15px;
7    color: darkslategray;
8  }

```

darkslategray olarak yazdığımız değer rengin adı ile verilmiştir.

```
1 .deneme {
2   width: 600px;
3   background-color: skyblue;
4   font-size: 14px;
5   padding: 50px;
6   margin-top: 15px;
7   color: #2F4F4F;
8 }
```

#2F4F4F değeri rengin HEX kodu ile verildiğini gösterir. HEX kodları daima diez (#) işareti ile başlar.

```
1 .deneme {
2   width: 600px;
3   background-color: skyblue;
4   font-size: 14px;
5   padding: 50px;
6   margin-top: 15px;
7   color: rgb(47,79,79);
8 }
```

rgb(47,79,79) değeri ise rgb kodu ile metinlere renk değeri verildiğini gösterir. Bu üç sayı sırasıyla kırmızı, yeşil ve mavi renkleri simgeler ve bu renklerin karışımı ile web renk renkleri elde edilir.

```
1 .deneme {
2   width: 600px;
3   background-color: skyblue;
4   font-size: 14px;
5   padding: 50px;
6   margin-top: 15px;
7   color: rgba(47,79,79,.5);
8 }
```

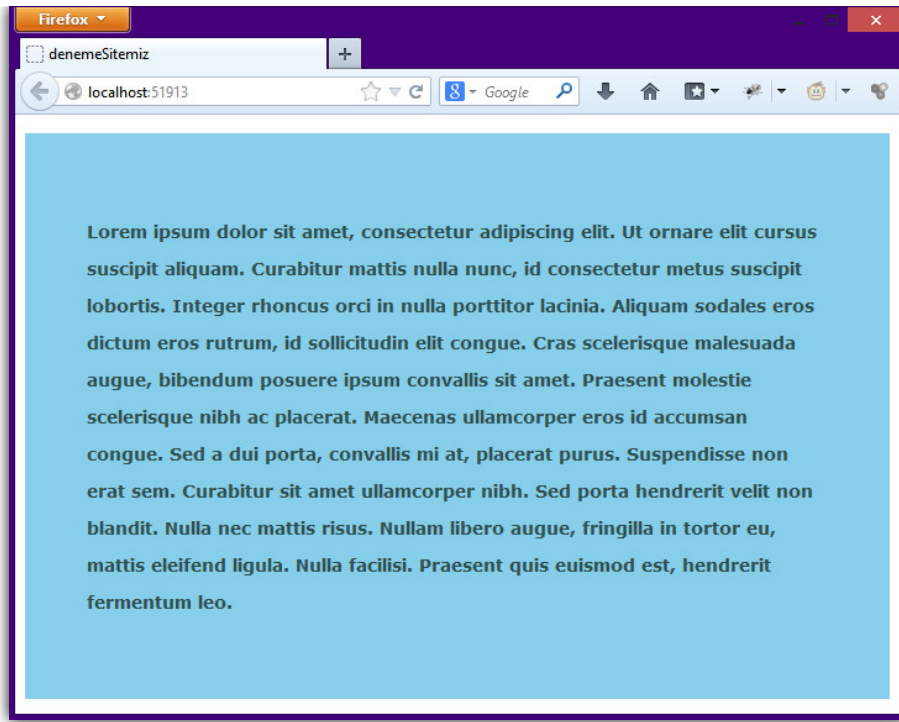
rgba(47,79,79,.5) değeri ise rgba kodunu belirtir. a harfinin açılımı *alpha*'dır ve renge şeffaflık vermek için kullanılır. Örneğin; bu rengimizi %50 oranında şeffaf yaptık. Son değere .49 yazsaydık %49 oranında şeffaf yapacaktık. 1 yazsaydık rengimiz hiçbir şekilde şeffaf olmayacaktı. Ama .1 yazsaydık %10 oranında şeffaf olacaktı.

line-height

Bazen satırlar arası çok dar olabilir. Bunun önüne geçebilmek için *line-height* yani *satır yüksekliği* özelliğini kullanabiliriz.

```
1 .deneme {
2   width: 600px;
3   background-color: skyblue;
4   font-size: 14px;
5   padding: 50px;
6   margin-top: 15px;
7   color: rgba(47,79,79,1);
8   line-height: 30px;
9 }
```

Yukarıdaki örnekte 30px'lik bir satır yüksekliği kullandık.



Büyütmek için resme tıklayınız.

