

CSS(Stil Şablonları)

Başlangıç için

Fatih HAYRİOĞLU



CSS(Stil Şablonları)

Başlangıç için

Fatih HAYRİOĞLU



Yazar Hakkında

Fatih Hayriođlu, 1978 Trabzon/Sürmene doğmuştur.

İlk, orta ve lise eğitimini İstanbul Zeytinburnu'nda tamamladıktan sonra Üniversiteyi Fırat Üniversitesi'nde okudu. Bölümü her ne kadar Çevre Mühendisliği olsa da hayatını web sayfası kodlaması yaparak geçirmektedir.

Web kodlama işine 1999 Ağustos ayında başladı. 2002-2007 yılları arasında Turkline'da çalışmıştır. Çalışma hayatına 2007'den itibaren Pixelplus Interactive şirketinde devam etmektedir.

Akbank, Akbank Yatırımcı , AmericanExpress, Axess, Beko , BonusCard.ro, Buyukkirmiziev, Exi26, EmlakveBen, Fiat, Grundig, Garanti Ödeme Noktaları, Maya, MSGSU(Mimar Sinan Güzel Sanatlar Üniversitesi), GNC, Koç.net, TEB-Focus, Tofaş, Turkcell, THY, VakıfBank vd. web sitelerinde emeđi geçmiştir ve bir kısmında hala geçmeye devam etmektedir.

Önsöz

İlk defa notlarımı bir yerde tutma ihtiyacı ile başlayan bir çalışmanın son noktasıdır bu kitap.

Web sayfası kodlarken bazı problemlerle karşılaşıyordum ve bu sorunlara çözüm bulup yoluma devam ediyordum. Belli bir süre sonra aynı sorunlarla karşılaştığımı gördüm ve karşılaştığım sorunları ve çözümlerini bir kenara not etmeye başladım. Bu notlar belli bir hacim kaplamaya başlayınca bunları bir sitede yayınlama gereği duydum. Ancak "Terzi kendi söküğünü dikemez" atasözünün doğruluğunu kanıtlarcasına bu siteyi yaklaşık 1,5 sene gecikmeden sonra Nisan 2006'da açtım ve notlarımı bir düzene sokup yayınlamaya başladım.

Notlarımı yayınlamamdaki etkenlerden biri CSS hakkında yeterli Türkçe kaynağın olmamasıydı. Daha sonra bu notları bir kitap haline getirmek istediysem de bu amacıma ulaşamadım ta ki şimdiye değin. Şimdi sitemdeki notları derleyip bir kitap haline getirmek nasip oldu. Kitabımı yazarken sitemdeki yazılarıma yazılan yorumları ve bana yazın kısmından bana gönderdiğiniz yorum ve önerileri dikkate aldım.

Bu kitabı normalde tek bir CSS kitabı olarak yazmayı düşündüm ancak notları kitap haline getirip birkaç şey daha ekleyince hacim bayağı arttı. Bu nedenle birden fazla kitap yazmanın daha mantıklı olacağını düşündüm. İlk kitap yani bu okuduğunuz kitap CSS'e başlangıç kitabıdır. Nasip olursa ikinci kitabıda derleyip en kısa zamanda yayınlamayı düşünüyorum.

Siteden farklı olarak özelliklerin ve önemli tanımlamaların Türkçe karşılıklarını yazdım. Bu konuda <http://www.belgeler.org/recs/css2/index.html#index-toc> sitesinden çok yardım aldım, kendilerine teşekkürü bir borç bilirim.

Bu kitap kimin için?

CSS'e başlangıç kitabı CSS'in teorik kısmını içerir. Tanımlamalar, özellikler ve CSS'e başlangıca ait bilgiler içerir.

Kitap içindeki konuları anlamak ve örnekleri uygulamak için genel web bilgisi ve HTML veya XHTML hakkında bilgi sahibi olunmalıdır. Bu konular hakkında birçok Türkçe ve İngilizce kaynaklar bulunmaktadır. Gerekli ve yeterli bilgi internet üzerinden edinilebilir.

Bu kitap gerek web programlamasına yeni başlayanlar ve gerekse konu hakkında bilgi sahibi olanlar için bir açıklama kitabıdır. CSS'i ve özelliklerini açıklayan bir kitap.

Kitabın yapısı

Kitap genel olarak iki ana kısımdan oluşmaktadır.

1. Kısım CSS'in Temelleri

- **CSS'e giriş:** CSS'in tanımı genel bilgiler ve avantajlarını içeriyor.
- **CSS'in yapısı:** CSS'in yapısını oluşturan elamanlar tek tek incelendi. Seçiciler, bildirim bloğu, gruplama, değerler ve yorumlar hakkında bilgi verildi.
- **Seçiciler:** CSS'deki önemi dikkate alınarak seçiciler için ayrı bir bölüm yazıldı. Tüm seçici türleri tanımları ve örnekleri ile anlatıldı.
- **Basamaklı yapı ve kalıtsallık:** CSS'in C(Cascading) harfinin ifade ettiği basamaklı yapı ve anlamı hakkında ayrıntılı bir bilgi verildi.

2. CSS özellikleri

- **Yazıtipi(font) özellikleri:** Metinleri oluşturan yazıtiplerinin CSS ile yönetilmesini sağlayan özellikleri içerir.
- **Metin(text) özellikleri:** Metinlere ait özellikler hakkında bilgi verildi.
- **Artalan(background):** Web sitelerindeki elemanların artalan özelliklerini içerir.
- **Listeleme(list) özellikleri:** Web sitelerindeki listelerin CSS ile yönetilmesini sağlayan özellikleri içerir.
- **Kutu modeli özellikleri:** CSS ile web sayfası kodlamanın önemli özelliklerini içeren kutu modeli ve özellikleri hakkında bilgi içerir.
- **Yüzen kutu ve konumlandırma özellikleri:** CSS ile web sayfası kodlamanın diğer önemli özellikleri olan float, clear ve position özellikleri hakkında bilgi içerir.
- **Tablo özellikleri:** CSS ile web sayfalarındaki tabloları yönetmek için kullanılan özellikleri içerir.
- **Kullanıcı arayüzü özellikleri:** CSS ile web tarayıcılarının belli özellikleri kontrol etmemize yarayan arayüz özelliklerini içerir.

Ayrıca Tüm CSS özelliklerinin referansı, kaynakça ve sözlük bölümü ek olarak sunuldu.

Kitabın düzeni?

Kitapta anlaşılabilirliği arttırmak ve görünümü güzelleştirmek için bazı işaretler ve şablonlar kullanılmıştır. Kitapta yer alan tüm şablonlar aşağıda sıralanmıştır.

Sırası ile bu şablon ve işaretleri incelersek:

1. Kodlar için mavi zeminli kalın yazılmış içerikler kullanıldı

```
p {  
font-style: italic;  
}  
h4 {  
font-style: oblique;  
}
```

2. Özelliklerin web tarayıcıları destekleme çizelgeleri mavi ve koyu mavi alanlar ile gösterildi. İnternet tarayıcılarının listesi çıkarıldı; İnternet Explorer(IE), Firefox(ff), Safari(S), Opera(O)

İnternet Explorer 3+ işaretinin anlamı belirtilen özelliğin Microsoft İnternet Explorer tarayıcısının 3. Sürümünden itibaren desteklendiğidir. Firefox, Safari ve Opera içinde aynı tanımla geçerlidir.

CSS1+ tanımının anlamı ise W3C'nin belirlenen özelliği hangi CSS sürümünde eklediğini belirtir.

Tarayıcı Uyumu:

İnternet Explorer 3+
Firefox 4+
Safari 1.3+
Opera 3.6+
CSS 1+

3. Özellik tanımları turkuaz zemin renkli alanlar ile belirtildi

Yapısı: **color: <değer>**
Aldığı Değerler : **<renk> | inherit**
Başlangıç değeri: **web tarayıcısı belirler**
Uygulanabilen elemanlar: **tüm elementler**
Kalıtsallık: **Var**

- Yapısı kısmında uygulanacak özelliğin yapısal özellikleri gösterilmiştir.
- Aldığı değerler alanı, özelliğin alabileceği değerleri gösterir. Değerler arasında ayraç(|) kullanılır.
- Başlangıç değeri alanı, özelliğe hiçbir tanım yapılmadan kendi aldığı değerdir.
- Uygulanabilen elemanlar alanı, özelliğin uygulanabileceği HTML elemanlarını

gösterir. Bu HTML elemanları belli gruplara ayrılır (blok seviyeli, satırıcı, yerleştirilen elemanlar gibi) bu gruplarda burada belirtilebilir.
-Kalıtsallık alanı, özelliğin kalıtsal özelliğine sahip olup olmayacağını gösterir.

4. Önemli not alanları için farklı bir şekil kullanıldı.

XML de Seçiciler HTML elementleri dışında yeni oluşturulan elementlerde olabilir.

Kaynak kodlar

Kitap içinde verilen örnekler özelliğın veya tanımın kavranması için verilen genel özelliklerdir.

Kaynak kodları web sitemden indirebilirsiniz. Konuların daha iyi anlaşılabilmesi için kodların incelenmesi önemli.

Hatalar

Kitabı hazırlarken her ne kadar hata yapmamaya çalışsam da muhakkak bir yerde bir sorun çıkacaktır.

Konu anlatımındaki hatalarım, yazım hatalarım ve diđer hatalarımı bana site üzerinden iletirseniz sevinirim.

Hatasız bir kitap olması temennisi ile.

Teşekkür

Bu kitabı yazarken devamlı beni oğlum yeter diye bilgisayar başından kaldırmaya çalışan biricik anneciğime.

İçindekiler

İçindekiler.....	1
1. CSS'E GİRİŞ	7
1.1. CSS'in tanımı ve genel bilgi	7
1.2. CSS'in Avantajları	7
1.2.1. Görünüm Avantajları.....	8
1.2.2. Kullanım Kolaylığı.....	8
1.2.3. Tasarım Tutarlılığı.....	9
1.2.4. Daha az Dosya Boyutları	9
1.2.5. Gelecek Vaat ediyor.....	9
1.3. CSS Editörleri.....	9
1.3.1. Metin Editörleri	9
1.3.1. CSS Editörleri	10
1.3.3. HTML ve Web Programlama Editörleri	10
1.4. CSS'i Web Dokümanlarına Ekleme	11
1.4.1. Kod içinde (In-line).....	11
1.4.2. <style> Elemanı Kullanılarak	11
1.4.3. Harici CSS Dosyası Kullanımı.....	11
1.4.4. @import İle Ekleme	12
1.5. Standart mod, garip mod ve Doctype Kullanımı	13
1.5.1. Web Sayfalarını Doğrulama "Onaylama"	13
1.5.2. Web tarayıcı mod'ları	14
1.5.3. DOCTYPE Geçişleri.....	14
2. CSS'in Yapısı	16
2.1. Seçiciler ve Bildirim Bloğu	16
2.2. Gruplama	18
2.3. Değerler	20
2.3.1. Anahtar sözcükler.....	20
2.3.2. Uzunluk Değerleri.....	21
2.3.3. Renk Değerleri	22
2.3.4. URL Değerleri	22
2.3.5. Dizgeler(Strings).....	23

2.4. CSS Yorumları	23
3. Seçiciler	25
3.1. Sınıf Seçicisi(Class Selector)	25
3.2. Tekil Seçiciler(Id Selector)	29
3.3. Sınıf(Çoğul) mı? Tekil mi?	30
3.4. (X)HTML Doküman Hiyerarşisini Anlamak	32
3.5. Torun Seçicileri (Descendant Selectors)	34
3.6. Çocuk Seçicileri(Direct Child Selectors)	35
3.7. Bitişik Kardeş Seçiciler(Next Sibling Selector)	37
3.8. Evrensel Seçiciler(Universal Selector)	39
3.9. Özellik Seçicileri(Attribute Selectors)	41
3.9.1. Basit Özellik Seçicisi	42
3.9.2. Özellik Değeri Seçicisi	43
3.9.3. Kısmi Özellik Değeri Seçicisi.....	45
3.9.4. Dikkate Değer Özellik Seçicisi.....	47
3.10. Sözde(Pseudo)-sınıfları ve Sözde(Pseudo)-elemanları	49
3.10.1. Sözde Sınıf Seçicileri	50
Sözde Link Sınıfı.....	50
Sözde Dinamik Sınıf Seçicileri.....	52
İlk Çocuk Elemanını seçmek	54
:lang sözde sınıfı	56
Sözde sınıfları birleştirmek	57
3.10.2. Sözde eleman seçicileri.....	59
first-letter (ilk harf)	59
first-line (ilk satır)	61
before ve after elemanları.....	62
4. Basamaklı Yapı ve Kalıtsallık	65
4.1. CSS'de Tanımlamalar ve Etkinlikleri(Specificity)	65
4.2. !important	70
4.3. Kalıtsallık(Inheritance).....	71
4.3.1. inherit (kalıtsallık)değeri.....	73
5. Yazıtipi(Font) Özellikleri	75
5.1. Renk (color) özelliği	75
5.2. Yazıtipi ailesi (font-family)	76

5.3. Yazıtipi boyutu (font-size)	78
5.4. Yazıtipi kalınlığı (font-weight)	80
5.5. Yazıtipi biçemi (font-style)	82
5.6. Yazıtipi farklılıkları (font-variant)	84
5.7. Yazıtipi (font)	85
6. Metin(Text) Özellikleri	88
6.1. Metin Girintileme (text-indent).....	88
6.2. Metin hizalama (text-align)	90
6.3. Dikey hizalama (vertical-align)	91
6.4. Satırlar arası yükseklik (line-height)	93
6.5. Sözcükler arası mesafe (word-spacing)	95
6.6. Harfler arası mesafe (letter-spacing)	97
6.7. Harf Büyüklüğü (text-transform)	99
6.8. Satırın altını, üstünü, ortasını çizme ve yanıp sönme (text-decoration) .	101
6.9. Metin gölgelendirme (text-shadow)	103
6.10. Boşlukların korunması (white-space).....	104
6.11. Metin yazma yönü(direction)	105
6.12. Çok yönlülük algoritması (unicode-bidi) özelliği	107
7. Artaalan (Background) Özellikleri	109
7.1. Artaalan rengi (background-color)	109
7.2. Artaalan resmi (background-image)	111
7.4. Artaalan resim bağıllığı (background-attachment)	114
7.5. Artaalan resmi konumu (background-position).....	116
7.6. Artaalan (background)	118
8. Listeleme Özellikleri.....	121
8.1. Liste imleyici tipi (list-style-type)	121
8.2. Liste imleyici resmi (list-style-image).....	124
8.3. Liste imleyici konumu (list-style-position).....	125
8.4. Liste biçemi (list-style)	127
9. Kutu Modeli Özellikleri.....	129
9.1. Genel Tanımlar	130
9.2. Kenarlık Çizgisi(Border) Özellikleri.....	131
9.2.1. Kenar çizgisi biçemi (border-style).....	131

9.2.2. border-top-style, border-right-style, border-bottom-style, border-left-style	133
9.2.3. Kenar çizgisi genişliği (border-width).....	134
9.2.4. border-top-width, border-right-width, border-bottom-width, border-left-width	136
9.2.5. Kenar Çizgisi Rengi (border-color)	138
9.2.6. border-top-color, border-right-color, border-bottom-color, border-left-color.....	139
9.2.7. Kenar Çizgisi (border)	141
9.3. Kenar Dış Boşluğu(Margin) Özellikleri	144
9.3.1. Üst Kenar dış boşluğu (margin-top)	144
9.3.2. Sağ kenar dış boşluğu (margin-right)	146
9.3.3. Alt kenar dış boşluğu (margin-bottom)	147
9.3.4. Sol kenar dış boşluğu (margin-left)	149
9.3.5. Kenar dış boşluğu (margin).....	151
9.4. Kenar iç boşluğu (Padding) özellikleri.....	155
9.4.1. Üst kenar içi boşluğu(padding-top)	155
9.4.2. Sağ kenar içi boşluğu(padding-right).....	157
9.4.3. Alt kenar içi boşluğu (padding-bottom).....	158
9.4.4. Sol kenar içi boşluğu (padding-left).....	160
9.4.5. Kenar içi boşluğu (padding)	162
9.5. Boyut Tanımlama Özellikleri	165
9.5.1. Genişlik (width).....	165
9.5.2. Yükseklik (height)	166
9.5.3. Asgari genişlik ve yükseklikler (min-width, min-height).....	168
9.5.4. Azami genişlik ve yükseklik (max-width, max-height).....	170
9.6. Görsel Sonuç Özellikleri	174
9.6.1. Taşma (Overflow).....	174
9.6.2. Kırpma (clip)	182
9.6.3. Görünürlük (visibility)	185
10. Yüzen kutu ve Konumlandırma Özellikleri.....	190
10.1. Yüzen kutu özellikleri.....	190
10.1.1. float özelliği	190
10.1.2. clear özeliği	193
10.2. Konumlandırma Özellikleri	195

10.2.1. Konumlandırma özelliği (position)	196
10.2.2. Üst, sağ, alt ve sol (top, right, bottom, left)	196
Göreceli Konumlandırma(position:relative)	197
Mutlak(Absolute) Konumlandırma	197
Sabit Konumlandırma (position:fixed).....	199
10.3. z-index özelliği	203
10.4. display özelliği	209
11. Tablo Özellikleri.....	211
11.1. Başlık Konumu Özelliği (table-layout).....	211
11.2. Tablo planı özelliği (caption-side)	214
11.3. Hücre Boşluklarını ayarlama (border-collapse).....	215
11.4. Hücre dışı boşluklarını ayarlama (border-spacing)	217
11.5. Boş hücre ayarları (empty-cells).....	219
12. Kullanıcı Arayüzü Özellikleri	222
12.1. Fare imleci özellikleri tanımlama (cursor).....	222
12.2. Dış hat çizgisi özellikleri(outline)	223
12.2.1 Dış hat çizgisi biçemi (outline-style)	224
12.2.2. Dış hat çizgisi genişliği (outline-width)	225
12.2.3. Dış hat çizgisi rengi (outline-color)	225
12.2.4. Dış hat çizgisi (outline).....	226
12.3. Sistem yazıtipi ve rengi özellikleri	227
Ek – A Tüm CSS özellikleri	228
Ek – B Kaynakça	241
Ek – C Sözlük	243
İndex Tablosu	248

1. CSS'E GİRİŞ

1.1. CSS'in tanımı ve genel bilgi

CSS (Cascading Style Sheets – Stil Şablonları), Web dokümanlarına stil eklemek için kullanılan basit ve güçlü bir dildir.

Web sayfalarının yapılarını düşündüğümüzde 3 kısma ayırabiliriz. İçerik kısmı, biçimlendirme kısmı ve dinamik kısım.

İçerik kısmı, HTML kodlarını içeren sayfa asıl içeriğini oluşturan birimleri oluşturma kısmıdır. Bu kısım genelde metinlerden oluşsa da, resim, animasyon ve video gibi elemanları da içerir.

Biçimlendirme kısmı, içerik kısmını kullanıcıya nasıl görüntüleneceğini gösteren kısımdır. Sayfanın biçimini belirler. Kullanıcıya daha okunaklı ve kullanışlı sayfa sunmak için gerekli elemanları içerir.

Dinamik kısım, Javascript ile yapılan dinamik işlemleri içerir. Etkileşimli sayfalar oluşturmak için kullanılır.

Eskiden ki hala bu şekilde kodlama yapanlar mevcut. İçerik kısmı ve biçimlendirme kısmı için iç içe kodlama yapardık, örneğin yazıtipi tanımı için etiketini kullanırdık. Eski kodlama yöntemi ile zaman, hız ve erişilebilirlik açısından büyük kayıplarımız vardı.

CSS biçimlendirme kısmının yönetimini sağlamak için kullanılır. İçerik kısmı ile biçimlendirme kısmının ayrılmasını sağlayan CSS bize birçok avantajlar sağlar. Örneği 10.000 sayfası olan bir siteyi tek css dokümanı yardımı ile biçimlendirmemizi sağlar.

CSS 1996'da [W3C](http://www.w3c.org) tarafından duyuruldu. Son olarak(Ocak 2003'den beri) CSS2.1 sürümü yürürlüktedir. CSS3.0'da çalışmaları devam etmektedir. CSS3.0 için ayrıntılı bilgiye <http://www.w3c.org/Style/CSS/current-work> adresinden ulaşabilirsiniz.

Tablosuz Web Sayfası kodlama tekniğini kullanmayanlar CSS'in önemiyetini tam olarak anlayamayacaktır.

1.2. CSS'in Avantajları

CSS kullanımının önemli avantajlarından bazıları:

1.2.1. Görünüm Avantajları

CSS HTML'e göre birçok stil özelliğine sahiptir. CSS'in sayfa içeriği öğelerinin sayfa görünümü öğelerinden ayrılması için geliştirildiğini düşünürsek avantajı baştan anlaşılmış olur.

<h1>CSS'e Giriş</h1>

HTML'de bu elemanın(h1) kalın, altı çizili, Artalan rengi kırmızı olarak atama gibi çeşitli özellikler verebiliriz. Ancak bu özellikleri atamak içinde ayrı HTML elemanları kullanmak zorundayız (, <u>), ancak CSS'de bunu tek bir elemanla yapabiliriz ve ayrıca daha fazla stil özellikleri de atayabiliriz.(örn: kenarlık, rollover vs stillerini ekleyebiliriz.)

```
h1 {
color: white;
font: italic 11px Arial, serif;
text-decoration: underline;
background: yellow url(titlebg.gif) repeat-x;
border: 1px solid red;
margin-bottom: 0;
padding: 5px;}
```

1.2.2. Kullanım Kolaylığı

HTML'de her etikete artı özellikler eklemek için başka bir etiket ve/veya özellik eklememiz gerekmektedir. Bu işlemi geniş çaplı bir sitede yaptığımızı düşünürseniz çok büyük zaman kaybı ve uğraş gerektiğini göreceksiniz.

<h1>Başlık</h1>

Bunun gibi onlarca veya yüzlerce başlığınız olduğunu düşünün, gerçekten çok zor. CSS'de aynı işlem için

```
h1 {color: blue}
```

Kodunu yazmamız yeterli. Bu elemanın özelliklerinde değişiklik yapmak istediğimizde sadece burada değiştirerek tüm sitede bu elemanın özelliklerini değiştirmiş olacağız. Ayrıca bu işlemi sadece bu etiket için değil diğer etiketlere de uygulayabiliriz.

```
h1, h2 {color: blue}
```

Ayrıca tek bir CSS dosyası ile sitenin tamamını yönetmek web kodlamacıları için çok büyük kolaylıktır. Sadece bir dosyada değişiklik yaparak tüm sitemizi ara yüzünü yönetebiliriz isterse yüzlerce sayfa olsun.

1.2.3. Tasarım Tutarlılığı

Tek CSS dosyası ile tüm sitenizi yönetebilirsiniz, bu özellik sayesinde siteniz tutarlılık kazanacaktır. Web sayfanızdaki ilk sayfadan son sayfaya kadar tutarlılığınızı koruyarak ziyaretçinize düzenli bir içerik sunmuş olacak ve sitenizin kendine has özelliklerini ziyaretçiye benimsetmiş olacaksınız. Sayfalarımız hızlı yüklenecektir, çünkü aynı elemanları diğer sayfalarda tekrar yüklemeyecek ve bu ziyaretçiye zaman kazandıracaktır.

1.2.4. Daha az Dosya Boyutları

CSS yardımı ile kodlanmış HTML'in normal HTML kodlamasına göre %50'ye varan performans ve hız artışları olduğu belirlenmiştir. CSS HTML'de kullanılan görünüm özelliklerini(iskeleti oluşturan tablolar, boş resimler, süsleyici resimler, yazı tipleri, renkler, genişlikler, yükseklikler ve Artalan resimleri) azaltacağı için dokümanlarınızın boyutunu ciddi oranda azaltacaktır.

1.2.5. Gelecek Vaat ediyor

HTML giderek işlevselliğini kaybedeceği ve XML ve dolayısı ile XHTML'in işlevselliğini artıracığı düşünülürse, CSS gibi tümleşik bir standardın önemi daha çok anlaşılıyor.

XHTML ile daha temiz kodlar üretilecek, farklı platformlara(pda, cep telefonu vs.) uyum sayesinde bilgi dolaşımı kolaylaşacak, sayfalarımız arama motorları programları tarafından daha anlaşılır olacaktır. CSS, XHTML teknolojisi ile tümleşik olarak çalışarak bizlere gelecek vaat ediyor.

1.3. CSS Editörleri

HTML yazarken ihtiyacımız olduğu gibi CSS yazarken de bir yazım aracına, editöre ihtiyacımız olacaktır. Burada sizlere basit editörlerden profesyonel editörlere bir liste sunacağız. Tüm editörleri burada sunamayacağımıza göre belli başlılarına yazmayı uygun gördük.

1.3.1. Metin Editörleri

Eğer ben bu iş için ayrıca bir program kurmak istemiyorum. Zaten o kadar da kullanmıyorum. Ayrıca böyle hızlı oluyor diyorsanız. İşte size CSS yazabileceğiniz editörler.

- Windows için NotePad veya WordPad
- MacOS için TextEdit
- Linux için vi, vim veya emacs

Ancak şunu da belirtmeliyim ki bu editörlerden fazla bir özellik ve işlevsellik beklemeyin.

1.3.1. CSS Editörleri

Bu editörler sadece css yazmak için üretilen programlardır. Başlangıç seviyesi için kullanışlıdır. Kod yazmanız için size kolaylık sağlayan özellikleri mevcuttur. Örneğin kodlarınız renklendirme, kodlara hızlı erişim vb.

- Windows için **TopStyle** <http://www.bradsoft.com/topstyle/>
- Windows ve Mac OS X için **Style Master** http://www.westciv.com/style_master/
- Mac OS X **CSSEdit** <http://www.macrabbit.com/cssedit/>

1.3.3. HTML ve Web Programlama Editörleri

İşin aslı birçok kullanıcı CSS yazmak için bu editörleri tercih etmektedir. Web kodlamacıları için gereken tüm araçları içermeleri, CSS kodlama da yardımcı özellikleri, kod düzeltme gibi birçok özelliği barındırmalarından dolayı çoğu web kodlamacısı bu editörleri tercih etmektedir.

- Windows için **Microsoft FrontPage** <http://www.microsoft.com/frontpage/>
- Windows ve Mac OS X **Macromedia Dreamweaver** <http://www.macromedia.com/software/dreamweaver/>
- Windows için **CoffeeCup HTML Editor** <http://www.coffeecup.com/editor/>
- Mac OS X için **BEdit** <http://www.barebones.com/products/bbedit/>
- Son olarak Microsoft'un yeni gözdesi **Microsoft® Expression** <http://www.microsoft.com/products/expression/>

Ben Macromedia Dreamweaver kullanıyorum: D

Ayrıca Crimson Editor, Notepad 2, Aptana, Notepad++, e-texteditor diğer öneriler.

FireBug: Editörlerden daha önemli bir aracı sizlere şiddetle öneriyorum. Firefox, Safari ve Internet Explorer'a eklenebilen çok mükemmel bir araç FireBug(<http://www.getfirebug.com/>). Araç size sitelerinize tarayıcı üzerinden tam hâkimiyet sağlıyor. Anlık olarak değişiklik yapıp değişikliğinizi sonucu görebiliyorsunuz. Gelişmiş Javascript hata ayıklama özelliği, site bileşenlerinin performansına etkisi ve birçok özelliği ile web kodlamacısı olacak her kesin mutlaka kullanması gereken bir araç.

1.4. CSS'i Web Dokümanlarına Ekleme

CSS'i (X)HTML sayfalarımıza eklemenin 4 yöntemi vardır.

1.4.1. Kod içinde (In-line)

Direk olarak (X)HTML elemanın içine **style** özelliği kullanılarak uygulamak.

```
<div style="color:red">Deneme yazımız</div>
```

Tüm CSS komutlarını kodların içine direk uygulamak önerilen bir kodlama şekli değildir. Ancak özel durumlarda kullanılabilir.

1.4.2. <style> Elemanı Kullanılarak

<head> kısmında <style> elemanı içinde CSS kodumuzu yazarak uygulamak.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'i Uygulamak </title>
<style type="text/css">
div{
color:red;
}
</style>
</head>
```

Birinci yöntem göre avantajı (X)HTML kod ile CSS bir birinden ayrıştırılmış olmasıdır.

1.4.3. Harici CSS Dosyası Kullanımı

Bu metot da CSS kodlarımızı **.css** uzantılı bir dosyaya kaydederiz.
ornek.css

```
p {
color: red;
}
a {
color: blue;
}
```

Daha sonra bu kodu gereken sayfalarımıza uygularız.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'i Uygulamak</title>
<link rel="stylesheet" type="text/css" href="ornek.css" />
</head>
```

Bu yöntemin diğerlerine göre en büyük avantajı bir kere yazılan kod lazım olan tüm sayfalara eklenebilmesidir. Bu sayede harici eklenen css kodu bir kere yüklendikten sonra diğer kullandığımız sayfalarda tekrar yüklenemeyerek bize hız kazandıracaktır.

1.4.4. @import ile Ekleme

3üncü yöntem ile kullanımı benzerdir.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'i Uygulamak</title>
<style type="text/css">
@import "ornek.css";
</style>
</head>
```

Bu yöntemle eklenen harici css dosyası eski web tarayıcıları tarafından görüntülenemeyecektir.(Örn: NN4)

@import ile eklenen kod link ile eklenen kodlamadan önce yorumlanır web tarayıcıları tarafından. Ayrıca CSS dosyalarımızda @import özelliğini kullanarak devamlı kullandığımız kodları css dosyamıza harici olarak ekleyebiliriz böylelikle tekrarlardan kurtulmuş oluruz.

Sonuç olarak burada dört adet CSS uygulama metodu gördük ancak bu metotlardan dördüncüsü avantajları bakımından önerilen bir yöntemdir.

Ayrıca içeriği büyük olan sitelerde css kodunun parçalara ayrılması ve kullanılan sayfa CSS'inde hangi parçalar gerekli ise onların import edilmesi önerilir. Bu sayfada kodun bir kısmında yaptığımız değişiklik için tüm css kodu incelenip değiştirilmesi gerekmez ve kod yönetimi kolaylaşır.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'i Uygulamak</title>
<style type="text/css">
@import "urunler.css";
</style>
</head>
```

Ürünler bölümü için bir css dosyası ekleyelim sonra parçalara ayırdığımız css kodlarının ürünlere lazım olanlarını **urunler.css** içine ekleyelim.

urunler.css

```
@import url(/css/iskelet.css)
@import url(/css/fontlar.css);
@import url(/css/renkler.css);
@import url(/css/urunlereozel.css);
```

1.5. Standart mod, garip mod ve Doctype Kullanımı

DOCTYPE, HTML veya (X)HTML dokümanımızın tipini göstermek için kullandığımız bir koddur.

DOCTYPE bildirimi (X)HTML kodunun başına bir veya iki satır olarak eklenir. Genel kullanımı:

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

Burada dokümanımızın XHTML 1.0 sürümü dikkate alınarak yazıldığı ve kullanılan bu kuralların linki verilmiştir. Linkin kullanılmadığı şekilde vardır.

Bu tanımlamadan sonra web tarayıcıları bu kurallara göre sayfayı yorumlayacak ve buna göre bir görünümü kullanıcıya sunacaktır. Web tarayıcıları kullanılan Doküman tip tanımlamasına(DTD-document type definition)göre sayfayı analiz eder.

1.5.1. Web Sayfalarını Doğrulama "Onaylama"

Güzel bir kodlama, (X)HTML dokümanın onaylanmış halidir. Web sayfalarımız doğrulamak için genelde <http://validator.w3.org/> adresi

kullanılırız. Ancak birçok web sayfası hazırlama editörü bu aracı içinde barındırır. Web sayfalarımız doğrulattığımızda sonuç olarak bize sayfamızda bir sorun varsa bunu ve niçin olduğunu belirtir.

Doğrulama işlemi önemlidir çünkü bizim kodlarımızı doğru yazdığımızı kontrol eder. Kodlarınızı belirli aralıklarla doğrulmayı unutmayın. Şunu unutmayalım ki web sayfasını doğrulamamız bunu en mükemmel kod olduğu anlamına gelmez. Ayrıntı için [tıklayınız](#).

1.5.2. Web tarayıcı mod'ları

Web tarayıcı üreticileri geriye dönük uyumluluğu sağlamak için standartlara uygun tarayıcılar çıkardılar. Bunu başardılar, bu işlem için iki mod ürettiler: **standart mod**(standards mode) ve **garip mod**(Quirks mode). Standart mod sayfa ayrıntılı olarak ve en iyi şekilde yorumlayacaktır. Garip mod da ise eski moda tarayıcılardaki gibi daha duyarsız davranacaklardır. Örneğin IE4 ve NN4 gibi.

Bu iki mod arasındaki farkı görebilmek için Windows üzerindeki IE'de kutu modeline bakmalıyız. IE6 başlangıçta Standart Mod da [Kutu Modeli](#) doğru görünürken Garip Mod da ise sorun çıkarır. IE5 ve aşağısında bu sorun vardır, ayrıca Opera 7 ve üstü Garip Mod da IE gibi davranır. Bunun dışında ufak tefek sorunlar da vardır. Hekzedesimal renk tanımında # kullanımına ihtiyaç duymama, CSS'de değer kullanılmadığında birimi piksel olarak tanınması vb. sorunlar olarak sıralayabiliriz.

Firefox ve Safaride bir üçüncü mod vardır, ancak Standart Mod'dan çok önemli bir farkı yoktur.

1.5.3. DOCTYPE Geçişleri

Web tarayıcıları DOCTYPE bildirimini ve kullanılan DTD'ye göre tarama metodunu belirler. Bir dokümanda DOCTYPE tanımı yapılsa bu sayfa Standart Mod da, yapılmazsa Garip Mod da yorumlanacaktır bu geçişe **DOCTYPE Geçışı**(DOCTYPE switching) adı verilir.

- XHTML dokümanında tam bir DOCTYPE tanımı yapıldı ise web tarayıcısı taramayı Standart Mod da yapacaktır.
- HTML 4.01 dokümanı için DOCTYPE ve strict DTD tanımı yapıldığında sayfa web tarayıcısı tarafından Standart Mod da yorumlanacaktır.
- DOCTYPE ve transtional DTD URI tanımlı halin de Standart Mod da yorumlanacaktır.
- DOCTYPE ve transtional DTD URI'siz ise Garip Mod da yorumlanacaktır.
- Kötü formatlanan veya DOCTYPE kullanılmayan dokümanlar HTML ve XHTML tarafında Garip Mod da yorumlanacaklardır.

Web tarayıcılarının DOCTYPE tiplerine göre farklılıklarını görmek için <http://www.ericmeyeroncss.com/bonus/render-mode.html> bir göz atın.

Çok uğraşıp en iyi ve en doğru CSS kodunu yazdıktan sonra yanlış DOCTYPE kullandığınızda sayfanız Garip Mod da yorumlanacaktır ve bu size birçok sorun yaşatacaktır. Sitenizde tam DOCTYPE bildirimini ve strict DTD kullanmanız çok önemlidir.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

Bu kullanım sayfanın **HTML 4.01** kurallarının geçerli olduğu bir HTML dosyası olduğunu gösterir.

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

Bu kullanım sayfanın **hem HTML hem de XHTML** kurallarının geçerli olduğunu bir XHTML dosyası olduğunu gösterir.

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

Bu kullanım sayfanın **sadece XHTML1.0** kurallarının geçerli olduğu bir XHTML sayfası olduğunu gösterir. Bu kodlama yapıldığında tüm XHTML kurallarına harfiyen uyulmalıdır. Örneğin kodların hepsinin küçük harfle yazılması gerekir.

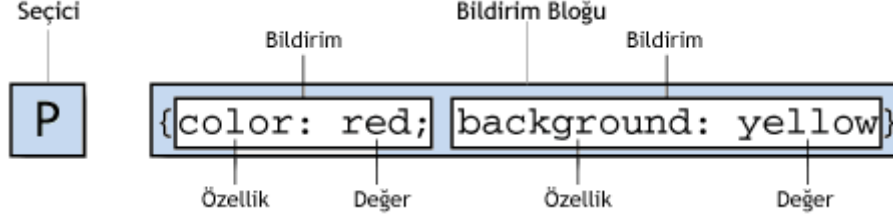
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

Bu kullanım sayfanın bir **XHTML çerçeve(frame)** sayfası olduğunu gösterir. Kullanımı azaldı.

Biz burada sizlere DOCTYPE tanımlamanın nasıl olduğunu ve önemini anlatmaya çalıştık. Birçok HTML editörü DOCTYPE tanımını otomatik eklemektedir.

2. CSS'in Yapısı

CSS'in yapısı iki ana kısımdan oluşur. Seçiciler(Selector) ve Bildirim Bloğu(Declaration Block). Bildirim Bloğuda ikiye ayrılır. Özellik(Property) ve Değer(Value).



Şekil2.1 CSS'in yapısı

2.1. Seçiciler ve Bildirim Bloğu

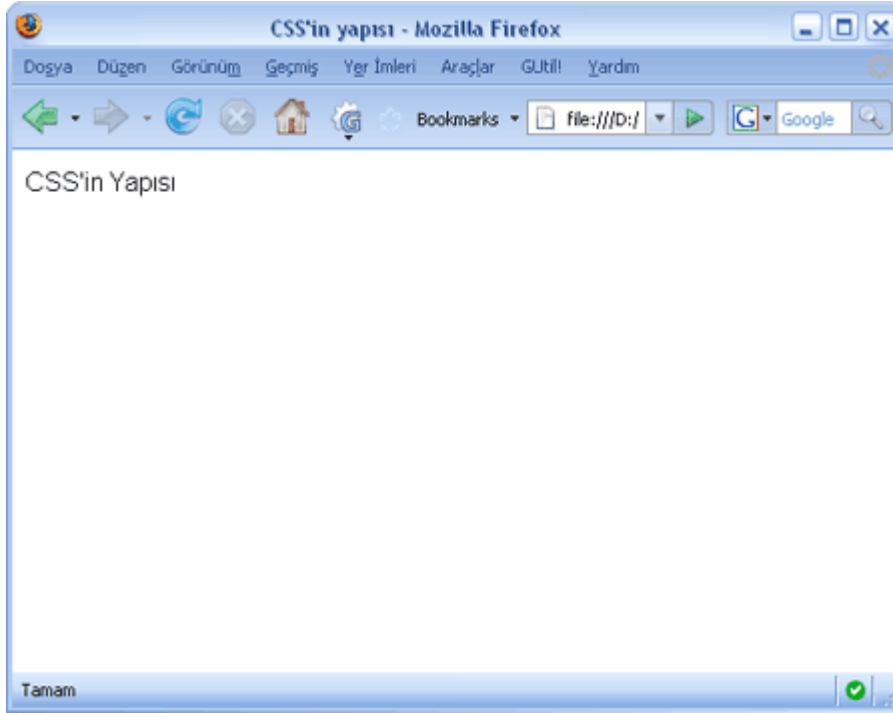
Tüm HTML elemanları(body, p vd.) potansiyel seçicilerdir. Seçiciler ismini buradan alır, seçilen HTML elemanı anlamındadır.

XML de Seçiciler HTML elemanları dışında yeni oluşturulan elemanlarda olabilir.