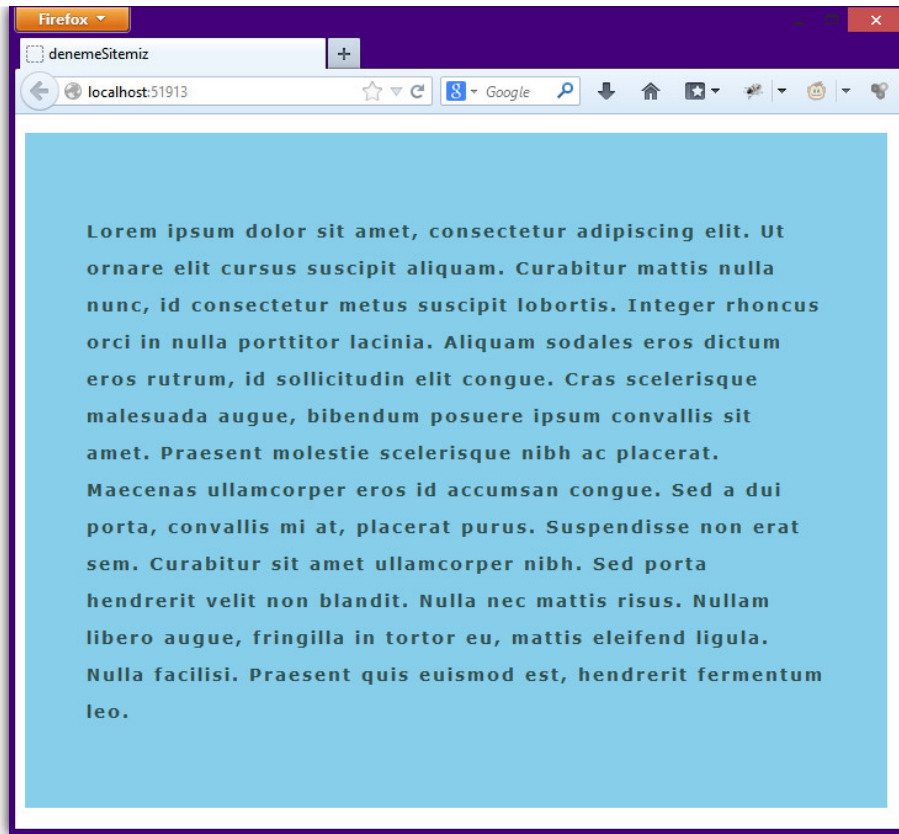
Çıktımız ise bu şekilde gözükecektir.

letter-spacing

Harfler arasında boşluk vermek için *letter-spacing* özelliği kullanılır. Dilimizde *harf boşluklama* gibi bir anlama gelmektedir.

```
1  .deneme {  
2    width: 600px;  
3    background-color: skyblue;  
4    font-size: 14px;  
5    padding: 50px;  
6    margin-top: 15px;  
7    color: rgba(47,79,79,1);  
8    line-height: 30px;  
9    letter-spacing: 2px;  
10 }
```

Örneğimize 2px bir harf boşluğu verildi.



Büyütmek için resme tıklayınız.

Çıktımız ise bu şekilde görünecektir.

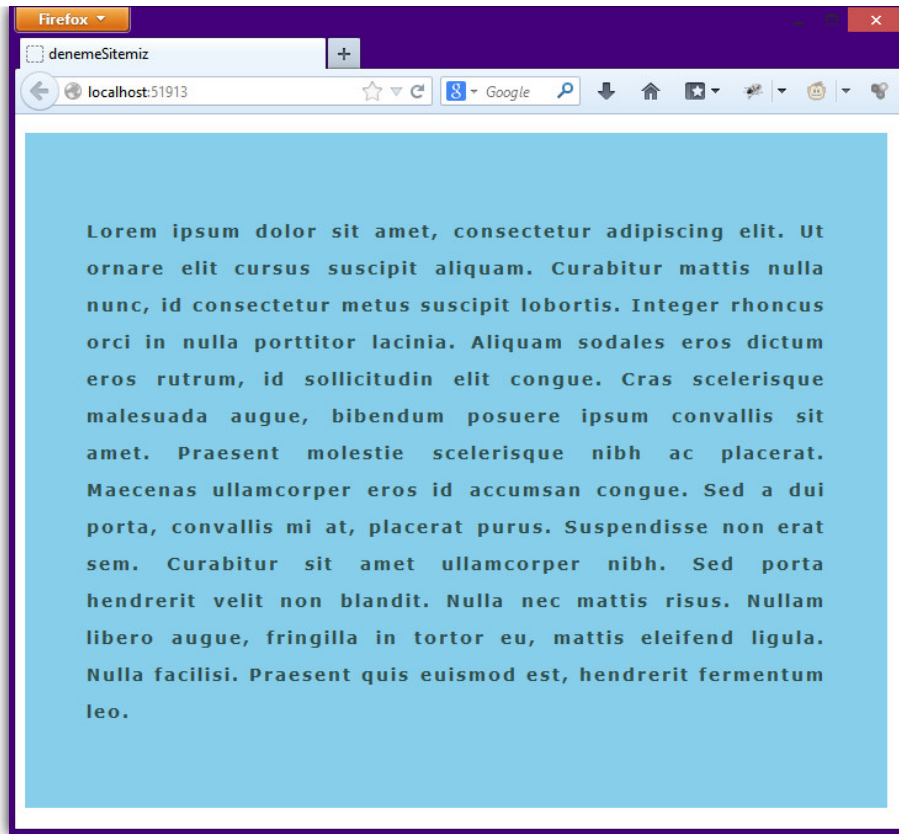
text-align

Metinleri sayfa içinde veya kullanıldığı element içinde hizalamak için kullanılır. *left*, *right*, *center* ve *justify* olmak üzere dört değer alır. Varsayılan olarak *left*'tir.

```
1  .deneme {  
2    width: 600px;  
3    background-color: skyblue;  
4    font-size: 14px;  
5    padding: 50px;  
6    margin-top: 15px;  
7    color: rgba(47,79,79,1);  
8    line-height: 30px;  
9    letter-spacing: 2px;  
10   text-align: justify;  
11 }
```

Metnimizi *justify* değeri ile div'in her iki yanına da yasladık.





Büyütmek için resme tıklayınız.

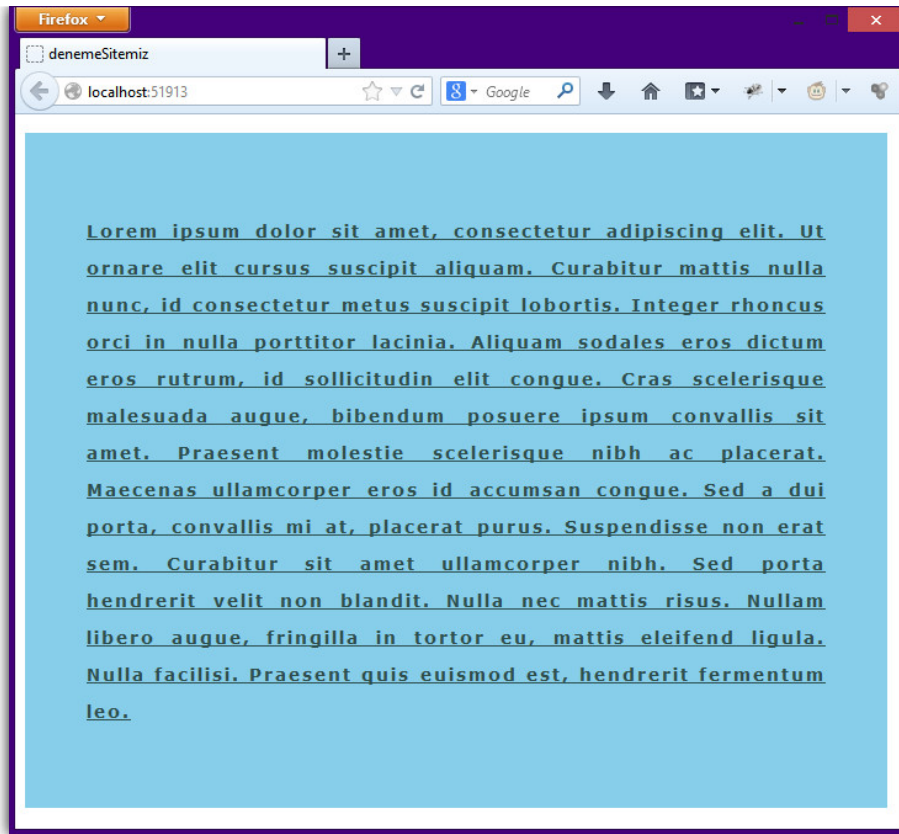
Ekran görüntümüz bu şekilde olacaktır.

text-decoration

Metinlerin altını çizmek için kullanılır. Varsayılan değeri *none*'dir.