Bildirim bloğu süslü parantezle "{ }" açılır ve kapanır. Bildirimler arasında "; " noktalı virgül kullanılır. Özellik ve değerler birbirinden " : " iki nokta üst üste ile ayrılır.

Örnek : 2-1

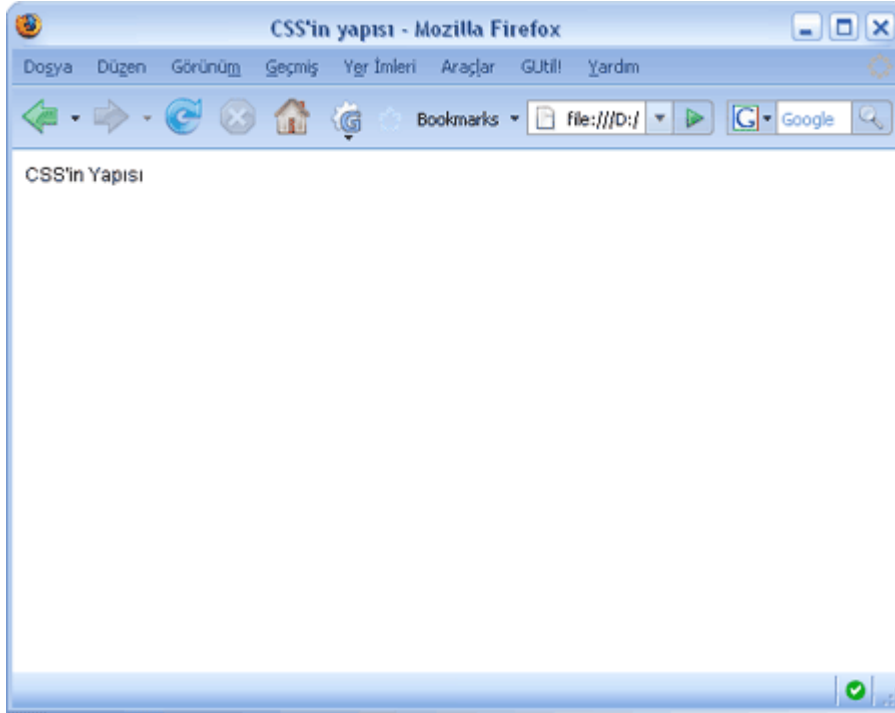
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
h1 {
    font: medium Arial;
}
</style>
</head>
<body>
<h1>CSS'in Yapısı</h1>
</body>
</html>
```



Şeklinde arada boşluk verilerek de bildirim yapılabilir. Burada ilk tanımlama yazı tipinin boyutunu ikinci tanımlama ise yazı tipinin ismini gösterir. İleride bu konuya daha ayrıntılı gireceğiz.(Bkz. Yazıtipi Özellikleri ve CSS'de Kısaltmalar) Ayrıca sadece yazıtipi etiketine özel " / " kullanımı vardır örnek aşağıda:

Örnek: 2-2

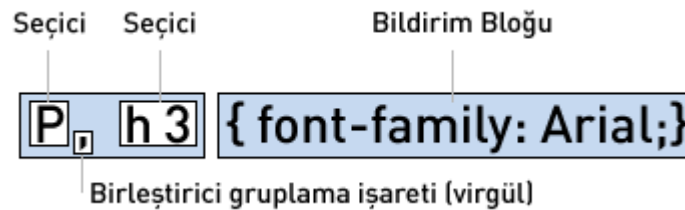
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
h1 {
    font: 12px/14px Arial;
}
</style>
</head>
<body>
<h1>CSS'in Yapısı</h1>
</body>
</html>
```



Burada " / " Seçicinin yazı tipi boyutunu ve satır yüksekliğini gösterir. (yazıtipi boyutu/satır yüksekliği)

2.2. Gruplama

Yukarıda hep bir seçiciyi sadece bir HTML elemanına atamayı gördük, Birden fazla HTML elemanına de atama yapabiliriz, buna **gruplama** denir. Gruplama seçicilerde yapıldığı gibi bildirimlerde de yapılabilir.



Şekil2.2 Gruplama Yapısı

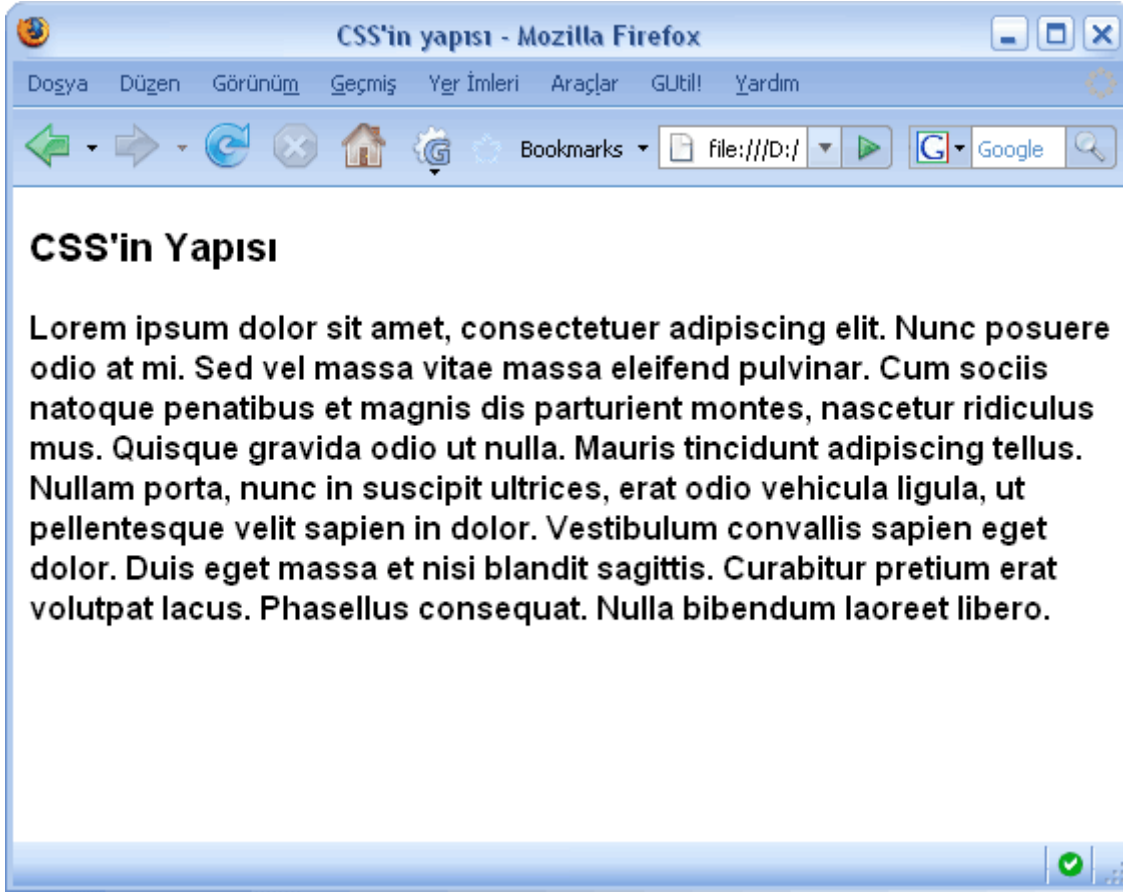
Burada HTML dokümanı içindeki paragraflarda(p) ve üçüncü seviye başlıklarda(h3) yazı tipinin **Arial** olacağını tek bildirim ile belirttik. Gruplama yapılan seçicileri ayırmak için " , " virgül kullanılır. Sınırsız sayıda seçici gruplanabilir. Gruplama bize büyük kolaylıklar sağlar.

Bildirimlerimizi de gruplayabiliriz, bununla ilgili yukarıda örnekler mevcut. Bir veya daha fazla seçiciye bir den fazla bildirim ekleyebiliriz.

Örnek: 2-3

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

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
p, h3
{
    font-family: Arial;
    font-size: 2;
    font-weight: bold;
}
</style>
</head>
<body>
<h3>CSS'in Yapısı</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc posuere
odio at mi. Sed vel massa vitae massa eleifend pulvinar. Cum sociis natoque
penatibus et magnis dis parturient montes, nascetur ridiculus mus. Quisque
gravida odio ut nulla. Mauris tincidunt adipiscing tellus. Nullam porta, nunc
in suscipit ultrices, erat odio vehicula ligula, ut pellentesque velit sapien in
dolor. Vestibulum convallis sapien eget dolor. Duis eget massa et nisi bland
sagittis. Curabitur pretium erat volutpat lacus. Phasellus consequat. Nulla
bibendum laoreet libero. </p>
</body>
</html>
```



2.3. Değerler

Değerler genelde mesafe ve renkleri tanımlamak için kullanılır.

Değerleri dört gruba ayıra biliriz. **Uzunluk** değerleri, **yüzde** değerleri, **renk** değerleri ve **URL** değerleri

2.3.1. Anahtar sözcükler

Birçok CSS özelliği anahtar sözcük değeri alabilir. Anahtar sözcükleri tırnaksız olarak atanır. Tırnaklı atamalar hatalıdır.

```
p{
color: red;
}
```

Doğru

```
p{
color: "red";
}
```

Hatalı bir tanımlamadır.

2.3.2. Uzunluk Değerleri

CSS'de tanımlanan 8 uzunluk değeri vardır. Bunların üç tanesi göreceli(relative) uzunluklar ve kalan beş tanesi kesin(absolute)uzunluklardır.

2.3.2.1. Göreceli Uzunluklar

Bu değerler diğer uzunluk değerlerine göre görecelidir. Sayfalarımızın ölçeğinin değiştiği ve çıktısının alındığı durumlarda bu uzunluk değerlerini kullanmamız bize avantaj sağlayacaktır.

em: Elemanın yazıtıpının yüksekliğidir. Mesela font-size değeri 14 px olarak atanmış ise 1em 14piksel eşit demektir.

ex: Elemanın "x" harfinin yüksekliğidir. Atanmış olan yazıtıpının küçük "x" değeri yüksekliğidir.

px: piksel değeri. Eğer monitörünüze yeterince yakından bakarsanız, çok küçük kutulardan oluşmuş bir ızgaraya benzer. Buradaki her kutu bir pikseldir. Bu da her monitöre göre farklı değer demektir.

2.3.2.2. Kesin Uzunluklar

Bu uzunluklar gerçek hayatta kullanılan birimlerdir.

in: inç. 1 inch=2,54 cm'dir. Örn: line-height:0.5in

cm: Santimetre. Bizim gerçek hayatta kullandığımız santimetre değeridir. Örnek: margin:2cm

mm: Milimetre. Bizim gerçek hayatta kullandığımız milimetre değeridir. Örnek: letter-spacing:1mm

pt: Point. Standart baskı birimidir. (1pt = 1/72in) Örn: font-size:14pt

pc: Pika. Bir inç'in 1/6 eşit olan bir baskı ölçü birimidir. Bir pika 12 noktaya bölünür. Örnek: font-size:2pc

2.3.2.3. Yüzde Değerleri

Yüzde değerleri göreceli değerlerdir. Yüzde değerleri uygulandığı nesnenin boyutuna bağlı hareket eder.

```
/* elemanın font-size değerinin %150 si dir */
h4 { line-height: 150% }
/* satır genişliğinin 10% nu */
p { text-indent: 10% }
```

2.3.3. Renk Değerleri

CSS'de renk değerleri isim olarak ve RGB değeri olarak tanımlanabilir. RGB tanımlamasıda dört çeşittir.

2.3.3.1. Renk İsimleri

Renklerin İngilizce isimleri kullanılarak tanımlama yapılabilir.

```
p {color: black} /* renk tanımlı siyah olarak yapıyor*/
h1{ background-color: red;} /* h1 başlığının ardalanı kırmızı tanımlanıyor */
```

2.3.3.2. RGB Renk Değerleri

CSS'de renk tanımlamak için genelde kullanılan yöntem budur. RGB tanımlamanında dört çeşidi vardır.

#RRGGBB : Burada, **RR**, **GG**, ve **BB**, sırasıyla kırmızı (red), yeşil (green) ve mavi (blue) renklerin toplam renk içindeki yoğunluğunu gösteren heksadesimal sayılardır. Örnek: color:#ff0000; (kırmızı)

#RGB: Yukarıdaki yöntemin kısaltmasıdır. Aynı değerleri taşıyan grup değerleri birleştirilir. Örnek: color:f00; (kırmızı)

rgb(R,G,B): Bu RGB renklerinin ondalık sayma düzeninde 0 ile 255 arasında bir tanımı vardır. Buna göre tanımlama yapılabilir. Örnek h1 {color: rgb(191,127,127);}

rgb(R%,G%,B%): Bu RGB renklerinin yüzdesel olarak %0 ile %100 arasında bir tanımı vardır. Buna göre tanımlama yapılır. Örnek: h1 {color: rgb(75%,0%,0%);}

CSS3 ile birlikte RGBA (RGB with Alpha channel) renk tanımında eklenmiştir. Renk tanımı ile birlikte alfa kanal saydamlık değeri de atanabilecek.

2.3.4. URL Değerleri

CSS veri dosyası ve kaynağın yerini belirlemek için URI(Universal Resource Indicator – Evrensel Kaynak) kullanır. CSS'de URI genel olarak iki yerde kullanılır stil şablonlarını eklemek veya Artalan resmi eklemek için. URL tanımlaması için aşağıdaki gibi bir tanımlama yapılır.

```
url("http://www.fatihhayrioglu.com/images/kucuk.gif")
```

Burada tırnaksız ve tek tırnaklı kullanımları da standarda uygundur.

2.3.5. Dizgeler(Strings)

Dizgeler çift ya da tek tırnak içinde yer alır.

```
div::before {
  content: "Dikkat";
  color: red;
}
```

Dizge içinde çift tırnak kullanmak için bölüm(\) işareti ile birlikte kullanılmalıdır.

```
div::before {
  content: "Dikkat \"burada\" köpek var ";
  color: red;
}
```

Dizgeler satır sonu elemanları içeremezler. Eğer satır sonu elemanı kullanacak isek öncesinde (\a) kullanmalıyız.

2.4. CSS Yorumları

CSS kodlarımızı yazdığımızda kod yapımız, kullandığımız teknikleri ve diğer özel uygulamalarımız için yorum satırları eklemeliyiz. Çünkü daha sonra kodumuzu incelerken bu yorumlar bize yardımcı olacaktır. Ayrıca bir proje üzerinde birden fazla kişi çalışıyorsa diğer çalışanlar için bu kodlar önemli bilgiler içerecektir. Kodlarınıza yorum satırı ekleyerek daha anlaşılır ve erişilebilir yaparız.

CSS'de yorum eklemek çok kolaydır.

Css dokümanları içindeki resim yolu css dokümanının konumuna göre verilir. Eklendiği (x)html sayfasının konumuna göre değil.

```
/* Genel Stiller */
body {
  font-size: 67.5%; /* genel yazıtipi tanımı */
}
/*
Birden fazla satır içeren
kodları bu şekilde kullanabiliriz.
```

*** /**

Internet Explorer 6 sürümünde float uygulanmış içi içe elemanlarda kullanılan yorum satırları anlamsız bir tekrarlama sorunu meydana getiriyor. Bunu engellemek için

<!--[if !IE]>icerik alani<![endif]-->

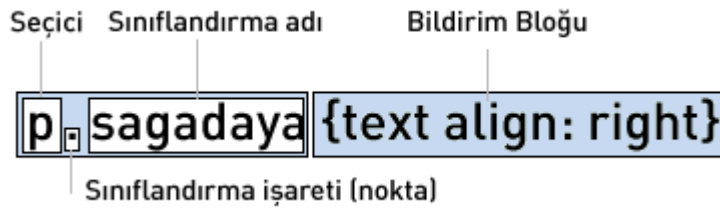
Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=232> inceleyiniz.

3. Seçiciler

Seçiciler Sınıf Seçicisi ve Tekil Seçicisi olarak ikiye ayrılır. CSS ile kod yazmaya başlarken bir plan yaparak hangi etikete Sınıf Seçicisi veya Tekil Seçicisi olacağını planlamalıyız. Zamanla bu ayrımı kodu yazarken yapmaya alışacağız.

3.1. Sınıf Seçicisi(Class Selector)

Aynı HTML elemanına farklı özellikler atamak için Sınıf Seçicisini kullanırız.



Şekil3.1 Sınıf seçicilerin yapısı

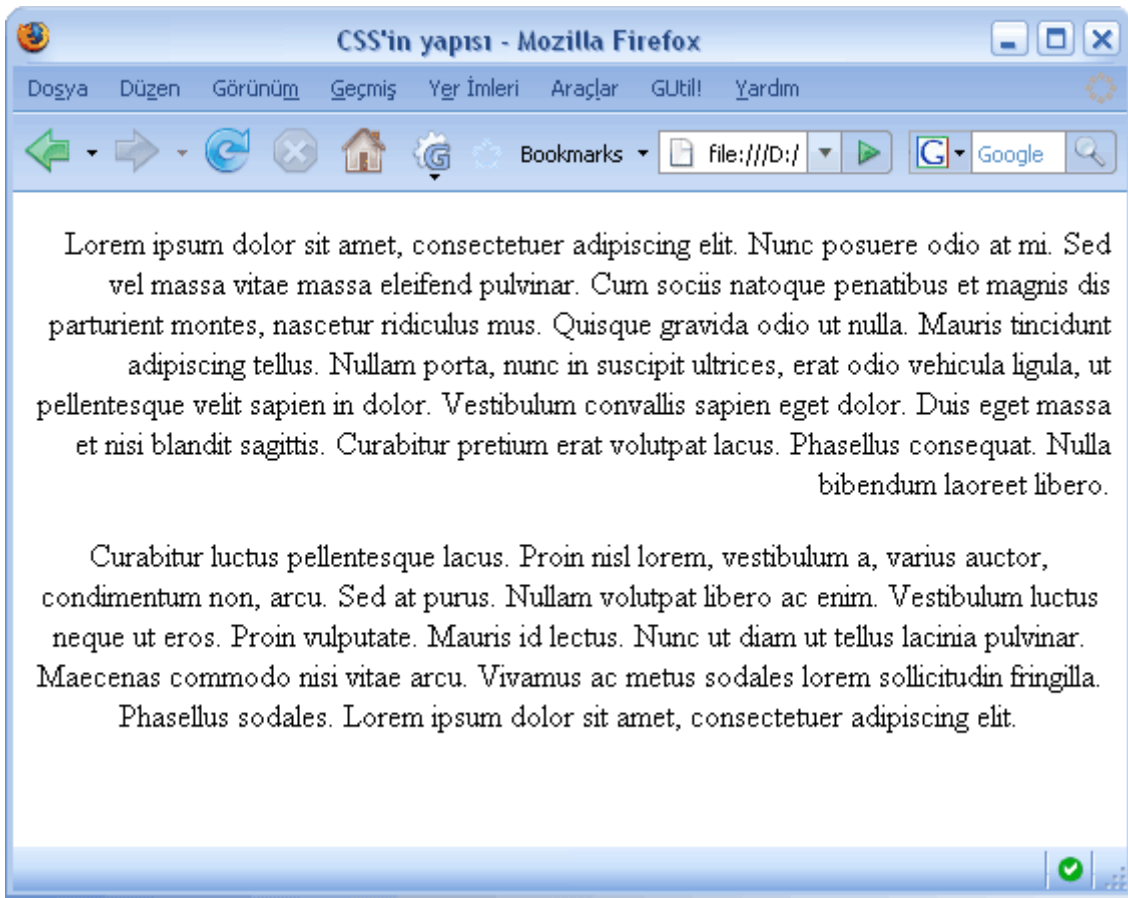
Bir örnek verecek olursak; hazırlayacağımız dokümanda iki adet paragraf tanımlaması yapacağımızı planlıyoruz. Bunlardan biri sağa dayalı, diğeri ise ortalı olmasını istiyoruz.

Örnek 3-1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
p.sagadaya {
    text-align: right;
}

p.ortala {
    text-align: center;
}
</style>
</head>
<body>
<p class="sagadaya">Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Nunc posuere odio at mi. Sed vel massa vitae massa eleifend pulvinar.
Cum sociis natoque penatibus et magnis dis parturient montes, nascetur
ridiculus mus. Quisque gravida odio ut nulla. Mauris tincidunt adipiscing
```

```
tellus. Nullam porta, nunc in suscipit ultrices, erat odio vehicula ligula, ut
pellentesque velit sapien in dolor. Vestibulum convallis sapien eget dolor.
Duis eget massa et nisi blandit sagittis. Curabitur pretium erat volutpat
lacus. Phasellus consequat. Nulla bibendum laoreet libero. </p>
<p class="ortala">Curabitur luctus pellentesque lacus. Proin nisl lorem,
vestibulum a, varius auctor, condimentum non, arcu. Sed at purus. Nullam
volutpat libero ac enim. Vestibulum luctus neque ut eros. Proin vulputate.
Mauris id lectus. Nunc ut diam ut tellus lacinia pulvinar. Maecenas
commodo nisi vitae arcu. Vivamus ac metus sodales lorem sollicitudin
fringilla. Phasellus sodales. Lorem ipsum dolor sit amet, consectetur
adipiscing elit. </p>
</body>
</html>
```



Birde önemli bir tanımlama yapacağınızı düşünün, ancak sadece bir HTML elemanına değil istediğiniz sayıda HTML elemanında bunu kullanmak istersek;

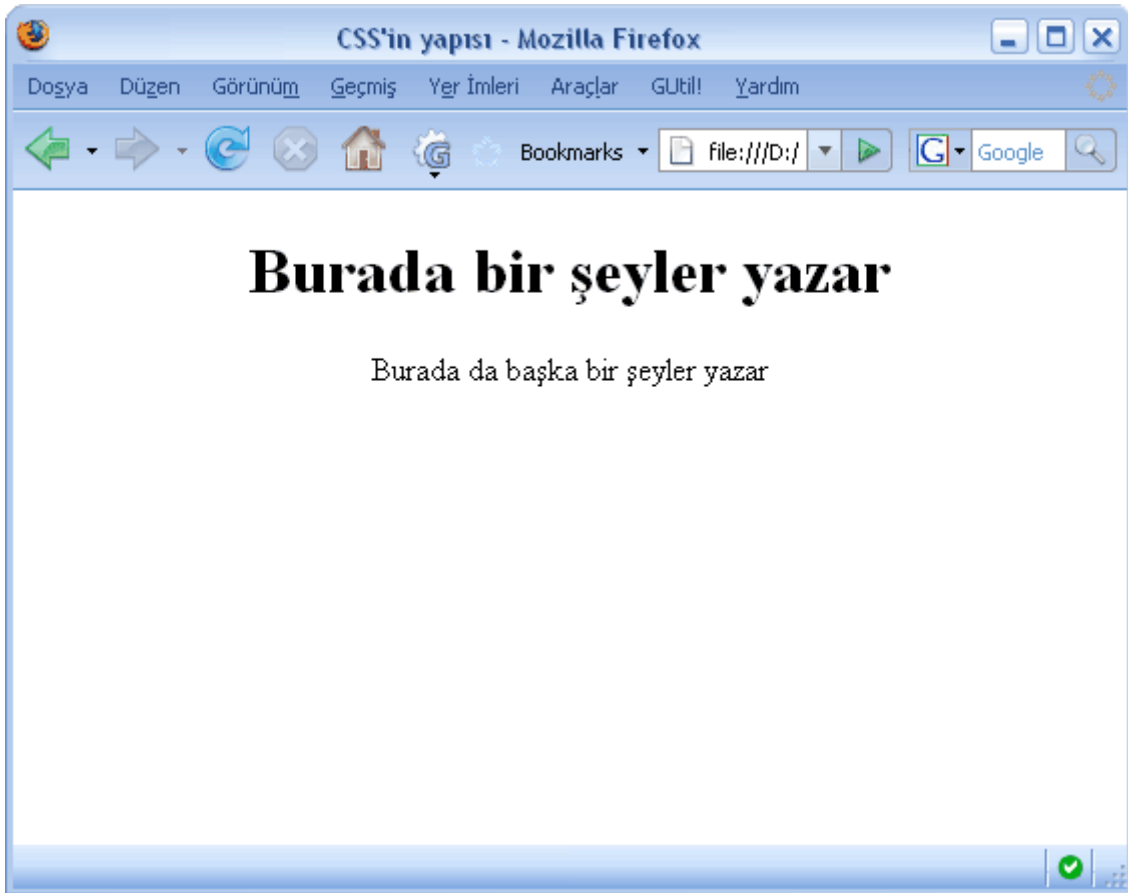
Örnek 3-2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```

<title>CSS'in yapısı</title>
<style type="text/css">
.ortala {
    text-align: center;
}
</style>
</head>
<body>
<h1 class="ortala"> Burada bir şeyler yazar </h1>
<p class="ortala">Burada da başka bir şeyler yazar</p> </body>
</html>

```



Bu tanımlamayı yaptıktan sonra istediğimiz her HTML elemanına bu sınıfı uygulayabiliriz.

Çoklu sınıflar, birden fazla sınıfı bir HTML elemanına uygulamak için kullanılır.

Örnek 3-3

```

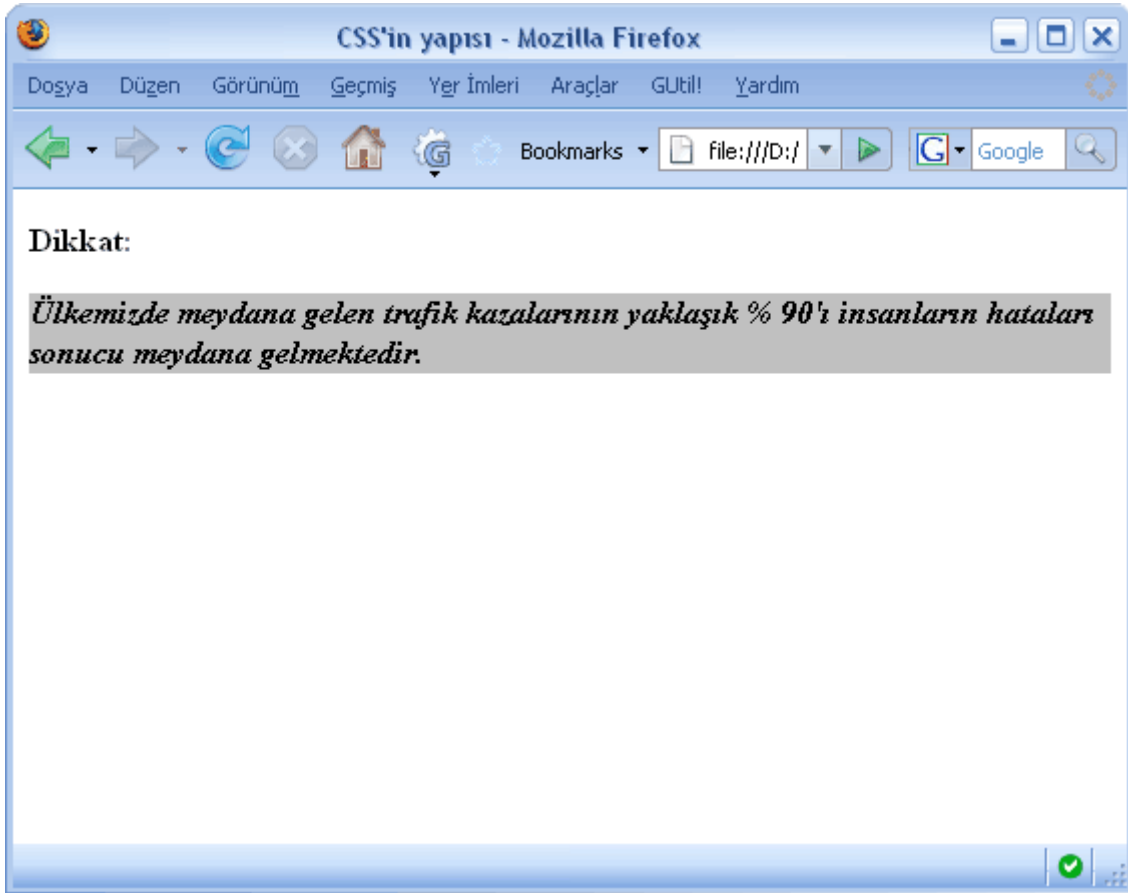
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"

```

```

/>
<title>CSS'in yapısı</title>
<style type="text/css">
.onemli {
    font-weight : bold;
}
.uyari {
    font-style: italic;
}
.onemli .uyari{
    background-color: silver;
}
</style>
</head>
<body>
<p class="onemli">Dikkat: </p>
<p class="onemli uyari">Ülkemizde meydana gelen trafik kazalarının
yaklaşık % 90'ı insanların hataları sonucu meydana gelmektedir.</p>
</body>
</html>

```

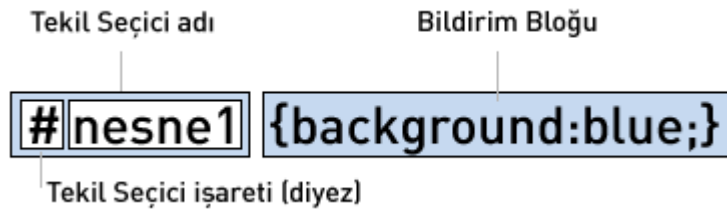


Yukarıdaki örnekte görüldüğü gibi bir **uyari** sınıfımız birde **onemli** sınıfımız mevcut. Bazı metinlerin **önemli uyari** olacağı düşüncesi ile böyle bir atama yapılabilir.

önemli metinler için kalın, **uyari** için yazının şeklini eğik olarak belirliyoruz. Birde her ikisine birden de atama yapabiliyoruz. Dikkat ederseniz **class="önemli uyari"** olarak atama yaparken CSS seçicisinde **".önemli .uyari"** şeklinde yazıyoruz.

3.2. Tekil Seçiciler(Id Selector)

Tekil Seçicisi Sınıf Seçicisinden farklıdır. Sınıf Seçicisi sayfada birden fazla elemana atanırken Tekil Seçicisi sadece bir tane elemana atanır. Tekil seçiciler tanımlanırken adının başına # (diyez) işareti konulur.



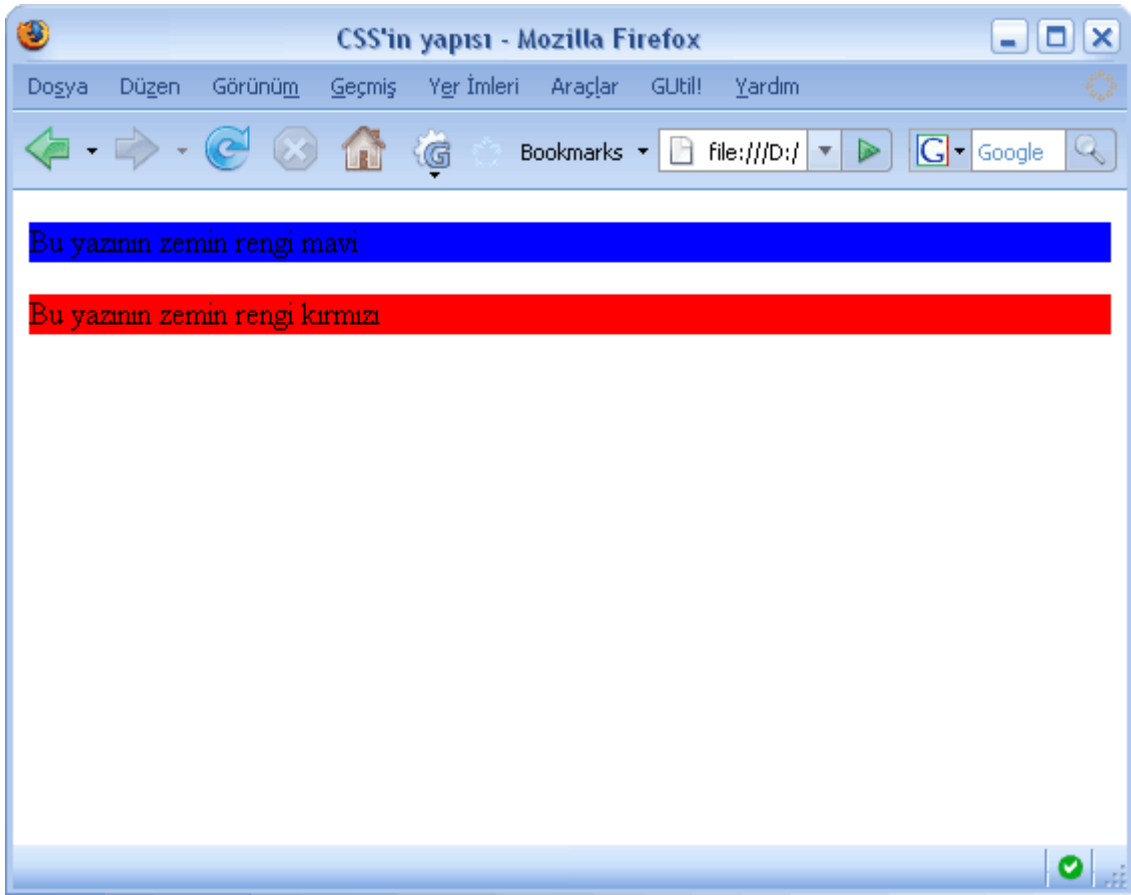
Şekil3.2 Tekil seçicilerin yapısı

Örnek yapalım:

Örnek 3-4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'in yapısı</title>
<style type="text/css">
#nesne1{
    background-color: blue;
}

#nesne2{
    background-color: red;
}
</style>
</head>
<body>
<p id="nesne1">Bu yazının Artalan rengi mavi</p>
<p id="nesne2">Bu yazının Artalan rengi kırmızı</p>
</body>
</html>
```



3.3. Sınıf(Çoğul) mı? Tekil mi?

Yukarıda Tekil Seçicisi için her ne kadar da bir sayfada sadece bir defa kullanılır diye tanımlasak de kodlamacının sayfada birçok yerde kullanmasına tarayıcılar ses çıkarmaz, yani hata mesajı vermez. Ancak Bu elemanları DOM(Belge Nesne Modeli) betiklerinde kullanırken **id** atamasını bir kaç yerde yaptığımızda hata meydana gelecektir. Bu nedenle her ne kadar tarayıcılar izin verse de Tekil Seçicilerini sayfalarımızda bir kez kullanmalıyız. Birden fazla kullanacağımız elemanlar için Sınıf Seçicisini kullanmalıyız.

Bu kodumuzu daha kullanışlı ve temiz yapacaktır.

Sınıf ve Tekil Seçicileri küçük-büyük harfe karşı duyarlıdır;

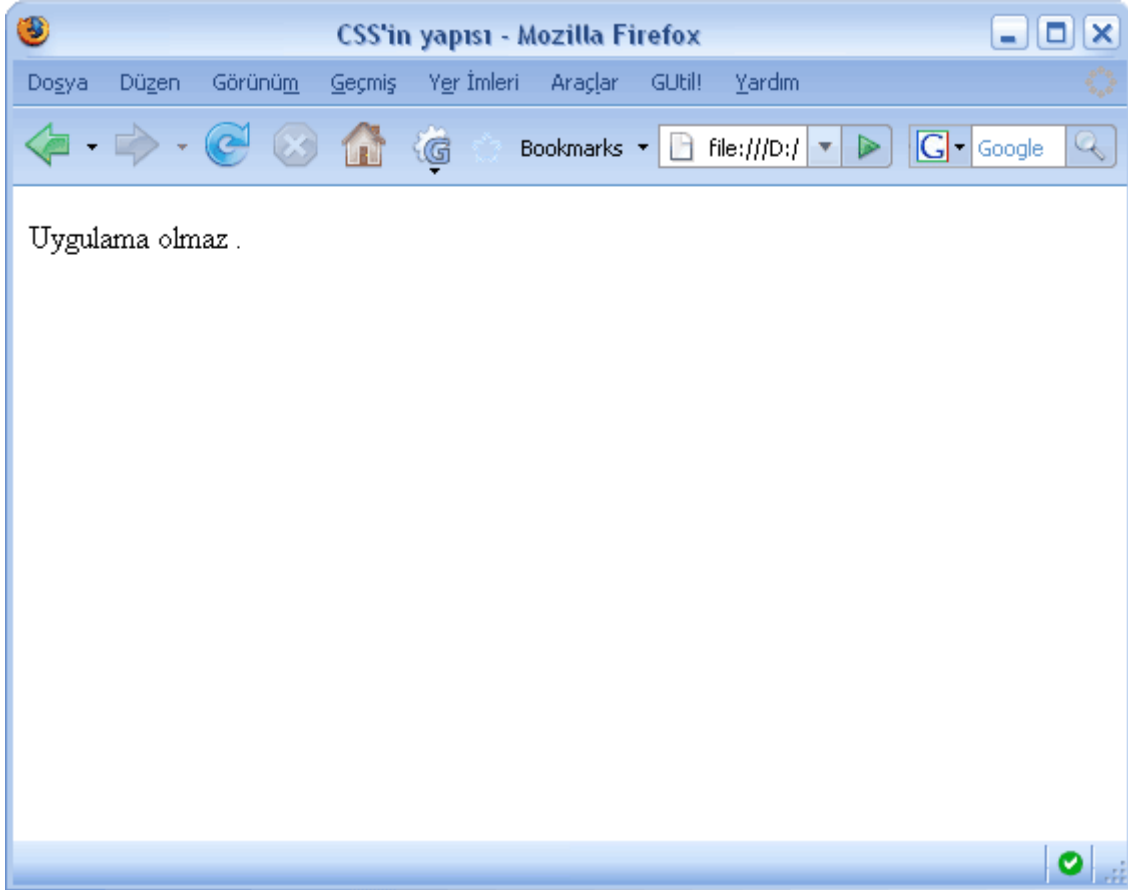
Örnek 3-5

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
```

```

p.onemliBilgi {
    font-weight: bold;
}
</style>
</head>
<body>
<p class="onemlibilgi">Uygulama olmaz .</p>
</body>
</html>

```



Yukarıdaki kod uygulanmayacaktır çünkü " **B** " bir yerde büyük birde küçük kullanılmıştır. Bazı eski tarayıcılar bu kuralın dışında kalabilir.

Şimdide diğer seçicilere göz atalım.

CSS güçlüdür, çünkü CSS uygun stilleri ve bu stilleri nasıl uygulanacağını belirlemek için (X)HTML dokümanının hiyerarşik yapısını çok iyi kullanır. (X)HTML Sayfa yapısına stil uygulamanın birçok yolu vardır. Ancak bu yolları öğrenmeden önce (X)HTML hiyerarşisini öğrenmemiz gerekir.

3.4. (X)HTML Doküman Hiyerarşisini Anlamak

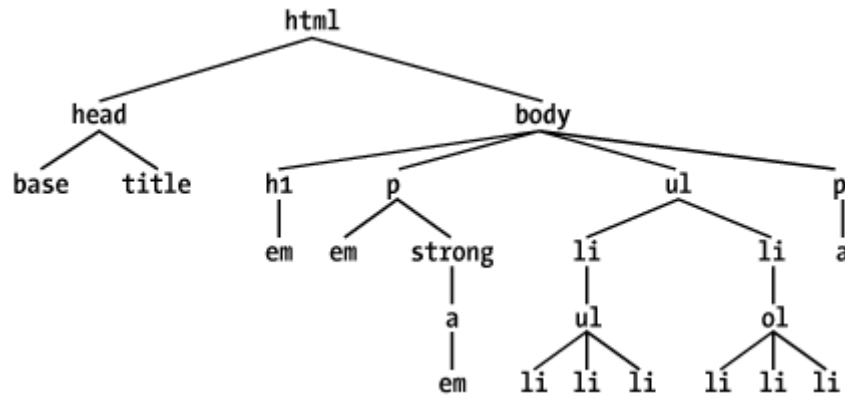
Seçici(Selector) ve doküman arasındaki ilişkiyi anlayabilmemiz için, dokümanların nasıl planlandığını bir göz atmamız gerekir. Aşağıdaki basit (X)HTML dokümanı üzerine biraz düşünelim:

Örnek 3-6

```
<html>
<head>
<title> CSS (X)HTML Hiyerarşisi </title>
</head>
<body>
<h1>CSS<em>(X)HTML Hiyerarşisi </em></h1>
Sitemize hoş geldiniz <em>Merhaba</em>biz <strong>sizler için <a
href="konu.html"> bazı <em>önemli</em> konular </a></strong>!
hazırladık.
<ul>
<li>Konu Başlıkları :
<ul>
<li><strong>Asıl</strong> Giriş</li>
<li>Önsöz</li>
<li><em>ilk</em> bölüm:
<ol>
<li>Elma</li>
<li>Armut</li>
<li>Karpuz</li>
</ol>
</li>
</ul>
</li>
<li>...vd.</li>
</ul>
<p>Ayrıntılı bilgi için <a href="mailto:bilgi@zerzebvat.com"> mesaj
gönder </a></p>
</body>
</html>
```



CSS'in güçlü olmasının en büyük nedeni (X)HTML elemanları arasındaki ilişkiden yararlanmasıdır. (X)HTML dokümanları gizli bir hiyerarşi ile oluşturulur. Bu hiyerarşi içinde tüm (X)HTML elemanları kendilerine uygun bir yer bulur. Bu ilişkiyi biz insanların **soy ağacına** benzete biliriz. Yukarıdaki HTML kodunun soyağacını çıkarırsak:



Şekil3.3 XHTML doküman hiyerarşisi

Yukarıdaki (X)html yapısını bir soy ağacı gibi düşünüp değerlendirebilirsek:

Bir elemanın ebeveyn(anne-baba) olduğunu anlamak için o elemanın altında başka eleman olup olmadığına bakmalıyız, eğer varsa o eleman ebeveynidir. Yukarıdaki şekilde **p** elemanına dikkat edelim. **p** bir ebeveynidir çünkü altında **em** ve **strong** elemanları bulunmaktadır. **strong** da bir ebeveynidir ki onun altında **a** elemanı vardır. Bir elemanın **çocuk eleman** olduğunu anlamak için de üstünde bir elemanın olup olmadığına bakarız. Yani yukarıda anlattığımız ebeveyn olayının tersi. Buna göre **strong** elemanı **p** elemanının bir çocuğudur.

Daha karmaşık yapılarda ata(*ancestor*) ve torun(*descendant*) ilişkisi vardır. Aradaki fark eğer bir eleman diğer elemanın tam olarak bir seviye üstünde ise ona **ata** diğeri de **torun** durumundadır. Yukarıdaki şekli

incelersek ilk **ul** elemanın iki adet **li** çocuk elemanı bulunmaktadır ve bu iki **li** elemanın altındaki tüm elemanlar ilk **ul** elemanın torun elemanlarıdır.

3.5. Torun Seçicileri (Descendant Selectors)

Bu modelin avantajlarından birisi Torun Seçicileridir (diğer isimleri iç içe seçicilerdir). Torun Seçicilere yapılan tanımlama sadece belirtilen elemanlara uygulanır bu kuralların haricinde kalanlara ise uygulanmaz. Bir örnek verecek olursak bir **h1** elemanın **em** torun elemanına belirli bir stil uygulamak istesek. Normalde bir sınıf tanımlaması yapılarak her **h1** elemanı altındaki **em** elemanını tek tek seçerek bu sınıfı uygulamamız gerekir ki bunun **font** etiketi uygulamasından farkı yoktur. Açıkçası bu işlemi yapmak uzun zaman alacaktır. Ancak Torun Seçicisi kullanarak bu işi kolayca yapabiliriz. Örnek kod yazarsak:

```
h1 em {color: gray}
```

Bu kod dokümandaki atası **h1** olan tüm **em** torun elemanlarını gri yapacaktır. Diğer **em**'ler ise bu kuralı uygulamayacaktır.

Torun Seçicilerde, seçici kısmı birbirinden bir boşlukla ayrılmış iki veya daha fazla seçiciden oluşur. Aradaki boşluk bağlayıcı özellik taşır.

Sadece iki elemanla sınırlı değiliz, sınırsız eleman tanımlayabiliriz.