```
1  .deneme {  
2    width: 600px;  
3    background-color: skyblue;  
4    font-size: 14px;  
5    padding: 50px;  
6    margin-top: 15px;  
7    color: rgba(47,79,79,1);  
8    line-height: 30px;  
9    letter-spacing: 2px;  
10   text-align: justify;  
11   text-decoration: underline;  
12 }
```

underline değeri ile metnimizdeki tüm cümlelerin altını çizili hâle getirdik.



Büyütmek için resme tıklayınız.

Ekran görüntümüz ise bu şekilde olacaktır.

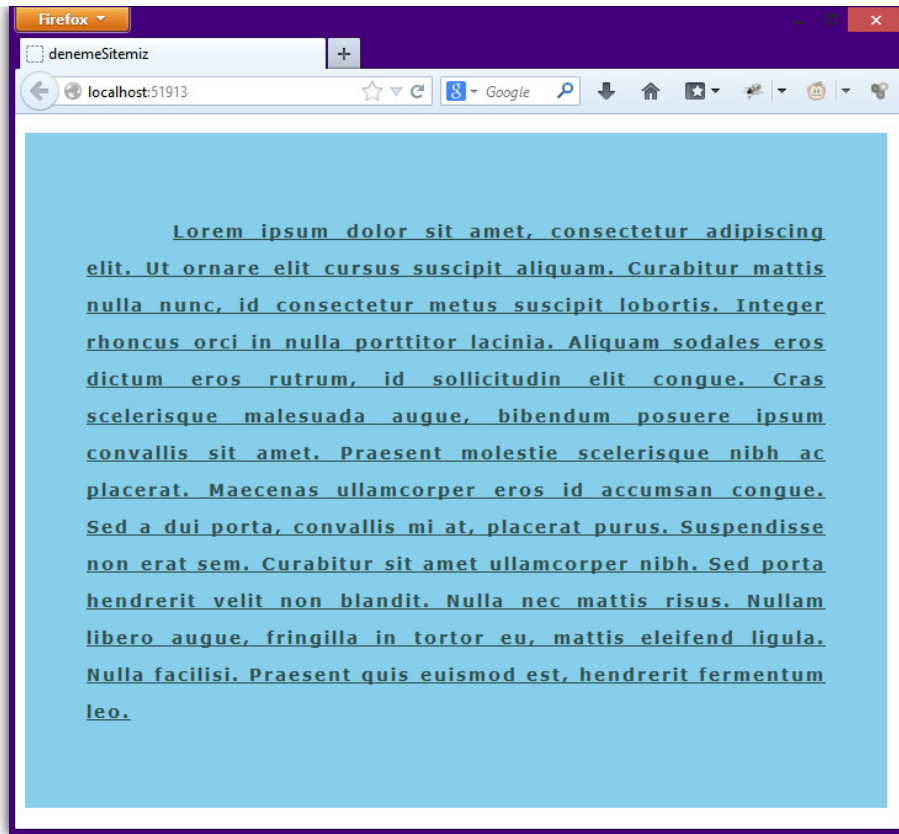
text-indent

Metinlerde kullanılan bir paragrafın ilk cümlesine girinti vermek için kullanılır.

```
1  .deneme {  
2    width: 600px;  
3    background-color: skyblue;  
4    font-size: 14px;  
5    padding: 50px;  
6    margin-top: 15px;  
7    color: rgba(47,79,79,1);  
8    line-height: 30px;  
9    letter-spacing: 2px;  
10   text-align: justify;  
11   text-decoration: underline;  
12   text-indent: 70px;  
13 }
```

Paragrafımızın ilk cümlesine 70px'lik bir girinti verdik.





Büyütmek için resme tıklayınız.

Ekran görüntümüz bu şekilde olacaktır.