```
ul ol ul em {color: gray}
```

Torun Seçiciler çok kuvvetli olabilirler. Onlar (X)HTML ile yapılması imkânı olmayan işler yaparlar. Örnek bir dokümanda iki adet alan oluşturduğumuzu düşünün birinci alanın artalanı (background) mavi, ikinci alanın artalanının beyaz olduğunu düşünün ve her iki alanın içindedeki linkler olduğunu farz edelim. Tüm linkleri mavi olarak atamamız mümkün olmayacaktır çünkü ilk alanın artalanı mavi olduğu için linkler görünmeyecektir.

Çözüm Torun Seçicilerindedir; ilk alan içindeki linklere farklı renk diğerlerine farklı renk tanımlaması yaparak bu işi halledebiliriz.

```
td.sidebar {background: blue;}
td.main {background: white;}
td.sidebar a:link {color: white;}
td.main a:link {color: blue;}
```

Bir örnek daha verelim. **blockquote** ve **p** elemanları içindeki **b** elemanı ile **blockquote** içinde ve normal paragraf içinde geçen **b** elemanlarına özel bir atama yapmak istiyoruz bunun için kod yazarsak:

```
blockquote b, p b {color: gray}
```

Koda dikkat edersek her iki atamayı ayrı ayrı yapmak yerine araya bir virgül koyarak birlikte yaptığımızı görürsünüz.

Tarayıcı Uyumu:

Internet Explorer 5.0+

Firefox 1+

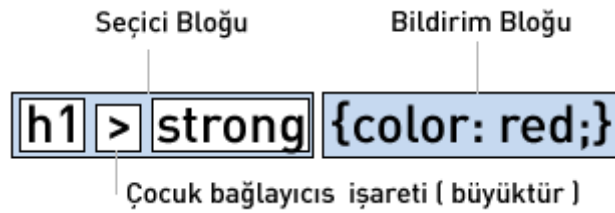
Safari 1.3+

Opera 5.0+

CSS 2+

3.6. Çocuk Seçicileri(Direct Child Selectors)

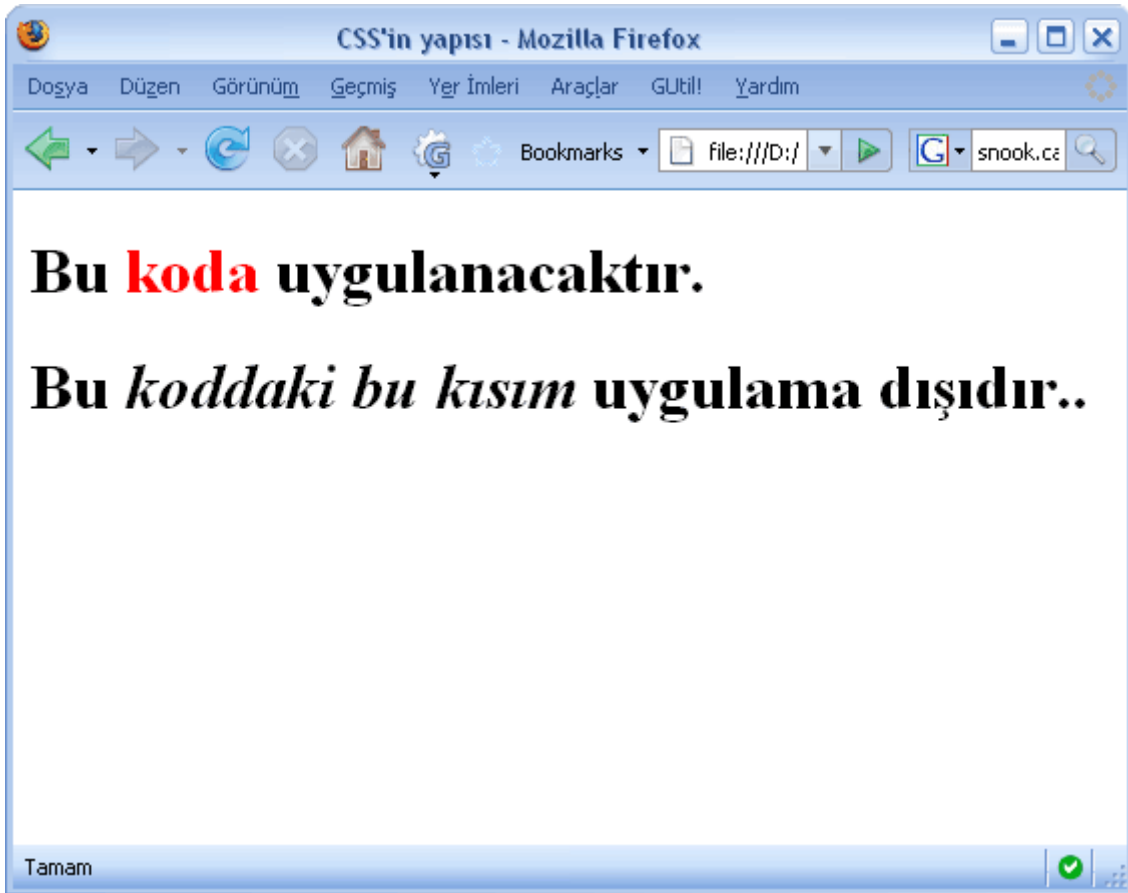
Bazı durumlarda keyfi olarak bir torun seçicisi kullanmaktansa daha ayrıntılı bir ayırım yapmak isteyebiliriz. Örneğin bir **h1** elemanının altındaki **strong** elemanlarından **sadece** Çocuk Elemanı için tanımlamalar yapmak istersek (torun elemanları hariç), Bunun için çocuk bağlayıcısını kullanırız (>) büyüktür işareti



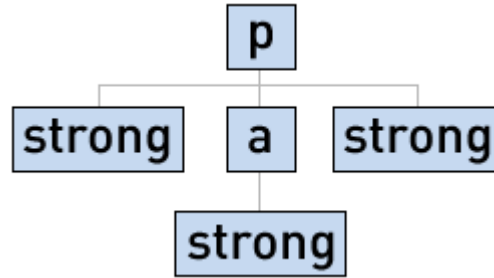
Şekil3.4 Çocuk seçicilerin yapısı

Örnek 3-7

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
h1 > strong {
    color: red
}
</style>
</head>
<body>
<h1>Bu <strong>koda</strong> uygulanacaktır.</h1>
<h1>Bu <em>koddaki <strong> bu kısım </strong></em> uygulama
dışındır..</h1>
</body>
</html>
```



Yukarıdaki durumda sadece ilk **strong** elemanı için tanımlama gerçekleşecektir. İkincisi için herhangi bir stil tanımı uygulanmayacaktır.



Yukarıda ana örneğimizin bir kısmı gösterilmektedir. Ebeveyn-çocuk ilişkisini daha iyi anlamak için hazırlanmıştır. **a** elemanı **strong**'un ebeveynidir. **p** elemanında **a** elemanının ebeveynidir. Ancak **p** elemanı **strong**'un ebeveyni değildir. Biz burada $p > a$ ve $a > strong$ şeklinde tanımlama yapabiliriz ancak $p > strong$ tanımlamasını yapamayız.

Birde aynı seçici içinde hem torun birleştiricisini hemde çocuk birleştiricisini kullanabiliriz.

table.summary td > p

Tarayıcı Uyumu :

Internet Explorer 6.0: *desteklemiyor*, 7.0 destekliyor
 Firefox1+
 Safari1.3+
 Opera 9.2+
 CSS 2+

3.7. Bitişik Kardeş Seçiciler(Next Sibling Selector)

Bitişik Kardeş Seçicileri biri biri ardına gelen aynı seviyedeki elemanlara stil tanımlaması için kullanılır.

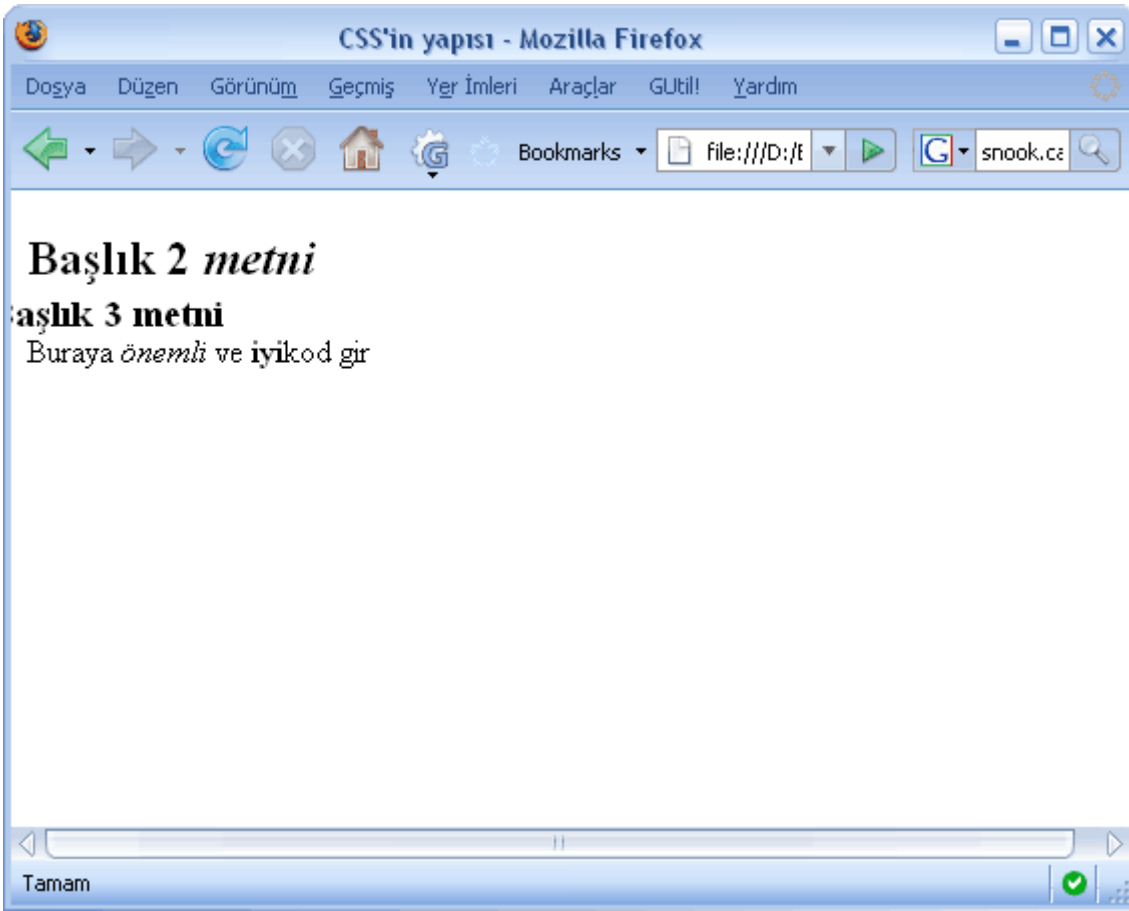


Şekil3.5 Bitişik kardeş seçicilerin yapısı

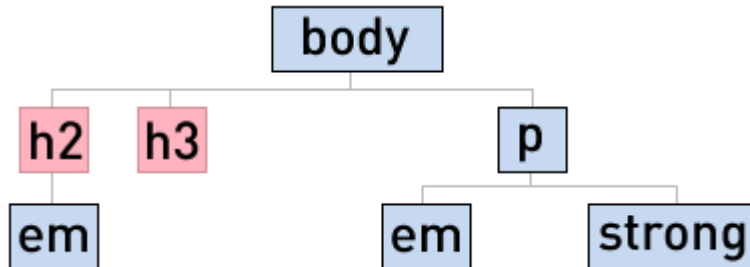
Örneğin bir **h3** elemanına stil uygulamak istiyoruz ayrıca bir sonraki h2 elemanında aynı stili uygulamak istediğimiz durumlarda kullanılır. Genel uygulamalarda kullanılan bir durumdur. Biz genelde **h2** ve **h3** arasındaki aralığı ayarlamak için bu durumla karşılaşırız. Örnek verecek olursak:

Örnek 3-8

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
h2 + h3 {
  margin: -1em;
}
</style>
</head>
<body>
<h2>Başlık 2 <em>metni</em></h2>
<h3>Başlık 3 metni</h3>
<p>Buraya <em>önemli</em> ve <strong>iyi</strong>kod gir</p>
</body>
</html>
```



Dokümanın yapısı:



Biz burada **h2** ve **h3** arasında boşluk vermek için yukarıdaki kodu kullandık

Tarayıcı Uyumu:

Internet Explorer 5.5, 6.0: *desteklemiyor* (7.0 destekliyor)

Firefox 1+

Safari 1.3+

Opera 5.0+

CSS 2

Sonuç olarak konunun başında söylediğimiz gibi CSS, (X)HTML elemanlarının kendi aralarındaki ilişkileri kullanarak bize birçok avantaj

sağlar. Bu sayede birçok kod yığınınından kurtulmuş oluruz ve kodlarımız bir düzene girer. Bu kuralları uyguladıkça önemini daha iyi anlayacağız.

Çocuk Seçicileri ve **Bitişik Kardeş Seçicileri**'ni Internet Explorer 7 sürümünden itibaren desteklemeye başlamıştır. Bu nedenle kullanımı konusunda dikkat etmeliyiz.

3.8. Evrensel Seçiciler(Universal Selector)

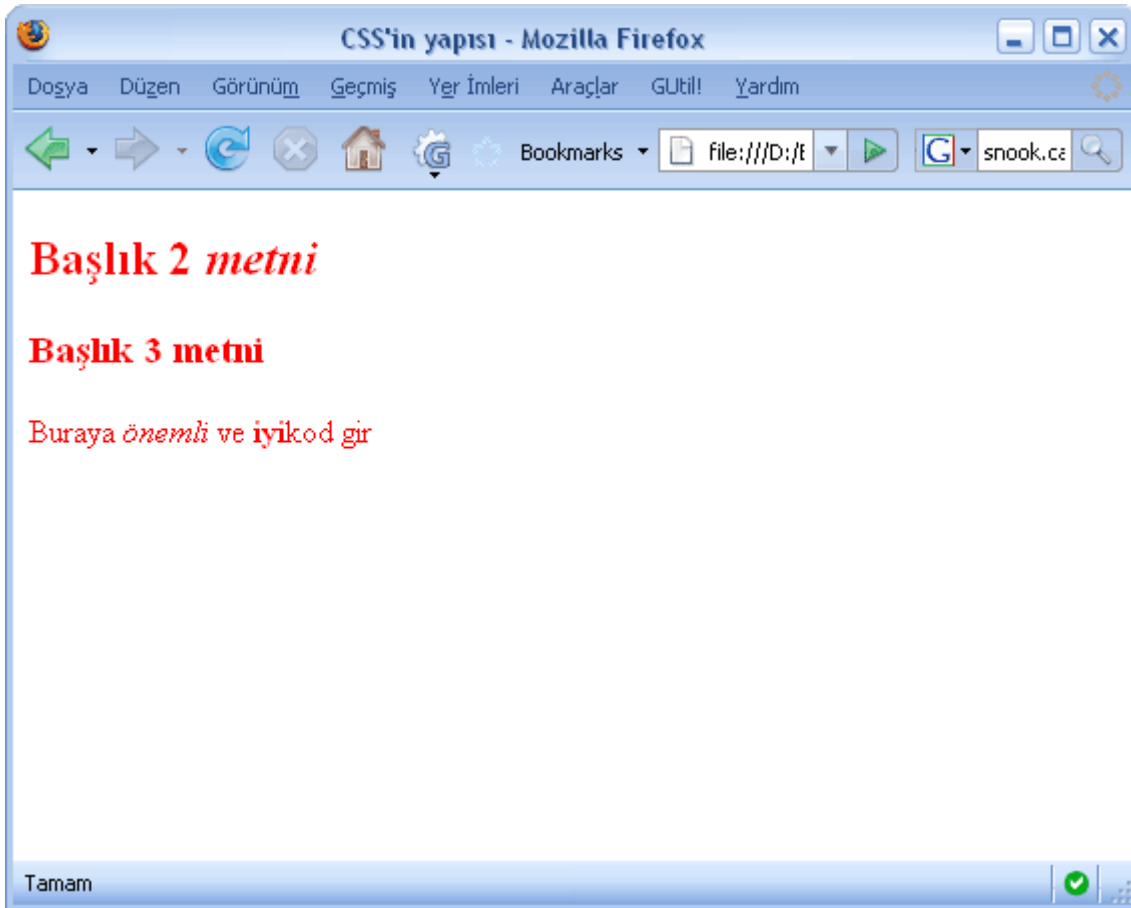
Evrensel Seçiciler bir elemanı bütüne uygulamak için kullanılır bu işlem için * (asteriks) işareti kullanılır. Aşağıdaki kod tüm elemanları kırmızı yapacaktır.



Şekil3.6 Evrensel seçicilerin yapısı

Örnek 3-9

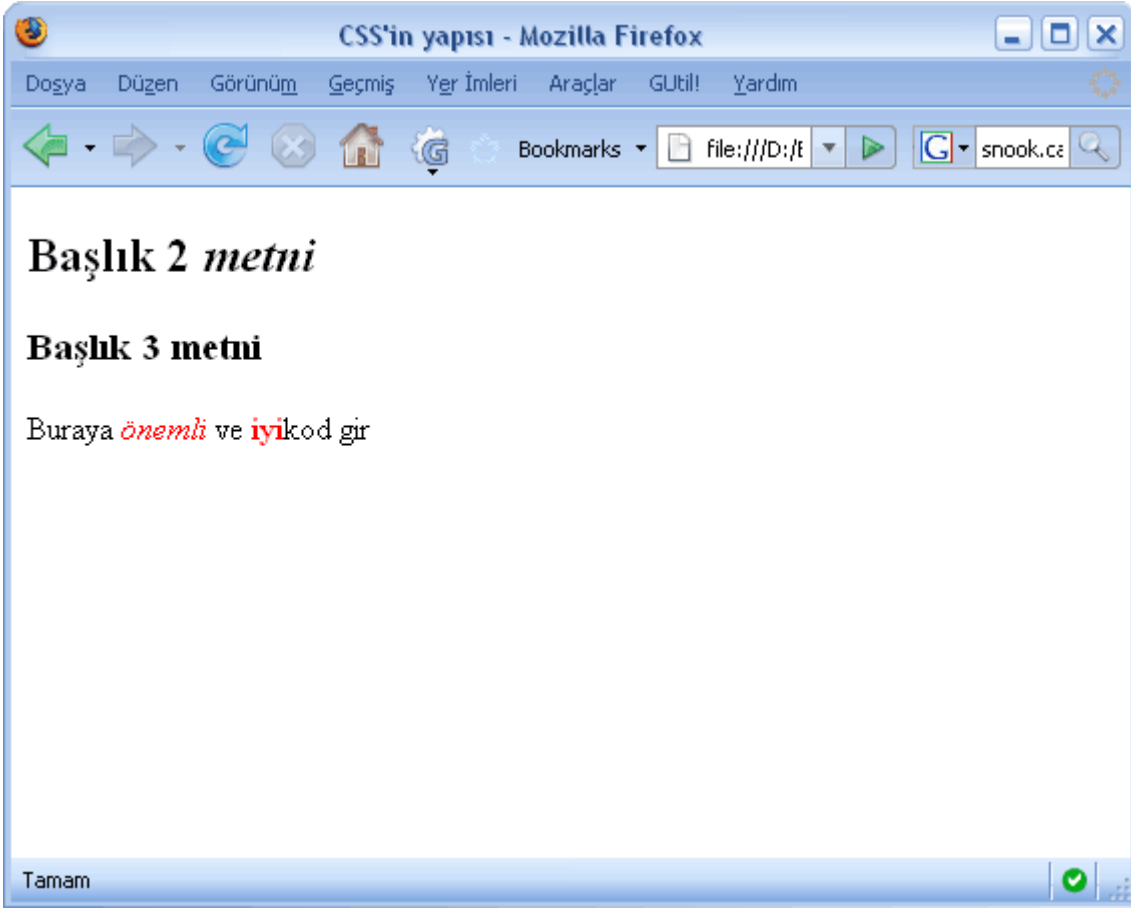
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
* {
    color:red
}
</style>
</head>
<body>
<h2>Başlık 2 <em>metni</em></h2>
<h3>Başlık 3 metni</h3>
<p>Buraya <em>önemli</em> ve <strong>iyi</strong>kod gir</p>
</body>
</html>
```



Ayrıca bir eleman içindeki tüm elemanlara uygulamak istersek

Örnek 3-10

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
p * {
    color:red
}
</style>
</head>
<body>
<h2>Başlık 2 <em>metni</em></h2>
<h3>Başlık 3 metni</h3>
<p>Buraya <em>önemli</em> ve <strong>iyi</strong>kod gir</p>
</body>
</html>
```



Paragraf içindeki tüm elemanları kırmızı yapar.

Tarayıcı Uyumu :

Internet Explorer 5.0+
 Firefox 1+
 Safari 1.3+
 Opera 5.0+
 CSS 2+

3.9. Özellik Seçicileri(Attribute Selectors)

CSS ile tanımlama yaparken Html nesnelere özellik atama yapmakta bazen sıkıntı çekeriz. Ne yapsakda bu elemana atama yapsak diye düşündüğümüz durumlar olur. Bu sorunu kökünden çözen bir yöntemdir **Özellik Seçicileri**.

Javascript gibi id'si olan her elemana(hatta daha fazlası da var) css özelliği atanması bize birçok bakımdan avantaj sağlar.

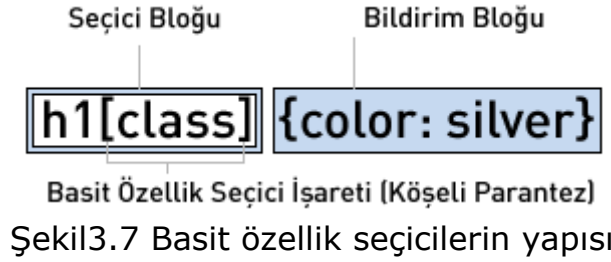
Özellik Seçicileri HTML, XHTML veya XML dokümanlarındaki seçilen etiketin özelliklerine veya özellik değerlerine göre tanımlama yapmamızı sağlar. Bir örnek verecek olursak sayfamızdaki bir resme(örn: grafik.gif) kenarlık ataması yapmak istersek:

```
img[src="grafik.gif"] { border: 1px solid #000 }
```

Dört çeşit Özellik Seçicisi vardır.

3.9.1. Basit Özellik Seçicisi

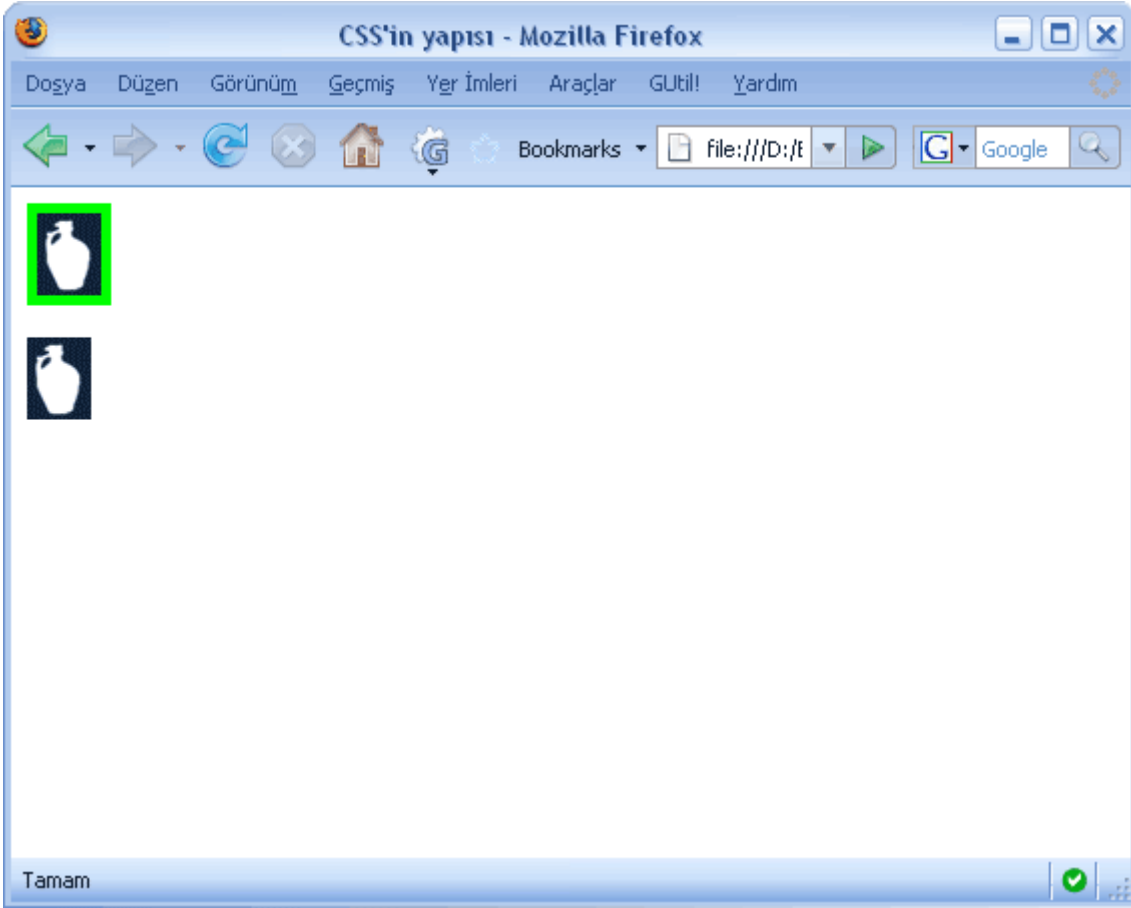
Belirli bir etiketin(örn:img) özelliğine(örn:alt) göre tanımlama yapmamızı sağlar.



Örnek 3-11

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
img[alt] {
border: 5px solid lime
}
</style>
</head>
<body>

</body>
</html>
```



Yukarıdaki kodlamada birden fazla `` tanımı vardır. Ama biz sadece alt tanımı yaptıklarımıza belli özellikler atamak istiyoruz. Bunun için Basit Özellik Seçicisini kullanabiliriz.

Tarayıcı Uyumu:

Internet Explorer 5.5, 6.0: **desteklemiyor** 7.0: destekliyor ;
 Firefox 1+
 Safari 1.3+
 Opera 7.0+
 CSS 2+

3.9.2. Özellik Değeri Seçicisi

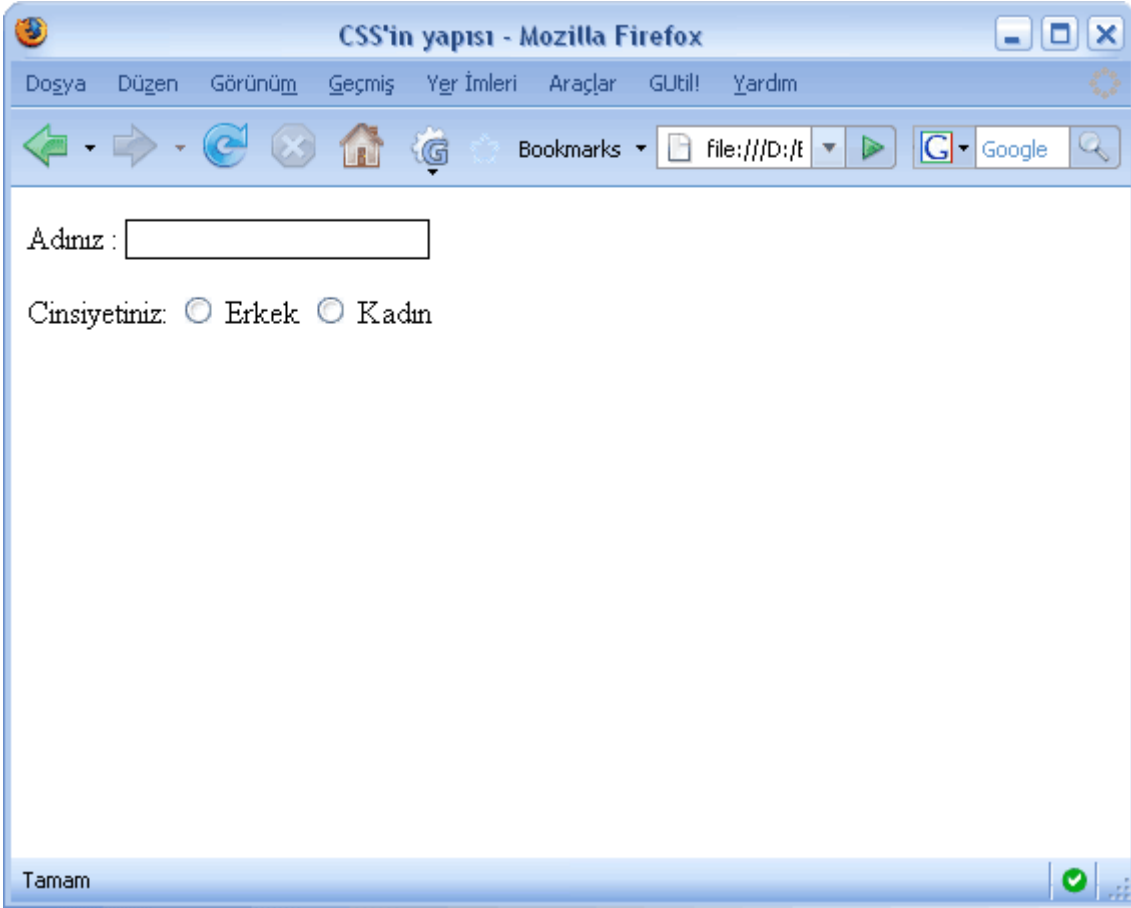
Bir etiketin özellik değerine göre tanımlama yapmamızı sağlar.



Şekil3.8 Özellik değeri seçicilerin yapısı

Örnek 3-12

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'in yapısı</title>
<style type="text/css">
input[type="text"] {
    border: 1px solid #000;
    width: 150px;
}
</style>
</head>
<body>
<p>Adınız : <input type="text" name="ad" /></p>
<p>Cinsiyetiniz: <input type="radio" name="cinsiyet" id="erkek" />
Erkek <input type="radio" name="cinsiyet" id="kadin" /> Kadın
</p>
</body>
</html>
```



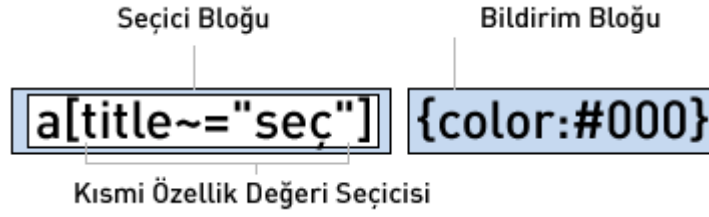
Genelde formlarda tanımlama yaparken sadece metin girişi yaptığımız alanlara atama yapmakta zorlanırsınız. Bunun sebebi **<input>** etiketini sadece metin girişi için kullanılmamasıdır, radyo buton ve işaretleme kutuları içinde bu etiketin kullanılmasıdır. Özellik Değeri Seçicisi bu sorunu kökten çözer. Yukarıdaki örnekte görüldüğü gibi

Tarayıcı Uyumu:

Internet Explorer 5.0, 5.5, 6.0: desteklemiyor 7.0: destekliyor ;
 Firefox 1+
 Safari 1.3+
 Opera 7.0+
 CSS 2+

3.9.3. Kısmi Özellik Değeri Seçicisi

Bir etiketin özellik değerindeki boşluklarla ayrılmış kısımlarına göre tanımlama yapmamızı sağlar.



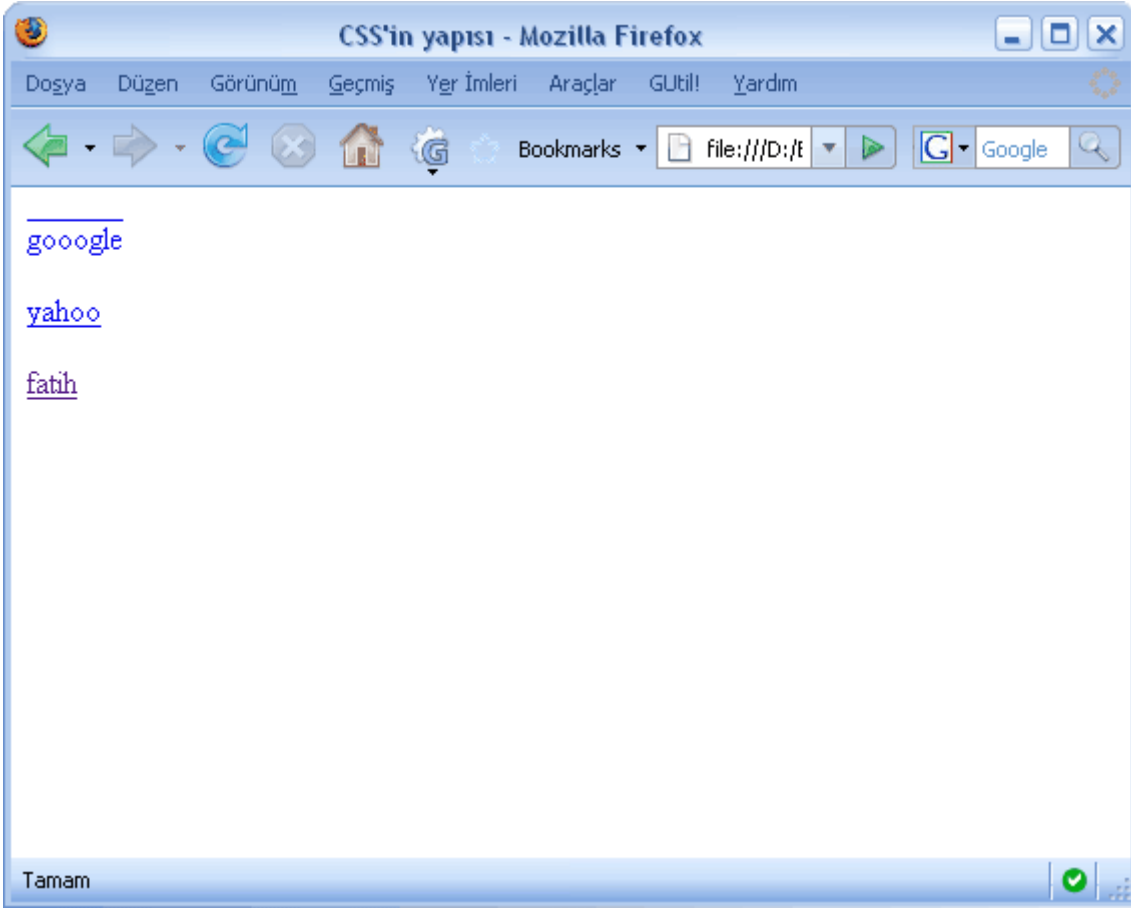
Şekil3.9 Kısmi özellik seçicilerin yapısı

Örnek 3-13

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'in yapısı</title>
<style type="text/css">
a[title~="Google"] {
    text-decoration:underline;
}
</style>
</head>
<body>
<p><a href="http://www.google.com" title="Google seç">google
</a> </p>
<p><a href="http://www.yahoo.com" title="Yahoo seç">yahoo </a>
</p>
<p><a href="http://www.fatihhayrioglu.com" title="Fatih">fatih </a>
</p>
</body>
</html>

```



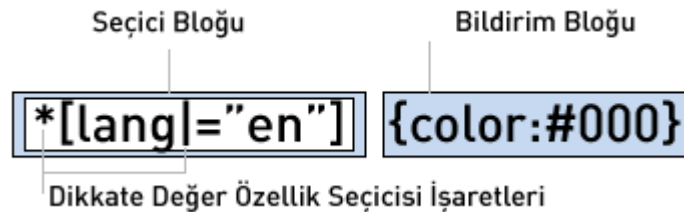
Türkçe karakterde sorunsuz çalışıyor. Siz kullanabileceğimiz yerleri hesap edin artık.

Tarayıcı Uyumu:

Internet Explorer 5.5, 6.0: **desteklemiyor** 7.0: destekliyor ;
 Firefox 1+
 Safari 1.3+
 Opera 7.0+
 CSS 2+

3.9.4. Dikkate Değer Özellik Seçicisi

Bir etiketin özellik değerindeki kesik çizgi(-) ile ayrılmış kısımlarına göre tanımlama yapmamızı sağlar.



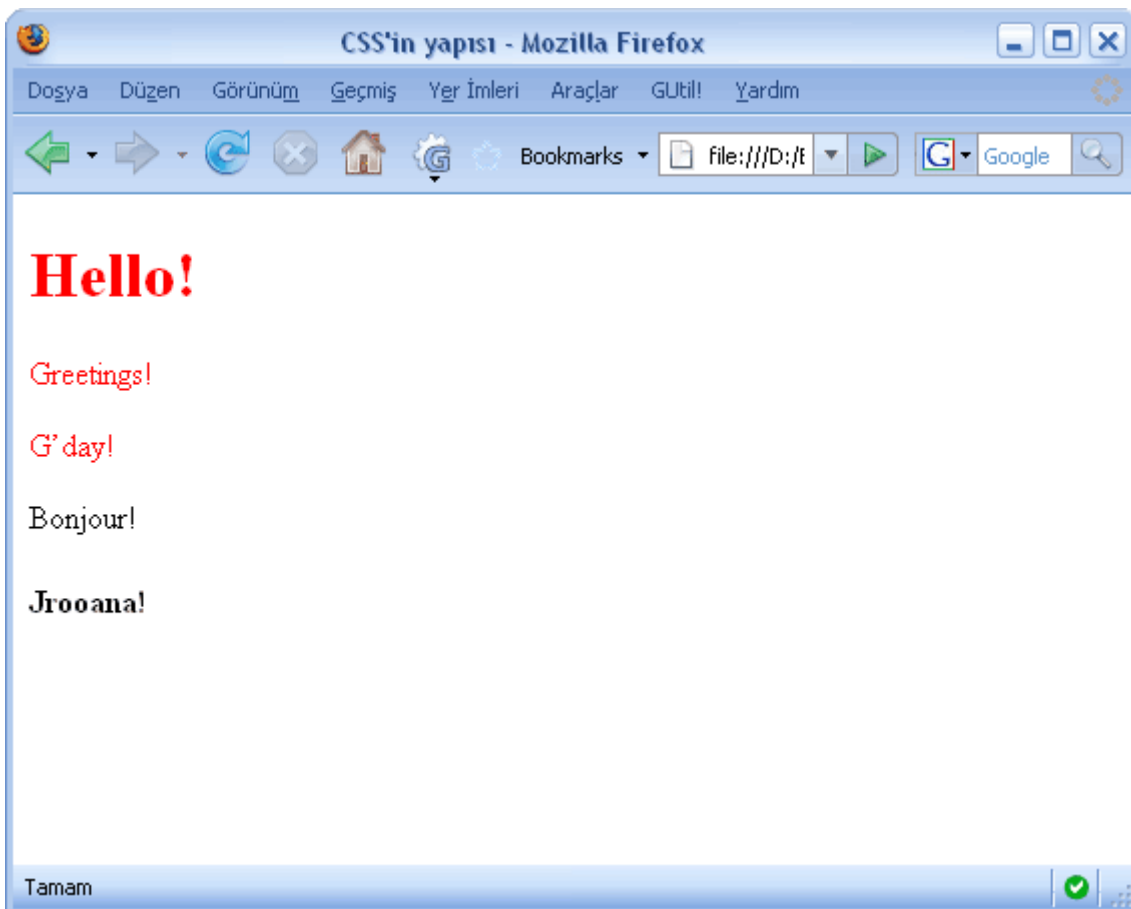
Şekil3.10 Dikkate değer özellik seçicilerin yapısı

Örnek 3-14

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'in yapısı</title>
<style type="text/css">
*[lang="en"] {
    color: red;
}
</style>
</head>
<body>
<h1 lang="en">Hello!</h1>
<p lang="en-us">Greetings!</p>
<div lang="en-au">G'day!</div>
<p lang="fr">Bonjour!</p>
<h4 lang="cy-en">Jrooana!</h4>
</body>
</html>

```



Tarayıcı Uyumu:

Internet Explorer 5.5, 6.0: **desteklemiyor** 7.0: destekliyor ;
 Firefox 1+
 Safari 1.3+
 Opera 7.0+
 CSS 2+

Bunların dışında CSS3 ile birlikte gelen dört tane daha yeni Özellik Seçicisi vardır. "Belirlenen Sözdizimi ile Başlayan Özellik Değer Seçicisi", "Belirlenen Sözdizimi ile Biten Özellik Değer Seçicisi", "İçinde Belirlenen Sözdizimi Geçen Özellik Değer Seçicisi" ve Aduzayı Özellik Değer Seçicisi diye 4 adet yeni Özellik Seçicisi daha vardır. Amaç CSS ve XHTML arasındaki iletişimi en üst seviyeye çıkararak kolay ve etkileşimli kod yazmamızı sağlamak.

3.10. Sözde(Pseudo)-sınıfları ve Sözde(Pseudo)-elemanları

Sözde(Pseudo) sınıflar ve Sözde Elemanları CSS'i destekleyen web tarayıcıları tarafından otomatik olarak tanınan özel sınıf ve elemanlardır. Sözde denmesinin nedeni normal HTML hiyerarşisi içinde bu sınıf elemanların bulunmamasındandır. Bu sınıf ve elemanlar (X)HTML hiyerarşisi ile erişemediğimiz eleman ve sınıflara erişmemizi sağlar.

Sözde(Pseudo) sınıf; bir elemanı farklı sınıflara böler(örn: link elemanını active, visited vd. sınıflarına böler)



Sözde(Pseudo) elemanı; bir elemanı alt kısımlara böler (örneğin bir paragrafın ilk harfi, bir paragrafın ilk satırı gibi.)

Sözde(Pseudo) sınıfa örnek:

```
a:visited {color: red}
```

Sözde(Pseudo) elemana örnek:

```
p:first-line { font-weight: bold; }
```

Sözde(Pseudo) sınıf ve elemanları HTML sınıf özelliği olarak belirtilmemiştir. Normal sınıflar sözde sınıfları ve sözde elemanları ile kullanılabilir.

```
a.disariyalinkler:link, a.disariyalinkler:visited {color: maroon}
```

Aynı şekilde tekil seçicileri ile birlikte de kullanılabilirler

```
a#altkisim:link{font-weight: bold;}
```

3.10.1. Sözde Sınıf Seçicileri

Sözde sınıflarını ikiye ayırabiliriz. **Sözde Link Sınıfları** ve **Sözde Dinamik Sınıfları**

Sözde Link Sınıfı

Sadece linklere uygulanan iki Sözde Link sınıfı vardır. **:link** ve **:visited**

:link - Ziyaret edilmemiş sayfanın linkine stil tanımlaması yapmak için kullanılır. Ancak birçok web tarayıcısı yapılan stil tanımlarını ziyaret edilmiş sayfa linklerine de uygular.

:visited - Ziyaret edilmiş sayfa linklerine stil tanımlaması yapmak için kullanılır.

```
a:link {color: blue}
a:visited {color: red}
```

Bunun yerine genelde aşağıdaki gibi bir kod da kullanılır

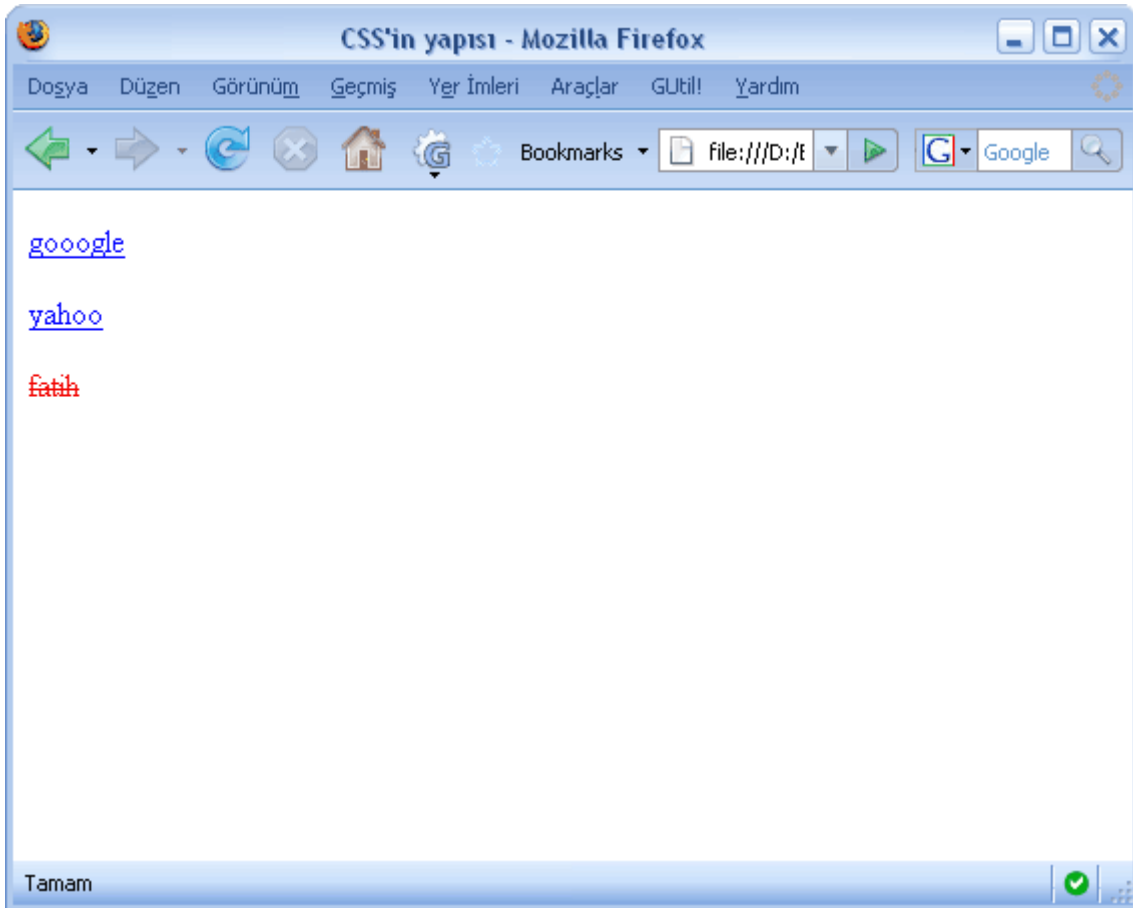
Örnek 3-15

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'in yapısı</title>
<style type="text/css">
a {
    color: blue;
}
a:visited {
    color: red;
    text-decoration: line-through;
```

```

}
</style>
</head>
<body>
<p><a href="http://www.google.com" title="Google seç">google </a>
</p>
<p><a href="http://www.yahoo.com" title="Yahoo seç">yahoo </a>
</p>
<p><a href="http://www.fatihhayrioglu.com" title="Fatih">fatih </a>
</p>
</body>
</html>

```



Bu kodlama ile kullanıcıya ziyaret ettiği sayfa linkleri farklı renkte ve üzeri çizili gösterilerek içeriksel bir bilgi görsel olarak verilebilir.

Tarayıcı Uyumu:

Sözde sınıflarda sıralama önemlidir. Genel kullanımda **"link-visited-hover-active,"** sıralaması yapılmalıdır. Tüm sınıflar içinse **"link-visited-focus-hover-active."** sıralaması yapılmalıdır.

Internet Explorer 5.0+
 Firefox 1+
 Safari 1.3+
 Opera 5.0+
 CSS 1+

Sözde Dinamik Sınıf Seçicileri

Sözde Dinamik Sınıfları sayfa görünümüne çok büyük katkılar yapabilir. Bu sınıflar genelde linklere uygulanır ancak birçok kullanım alanları vardır. **:focus**, **:active** ve **:hover**

:focus - Odaklanan elemana stil tanımlaması yapmak için kullanılır Örn: <input> alanına içerik girerken metin giriş alanına odaklandığında renk değişimi yapılması gibi.

:active - Aktif olan elemana stil atamak için kullanılır.

:hover - Bir elemanın üzerine Farenin imleci geldiğinde yapılacak tanımlama için Örn: bir linkin üzerine fare ile geldiğimizde renk değiştirmesini sağlamak için kullanılabilir.

Linkler için genel kullanım

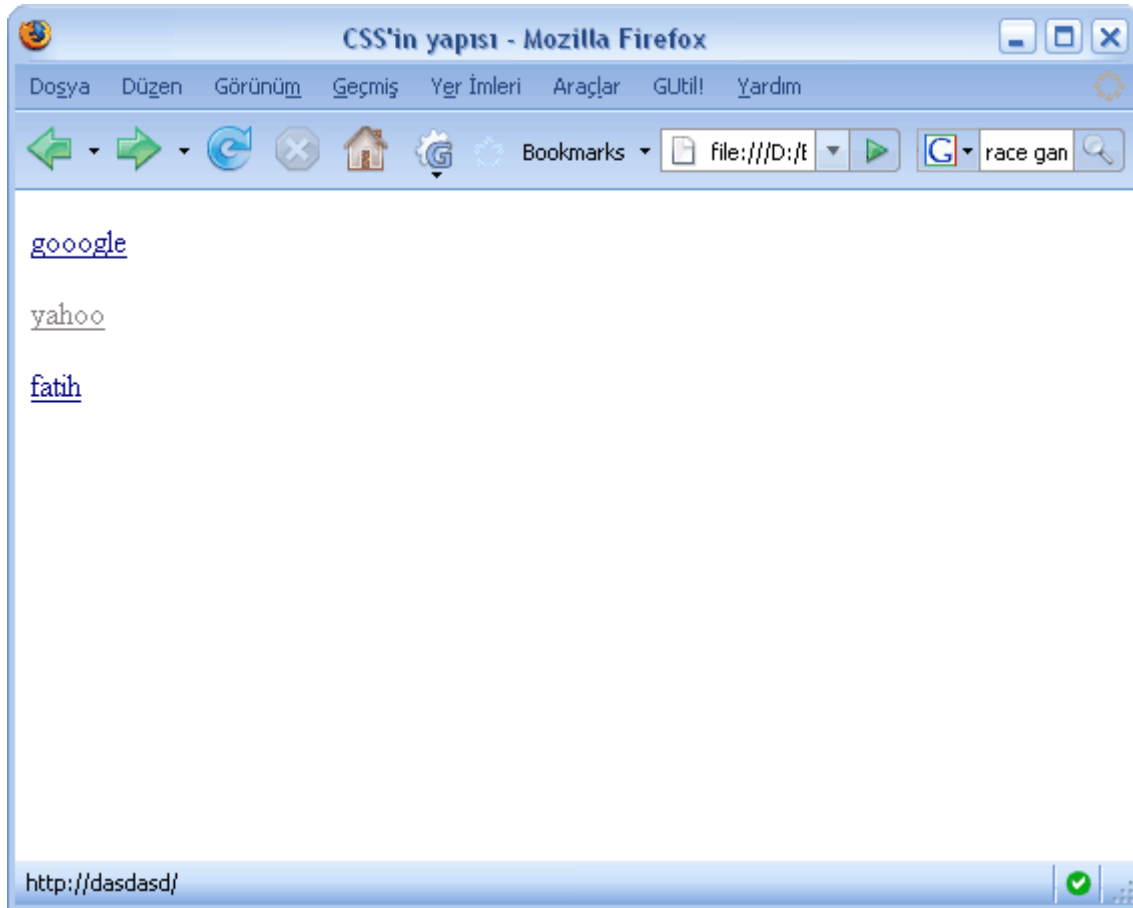
Örnek 3-16

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
a:link {
    color: navy;
}
a:visited {
    color: gray;
}
a:hover {
    color: red;
}
a:active {
    color: yellow;
}
</style>
</head>
<body>
<p><a href="http://www.google.com" title="Google seç">google </a>
```

```

</p>
<p><a href="http://www.yahoo.com" title="Yahoo seç">yahoo </a>
</p>
<p><a href="http://dasdasd" title="Fatih">fatih </a> </p>
</body>
</html>

```



:focus için bir örnek verecek olursak

Örnek 3-17

```

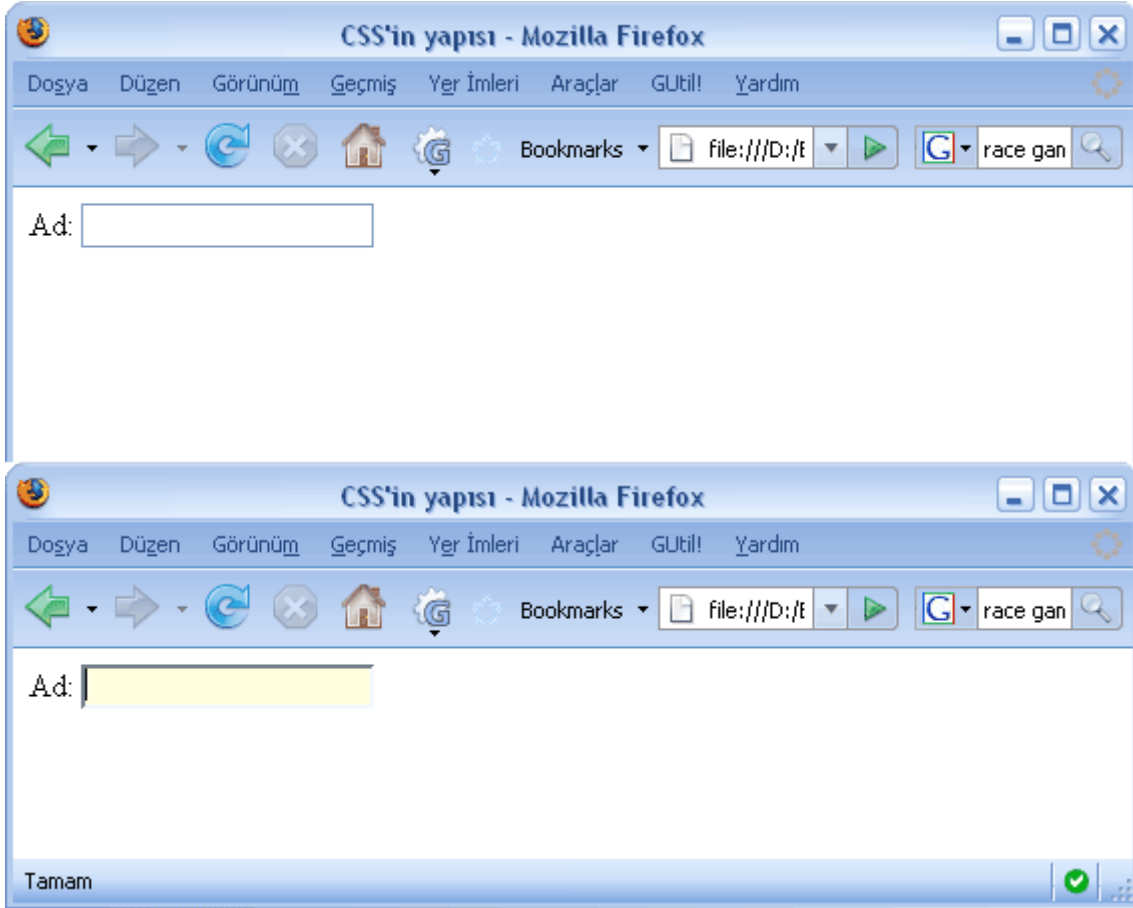
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>CSS'in yapısı</title>
<style type="text/css">
input:focus
{
    background-color: #FFFFE0;
    color: white;
}
</style>

```

```

</head>
<body>
Ad: <input type="text" id="ad" />
</body>
</html>

```



Tarayıcı Uyumu:

Internet Explorer 5.0+
 Firefox 1+
 Safari 1.3+
 Opera 5.0+
 CSS 1+

İlk Çocuk Elemanını seçmek

Diğer bir sözde sınıfımız **:first-child**'dir.

first:child: Belirtilen elemanın ilk Çocuk Elemanına stil atamak için kullanılır.

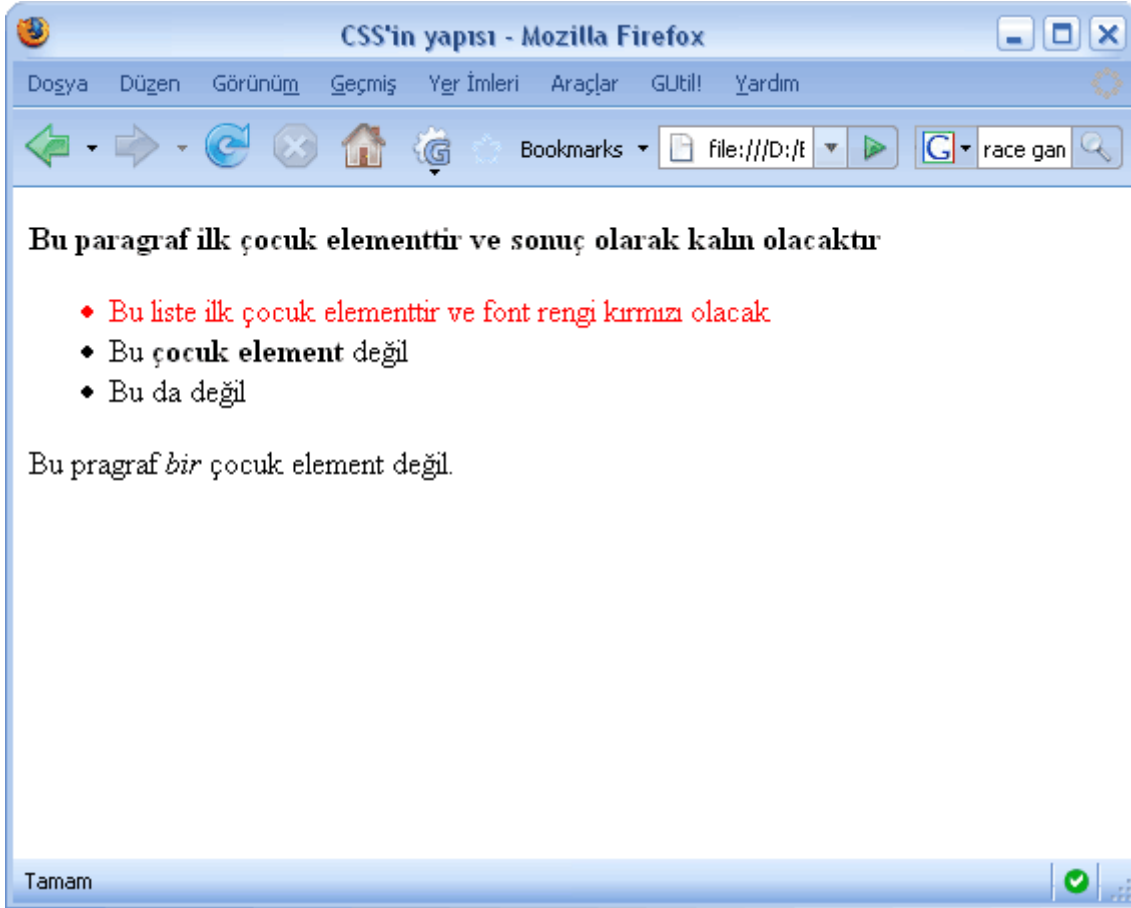
Örnek 3-18

```

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

```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'in yapısı</title>
<style type="text/css">
p:first-child {
    font-weight: bold;
}
li:first-child {
    color:#f00;
}
</style>
</head>
<body>
<div>
<p>Bu paragraf ilk çocuk elemanıdır ve sonuç olarak kalın olacaktır</p>
<ul> <li>Bu liste ilk çocuk elemanıdır ve yazıtipi rengi kırmızı olacak</li>
<li>Bu<strong>çocuk eleman </strong>değil</li> <li>Bu da değil</li>
</ul>
<p>Bu paragraf <em>bir</em> çocuk eleman değil.</p>
</div>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 5.5, 6.0 **desteklemiyor** 7.0: destekliyor ;
 Firefox 1+
 Safari 1.3+
 Opera 5.0+
 CSS 1+

:lang sözde sınıfı

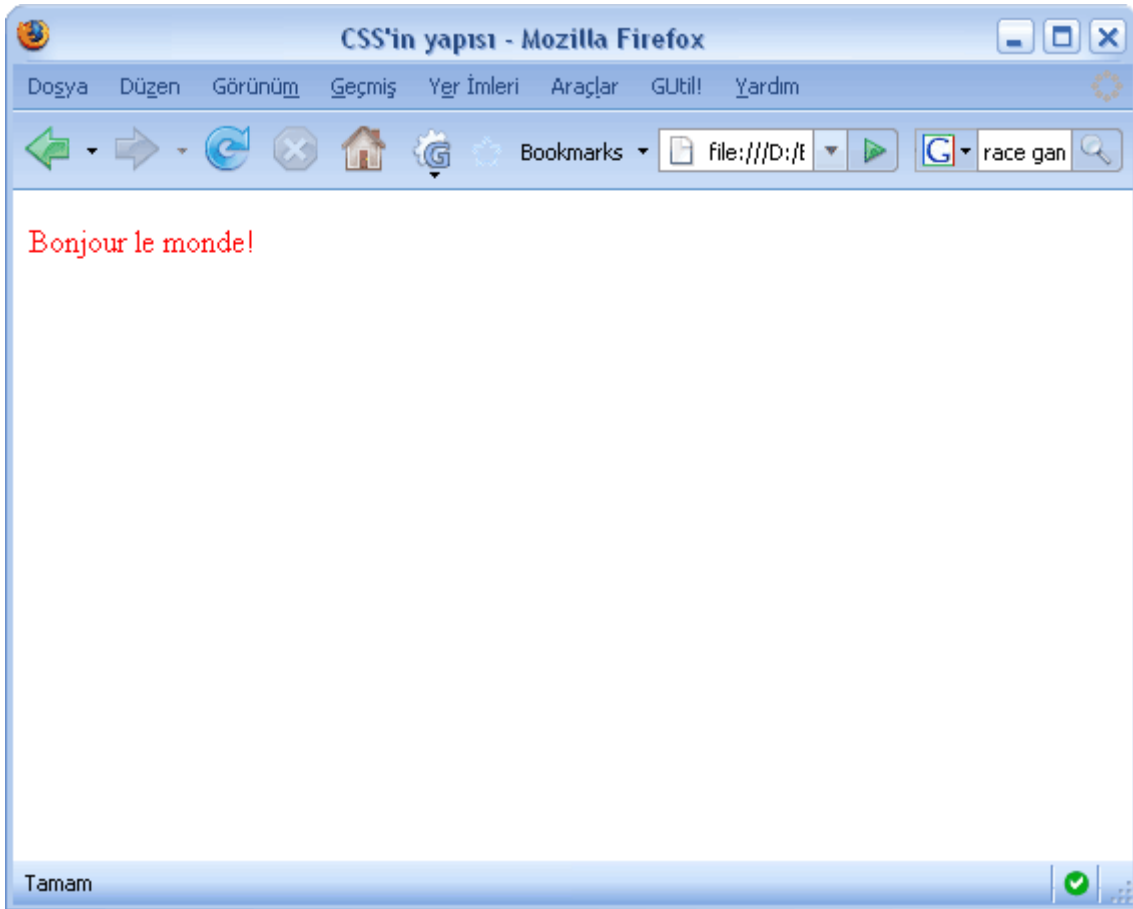
:lang sözde sınıfı içerikteki bir elemana farklı bir dilde yazmamızı sağlar.

Atanabilecek dil listesi ve kullanılacak kısaltmalara [ISO 639 and RFC 1776 standards](http://www.iso.org/iso/639-1.html) sitesinden erişebilirsiniz.

Örnek 3-19

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'in yapısı</title>
<style type="text/css">
```

```
p:lang(fr) {  
    color: red;  
}  
</style>  
</head>  
<body>  
<div>  
<p lang="fr">Bonjour le monde!</p>  
</div>  
</body>  
</html>
```



Tarayıcı Uyumu:

Internet Explorer 5.5, 6.0, 7.0 **desteklemiyor**

Firefox 1+

Safari 1.3+

Opera 7.5+

CSS 1+

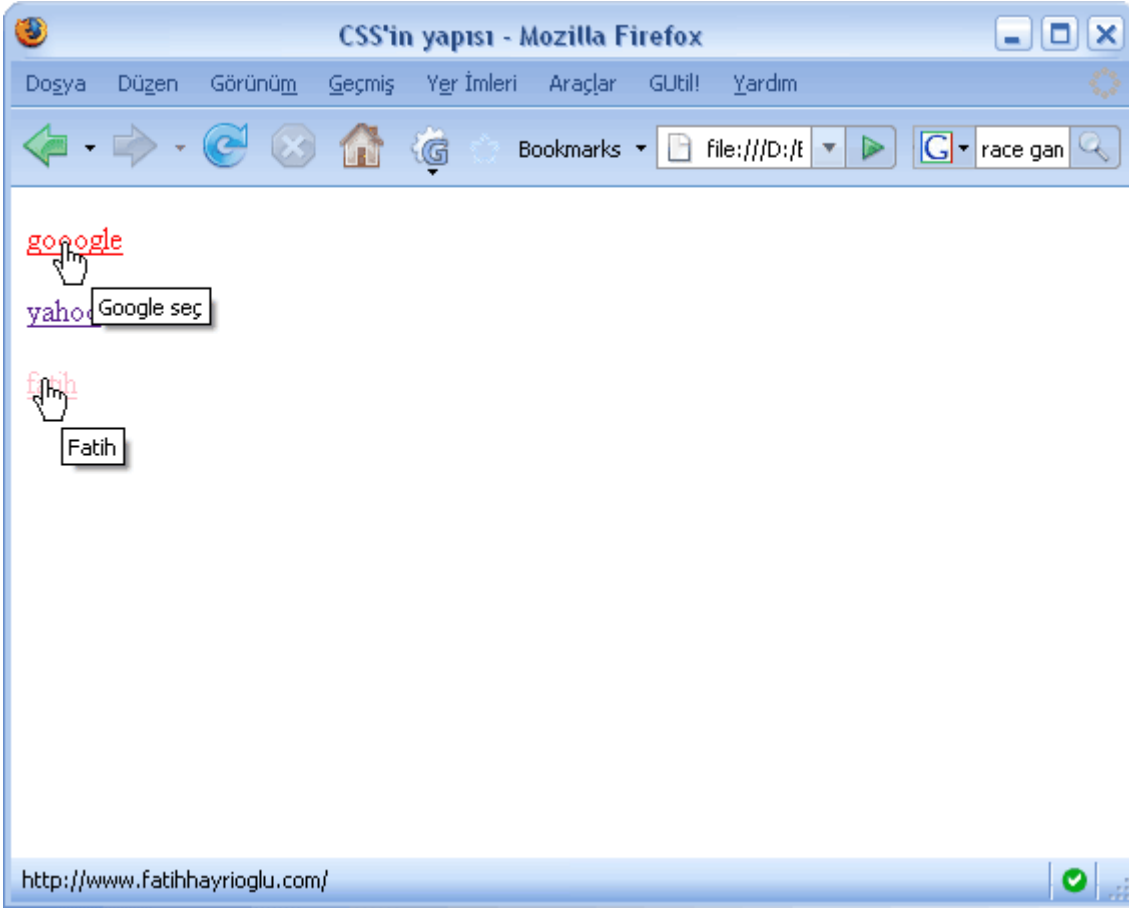
Sözde sınıfları birleştirmek

CSS2.1 ile birlikte aynı seçiciye ait sözde sınıfları birleştirme özelliğide gelmiştir. Örneğin ziyaret edilmiş linklerin :hover özelliği ile ziyaret

edilmemiş linklerin :hover özelliğini farklı atamak istesek.

Örnek 3-20

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'in yapısı</title>
<style type="text/css">
a:link:hover {
    color: red;
}
a:visited:hover {
    color: pink;
}
</style>
</head>
<body>
<p><a href="http://www.google.com" title="Google seç">google </a>
</p>
<p><a href="http://www.yahoo.com" title="Yahoo seç">yahoo </a>
</p>
<p><a href="http://www.fatihhayrioglu.com" title="Fatih">fatih </a>
</p>
</body>
</html>
```



Sıralama önemli değildir.

Tarayıcı Uyumu:

Internet Explorer 5.5, 6.0, 7.0 **desteklemiyor**
 Firefox 1+
 Safari 1.3+
 Opera 9.0: desteklemiyor
 CSS 1+

3.10.2. Sözde eleman seçicileri

Bu elemanlar hayali elemanlardır. (X)HTML hiyerarşisi içinde bu elemanlar yoktur. CSS2.1'de dört adet Sözde Eleman Seçicisi vardır: **first-letter**, **first-line**, **before** ve **after**

first-letter (ilk harf)

Bir blok-level elemanın ilk harfine stil tanımlaması yapmak için kullanılır. Örnek verecek olursak h1 elemanının baş harfinin büyük olması için kullanılabilir.

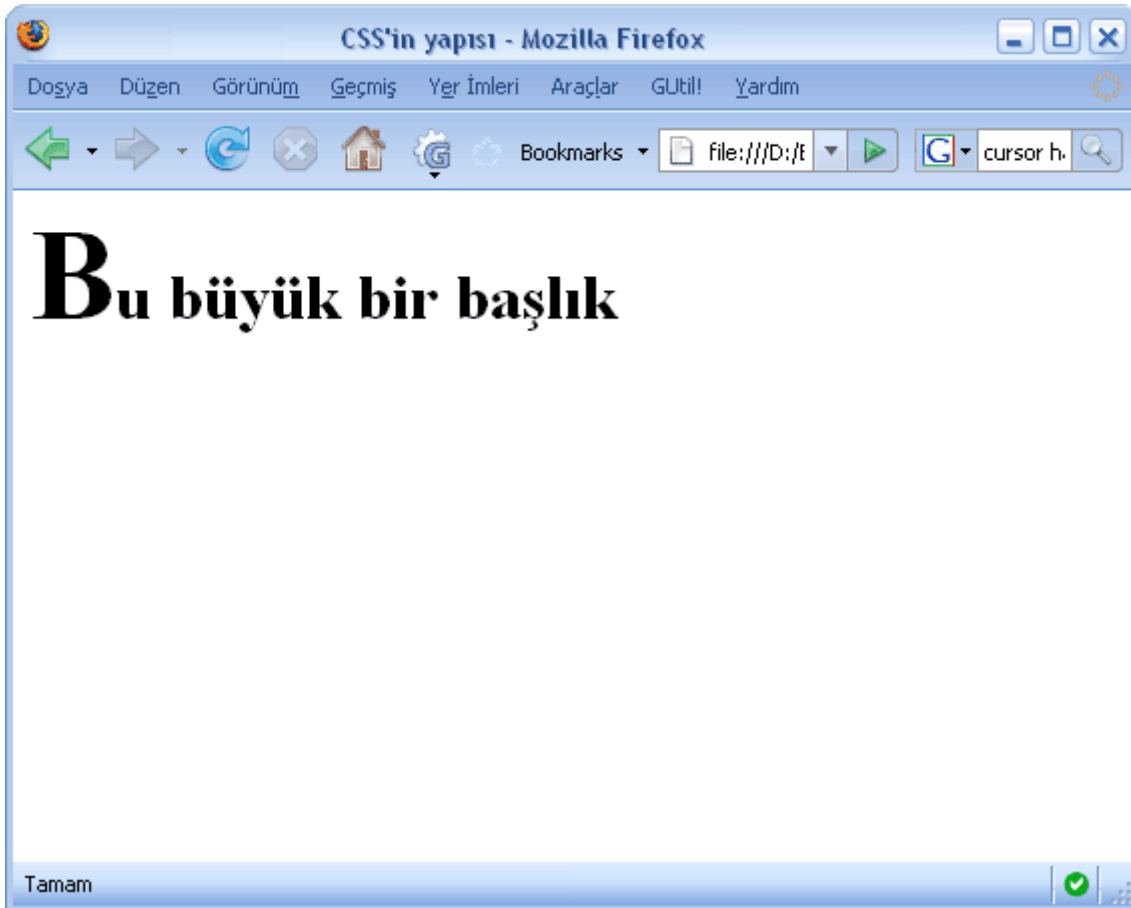
Örnek 3-21

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'in yapısı</title>
<style type="text/css">
h1:first-letter {
    font-size: 200%;
}
</style>
</head>
<body>
<h1>Bu büyük bir başlık</h1>
</body>
</html>

```



Tarayıcı Uyumu:

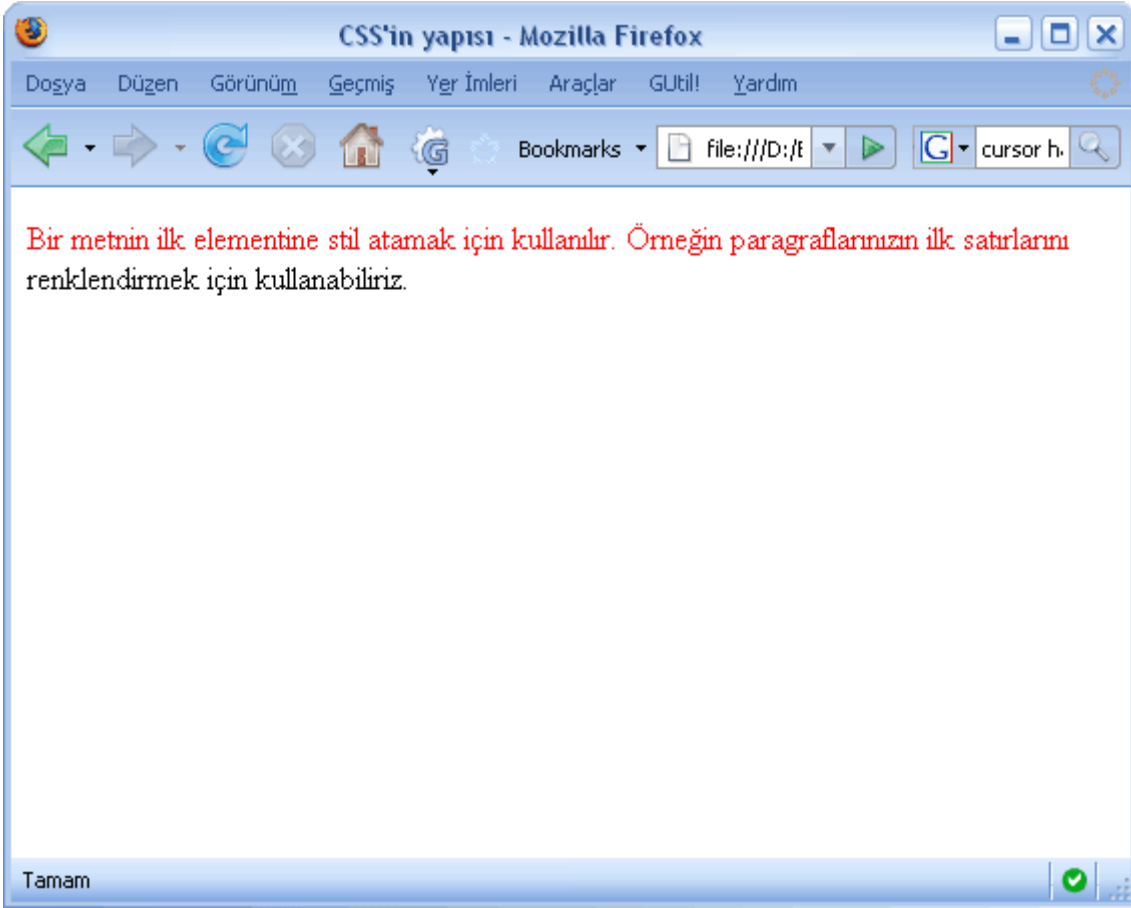
Internet Explorer 5.5, 6.0 **desteklemiyor** 7.0: destekliyor ;
 Firefox 1+
 Safari 1.3+
 Opera 5.0+
 CSS 1+

first-line (ilk satır)

Bir metnin ilk elemanına stil atamak için kullanılır. Örneğin paragraflarınızın ilk satırlarını renklendirmek için kullanabiliriz.