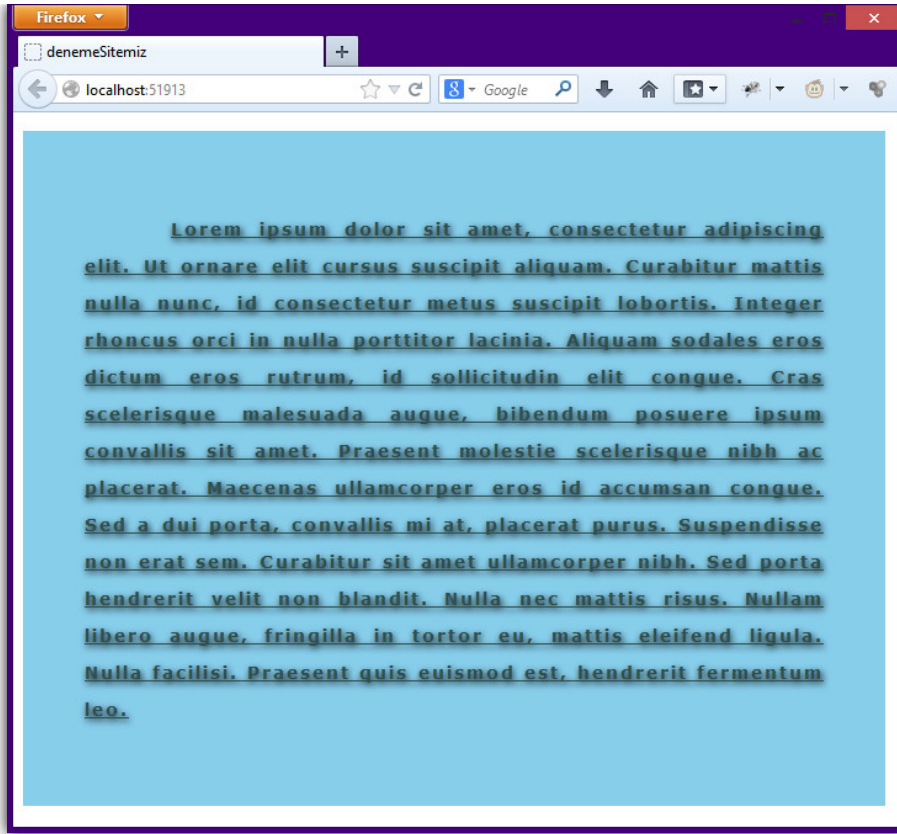
text-shadow

Metinleri gölgelendirmek için kullanılır.

```
1  .deneme {  
2    width: 600px;  
3    background-color: skyblue;  
4    font-size: 14px;  
5    padding: 50px;  
6    margin-top: 15px;  
7    color: rgba(47,79,79,1);  
8    line-height: 30px;  
9    letter-spacing: 2px;  
10   text-align: justify;  
11   text-decoration: underline;  
12   text-indent: 70px;  
13   text-shadow: 1px 2px 5px black;  
14 }
```

1px değeri soldan sağa doğru, 2px değeri ise yukarıdan aşağıya doğru hizalamayı gösterir. 5px değeri *blur* yani bulanıklaştırma için kullanılır. Son değer ise gölgenin rengini belirler.





Büyütmek için resme tıklayınız.

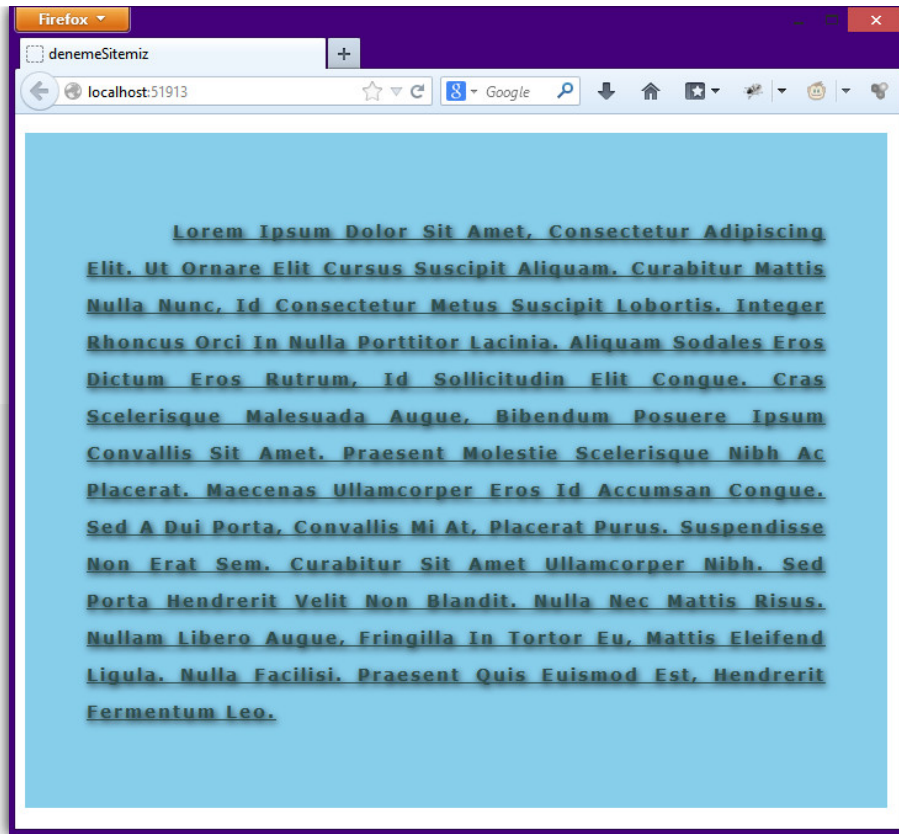
Ekran çıktımız bu şekilde olacaktır.