Örnek 3-22

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>CSS'in yapısı</title>
<style type="text/css">
p:first-line {
    color: red;
}
</style>
</head>
<body>
<p>Bir metnin ilk elemanına stil atamak için kullanılır. Örneğin
paragraflarınızın ilk satırlarını renklendirmek için kullanabiliriz.</p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 5.0+
 Firefox 1+(kısmen)
 Safari 1.3+
 Opera 5.0+
 CSS 1+

before ve after elemanları

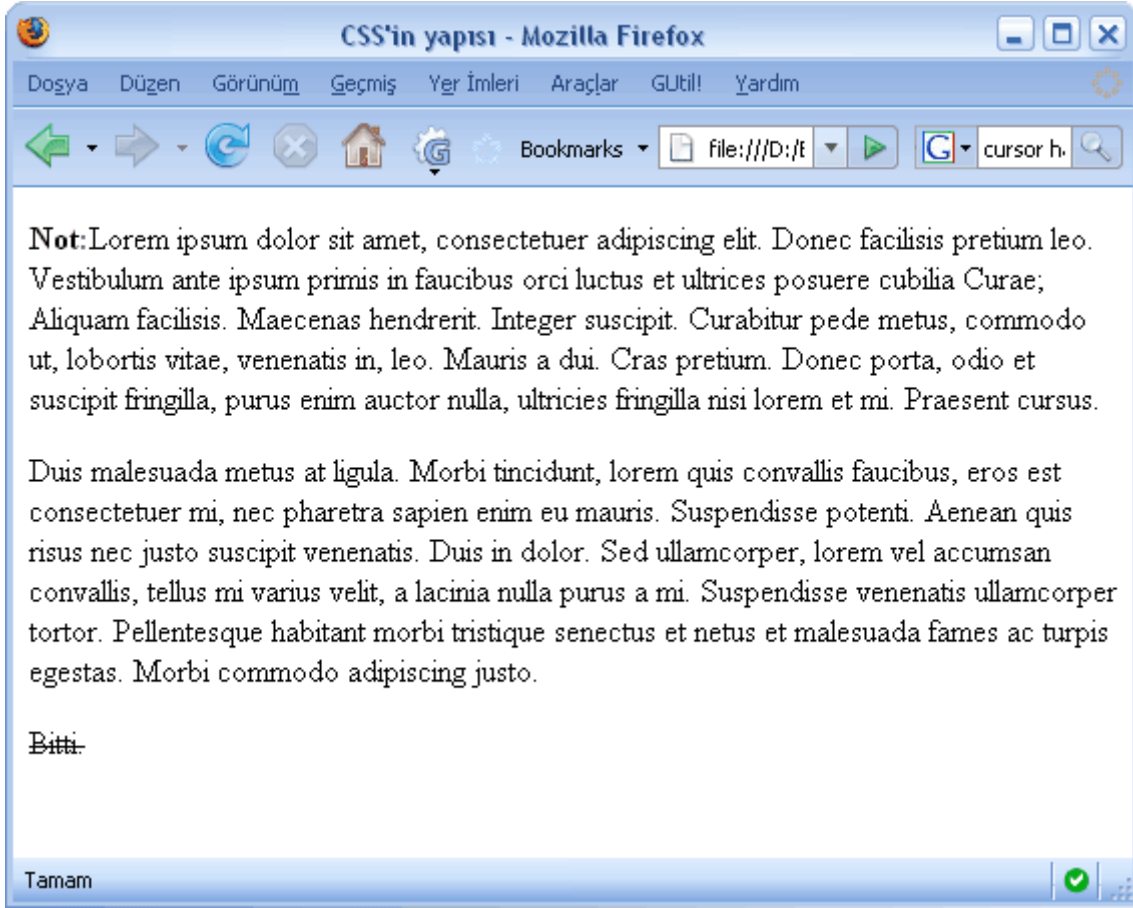
Bir elemanın öncesine ve sonrasına bir içerik veya özellik eklemek için kullanılır.

Notlarımızın başına otomatik olarak Not yazmak için

Örnek 3-23

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
```

```
<title>CSS'in yapısı</title>
<style type="text/css">
p.not:before {
    content: "Not: ";
    font-weight: bold;
}
body:after {
    content: "Bitti.";
    text-decoration: line-through;
}
</style>
</head>
<body>
<p class="not">Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Donec facilisis pretium leo. Vestibulum ante ipsum primis in faucibus orci
luctus et ultrices posuere cubilia Curae; Aliquam facilisis. Maecenas
hendrerit. Integer suscipit. Curabitur pede metus, commodo ut, lobortis
vitae, venenatis in, leo. Mauris a dui. Cras pretium. Donec porta, odio et
suscepit fringilla, purus enim auctor nulla, ultricies fringilla nisi lorem et
mi. Praesent cursus. </p>
<p>Duis malesuada metus at ligula. Morbi tincidunt, lorem quis convallis
faucibus, eros est consectetur mi, nec pharetra sapien enim eu mauris.
Suspendisse potenti. Aenean quis risus nec justo suscipit venenatis. Duis
in dolor. Sed ullamcorper, lorem vel accumsan convallis, tellus mi varius
velit, a lacinia nulla purus a mi. Suspendisse venenatis ullamcorper
tortor. Pellentesque habitant morbi tristique senectus et netus et
malesuada fames ac turpis egestas. Morbi commodo adipiscing justo.
</body>
</html>
```



Kodu yeterlidir. Aynı şekilde paragraflarımızın sonuna otomatik bitti yazıp nokta koymak istersek

Kodlaması bunu yapmak için yeterlidir.

Tarayıcı Uyumu:

Internet Explorer 5.5, 6.0,7.0 desteklemiyor
 Firefox 1+(kısmen)
 Safari 1.3+
 Opera 7.5+
 CSS 2+

4. Basamaklı Yapı ve Kalıtsallık

CSS tanımı yaparken stil şablonu olarak çevrilir genelde Türkçeye. Baştaki C(Cascade) harfi biraz dışarıda tutulur, biz burada bu kısma odaklanacağız. "C" yani **Cascading**, stil tanımlamada öncelik sırasındır. CSS'de bir elemana birden fazla tanım yapıldığında hangi özelliğin etkin olacağını belirlemek için belli bir kuralı vardır. Bu kurala göre hangi özellik etkin ise o tanım elemana etki yapar, diğer özellikler etkisiz kalır.

Basamaklı yapı, etkinlik ve kalıtsallık ile ilişkilidir.

4.1. CSS'de Tanımlamalar ve Etkinlikleri(Specificity)

CSS'de etkinliğin anlamı stil çatışması(aynı elemana birden fazla tanım yapıldığında) olduğunda kullanılan hangi kodun web tarayıcıları tarafından yorumlanacağını belirlemektir.

Örnek 4-1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Basamaklı Yapı ve Kalıtsallık</title>
<style type="text/css">
h1{
    color:red;
}

h1{
    color:blue;
}
</style>
</head>
<body>
<h1>Sed ut perspiciatis</h1>
</body>
</html>
```



Yukarıdaki örnekte görüldüğü gibi aynı HTML elemanına(h1) iki adet tanım yapılmıştır. Tarayıcı ekranında görünen son tanımlamadır. Eğer aynı nesneye tanımlanan aynı özellik var ise son yapılan geçerlidir.

Burada iki tanımlamanın yakınlık dereceleri aynıdır. O zaman hangisi uygulanacak. Böyle durumlarda web tarayıcıları kendilerine has garip bir hesaplama sistemi kullanırlar. Kuralları aşağıdaki gibidir.

- Kod içinde tanımlama yapılmışsa (satır içi-kod)
- Stil dosyasındaki tanımlı ID'lerin sayısına
- Sınıf tanımlamalarının sayısına
- Seçicilerin sayısına bakar

Web tarayıcısı bu sayıları toplamaz. Yalnızca rakamları yan yana sıralar. Bu web tarayıcılarına özel bir aritmetik işlemidir. Bir örnek verecek olursak id değeri 2, sınıf değeri 3 ve seçici değeri 0 olan bir tanımlamada

0 (kod içinde değil), 2 (id), 3 (class), 0(seçici) = 0230

Kod içinde tanımlama yapılmışsa (inline-kod) kısmı CSS2.1 ile gelen yeni bir özelliktir ve etkinlik değerini 4 basamağa çıkarmıştır. CSS2 ve öncesinde üç basamaklı hesaplama yapılmakta idi, kod içindeki tanımlamalar üç basamağa göre 1,0,0 olarak belirlenirdi.

Değerini elde ederiz. Burada id değeri 100'ler basamağına sınıf değeri 10'lar basamağına yazılarak hesaplama yapılır ve iki yüz otuz değeri elde edilir.

Örnek 4-2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Basamaklı Yapı ve Kalıtsallık</title>
<style type="text/css">
body h1{
    color:red;
}

h1{
    color:blue;
}
</style>
</head>
<body>
<h1>Sed ut perspiciatis</h1>
</body>
</html>
```



Bir önceki örnekte mavi olan başlığımız şimdi kırmızı oldu. Bunun nedeni etkinlik değerinin daha büyük olmasıdır.

```
Body h1 = 0002
h1 = 0001
0002 > 0001
```

Olduğu için başlık kırmızı olmuştur.

Büyük içerikli siteleri kodlarken birçok defa karşılaştığımız sorunlardan bir tanesi bir elemana aynı özelliğin birden fazla tanımlanması sonucu istemediğimiz sonuçlarla karşılaşmaktır. Bu birden fazla tanımla genelde genel bir tanım yaptıktan sonra o elemana özel tanımlamalar yapmak istediğimizde karşılaşırız. Bu sorunu aşmak için benim uyguladığım yöntem etkin olmasını istediğim elemanın etkinlik değeri arttırmak için tanımlama yaparken üst elemanlarını yazmak veya id atamak oluyor. Yukarıdaki örnekten gidersek.

Eğer başlığın mavi olmasını istiyorsam. Etkinlik derecesini arttırmak için başına **html body** eklerim

```
html body h1{ color:blue}
```

Şeklinde veya

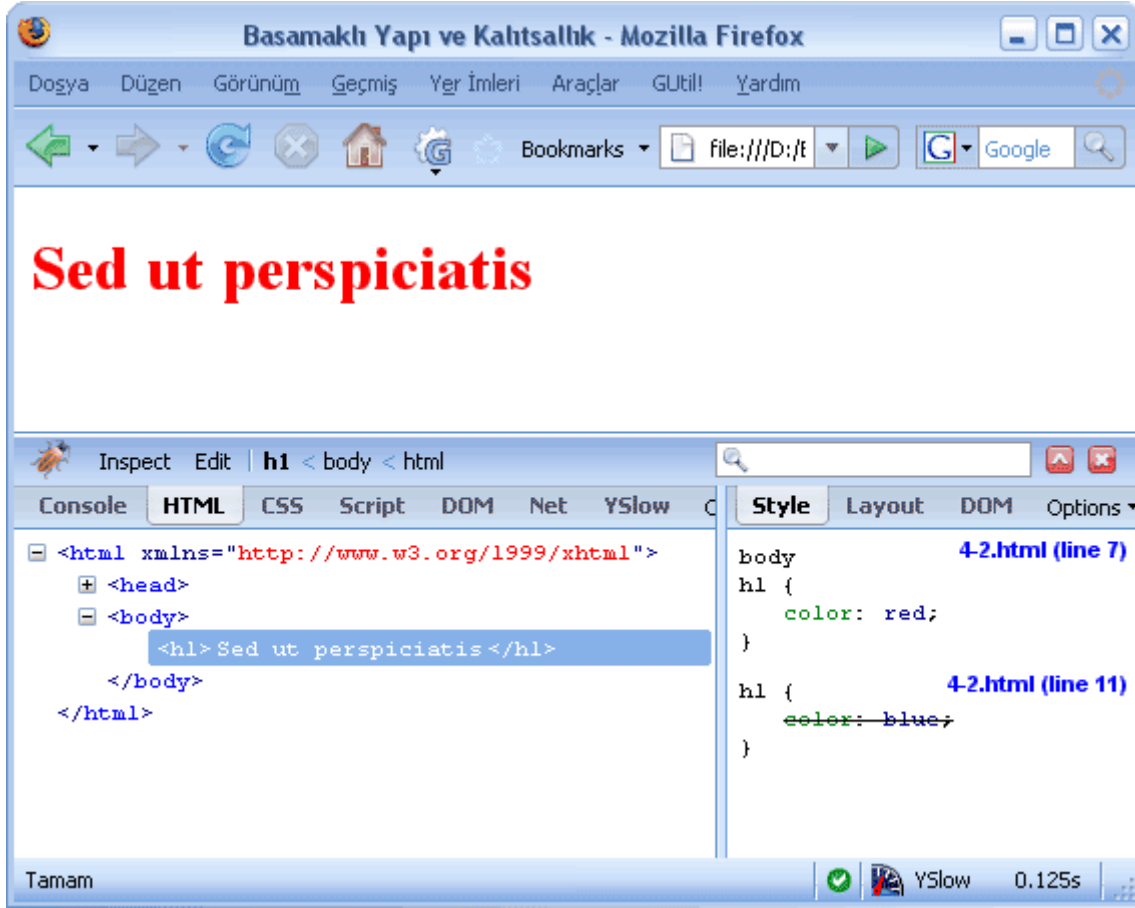
```
h1#baslik{ color:blue}
```

....

```
<h1 id="baslik">Sed ut perspiciatis</h1>
```

Peki site içeriği geniş ve css dokümanımızda uzunca biz bu üst üste basamaklı olarak binmiş tanımları nasıl anlayacağız. Bunun için size Firefox tarayıcı ile gelen FireBug(<http://www.getfirebug.com/>) eklentisini öneriyorum. Bu eklentini birçok özelliği var bu sadece biri.

Firebug eklentisi kurup etkinleştirdikten sonra sağ alt köşeye eklenen yeşil ok işaretine tıklayıp eklenti ekranını açarız ve soldaki Inspect'e tıkladıktan sonra sayfamızdan h1 elemanımızı seçeriz.



İkinci örnek üzerinde yaptığımız Firebug araştırmasında görüldüğü gibi 11.satırdaki h1 tanımımızın üzeri çizilmiştir, yani etkisizdir. Etkili olan 7. Satırdaki h1 tanımımızdır.

Etkinlik değerini anlamak için birkaç örnek daha verelim.

```
ul li.secili {color: yellow}
```

Yukarıdaki tanımlama da 0013 değerini elde ederiz.

0 (kod içinde değil), 0 (id), 1 (secili sınıfı), 3 (ul h1 li seçicileri) = 0013

h1 {color: blue}

Bu tanımlamanın değeri 0001'dir. CSS belirlemeye göre yüksek değer alanın özelliği uygulanır. Her ikisinde de h1'e bir tanımlama yapılmıştır. Ancak tanımlama sistemleri farklıdır. XHTML kodumuzdaki **h1** elemanını mavi değil **sarı**(yellow) olarak gösterecektir.

Seçici	Seçici Tipi	Etkinlik değeri
*	Evrensel seçiciler	0000 (a = 0, b = 0, c = 0, d = 0)
li	Eleman adı	0001 (a = 0, b = 0, c = 0, d = 1)
ul li	Eleman adı	0002 (a = 0, b = 0, c = 0, d = 2)
div h1 + p	Eleman adı	0003 (a = 0, b = 0, c = 0, d = 3)
input[type='text']	Eleman adı + özellik	0011 (a = 0, b = 0, c = 1, d = 1)
.siniftanimi	Sınıf adı	0010 (a = 0, b = 0, c = 1, d = 0)
div.siniftanimi	Eleman adı + sınıf tanımı	0011 (a = 0, b = 0, c = 1, d = 1)
#tekiltanim	Tekil seçici adı	0100 (a = 0, b = 1, c = 0, d = 0)
div#tekiltanim	Eleman adı + tekil seçici adı	0101 (a = 0, b = 1, c = 0, d = 1)
style (attribute)	Satır içi style tanımı	1000 (a = 1, b = 0, c = 0, d = 0)

4.2. !important

Bir de öncelik için **!important** tanımlaması vardır. Eğer yukarıdaki koda **!important** eklersek

h1 {color: blue !important;}

Tanımlama yukarıdaki gibi değiştirildiğinde h1 elemanı mavi renkte görünecektir.

Eğer değer aynı olursa(mesela 0013 e 0013) ne olur, bu durumda son yapılan tanımlamanın özellikleri uygulanır.

4.3. Kalıtsallık(Inheritance)

XHTML dokümanlarının yapısından ve CSS ile olan ilişkisinden bir önceki konuda bahsetmiştik. XHTML öğeleri bir biri ile bir ailenin birbiri ile olan bağı gibi bağlıdır. Hatırlıyorsanız bir soy ağacı benzetmesi yapmıştık. İşte bu soy ağacının öğeleri arasında bir kalıtsallık söz konusudur. Ebeveynden çocuğuna geçen özellikler gibi XHTML öğeleri arasında CSS özellikleride kalıtsal olarak alt elemanları(çocuk eleman) etkiler. Kalıtsallık; CSS ile web sayfası kodlarken tüm elemanlara tek tek atama yapabildiğimiz gibi birde bir elemana atama yaparız ve bu özellik tüm alt(çocuk elemanları) elemanlarınada uygulanır buna kalıtsallık denir. CSS'in birçok özelliğinin kalıtsallık özelliği vardır. Kitabın ilerleyen sayfalarında CSS özellikleri verilirken her özelliğin kalıtsallık özelliği hakkında bilgi verilecektir.

Bazı CSS özelliklerinde kalıtsallık yoktur. Örneğin margin, padding ve border özellikleri gibi tüm liste sağda

Örnek bir kodlama yaparsak:

Örnek 4-3

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Basamaklı Yapı ve Kalıtsallık</title>
<style type="text/css">
div#icerik {
text-align: center;
}
</style>
</head>
<body>
<div id="icerik">Buradaki içerik ortalancaktır <p>text-align özelliğinin
kalıtsallığı olduğu için buradaki içerikte ortalancaktır </p></div>
</body>
</html>
```



Kalıtsallık kodlama yaparken bazen bize yararlı olur, bazende sayfalarımızda sorun çıkmasına neden olur. Bundan dolayı elemanların kalıtsallığı olup olmadığını bilmeliyiz. Yoksa gerçekten zor durumlarda kalabilirsiniz, tecrübeyle sabit :)

Direk atamalar kalıtsallığı yok eder. Örneğin yukarıda örnekte

```
p{
text-align: left;
}
```

Tanımlaması ile üst eleman olan **#icerik**'den kalıtsal olarak gelen **text-align: center;** tanımlamasını etkisiz kılar.

NN4x ve IE5x ve alt sürümlerinde **body** içerisindeki **font** tanımlamasının **table** elemanına etki etmemesi gibi bir sorunumuz vardır.

```
body {font-size: 0.8em;}
```

Tanımlaması NN4x ve IE5x'de kodlama içindeki tabloları etkilemeyecektir. Bunun için

```
body, td, th {font-size: 0.8em;}
```

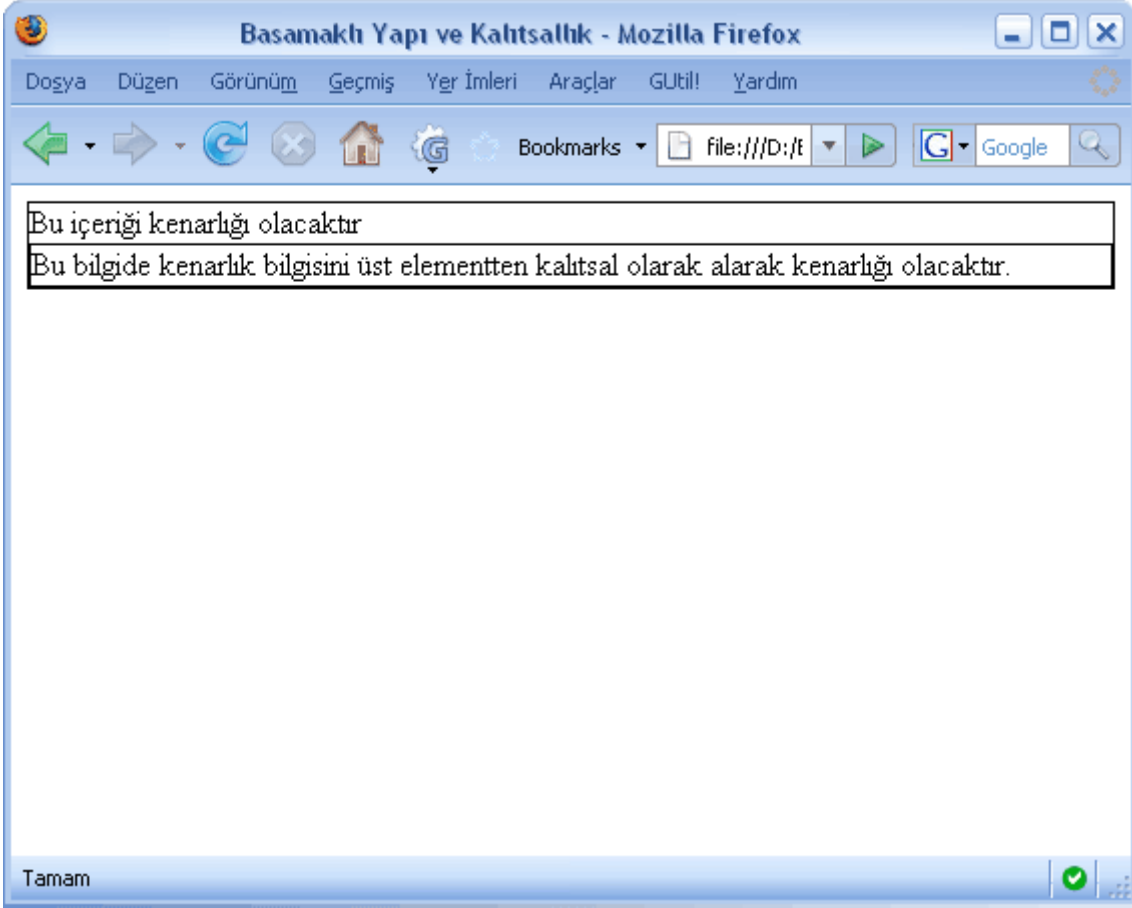
Şeklinde bir tanımlama daha uygun bir tanımlama şeklidir.

4.3.1. inherit (kalıtsallık)değeri

CSS2 ile birlikte CSS özelliklerine *inherit* değeri atamamıza olanak sağlamaktadır. Tüm elemanlara(kalıtsallık özelliği olmayanlar dâhil) *inherit* değeri atayabiliriz. *inherit* değeri atadığımız eleman üst elemanın değeri alacaktır. Bir örnek yapacak olursak:

Örnek 4-4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Basamaklı Yapı ve Kalıtsallık</title>
<style type="text/css">
div#icerik{
    border: 1px solid #000;
}
div#bilgi {
    border: inherit;
}
</style>
</head>
<body>
<div id="icerik">Bu içeriği kenarlığı olacaktır
<div id="bilgi">Bu bilgi de kenarlık bilgisini üst elemandan
kalıtsal olarak alarak kenarlığı olacaktır. </div>
</div>
</body>
</html>
```



Kalıtsallığı kullanarak kodlarımızı azaltabiliriz. Ancak kalıtsallık nedeni ile de birçok sorunla karşılaşacağımızı unutmamalıyız.

Internet Explorer 7 ve alt sürümlerinde **inherit** değerini desteklememektedir.

5. Yazıtipi(Font) Özellikleri

Yazıtipi özellikleri sayfalarımızdaki metinlerin yazıtipi ailesini, kalınlık ayarlarını, boyutunu, büyük-küçük harf olmasını ve stilini değiştirmemizi sağlar. CSS yazıtipleri üzerinde tam hâkim olmamızı sağlar.

- Renk (color) özelliği
- Yazıtipi ailesi (font-family)
- Yazıtipi boyutu (font-size)
- Yazıtipi kalınlığı (font-weight)
- Yazıtipi biçemi (font-style)
- Yazıtipi farklılıkları (font-variant)
- Yazıtipi (font)

5.1. Renk (color) özelliği

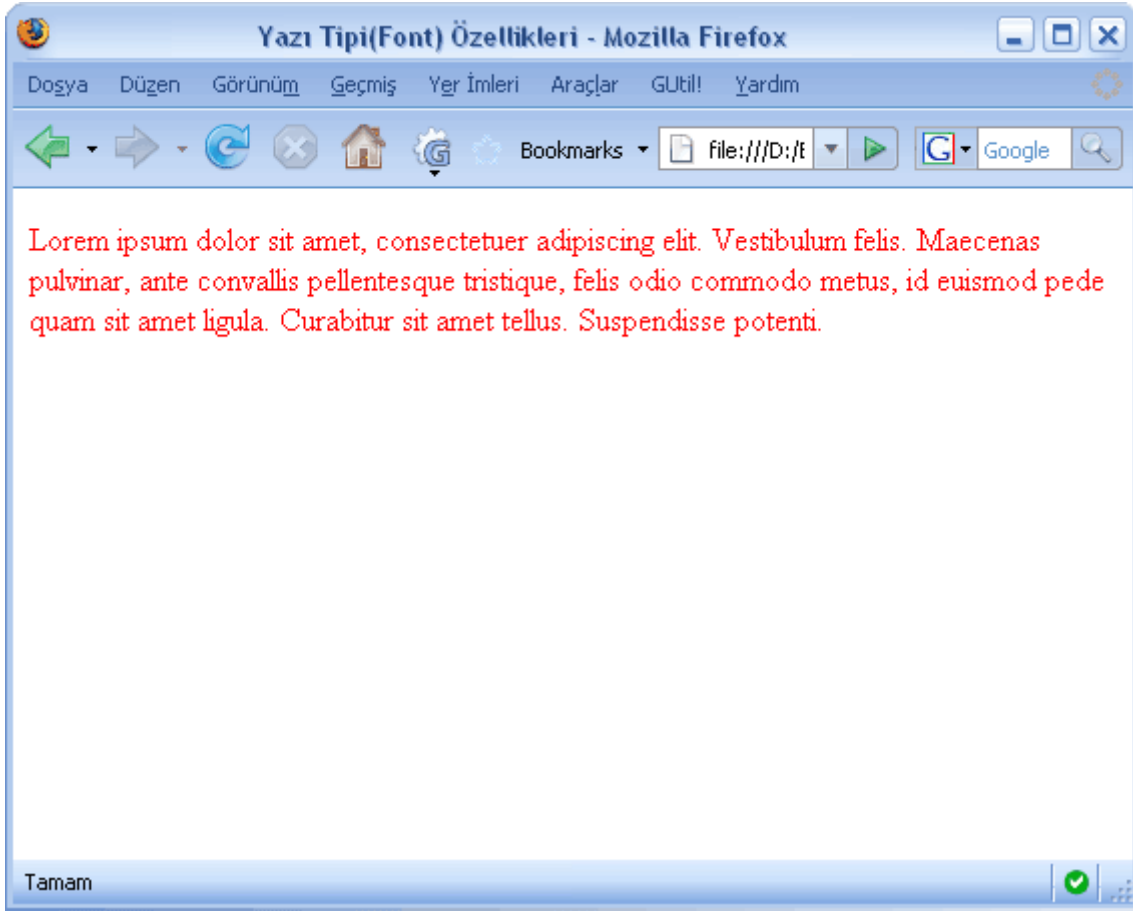
Yapısı: **color: <değer>**
 Aldığı Değerler: **<renk>** | kalıtsallık
 Başlangıç değeri: **web tarayıcısı belirler**
 Uygulanabilen elemanlar: **tüm elemanlar**
 Kalıtsallık: **Var**

Renk tanımlaması metinlerimizin rengini belirlemek için kullanılır.

Örnek 5-1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Yazı Tipi(Font) Özellikleri</title>
<style type="text/css">
p{
    color: #f00; /* kırmızı renk kodu */
}
</style>
</head>
<body>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum
felis. Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
```

```
amet tellus. Suspendisse potenti.</p>
</body>
</html>
```



CSS3 ile birlikte 147 renk tanımı gelmektedir. Tüm tarayıcılar bu renkleri desteklemektedir.

CSS3 ile birlikte RGBA (RGB with Alpha channel) renk tanımında eklenmiştir. Renk tanımı ile birlikte alfa kanal saydamlık değeri de atanabilecek.

Tarayıcı Uyumu:

Internet Explorer 3+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

5.2. Yazıtipi ailesi (font-family)

Yapısı: font **family** : [[<yazıtipi aile ismi > | <soysal aile ismi >],]*
[<yazıtipi aile ismi > | <soysal aile ismi >] | inherit

Aldığı Değerler:

**<yazıtipi aile ismi >- herhangi bir yazıtipi ailesi ismi kullanılabilir.
<soysal aile ismi >**

- serif (**örn: Times**)
- sans-serif (**örn: Arial veya Helvetica**)
- cursive (**örn: Zapf-Chancery**)
- fantasy (**örn: Western**)
- monospace (**örn: Courier**)

Başlangıç değeri: **web tarayıcısı belirler**
Uygulanabilen elemanlar: **tüm elemanlar**
Kalıtsallık: **Var**

Metinlerin kullanılacağı yazıtipi ailesini belirlemek için kullanılır. Birden fazla yazıtipi ailesi kullanılacaksa aralarına virgül (,) koyulur. Birden fazla yazılmasının nedeni birinci yazıtipi ailesi eğer kullanıcının bilgisayarında yok ise alternatif bir yazıtipi sunmak. Genelde son yazıtipi ailesi olarak soysal yazıtipi ailesi yazılır.

Soysal yazıtipi aileleri	Benzeyenler
serif	Times, Times New Roman
sans-serif	Helvetica, Arial
cursive	Zapf-Chancery
fantasy	Western

Eğer yazıtipi ismi bir den fazla kelimedenden oluşuyorsa çift tırnak içine alınır.

Örnek 5-2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Yazı Tipi(font) Özellikleri</title>
<style type="text/css">
p{
font-family: Verdana, Arial, Helvetica, sans-serif;
}
h1{
font-family: Georgia, "Times New Roman", serif;
}
</style>
</head>
<body>
<h1>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </h1>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
```

```
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 3+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

5.3. Yazıtipi boyutu (font-size)

Yapısı: **font-size:** <kesin değerler > | <göreceli değerler > | <uzunluk> | <yüzde>

Aldığı Değerler:

- <kesin değerler >
 - xx-small | x-small | small | medium | large | x-large | xx-large
- <göreceli değerler >
 - larger | smaller

- **<uzunluk>**
- **<yüzde>** (üst elemanlara(ebeveyn) bağlı olarak)

Başlangıç değeri: **medium**

Uygulanabilen elemanlar: **tüm elemanlar**

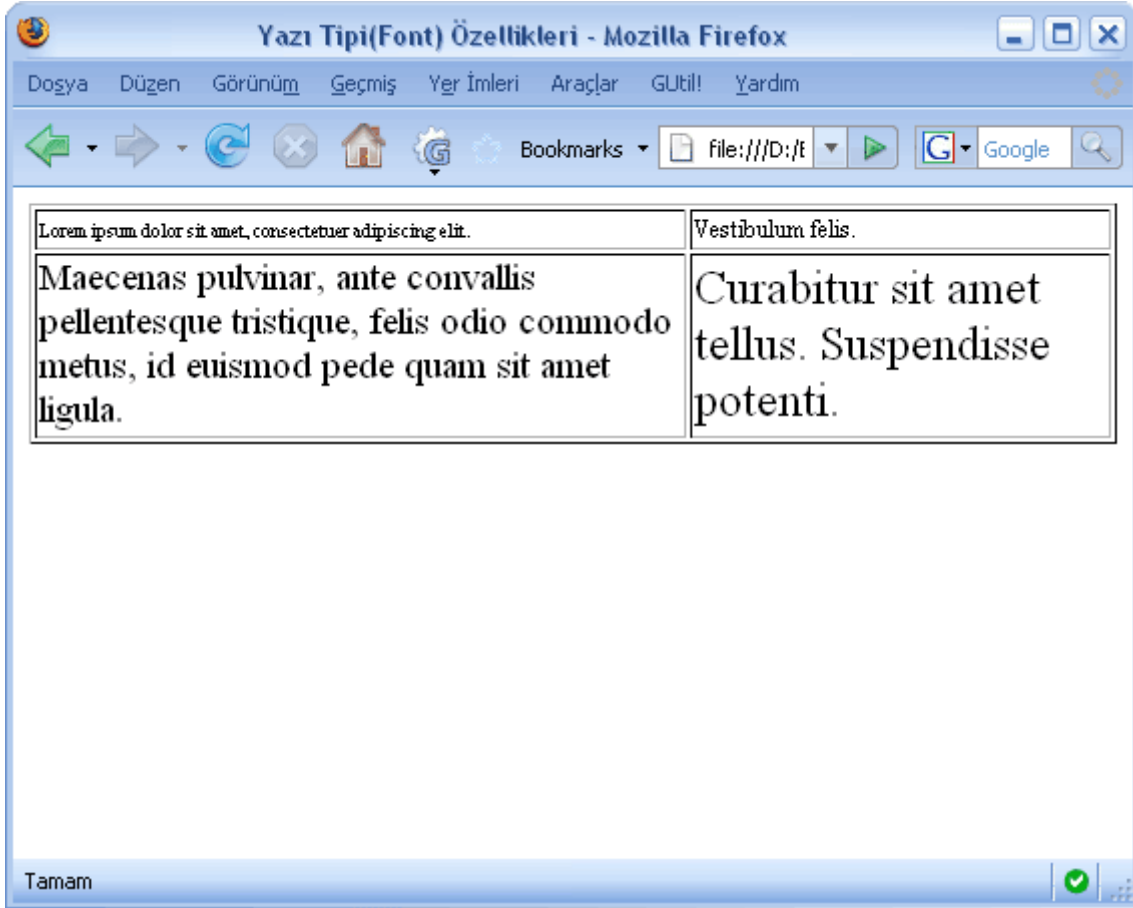
Kalıtımsallık: **Var**

CSS, yazı tipi boyutlandırılmada daha esnek tanımlama yapmamızı olanak sağlar. Mesela 11px değeri HTML'de 2 ve 3 değeri arasında bir değere denk geliyor ve HTML yazıtipi elemanı ile birlikte kullanılamıyordu. CSS'de yazı tipi boyutunu belirlemek için birçok birim kullanılır. Örn: px, em, pt, in, cm vd.