text-transform

Metnin boyutunu belirler. *capitalize*, *uppercase* ve *lowercase* olmak üzere üç değer alır. *capitalize* metindeki ilk harfleri büyük yapar. *uppercase* tüm harfleri büyük, *lowercase* ise tüm harfleri küçük yapar. Bu özellik varsayılan olarak *none* değerini alır.

```
1  .deneme {
2    width: 600px;
3    background-color: skyblue;
4    font-size: 14px;
5    padding: 50px;
6    margin-top: 15px;
7    color: rgba(47,79,79,1);
8    line-height: 30px;
9    letter-spacing: 2px;
10   text-align: justify;
11   text-decoration: underline;
12   text-indent: 70px;
13   text-shadow: 1px 2px 5px black;
14   text-transform: capitalize;
15 }
```

Metindeki tüm harflerin ilk kelimesini *capitalize* değeri ile büyük yaptık.



Büyütmek için resme tıklayınız.

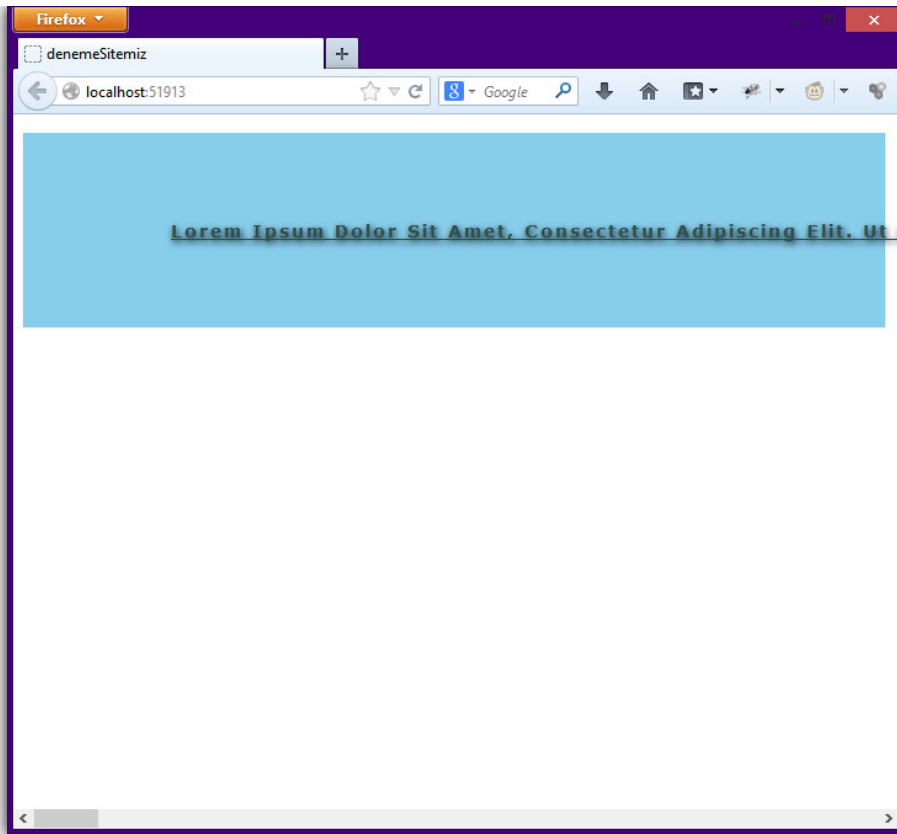
Ekran çıktımız bu şekilde olacaktır.

white-space

Beyaz boşluk tanımlaması yapar. Varsayılan olarak *normal* değerini alır. *pre*, metni boşluk ve satırları ile birlikte olduğu gibi gösterir. Kodların gösterimi için çok sık kullanılır. *nowrap*, metnin satırlarını yok sayar. Bütün satırları tek bir satırmiş gibi gösterir.