Örnek 5-3

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Yazı Tipi(font) Özellikleri</title>
<style type="text/css">
td.normal{
    font-size : 12px;
}
td.buyuk {
    font-size : 120%;
}
td.kucuk {
    font-size : xx-small;
}
td.enbuyuk {
    font-size : x-large;
}
</style>
</head>
<body>
<table width="100%" border="1">
<tr>
<td class="kucuk">Lorem ipsum dolor sit amet, consetetuer
adipiscing elit.</td>
<td class="normal">Vestibulum felis.</td>
</tr>
<tr>
<td class="buyuk">Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. </td>
<td class="enbuyuk">Curabitur sit amet tellus. Suspendisse
```

```
potenti.</td>
</tr>
</table>
</body>
</html>
```



Windows ortamındaki Internet Explorer 5.5 ve alt sürümlerinde başlangıç değeri **medium** yerine **small**'dir.

Tarayıcı Uyumu:

Internet Explorer 3+
Firefox1+
Safari 1.3+
Opera 3.6+
CSS 1+

5.4. Yazıtipi kalınlığı (font-weight)

Yapısı: **font-weight: <değer>**

Aldığı Değerler: **normal** | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

Başlangıç değeri: **normal**

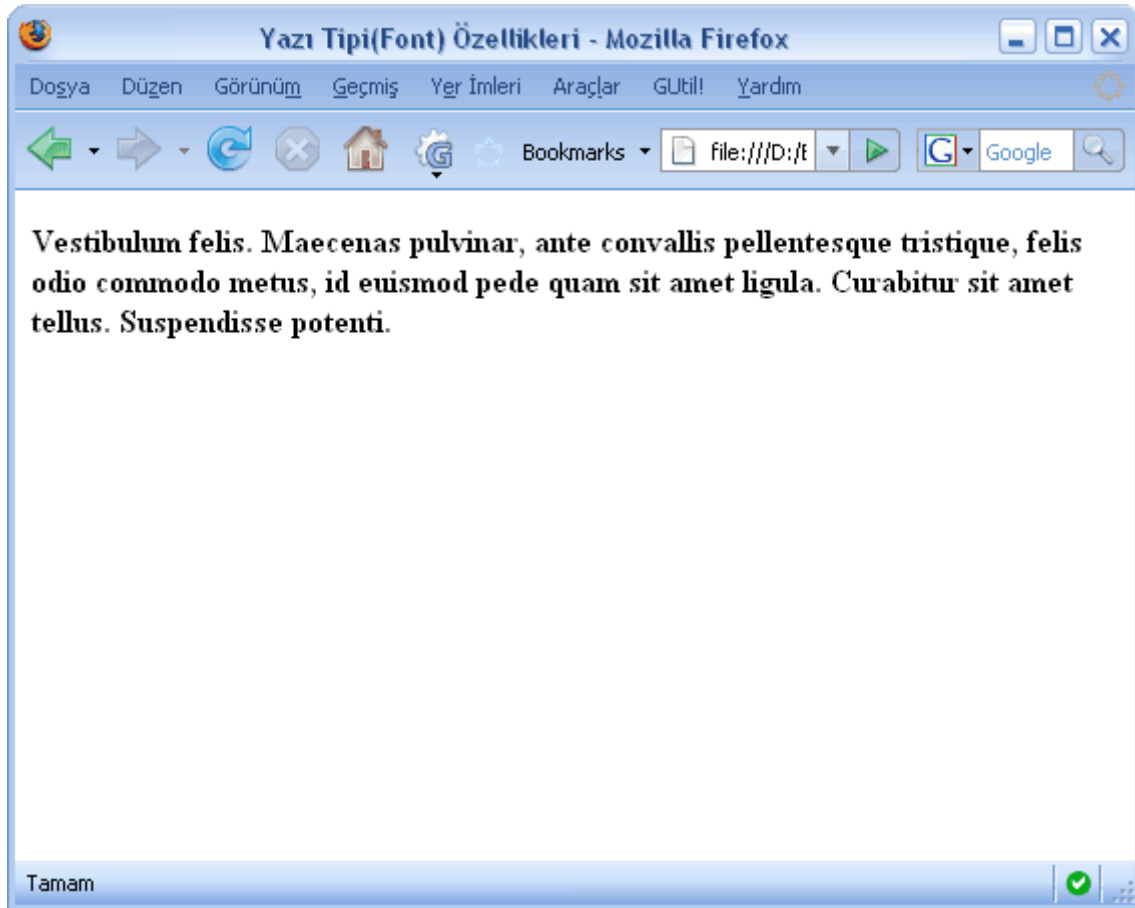
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtsallık: **Yok**

Yazıtipinin kalınlık incelik durumunu belirler. **100(ince)-900(kalın)** arasında bir değer alabildiği gibi kalın(bold), çok kalın(bolder) ve normal değerlerini de alır.

Örnek 5-4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Yazıtipi(font) Özellikleri</title>
<style type="text/css">
p{
font-weight: bolder;
}
</style>
</head>
<body>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 3+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

5.5. Yazıtipi biçemi (font-style)

Yapısı: **font-style: <değer>**

Aldığı Değerler: **normal | italic | oblique | inherit**

Başlangıç değeri: **normal**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Var**

Yazıtipinin eğik(italiktik) olup olmasını belirler.

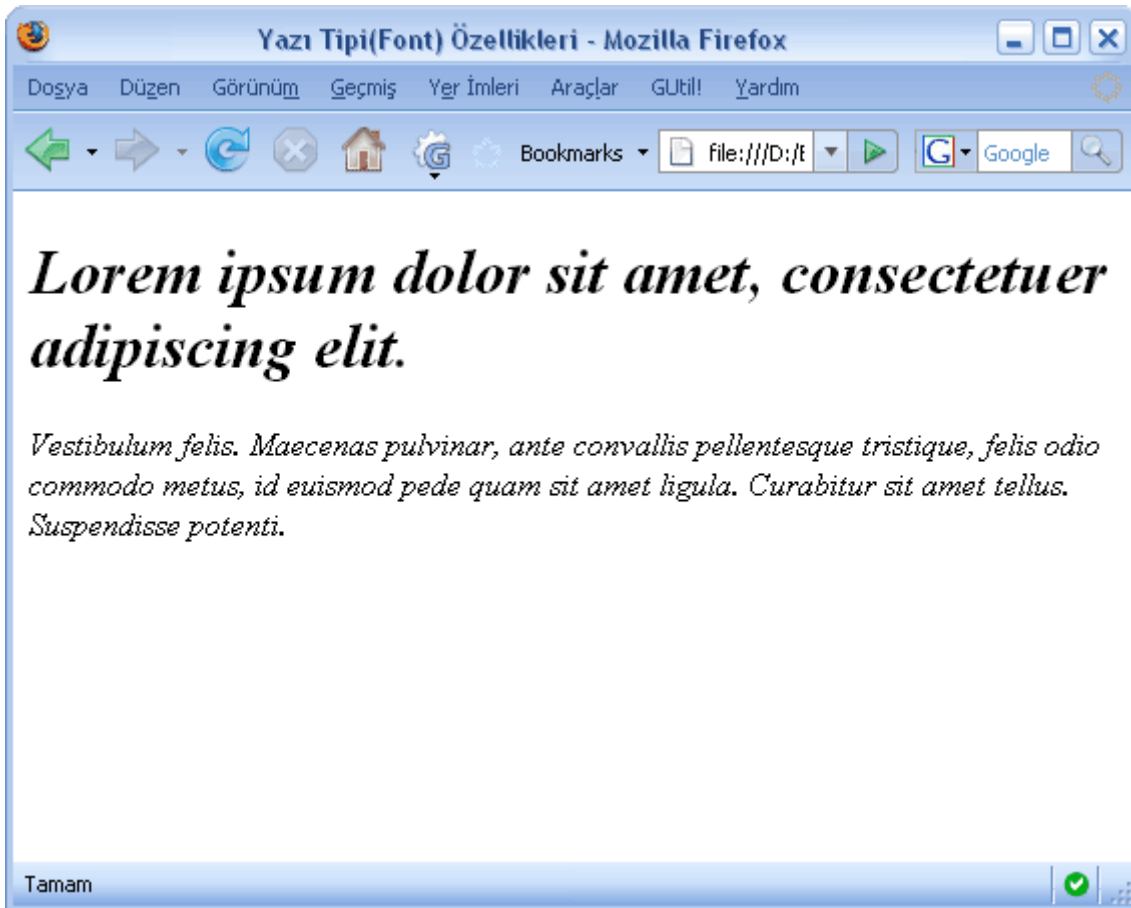
Örnek 5-5

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Yazıtipi(font) Özellikleri</title>
<style type="text/css">
h1{
    font-style: oblique;
}
p {
    font-style: italic;
}
</style>
</head>
<body>
<h1>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </h1>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
</body>
</html>

```



Tarayıcı Uyumu:

Internet Explorer 3+

Firefox 4+

Safari 1.3+
Opera 3.6+
CSS 1+

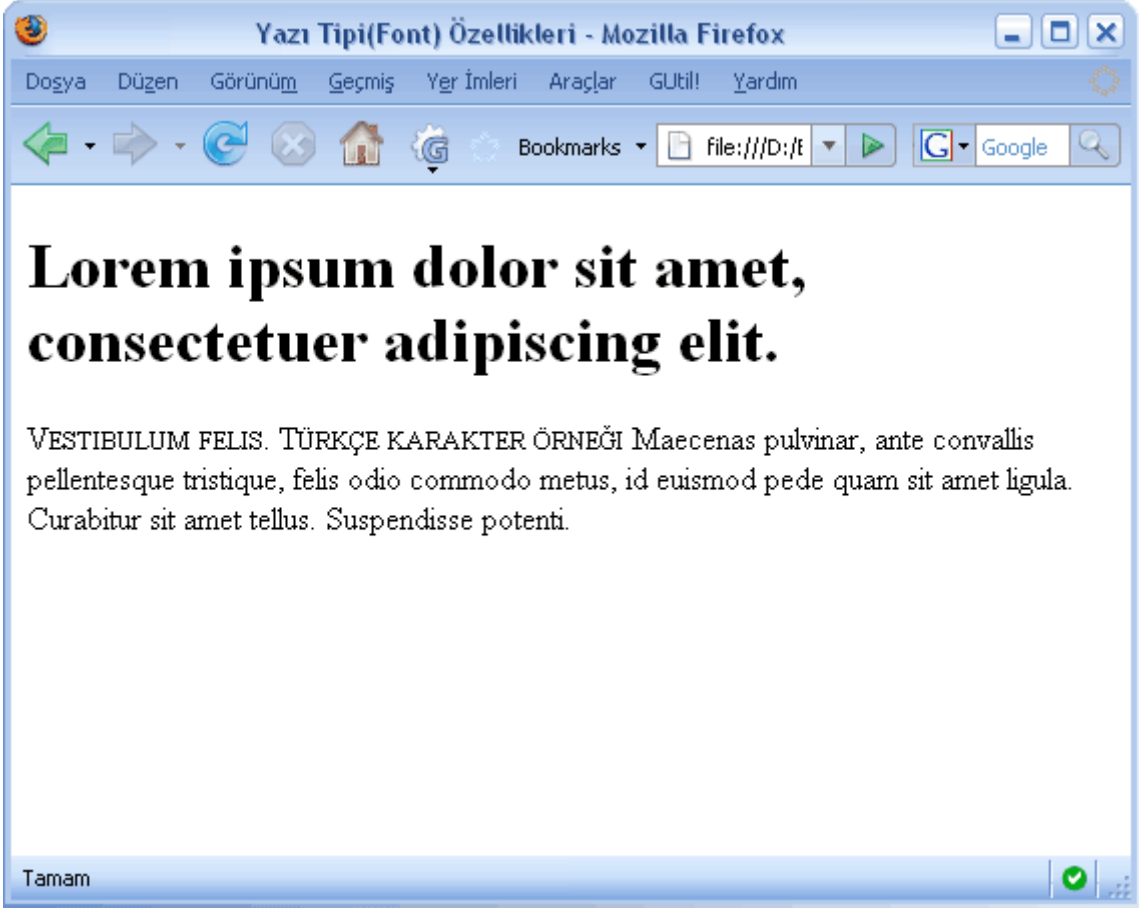
5.6. Yazıtipi farklılıkları (font-variant)

Yapısı: **font-variant: <değer>**
Aldığı Değerler: **normal | small-caps | inherit**
Başlangıç değeri: **normal**
Uygulanabilen elemanlar: **tüm elemanlar**
Kalıtsallık: **Var**

Metnin küçük-büyük harf ile gösterilmesini belirler. İki değer alır. **normal | small-caps.**

Örnek 5-6

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Yazı Tipi(Font) Özellikleri</title>
<style type="text/css">
span{
font-variant: small-caps;
}
</style>
</head>
<body>
<h1>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </h1>
<p><span>Vestibulum felis. Türkçe karakter örneği</span> Maecenas
pulvinar, ante convallis pellentesque tristique, felis odio commodo metus,
id euismod pede quam sit amet ligula. Curabitur sit amet tellus.
Suspendisse potenti.</p>
</body>
</html>
```



Windows ortamında Internet Explorer 7 ve alt sürümlerinde **font-variant** değeri **small-caps** atandığında **text-transform**'a atanan **lowercase** ve **uppercase** değerleri **none** gibi algılanacaktır.

Safari 3 ve alt sürümlerde bu özelliği desteklemez.

Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari **desteklemiyor**
 Opera 3.6+
 CSS 1+

Türkçe karakterlerde utf-8 karakter kodunda "i" de sorun çıkıyor.

5.7. Yazıtipi (font)

Yukarda anlatılan yazıtipi özelliklerinin hatta ek olarak **line-height** değerinde tek seferde tanımlamak için kullanılır.

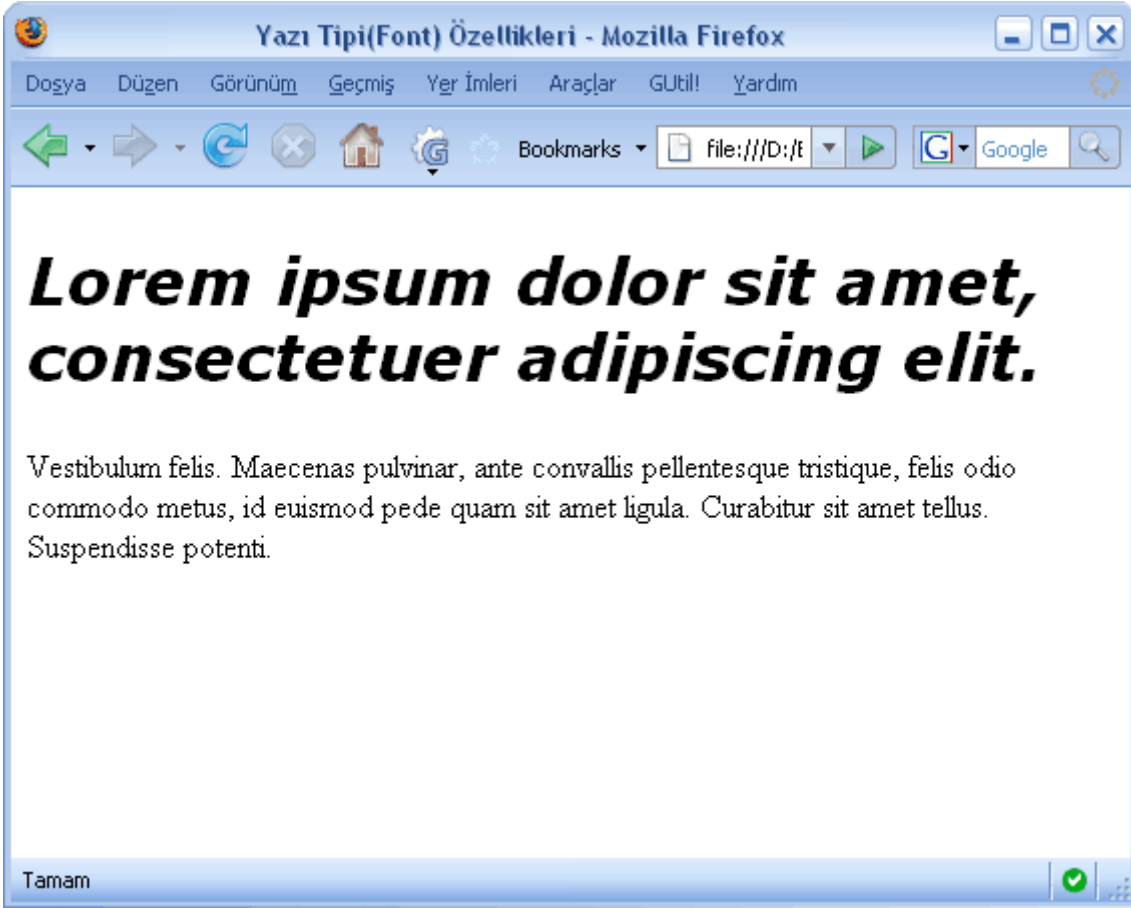
Yapısı: **font: <değer>**

Aldığı Değerler: [**<font-style> || <font-variant> || <font-weight>]? <font-size> [/ <line-height>]? <font-family>**

Başlangıç değeri: **tanımsız**
 Uygulanabilen elemanlar: **tüm elemanlar**
 Kalıtsallık: **Var**

Örnek 5-7

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Yazı Tipi(Font) Özellikleri</title>
<style type="text/css">
h2{
font:bold italic 200%/1.2em Verdana, Helvetica, Arial, sans-
serif;
}
</style>
</head>
<body>
<h2>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </h2>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 3+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

6. Metin(Text) Özellikleri

Yazı tipi(font) özellikleri ile Metin(Text) özellikleri arasındaki fark; yazı tipi özellikleri metni oluşturan karakterlerin nasıl olacağını belirlerken, Metin(Text) özellikleri sayfadaki metinlerin düzenini kontrol eder. Metin özellikleri metin düzeni için çok avantajlı özellikler getirmiştir.

- Metin Girintileme (text-indent)
- Hizalama (text-align)
- Dikey hizalama (vertical-align)
- Satırlar arası yükseklik (line-height)
- Kelimeler arası mesafe (word-spacing)
- Harfler arası mesafe (letter-spacing)
- Harf büyüklüğü (text-transform)
- Metin süsleme (text-decoration)
- Metin gölgeleme (text-shadow)
- Elemanlar arası boşluk (white-space)
- Metin yazma yönü(direction)
- Çok yönlülük algoritması(unicode-bidi) özelliği

6.1. Metin Girintileme (text-indent)

Paragrafların ilk cümlelerin soldan içeriye kaydırılması için kullanılan bir özelliktir.

Yapısı: **text-indent: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | inherit**

Başlangıç değeri: **0**

Uygulanabilen elemanlar: **Blok seviyesinden elemanlar**

Kalıtımsallık: **Var**

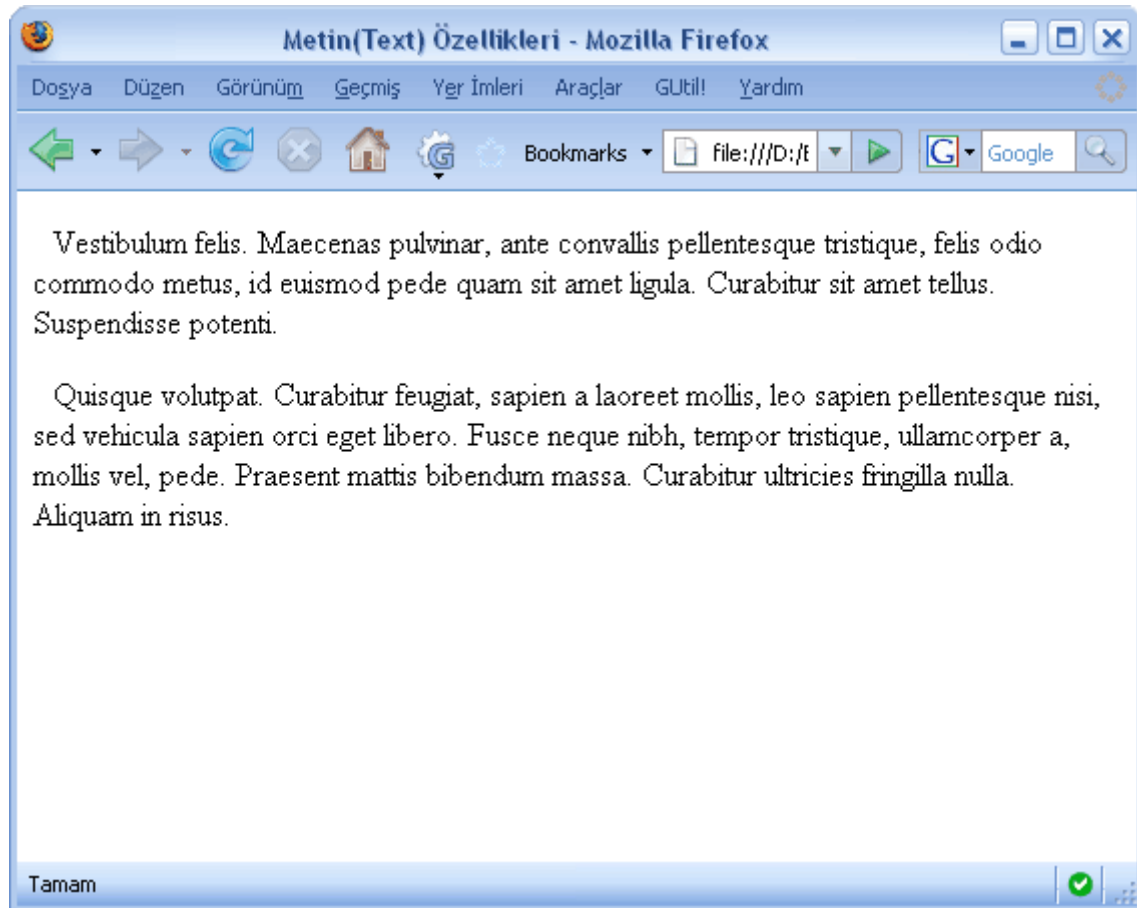
Örnek 6-1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Metin(Text) Özellikleri</title>
<style type="text/css">
```

```

p {
  text-indent: 10px;
}
</style>
</head>
<body>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce
neque nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent
mattis bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in
risus. </p>
</body>
</html>

```



Windows ortamında Internet Explorer 5.5 ve altı sürümlerinde satır içi elemanlara girinti özelliği eklemek için **width** değeri **auto** olarak atanmalıdır.

Opera uçuk değerleri yorumlamıyor. Örneğin resimli düğme(button) yaparken metni yok etmek için -999px; atması yaptığınızda Opera bunu algılamayacaktır.

Tarayıcı Uyumu:

Internet Explorer 3+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

6.2. Metin hizalama (text-align)

Bir elemanın diğerlerine göre hizasını belirlemek için kullanılır. Sadece metinler için kullanılmaz, diğer elemanlar içinde kullanılır.

Yapısı: **text-align: <değer>**

Aldığı Değerler: **left | center | right | justify | inherit**

Başlangıç değeri: **kullanıcı arayüzüne ve yazım yönüne bağlı**

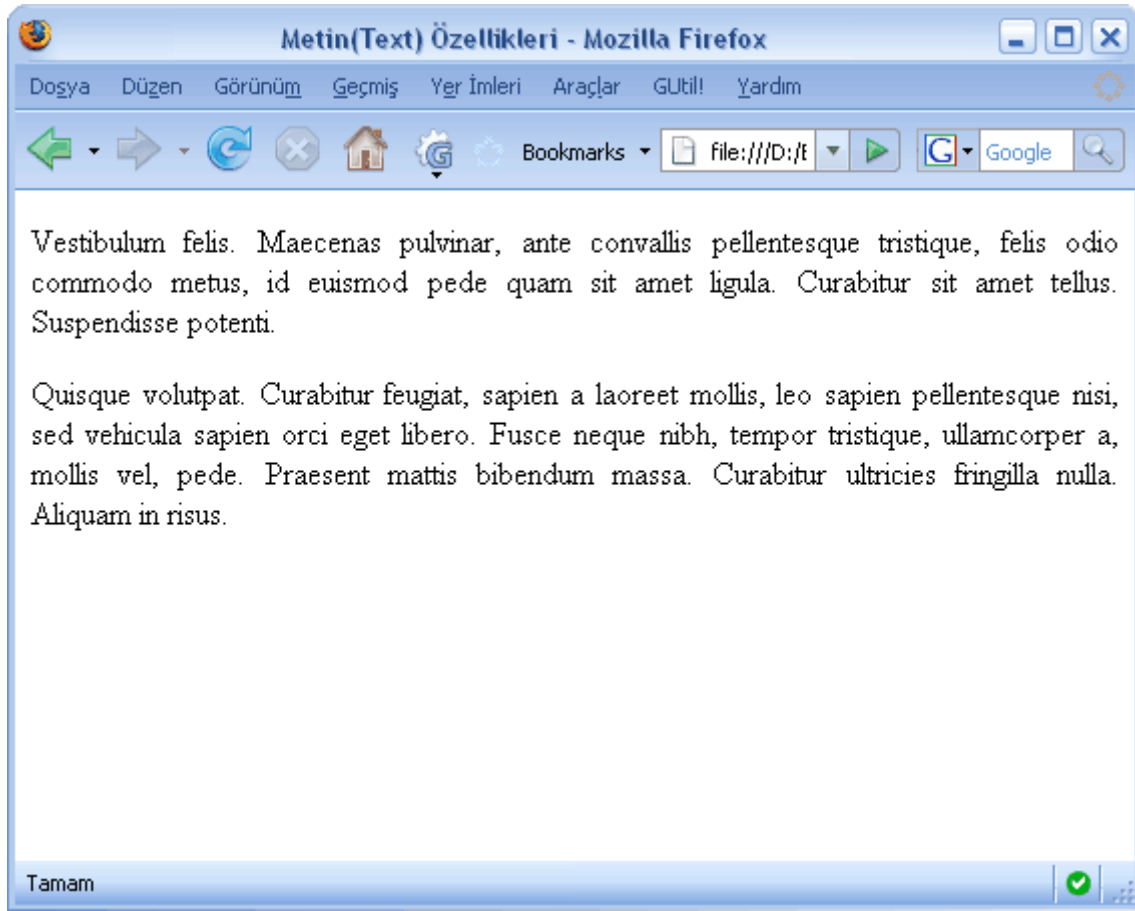
Uygulanabilen elemanlar: **blok seviyesinden elemanlar**

Kalıtıllık: **Var**

left, right, center veya justify: Metni sırayla sola, sağa yanaştırır, ortalar veya iki kenara yanaştırarak satıra yayar.

Örnek 6-2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Metin(Text) Özellikleri</title>
<style type="text/css">
p{
    text-align: justify;
}
</style>
</head>
<body>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 3+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

6.3. Dikey hizalama (vertical-align)

Bir elemanın içeriğinin dikey hizalaması için kullanılır.

- baseline: Orta(metin içeriğine göre)
- sub: Altsimge
- super: Üstsimge
- top: Yukarı(elemana göre)
- text-top: Yukarı (Metin içeriğine göre)
- middle: Orta (Elemana göre)
- bottom: Alt (Elemana göre)
- text-bottom: Alt(Metin içeriğine göre)

Yapısı: **vertical-align: <değer>**

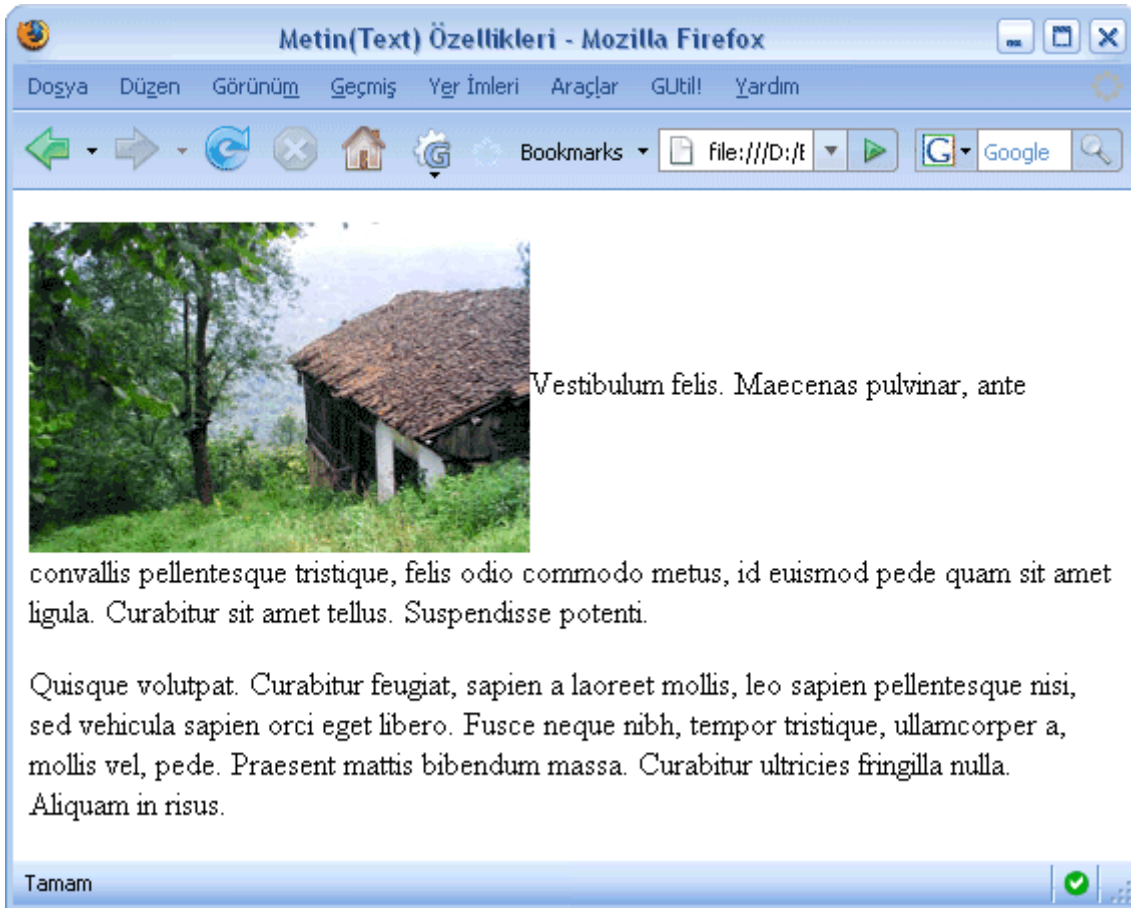
Aldığı Değerler: **<yüzde> | baseline | sub | super | top | text-top |**

middle | bottom | text-bottomBaşlangıç değeri: **baseline**Uygulanabilen elemanlar: **satır içi seviyeden ve table-cell elemanları**Kalıtsallık: **Yok****Örnek 6-3**

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Metin(Text) Özellikleri</title>
<style type="text/css">
img{
    vertical-align: middle;
}
</style>
</head>
<body>
<p>Vestibulum felis. Maecenas
pulvinar, ante convallis pellentesque tristique, felis odio commodo metus,
id euismod pede quam sit amet ligula. Curabitur sit amet tellus.
Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>

```



Internet Explorer 7 ve alt sürümlerinde **bottom** ve **top** değerleri **text-bottom** ve **text-top** gibi yorumlanır.

Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

6.4. Satırlar arası yükseklik (line-height)

Satırlar arasındaki yüksekliği belirler. **normal**, **piksel** veya **yüzde** değerlerin den birini alabilir.

Yapısı: **line-height: <değer>**

Aldığı Değerler: **normal** | **<sayı>** | **<uzunluk değeri >** | **<yüzde>** | **inherit**

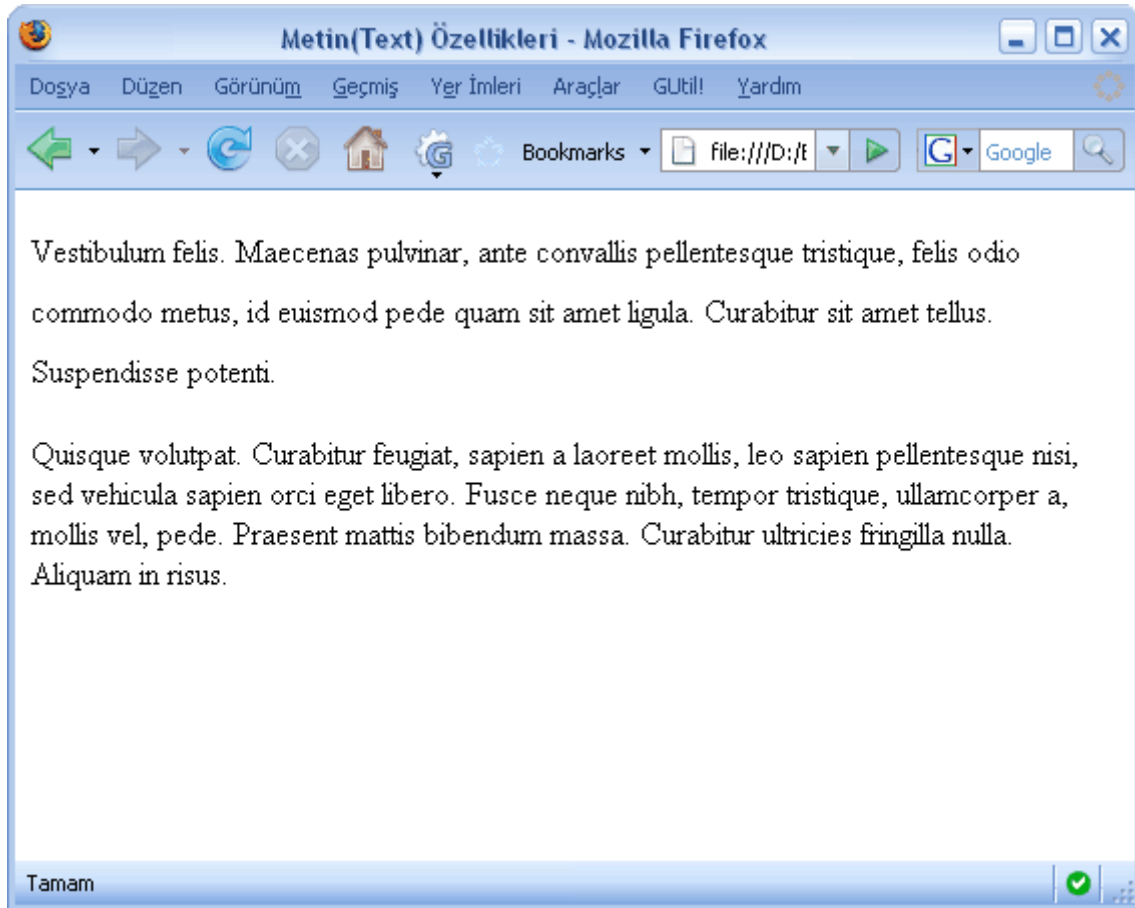
Başlangıç değeri: **normal**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Var**

Örnek 6-4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Metin(Text) Özellikleri</title>
<style type="text/css">
p.satir{
    line-height:30px;
}
</style>
</head>
<body>
<p class="satir">Vestibulum felis. Maecenas pulvinar, ante convallis
pellentesque tristique, felis odio commodo metus, id euismod pede quam
sit amet ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Internet Explorer 7 ve alt sürümlerinde eğer satır içinde yerleştirilen bir eleman(resim veya form elemanları gibi) varsa bu özelliği yanlış uygular.

Firefox 1.0.0.8 ve alt sürümlerde sayısal değerlerin uygulamasında sorun yaşanır.

Tarayıcı Uyumu:

Internet Explorer 3+

Firefox 1.5+

Safari 1.3+

Opera 3.6+

CSS 1+

6.5. Sözcükler arası mesafe (word-spacing)

Sözcükler arasındaki boşluk değerini belirler. Eksi değer alabilir.

Yapısı: **word-spacing: <değer>**

Aldığı Değerler: **normal** | **<uzunluk değeri >** | **inherit**

Başlangıç değeri: **normal**

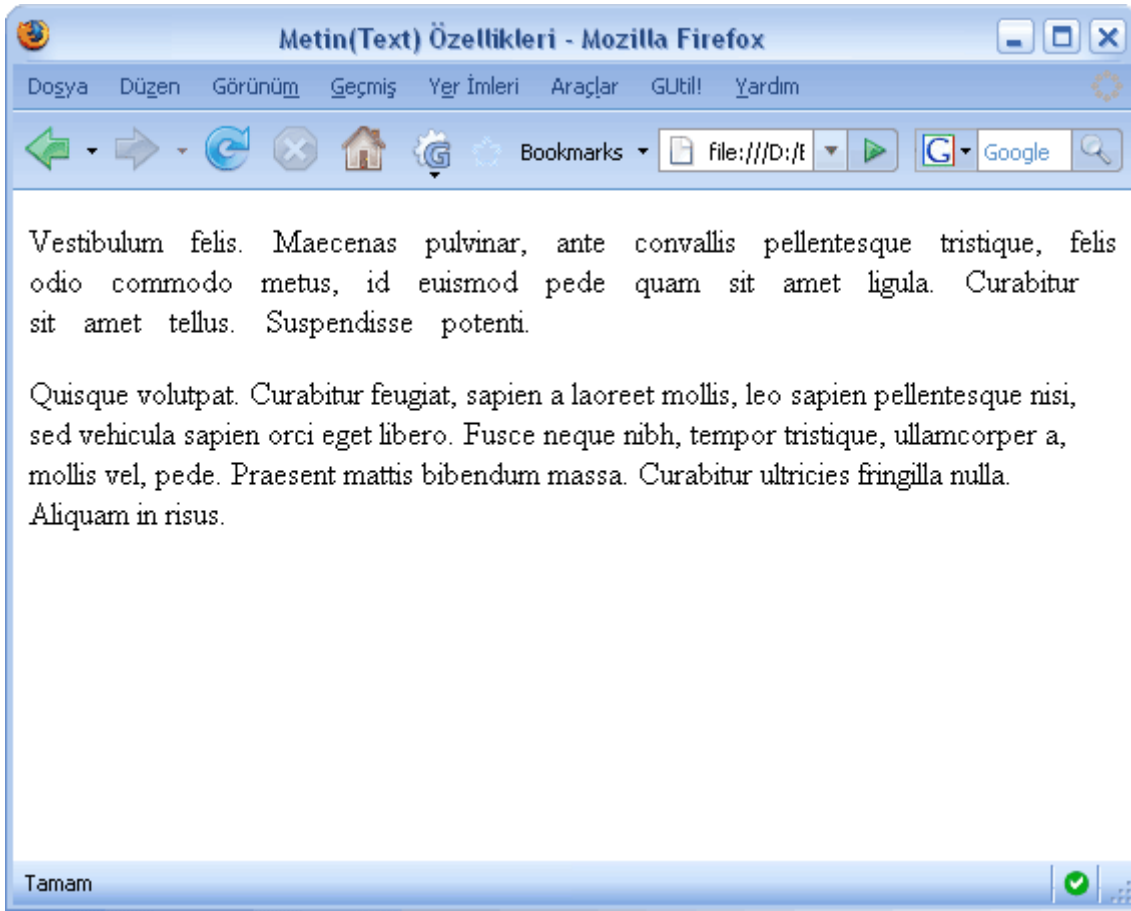
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Var**

Sözcükler arası boşluk algoritmaları kullanıcı arayüzüne bağlıdır. Sözcükler arası boşluklar ayrıca satır iki kenara yayılarak hizalama yapılırken değişebilir.

Örnek 6-5

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Metin(Text) Özellikleri</title>
<style type="text/css">
p.deneme{
    word-spacing: 10px;
}
</style>
</head>
<body>
<p class="deneme">Vestibulum felis. Maecenas pulvinar, ante convallis
pellentesque tristique, felis odio commodo metus, id euismod pede quam
sit amet ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Firefox 2 ve alt sürümlerinde ve Opera9.2 ve alt sürümlerinde Satır içi çocuk elemanların arasındaki boşluğun asgari genişliği sıfırdır.

Tarayıcı Uyumu:

Internet Explorer 6+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 2+

6.6. Harfler arası mesafe (letter-spacing)

Harfler arasındaki boşluk değerini belirler. Eksi değer alabilir.

Yapısı: **letter-spacing: <değer>**

Aldığı Değerler: **normal | <uzunluk değeri > | inherit**

Başlangıç değeri: **normal**

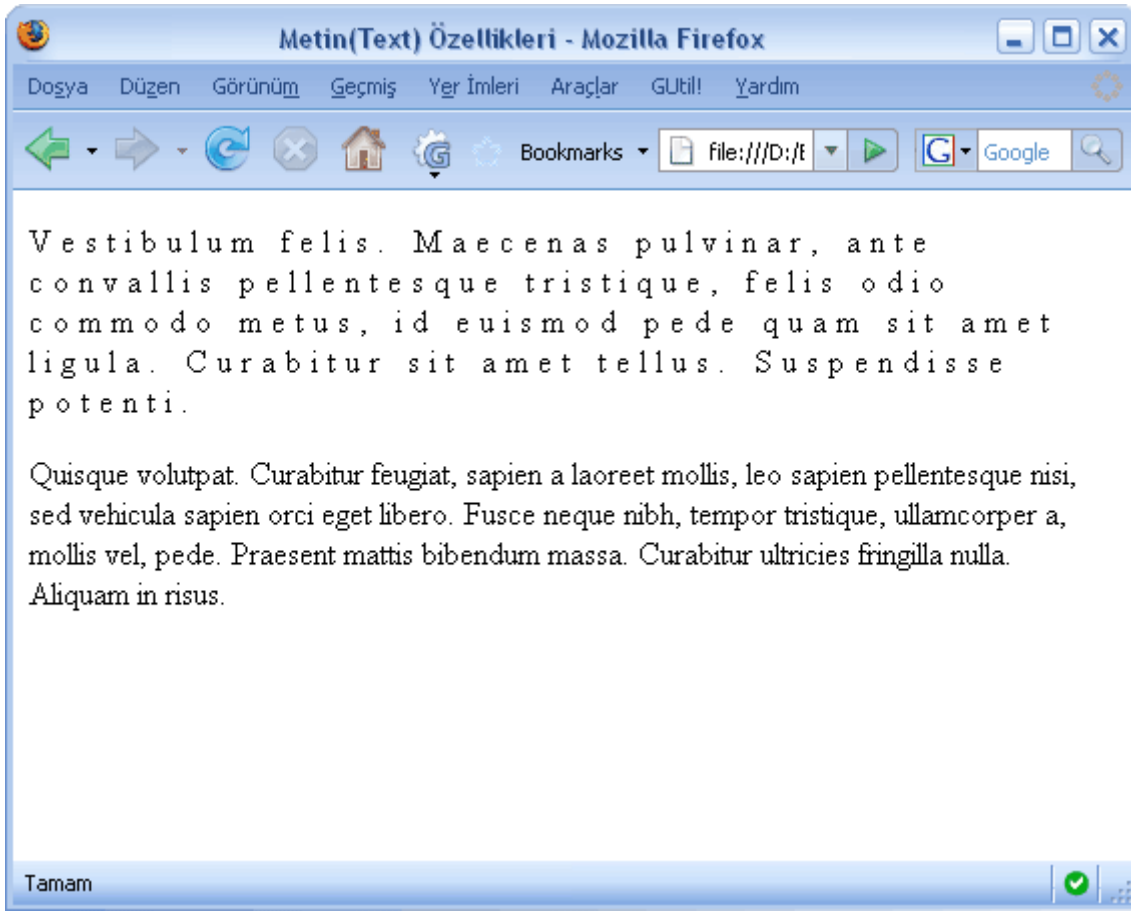
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Var**

Harfler arası boşluk algoritması kullanıcı arayüzüne bağlıdır. Harfler arası boşluklar ayrıca satır iki kenara yayılarak hizalama yapılırken değişebilir.

Örnek 6-6

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Metin(Text) Özellikleri</title>
<style type="text/css">
p.deneme{
    letter-spacing: 5px;
}
</style>
</head>
<body>
<p class="deneme">Vestibulum felis. Maecenas pulvinar, ante convallis
pellentesque tristique, felis odio commodo metus, id euismod pede quam
sit amet ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

6.7. Harf Büyüklüğü (text-transform)

Metnin büyük-küçük harf çevirmek için kullanılır.

- **uppercase**(hepsini büyük harf yap)
- **lowercase**(hepsini küçük harf yap)
- **capitalize**(sadece baş harfleri büyük yap)

Yapısı: **text-transform: <değer>**

Aldığı Değerler: **none | capitalize | uppercase | lowercase | inherit**

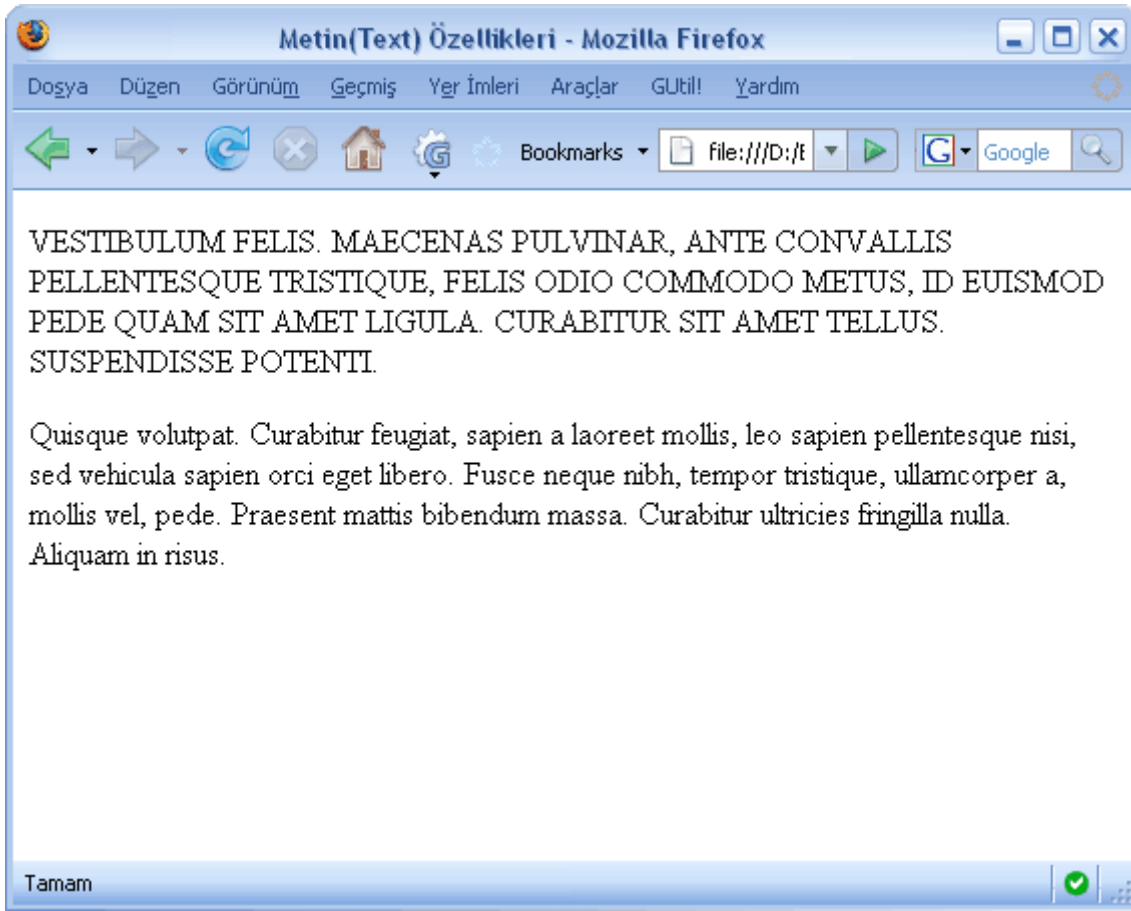
Başlangıç değeri: **none**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Var**

Örnek 6-7

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Metin(Text) Özellikleri</title>
<style type="text/css">
p.deneme{
    text-transform:uppercase;
}
</style>
</head>
<body>
<p class="deneme">Vestibulum felis. Maecenas pulvinar, ante convallis
pellentesque tristique, felis odio commodo metus, id euismod pede quam
sit amet ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

6.8. Satırın altını, üstünü, ortasını çizme ve yanıp sönmeye (text-decoration)

Bu özellik metinlerimize özel işaretler koymamızı sağlar. Metinlerinizin altını çizmek örnek olarak gösterilebilir.

Yapısı: **text-decoration: <değer>**

Aldığı Değerler: **none** | [**underline** || **overline** || **line-through** || **blink**] | **inherit**

Başlangıç değeri: **none**

Uygulanabilen elemanlar: **tüm elemanlar**

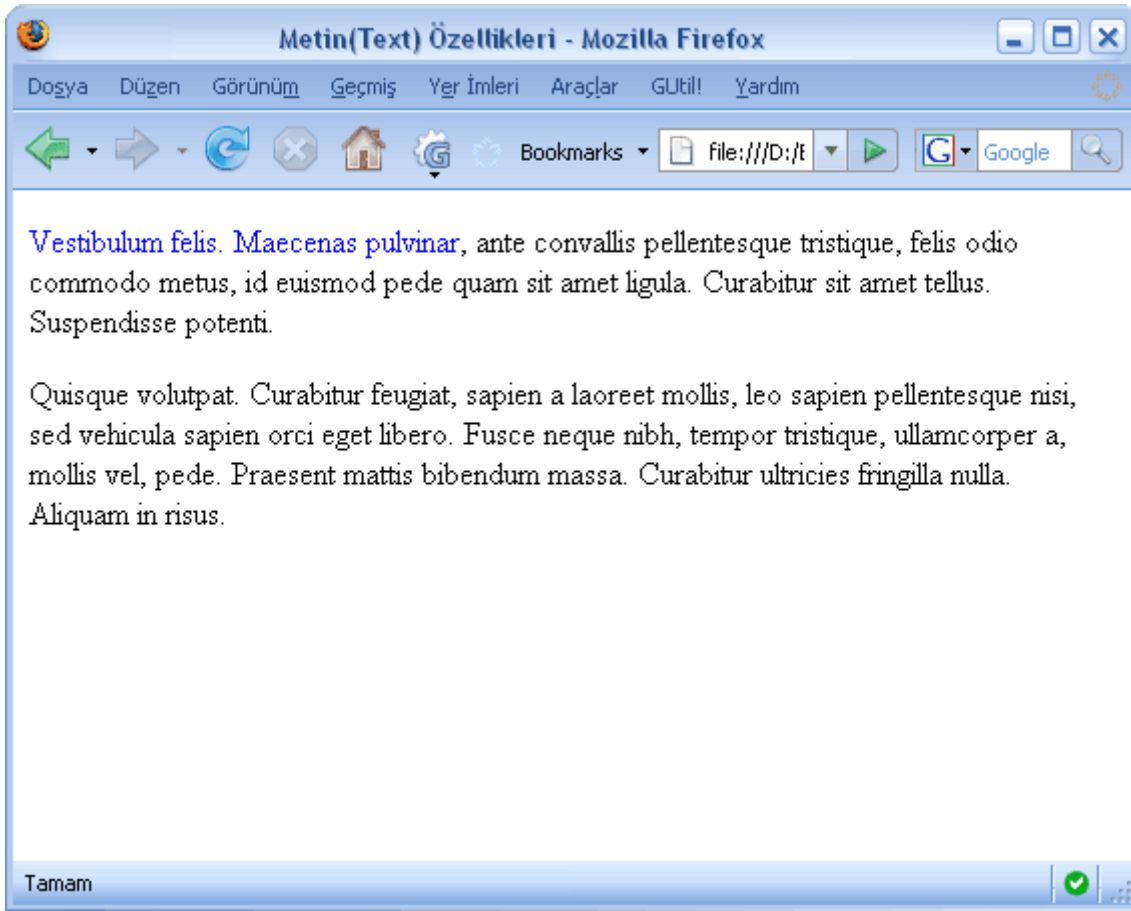
Kalıtımsallık: **Yok**

Örnek 6-8

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Metin(Text) Özellikleri</title>
<style type="text/css">
a:link, a:visited, a:active{
    text-decoration: none;
}
a:hover{
    text-decoration: underline;
}
</style>
</head>
<body>
<p class="deneme"><a href="#">Vestibulum felis. Maecenas
pulvinar</a>, ante convallis pellentesque tristique, felis odio commodo
metus, id euismod pede quam sit amet ligula. Curabitur sit amet tellus.
Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>

```



Internet Explorer 7 ve altı sürümlerinde **blink** değerini desteklemez.

Internet Explorer 6 ve altı sürümlerinde **line-through** ile çizilen çizgi diğer tarayıcılara göre daha yukarıdadır.

Firefox 2 ve alt sürümlerde metin süsleme atamalarını **float** uygulanmış alt elemanlarada uygular.

Tarayıcı Uyumu:

Internet Explorer 3+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

6.9. Metin gölgelendirme (text-shadow)

Bu özellik CSS2 ile birlikte geldi ancak internet tarayıcıları bu özelliği desteklemeyince CSS2.1 kaldırıldı. CSS3 ile birlikte yeniden eklenen gölge özelliği bu sefer 4 değer ile gerçeğe daha yakın gölgeler oluşturmamızı sağlayacak.

6.10. Boşlukların korunması (white-space)

Yapısı: **white-space: <değer>**

Aldığı Değerler: **normal | pre | nowrap | inherit**

Başlangıç değeri: **normal**

Uygulanabilen elemanlar: **Blok seviyeden elemanlar**

Kalıtıllık: **Var**

white-space özelliği elemanların boşluklarının nasıl işlem göreceğini belirler. 3 değer alır.

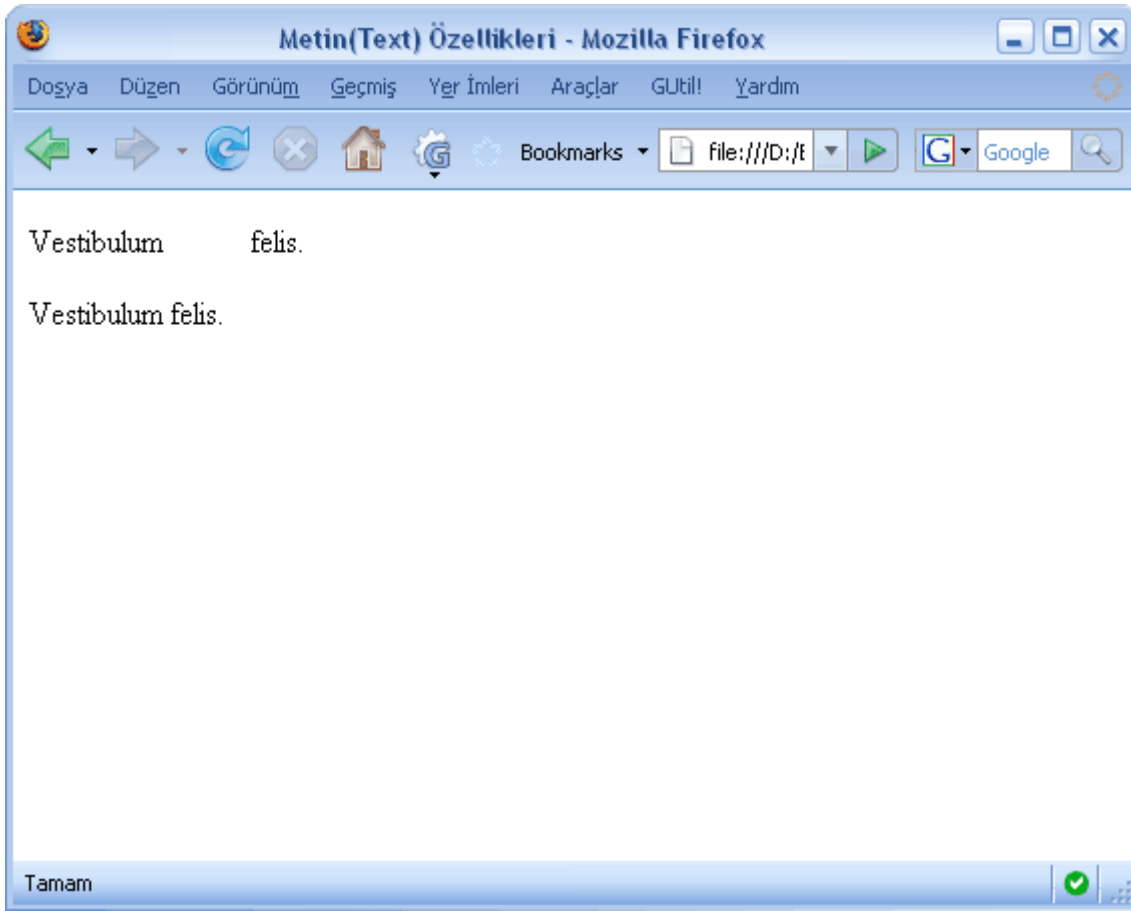
normal: Birden fazla boşluğu tek boşluk gibi sayar.

pre: Birden fazla boşluğu birleştirmez.

nowrap :
 etiketi hariç alt satıra geçişe izin vermez.

Örnek 6-9

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Metin(Text) Özellikleri</title>
<style type="text/css">
p.deneme{
    white-space: pre;
}
</style>
</head>
<body>
<p class="deneme">Vestibulum          felis. </p>
<p>Vestibulum          felis. </p>
</body>
</html>
```



Internet Explorer 7 ve alt sürümlerinde **pre-line** veya **pre-wrap** değerlerini desteklemiyor. **textarea** elemanı için **normal** ve **pre** atamaları **pre-wrap** gibi davranır. **textarea** elemanı için **nowrap** değeri **pre-line** gibi davranır.

Firefox 2 ve alt sürümlerinde **pre-line** ve **pre-wrap** değerlerini desteklemiyor(bunu yerine benzer **-moz-pre-wrap** uygulanır.) **textarea** elemanı için **normal** ve **pre** atamaları **pre-wrap** gibi davranır.

Opera 9.2. ve alt sürümlerinde **pre-line** özelliğini desteklemiyor. **textarea** elemanı için **normal** ve **pre** atamaları **pre-wrap** gibi davranır. **textarea** elemanı için **nowrap** değeri **pre-line** gibi davranır.

Tarayıcı Uyumu:

Internet Explorer 5.5+(kısmen)

Firefox 1+(kısmen)

Safari 1.3+

Opera 4+(kısmen)

CSS 2+

6.11. Metin yazma yönü(direction)

Yapısı: **direction: <değer>**

Aldığı Değerler: **ltr | rtl | inherit**
 Başlangıç değeri: **ltr**
 Uygulanabilen elemanlar: **tüm elemanlar**
 Kalıtsallık: **Var**

Bu özelliğin aşağıdakileri belirler:

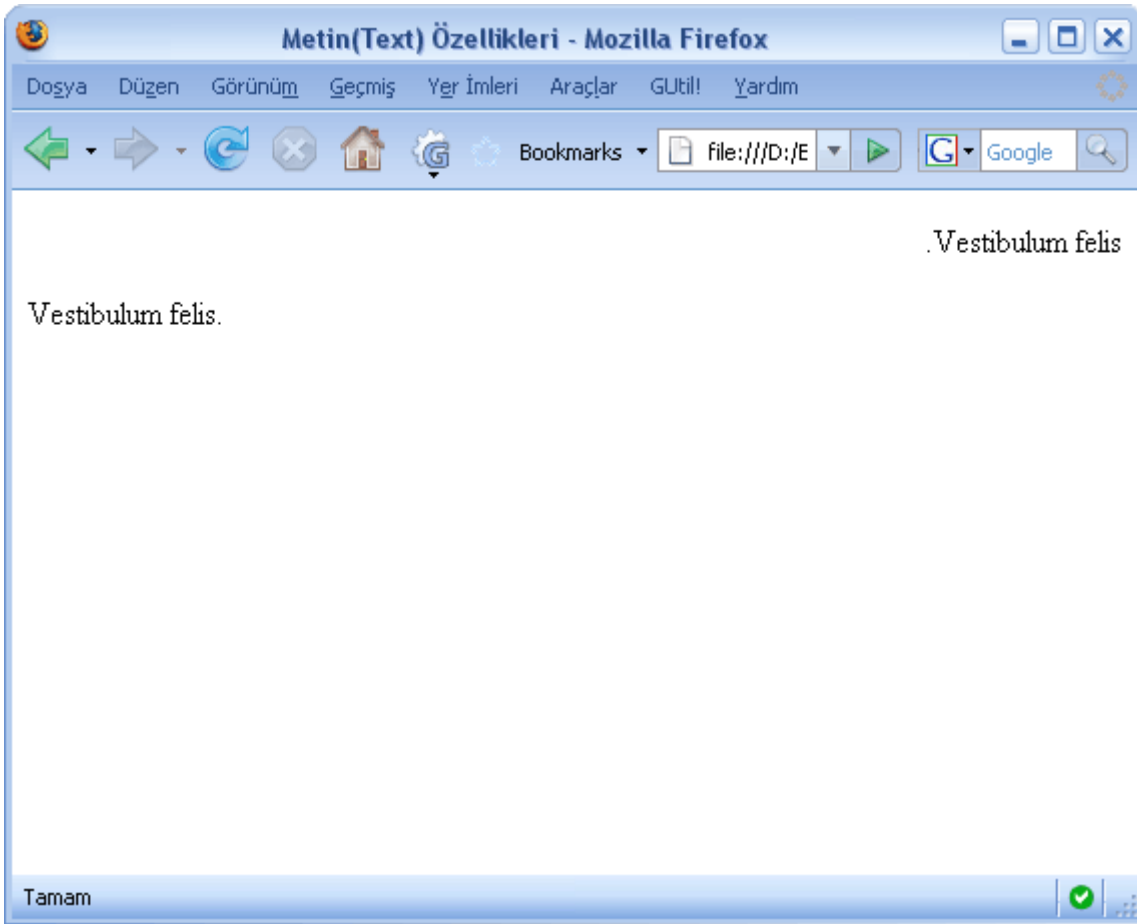
- Bloğun yazma yönünü belirler
- Unicode çift yönlülük(Unicode-bidi) için iç içelik ve üst üste gelme yönünü belirler.
- Tablo sütunlarının yerleşim yönünü belirler
- Yatay taşma yönünü belirler
- Bir blokta text-align: justify tanımlaması yapıldığında son satırın yönünü belirler.

direction özelliğinin satırıçi seviyeden elemanlar üzerinde etkili olabilmesi için **unicode-bidi** özelliğinin değerinin **embed** veya **override** olması gerekir.

Normalde bu özellik DTD içerisinde tanımlanır, ancak CSS2.1 önemine binaen bu özellikleri eklemiştir.

Örnek 6-10

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Metin(Text) Özellikleri</title>
<style type="text/css">
p.bir{
direction: rtl
}
p.iki{
direction: ltr
}
</style>
</head>
<body>
<p class="bir">Vestibulum felis. </p>
<p class="iki">Vestibulum felis. </p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 5.5+
 Firefox 1+
 Safari 1.3+
 Opera 9.2+
 CSS 2+

6.12. Çok yönlülük algoritması (unicode-bidi) özelliği

Yapısı: **unicode-bidi: <değer>**

Aldığı Değerler: **bidi-override | embed | normal | inherit**

Başlangıç değeri: **normal**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

Bazı diller sağdan sola yazılırken bazı diller solda sağa doğru yazılır. Bazı dokümanlarımızda her ikisini birden kullanabiliriz buna çok yönlülük denir kısaca bidi.

Bu özellik **direction** ile birlikte çok yönlü blokları idare etmemizi sağlar. Eğer bir sayfada birden fazla yönlü blok kullanacaksak buna özel

tanımlamalar yapmalıyız. Ayrıntılı bilgi için <http://www.unicode.org/reports/tr9/> adresini inceleyiniz. Bu özellik Unicode çok yönlülük algoritmasının iç içe seviyesini ve üst üsten binme kontrolünü belirler.

Normalde bu özellik DTD içerisinde tanımlanır, ancak CSS2.1 önemine binaen bu özellikleri eklemiştir.

Tarayıcı Uyumu:

Internet Explorer 5.5+(sorunlu)

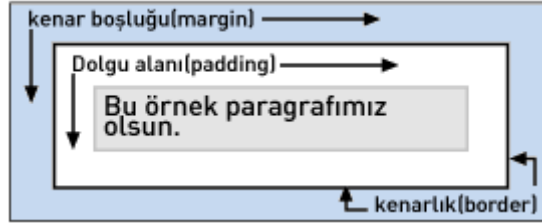
Firefox 1+

Safari **desteklemiyor**

Opera 9.2+

CSS 2+

7. Artalan (Background) Özellikleri



Artalan(Background) yukarıdaki resimde dolgu alanı(padding) ve içerik alanını(paragraf) kapsar.

Artalan özellikleri ile elemanlara tek bir renk atanabildiği gibi **background-image** özelliği ile (X)HTML'in çok üzerinde eklemelerde yapılabilir.

Artalan özelliklerini tek tek incelersek:

- Artalan rengi (background-color)
- Artalan resmi (background-image)
- Artalan resim yenileme (background-repeat)
- Artalan resim bağlantısı (background-attachment)
- Artalan resim konumu (background-position)
- Artalan (background)

7.1. Artalan rengi (background-color)

Elemanların artalanına bir renk atamak için kullanılır.

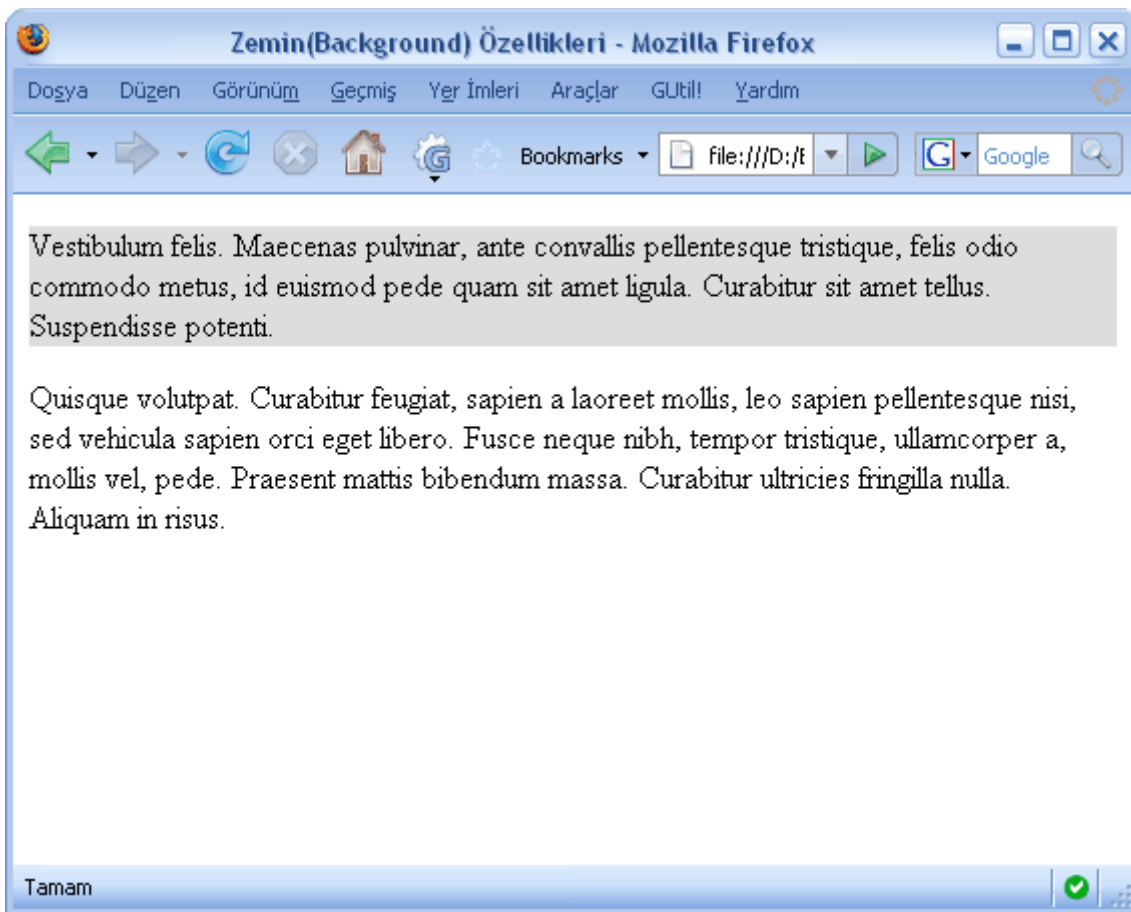
Yapısı: **background-color: <değer>**
 Aldığı Değerler: **renk | transparent | inherit**
 Başlangıç değeri: **transparent**
 Uygulanabilen elemanlar: **tüm elemanlar**
 Kalıtsallık: **Yok**

transparent değeri elemanın artalanının görünmesi için kullanılır.

Örnek 7-1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Aralan(Background) Özellikleri</title>
<style type="text/css">
p.deneme{
    background-color: #ddd;
}
</style>
</head>
<body>
<p class="deneme">Vestibulum felis. Maecenas pulvinar, ante convallis
pellentesque tristique, felis odio commodo metus, id euismod pede quam
sit amet ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 4+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

7.2. Artalan resmi (background-image)

Elemanların artalanına resim eklemek için kullanılır.

Yapısı: **background-image: <deger>**

Aldığı Değerler: **url | none | inherit**

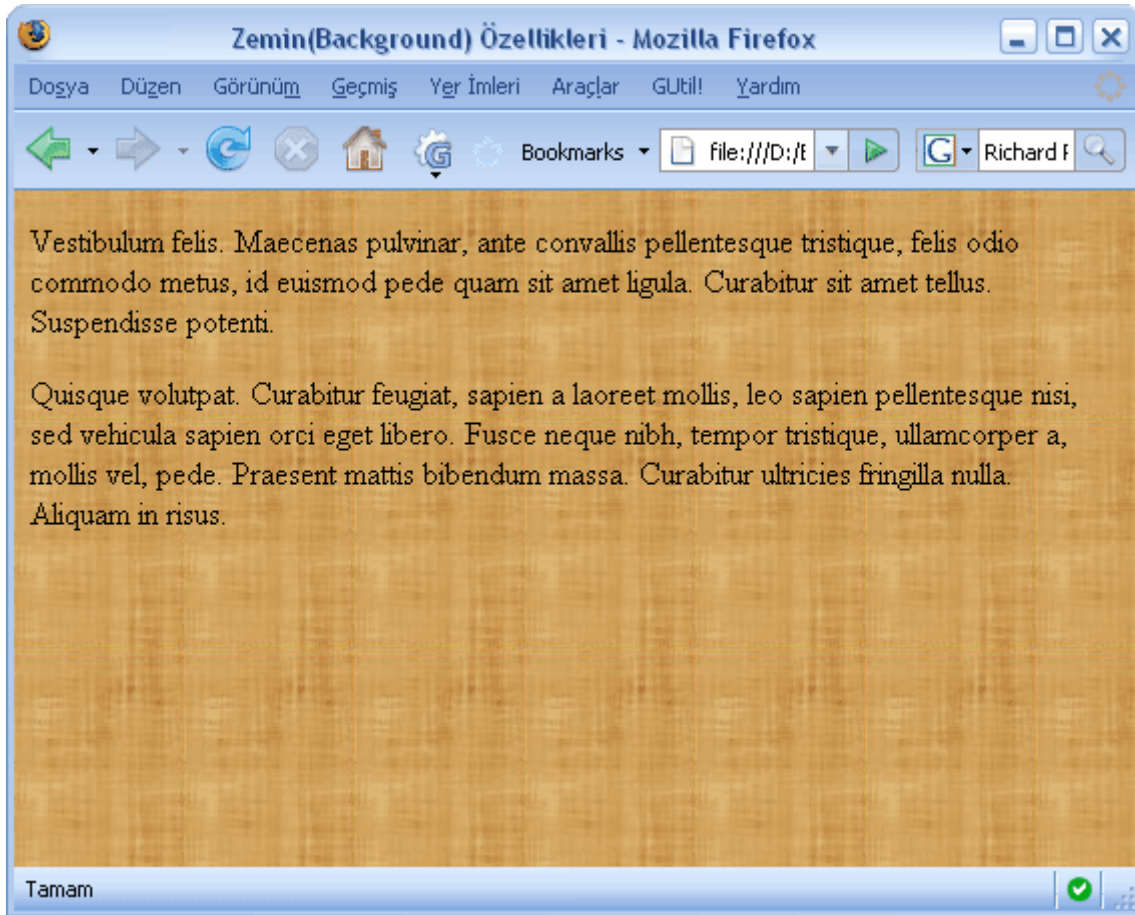
Başlangıç değeri: **none**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

Örnek 7-2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Artalan(Background) Özellikleri</title>
<style type="text/css">
body{
    background-image: url(Artalan_resmi.gif)
}
</style>
</head>
<body>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

7.3. Artalan resim yenileme (background-repeat)

background-repeat özelliği **background-image** ile artalana eklenen resmin tekrarı ile özellikleri belirler.

Yapısı: **background-image: <değer>**

Aldığı Değerler: **repeat | repeat-x | repeat-y | no-repeat**

Başlangıç değeri: **repeat**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

repeat [yinele]: Resim hem yatayda hem de düşeyde yinelenir.

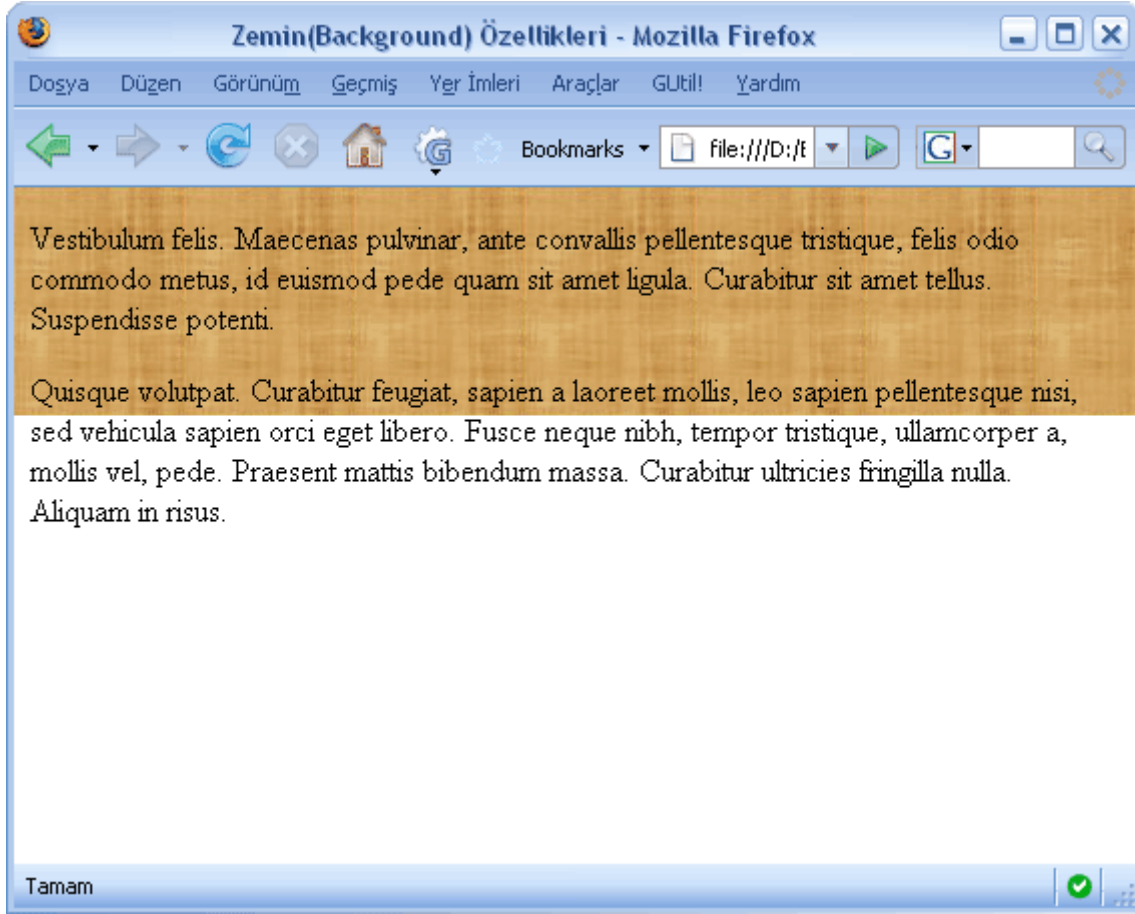
repeat-x: Resim sadece yatay olarak yinelenir.

repeat-y: Resim sadece düşey olarak yinelenir.

no-repeat [yineleme]: Resim yinelenmez; resmin sadece bir kopyası oluşturulur.

Örnek 7-3

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Artalan(Background) Özellikleri</title>
<style type="text/css">
body{
    background: white url(Artalan_resmi.gif);
    background-repeat: repeat-x;
}
</style>
</head>
<body>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Safari 2 ve alt sürümlerinde bu özellik ile ilgili bazı sorunlar vardır: **background-repeat** değeri **no-repeat** olarak atanmış olan artalan resimleri tekrarlamaması gerekirken tekrar yapar.

Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+(kısmen) 3+
 Opera 3.6+
 CSS 1+

7.4. Artalan resim bağılılığı (background-attachment)

background-attachment özelliği artalana eklenen resmin sayfa ile kaymasını(scroll) veya sayfanın artalanına çakılı(fixed) kalmasını sağlar.

Yapısı: **background-attachment: <değer>**

Aldığı Değerler: **scroll | fixed | inherit**

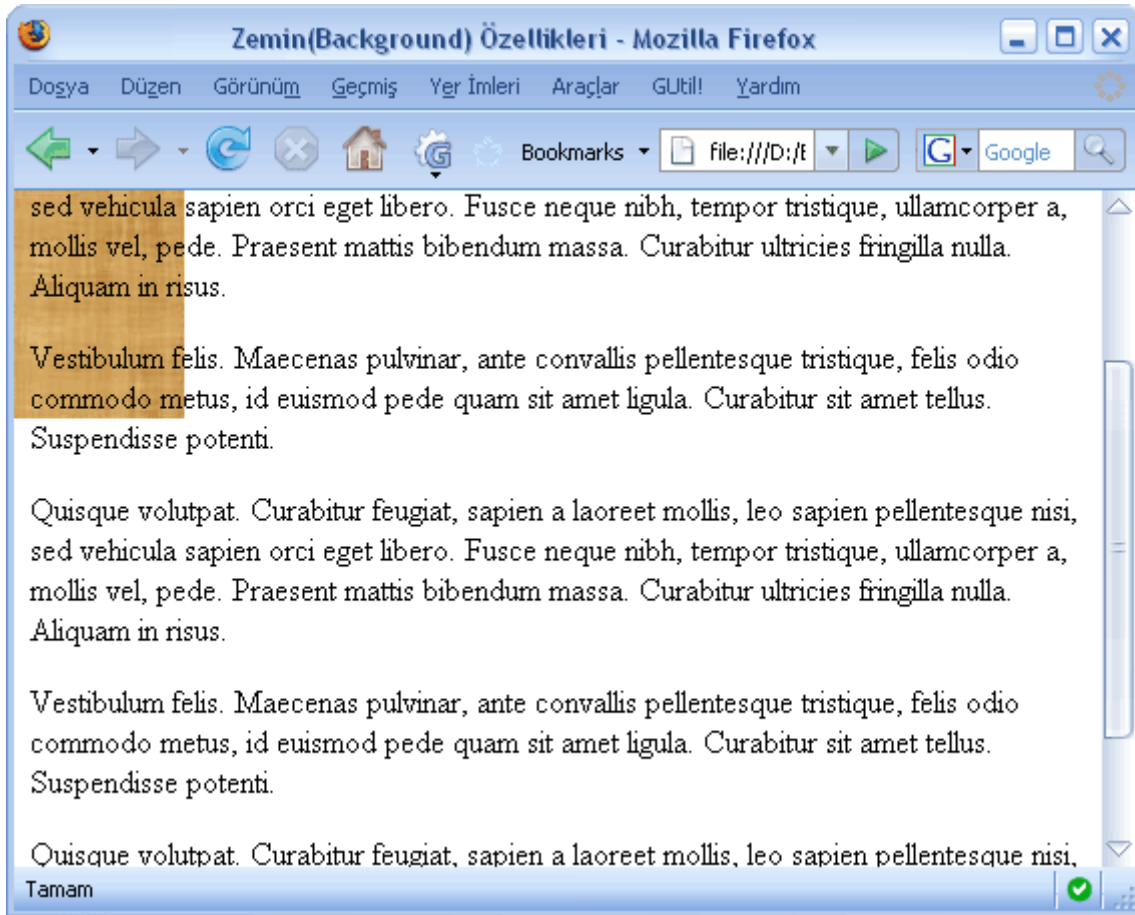
Başlangıç değeri: **scroll**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Yok**

Örnek 7-4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Artalan(Background) Özellikleri</title>
<style type="text/css">
body{
    background: white url(Artalan_resmi.gif);
    background-repeat: no-repeat;
    background-attachment: fixed;
}
</style>
</head>
<body>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

7.5. Artalan resmi konumu (background-position)

background-position özelliği **background-image** ile belirlenen resmin başlangıç noktasını belirler. Bu özellik sadece blok seviyeli ve yerleştirilen (Bu elemanlar kendine özgü boyutları olan elemanlar olarak tanımlanabilir. Örn: **IMG**, **INPUT**, **TEXTAREA**, **SELECT**, ve **OBJECT**) elemanlara uygulanır.

Yapısı: **background-position: <değer>**

Aldığı Değerler: [**<percentage>** | **<length>**]{1,2} | [**top** | **center** | **bottom**] | [**left** | **center** | **right**]

Başlangıç değeri: **0% 0%**

Uygulanabilen elemanlar: **Blok seviyeli ve yerleştirilen elemanlar**

Kalıtımsallık: **Yok**

En kolay kullanım şekli;
Yatay değerler için: left, center, right
Dikey değerler için: top, center, bottom

Yüzde değerleri ve uzunluk değerleri de kullanılır. Yüzde değerleri elemanın boyutuna bağlı olarak görecelidir. Uzunluk değerleri de kullanılabilir. Ancak farklı ekran çözünürlüklerinde farklı görüntülere sebebiyet vermesi nedeni ile pek önerilmez.

Yüzde değerler ve uzunluk değerleri verildiğinde ilk değer yatay içindir sonra gelen dikey değerdir. Örneğin %10 %60 değeri bir artalan resmi için verilmiş ise %10 değeri yataydaki değeri %60 ise dikey olarak değerini gösterir. 5px 10px gibi değerler verilmişse resmin sol üstten 5px sağa ve 10px aşağıdan başlayacağını belirler.

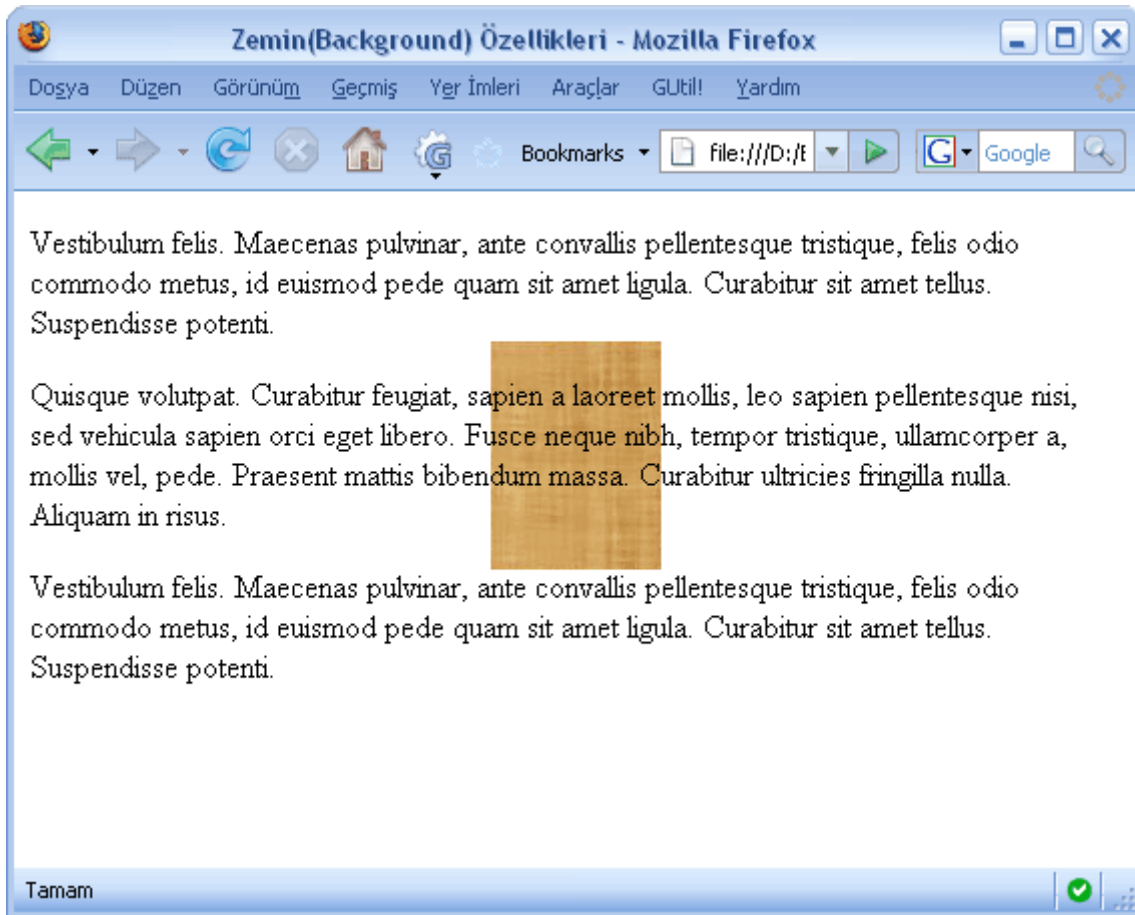
Eğer yalnızca yatay değer verilmiş ise, dikey değer %50 olarak kabul edilecektir. Yüzde değerler ve uzunluk değerleri eksi değerler alabilir. Örn -2px %10 gibi. Aşağıdaki örnekler genel kullanım için yararlıdır:

- **top left = left top = 0% 0%**
- **top = top center = center top = 50% 0%**
- **right top = top right = 100% 0%**
- **left = left center = center left = 0% 50%**
- **center = center center = 50% 50%**
- **right = right center = center right = 100% 50%**
- **bottom left = left bottom = 0% 100%**
- **bottom = bottom center = center bottom = 50% 100%**
- **bottom right = right bottom = 100% 100%**

Örnek 7-5

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Artalan(Background) Özellikleri</title>
<style type="text/css">
body{
    background: white url(Artalan_resmi.gif);
    background-repeat: no-repeat;
    background-position: center;
}
</style>
</head>
<body>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
```

```
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 4+

Firefox 1+

Safari 1.3+

Opera 3.6+

CSS 1+

7.6. Artalan (background)

Bu özellik Artalan(background) ile ilgili tüm özelliklerin bir arada kullanımı sağlar.

Yapısı: **background: <deger>**

Aldığı Değerler: **<background-color> || <background-image> || <background-repeat> || <background-attachment> || <background-position>**

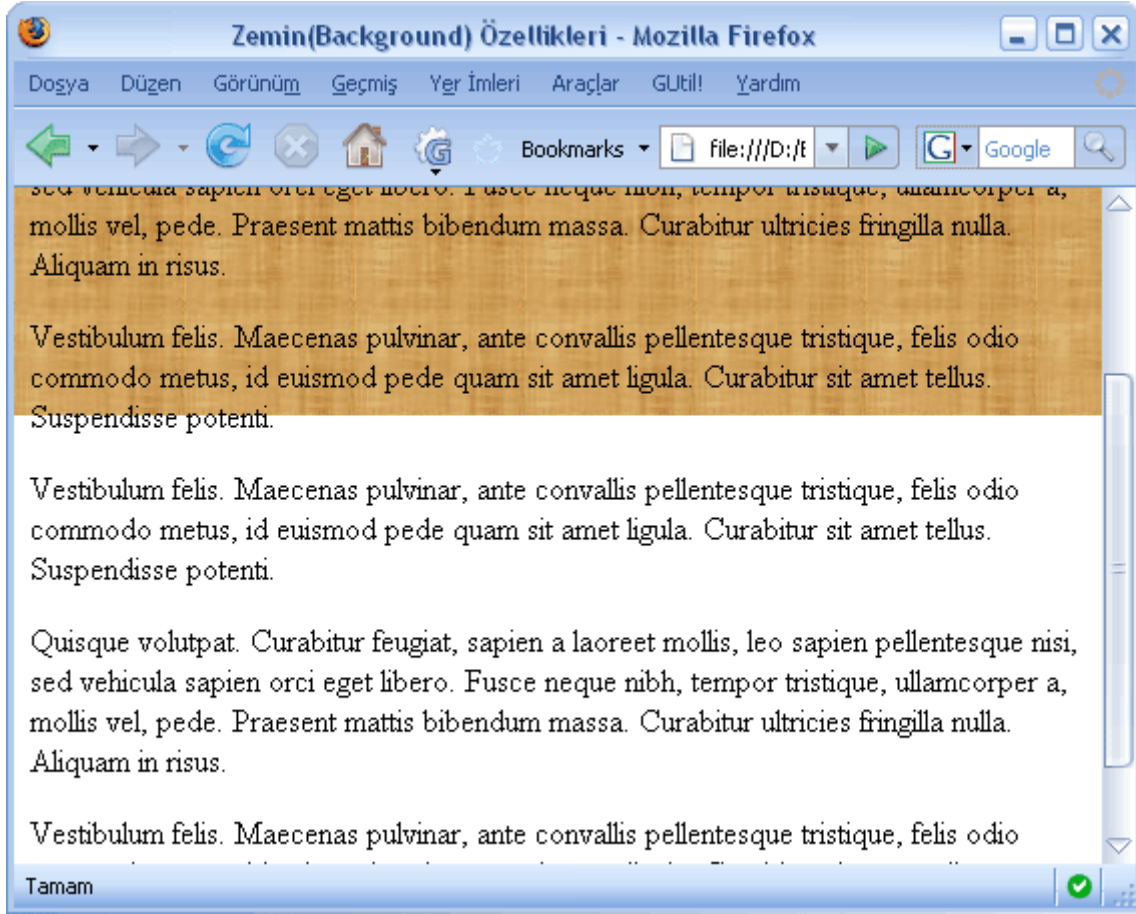
Başlangıç değeri: **Tanımsız**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

Örnek 7-6

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Artalan(Background) Özellikleri</title>
<style type="text/css">
body{
    background: white url(Artalan_resmi.gif) top left fixed
repeat-x;
}
</style>
</head>
<body>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
<p>Vestibulum felis. Maecenas pulvinar, ante convallis pellentesque
tristique, felis odio commodo metus, id euismod pede quam sit amet
ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 3+
 Firefox 1+
 Safari 1.3+(kısmen) 3+
 Opera 3.6+
 CSS 1+

8. Listeleme Özellikleri

Liste özellikleri listelerin genel görsel biçimi ayarlamamızı sağlar. İlk olarak listeleme amaçlı kullanılsa da şimdilerde menü yapımında kullanımı revaçtadır.

- Liste imleyici tipi (`list-style-type`)
- Liste imleyici resmi (`list-style-image`)
- Liste imleyici konumu (`list-style-position`)
- Liste biçemi (`list-style`)

8.1. Liste imleyici tipi (`list-style-type`)

Yapısı: **list-style-type: <değer>**

Aldığı Değerler: **disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha | none**

Başlangıç değeri: **disc**

Uygulanabilen elemanlar: **display** değeri **list-item** alan elemanlar

Kalıtımsallık: **Var**

list-style-type özelliği `list-item` işaretinin tipini belirler. `list-style-images` özelliği **none** değeri aldığı veya resim görüntülenemediğinde kullanılır.

Şekiller `disc` (içi dolu daire), `circle` (içi boş daire) ve `square` (kare) değerleriyle belirtilirler. Şekillerin yorumlanması kullanıcı arayüzüne bağlıdır.

Numaralama sistemleri şöyle belirtilir:

decimal: Onluk tabandaki sayılar 1'den başlar.

decimal-leading-zero: Önlerine sıfır konmuş onluk tabandaki sayılar (01, 02, 03, ..., 98, 99 gibi).

lower-roman: Küçük harflerle romen rakamları (i, ii, iii, iv, v, vs.).

upper-roman: Büyük harflerle romen rakamları (I, II, III, IV, V, vs.).

hebrew: Geleneksel İbranice numaralama (Alef, Bet, ... Tet Vav, Tet Zayin, ... Yod Tet, Kaf ...).

georgian: Geleneksel Gürcüce numaralama (an, ban, gan, ..., he, tan, in, in-an, ...).

armenian: Geleneksel Ermenice numaralama.

cjk-ideographic: Düz sembolik (Çince/Japonca) numaralama

hiragana: a, i, u, e, o, ka, ki, ...

katakana: A, I, U, E, O, KA, KI, ...

hiragana-iroha: i, ro, ha, ni, ho, he, to, ...

katakana-iroha: I, RO, HA, NI, HO, HE, TO, ...

Alfabetik sistemler şöyle belirtilir:

lower-latin

lower-alpha: Küçük ascii harfleri (a, b, c, ... z).

upper-latin

upper-alpha: Büyük ascii harfleri (A, B, C, ... Z).

lower-greek: Küçük Yunan harfleri (α , β , γ , ...)

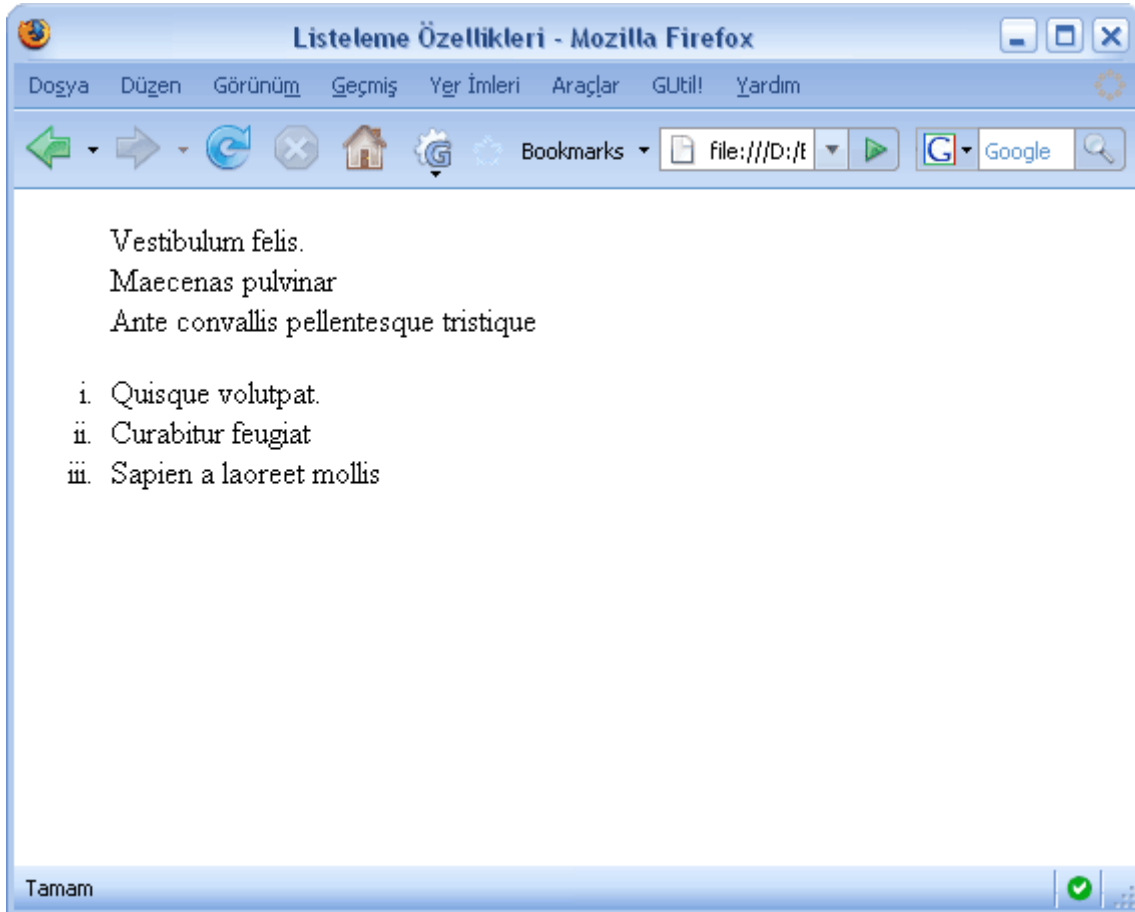
Örnek 8-1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Listeleme Özellikleri</title>
<style type="text/css">
ul{
    list-style-type: none
}
ol{
    list-style-type: lower-roman /* i ii iii iv v gibi. */
}
</style>
</head>
<body>
<ul>
<li>Vestibulum felis.</li>
<li>Maecenas pulvinar</li>
<li>Ante convallis pellentesque tristique</li>
```

```

</ul>
<ol>
<li>Quisque volutpat.</li>
<li>Curabitur feugiat</li>
<li>Sapien a laoreet mollis</li>
</ol>
</body>
</html>