```
1  .deneme {
2    width: 600px;
3    background-color: skyblue;
4    font-size: 14px;
5    padding: 50px;
6    margin-top: 15px;
7    color: rgba(47,79,79,1);
8    line-height: 30px;
9    letter-spacing: 2px;
10   text-align: justify;
11   text-decoration: underline;
12   text-indent: 70px;
13   text-shadow: 1px 2px 5px black;
14   text-transform: capitalize;
15   white-space: nowrap;
16 }
```

nowrap değeri ile tüm satırları tek bir satırmiş gibi gösterdik.



Büyütmek için resme tıklayınız.

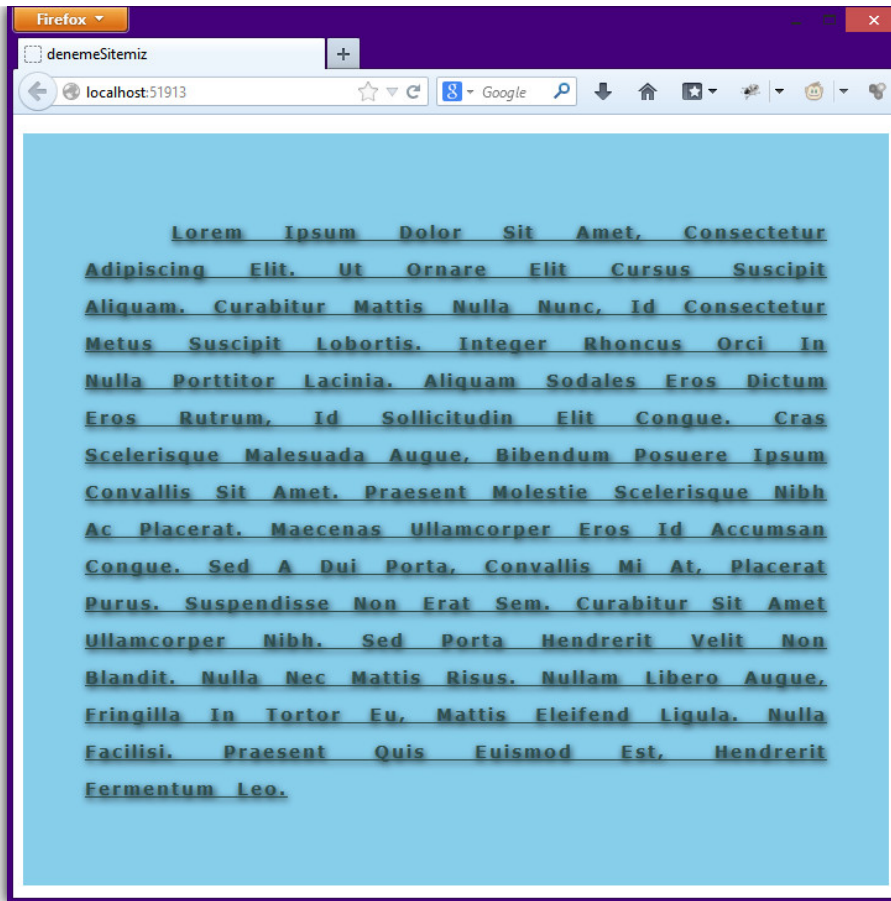
Ekran çıktımız ise bu şekilde olacaktır.

word-spacing

Kelimeler arasında boşluk vermek için kullanılır. Varsayılan olarak *normal* değerini alır.

```
1  .deneme {  
2    width: 600px;  
3    background-color: skyblue;  
4    font-size: 14px;  
5    padding: 50px;  
6    margin-top: 15px;  
7    color: rgba(47,79,79,1);  
8    line-height: 30px;  
9    letter-spacing: 2px;  
10   text-align: justify;  
11   text-decoration: underline;  
12   text-indent: 70px;  
13   text-shadow: 1px 2px 5px black;  
14   text-transform: capitalize;  
15   word-spacing: 10px;  
16 }
```

Her bir kelime arasına 10px'lik bir boşluk verdik.



Büyütmek için resme tıklayınız.

Ekran çıktımız ise bu şekilde olacaktır.

@font-face

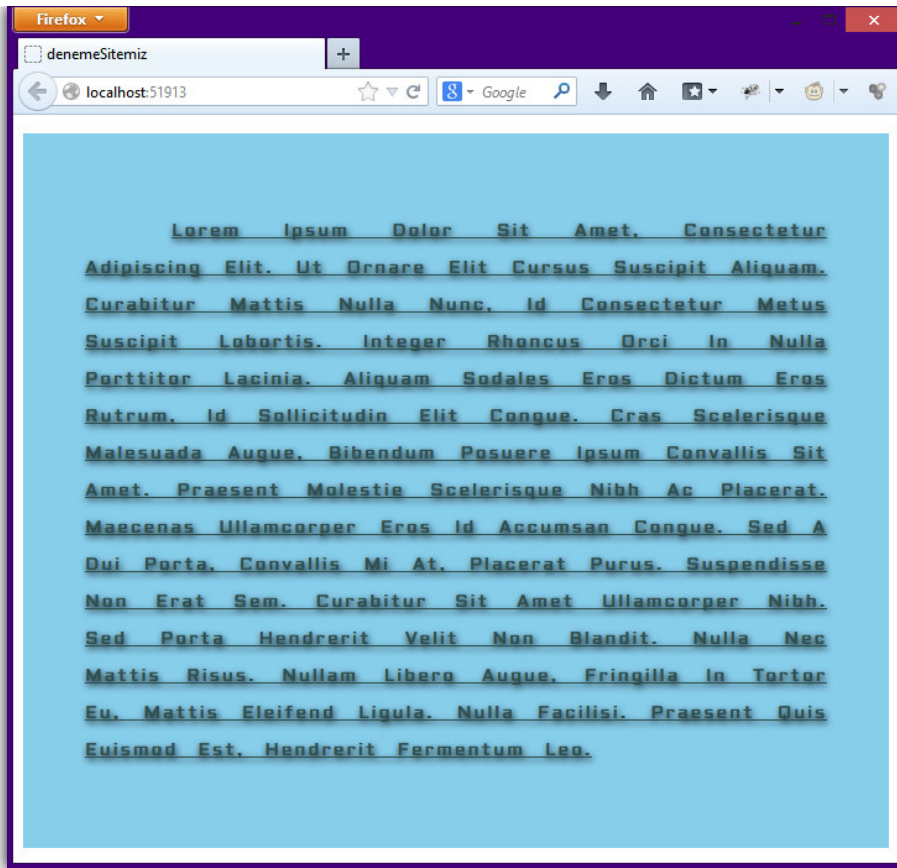
Kullanıcıların işletim sistemlerinde olmayan font'ların gösterilmesi için kullanılır. Örneğin; *Aldrich* isimli bir fontumuz olsun. İlk olarak font2web.com adresine giriyoruz. Daha sonra font'umuzu seçiyoruz ve *convert* ediyoruz. Yani dönüştürüyoruz. Masaüstümüze bir dosya inecek. Bu dosyayı açıyoruz. Dosyanın içindeki fonts klasöründeki tüm fontları projemizdeki font klasörüne ekliyoruz. Daha sonra fonts.css dosyasını açıyoruz. Bu dosyanın içindekileri kendi style.css dosyamıza ekliyoruz. Gerekli url düzenlemelerini yaptıktan sonra .deneme div'inin font-family'sini eklediğimiz fontun değeri olarak değiştiriyoruz.

```

1  .deneme {
2    width: 600px;
3    background-color: skyblue;
4    font-size: 14px;
5    padding: 50px;
6    margin-top: 15px;
7    color: rgba(47,79,79,1);
8    line-height: 30px;
9    letter-spacing: 2px;
10   text-align: justify;
11   text-decoration: underline;
12   text-indent: 70px;
13   text-shadow: 1px 2px 5px black;
14   text-transform: capitalize;
15   word-spacing: 10px;
16   font-family: 'Conv_Aldrich';
17 }
18
19 @font-face {
20   font-family: 'Conv_Aldrich';
21   src: url('/Content/font/Aldrich.eot');
22   src: local('@'), url('/Content/font/Aldrich.woff') format('woff'),
23        url('/Content/font/Aldrich.ttf') format('truetype'),
24        url('/Content/font/Aldrich.svg') format('svg');
25   font-weight: normal;
26   font-style: normal;
27 }

```

Bu şekilde yeni fontumuzu CSS dosyamıza ekledik.



Büyütmek için resme tıklayınız.

Ekran çıktımız bu şekilde olacaktır ve yeni fontumuzu kullanabileceğiz.

Bu dersin videosunu 1080p HD kalitesinde izleyebilirsiniz.

CSS3 - CSS'te Metin Biçimlendirme



Bu derslikte bu kadar arkadaşlar. Bir sonraki dersimizde CSS'te konumlandırmayı inceleyeceğiz.

Diğer derse geçmek için tıklayınız. >>>

TOP FONTS FOR EVERY PROJECT