```



Internet Explorer 7 ve alt sürümlerinde **armenian**, **decimal-leading-zero**, **georgian** ve **lower-greek** değerlerini desteklemiyor. **lower-latin** ve **upper-latin** yerine **lower-alpha** ve **upper-alpha** değerlerini destekliyor.

Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

8.2. Liste imleyici resmi (list-style-image)

Yapısı: **list-style-image:** <değer>

Aldığı Değerler: <url> | none | inherit

Başlangıç değeri: none

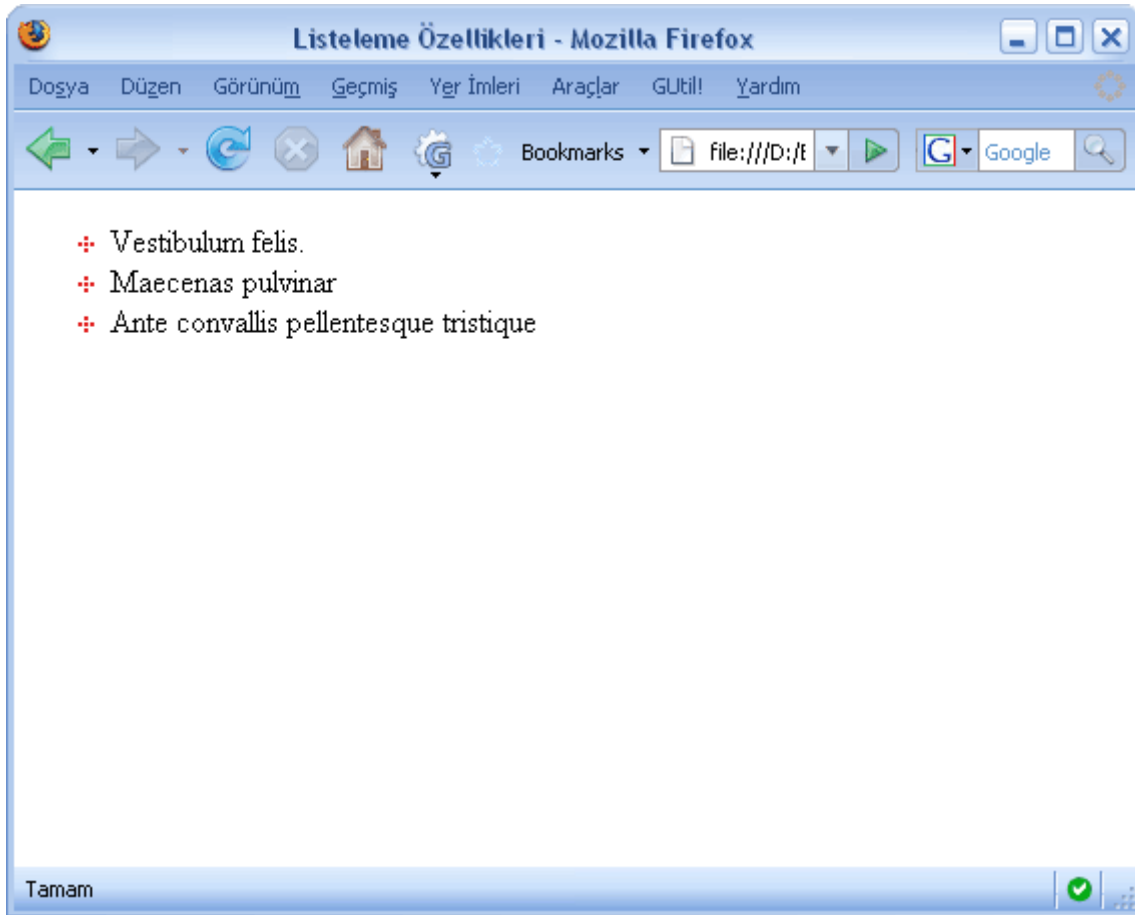
Uygulanabilen elemanlar: **display değeri list-item alan elemanlar**

Kalıtıllık: **Var**

list-style-image özelliği list-style işaretinin yerine resim koymak için kullanılır.

Örnek 8-2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Listeleme Özellikleri</title>
<style type="text/css">
ul{
    list-style-image: url(liste_imgesi.gif);
}
</style>
</head>
<body>
<ul>
<li>Vestibulum felis.</li>
<li>Maecenas pulvinar</li>
<li>Ante convallis pellentesque tristique</li>
</ul>
</body>
</html>
```



Internet Explorer 7 ve alt sürümlerinde **float** uygulanmış liste elemanının **list-style-image** ile atanmış işaretleri görünmez.

Tarayıcı Uyumu:

Internet Explorer 4+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

8.3. Liste imleyici konumu (**list-style-position**)

Yapısı: **list-style-position: <deger>**
Aldığı Değerler: **inside | outside | inherit**
Başlangıç değeri: **outside**
Uygulanabilen elemanlar: **tüm elemanlar**
Kalıtsallık: **Yok**

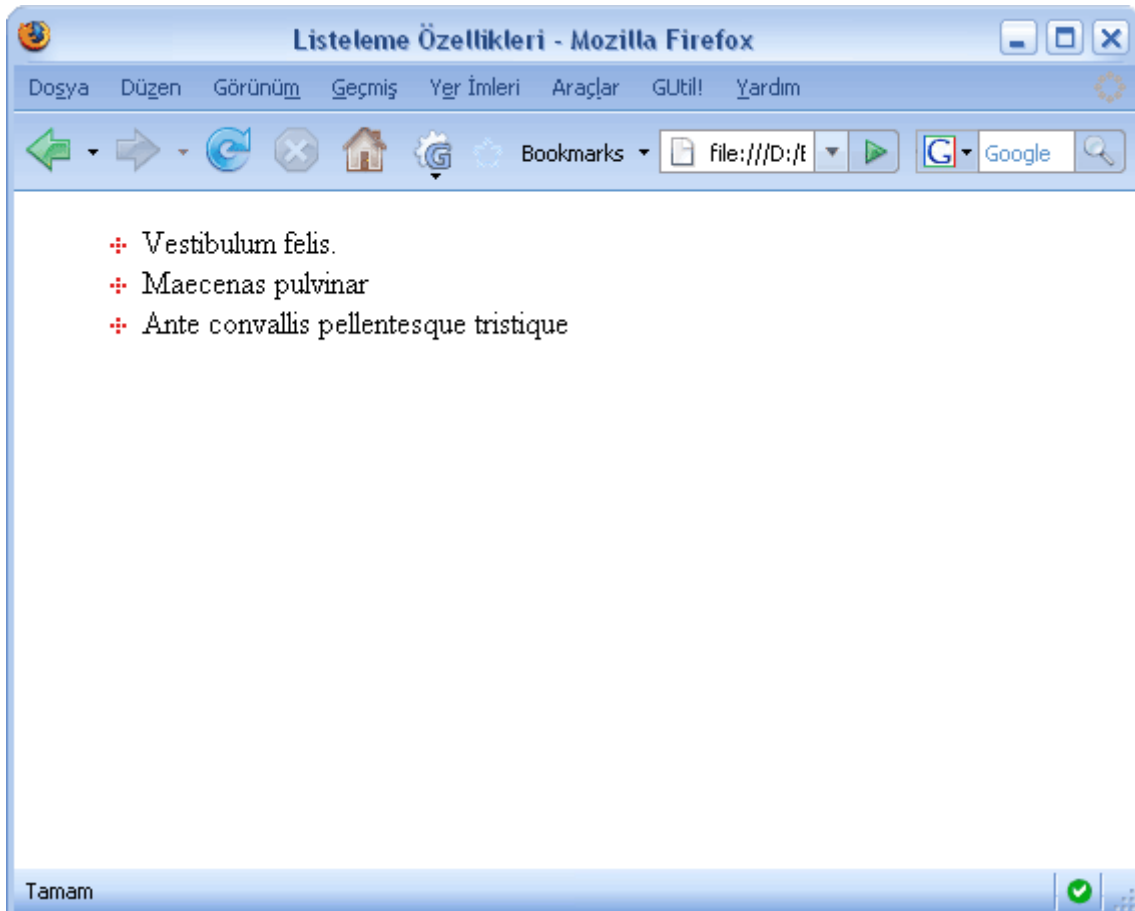
list-style-position özelliği **list-item** işaretlerin metnin içinden(inside) veya soldan dışında(outside) mı olacağını belirler.

Örnek 8-3

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Listeleme Özellikleri</title>
<style type="text/css">
ul{
list-style-image: url(liste_imgesi.gif);
list-style-position: inside;
}
</style>
</head>
<body>
<ul>
<li>Vestibulum felis.</li>
<li>Maecenas pulvinar</li>
<li>Ante convallis pellentesque tristique</li>
</ul>
</body>
</html>

```



Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

8.4. Liste biçemi (list-style)

Yapısı: **list-style: <değer>**

Aldığı Değerler: **<list-style-type> | <list-style-position> | <url>**

Başlangıç değeri: **0**

Uygulanabilen elemanlar: **display değeri list-item alan elemanlar**

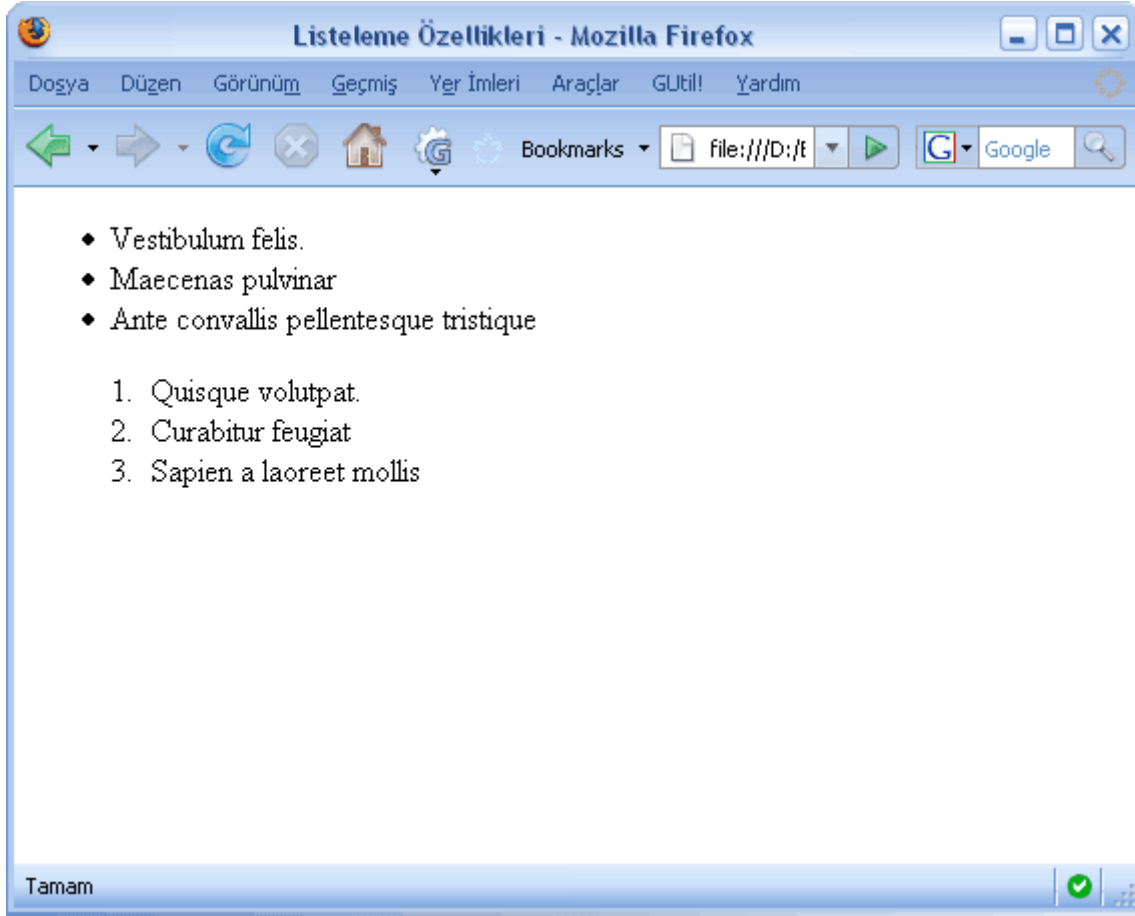
Kalıtıllık: **Var**

list-style özelliği list-style-type, list-style-position ve list-style-image özelliklerinin kısa yoludur.

Örnek 8-4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Listeleme Özellikleri</title>
<style type="text/css">
ul {
list-style: disc outside;
}
ol {
list-style: decimal inside;
}
</style>
</head>
<body>
<ul>
<li>Vestibulum felis.</li>
<li>Maecenas pulvinar</li>
<li>Ante convallis pellentesque tristique</li>
</ul>
<ol>
<li>Quisque volutpat.</li>
<li>Curabitur feugiat</li>
<li>Sapien a laoreet mollis</li>
</ol>
</body>
```

</html>

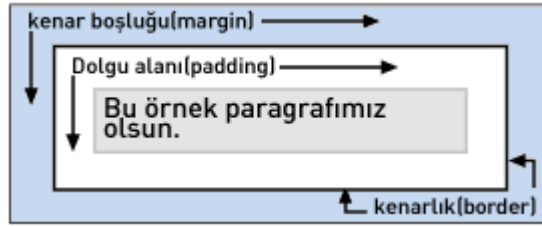
**Tarayıcı Uyumu:**

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

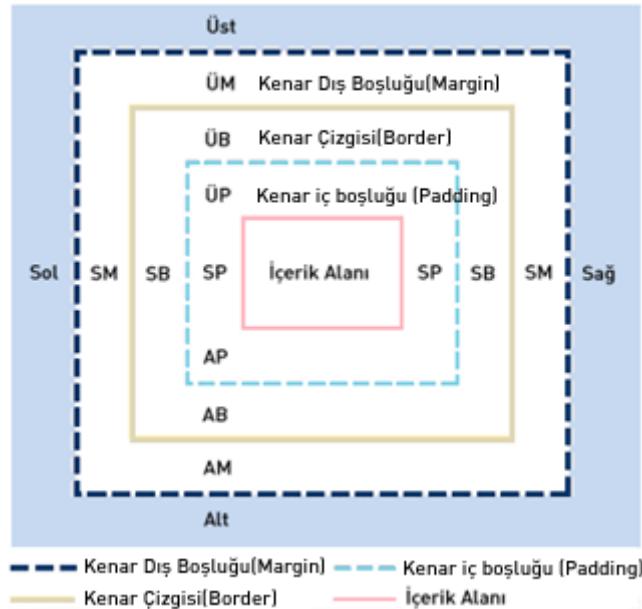
Yukarıdaki özellikler genel bir bilgi vermesi açısından anlatılmıştır. Günümüzde listeleme elemanları listeleme, menü veya değişik amaçlarla kullanılırken yukarıdaki imleyici özelliklerinden yararlanılmamaktadır. Genel kullanım liste imleyici özelliklerini kaldırıp onun yerine artalan resmi kullanılması şeklindedir. Buradaki amaç liste özelliklerinin bize sağlayamadığı yetenekleri artalan özelliği ile erişebilmemizdir. Web sitemdeki menü yapımı örnekleri bu konuyu anlamınızda size daha fazla yardımcı olacaktır.

9. Kutu Modeli Özellikleri

CSS tüm HTML elemanlarını bir dikdörtgen kutu içine konmuş varsayarak ona göre pozisyonları ve boyutlarını kontrol eder. Bu elemanlar her zaman bir kenar içi boşluğu(padding), kenar dışı boşluğu(margin), kenar çizgisi(border) ve içerik alanını kapsar. Aşağıdaki şekle bir bakınız:



Kenar dışı boşlukları(margin) her zaman saydamdır. Kenar çizgileri üç şekilde(solid, dashed ve dotted) olabilir. Artılan sadece kenar çizgisi alanı içerisine uygulanabilir, yani kenar içi boşluğu(padding) ve içerik alanı kısmını içerir.



Bir elemanın toplam genişliği; sol+sağ kenar dışı boşlukları, sol+sağ kenar içi boşlukları, sol+sağ kenar çizgileri ve elemanın içeriğini kapsar. Kenar dışı boşluklar, kenar içi boşluklar ve kenar çizgileri tercihe dayalı elemanlardır fakat başlangıçta bunların değerleri '0' olarak belirlenmiştir. Yükseklik içinde durum aynıdır. Üst-alt kenar dışı boşluğu değerleri, üst-alt

kenar içi boşluğu değerleri, üst-alt kenar çizgisi değerleri ve içerik alanı yüksekliği toplamı yükseklik değerini oluşturur.

Kutu modelinin iki elemanı bulunur. Blok seviyesinden elemanlar ve satırıçi seviyeden elemanlar.

9.1. Genel Tanımlar

Bundan sonraki kısımları anlamak için kutu modelinin genel tanımlarına açıklık getirelim.

Blok seviyesinden elemanlar(Block-level element): Blok seviyesinden elemanın görünüş formatı bir kutu gibidir. Blok seviyesinden elemanlar satırıçi elemanları ve diğer blok seviyesinden elemanları içerebilir. Tipik özellikleri yeni bir satır başlangıcı yapmalarıdır. Örnek: h6, p, li vd.

Satırıçi seviyeden elemanlar(Inline element): Satırıçi seviyeden elemanlar bir içerik gibi görünür, diğer bir satırıçi eleman veya bir metni içerebilir. Genelde yeni satır veya boşluk oluşturmazlar. Örnek: a, strong, span vd.

Normal akış(Normal flow): Batı dillerinde ve genel HTML dokümanlarında metin akışı soldan sağa, yukarıdan aşağı doğrudur. Diğer diller için akış yönü değiştirilebilir. HTML elemanları normal akış dışına float ve konumlandırma özellikleri atanarak çıkarılabilirler.

Yerleştirilmeyen elemanlar(Nonreplaced element): Belli bir boyutu olmayan doküman içeriğine göre boyutu değişen elemanlardır. Örnek olarak paragraf verilebilir.

Yerleştirilen eleman(Replaced element): Sunucunun kendilerine belli bir yer ayırdığı elemanlardır. Klasik örneği resimlerdir resimler eklendiği dokümanın eklendiği noktasında bulunur. Ayrıca bir çok form elemanında bu sınıflandırmaya girer.

Kök Eleman(Root element): Doküman hiyerarşisinde en üstteki elemandır. HTML için <html> elemanıdır. XML herhangi bir eleman kök eleman olma izni verir.

Tüm kutu modelleri aşağıdaki bölümleri içerir:

İçerik Alanı: Elemanın asıl içeriğini oluşturan kısımdır. Metin ve nesnelere içerir.

Kenar içi boşluğu (Padding): Elemanın kenar dışı boşluğudur.

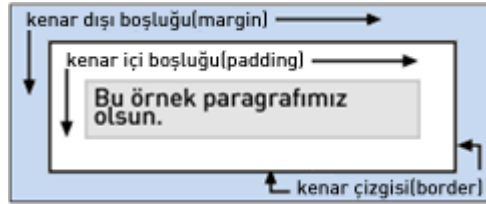
Kenar çizgisi (Border): Elemanın kenar çizgilerini gösterir.

Kenar dışı boşluğu (Margin): Elemanın kenar dışı boşluklarıdır.

Sırası ile bu özellikleri inceleyelim.

9.2. Kenarlık Çizgisi(Border) Özellikleri

Kenar çizgisi(border), içerik alanı ve kenar içi boşluğu(padding) etrafındaki bir veya daha fazla çizgiye denir.



- Kenar çizgisi biçemi (border-style)
- border-top-style, border-right-style, border-bottom-style, border-left-style
- Kenar çizgisi genişliği (border-width)
- border-top-width, border-right-width, border-bottom-width, border-left-width
- Kenar çizgisi rengi (border-color)
- border-top-color, border-right-color, border-bottom-color, border-left-color
- Kenar çizgisi (border)

Sırasıyla incelersek:

9.2.1. Kenar çizgisi biçemi (border-style)

Yapısı: **border-style: <değer>**

Aldığı Değerler: [**none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset**]{1,4} | **inherit**

Başlangıç değeri: **tanımlama yok**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

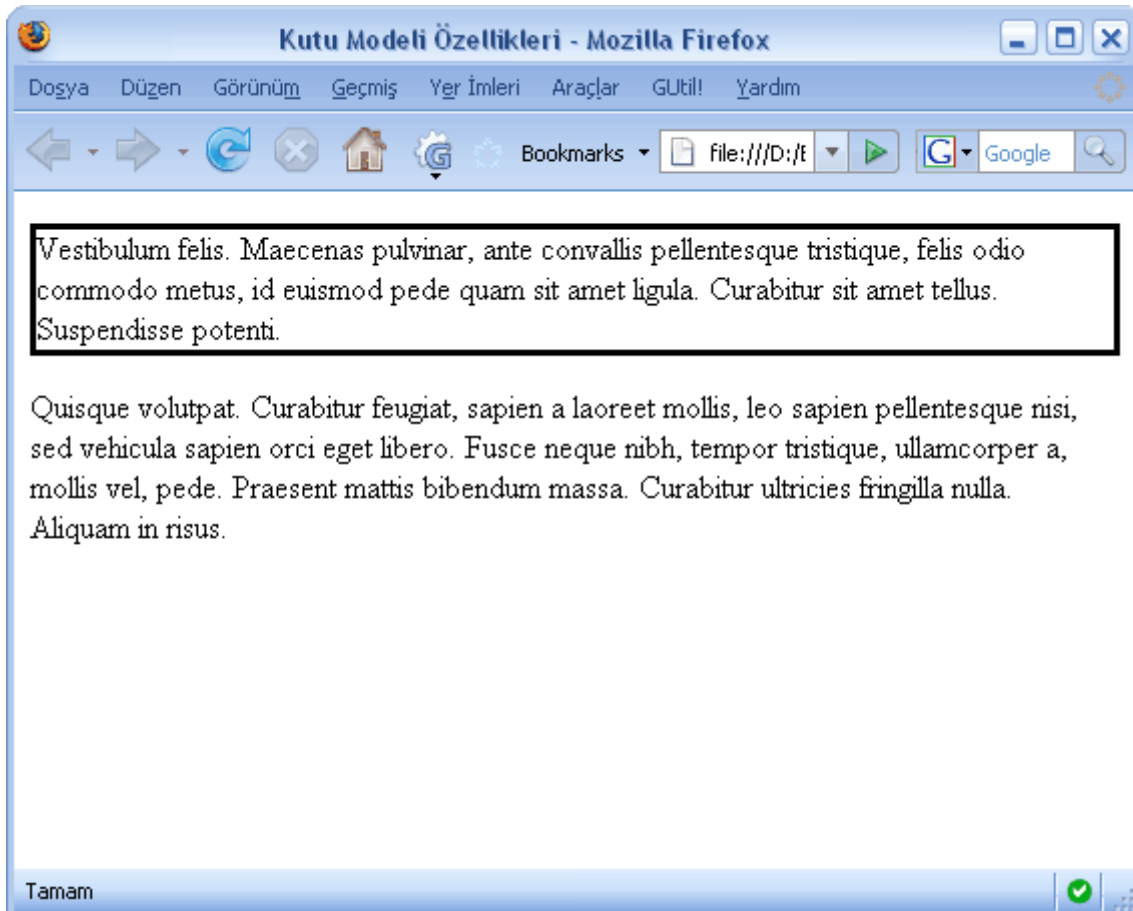
border-style özelliği kenarlık biçimini belirlememizi sağlar.(Örn: dotted(noktalı), double(çift) vd.) 10 farklı değeri CSS desteklemektedir. Bu özellik kenarlığın görünmesi için tanımlanması gereklidir. Bir veya dört değer alabilir eğer dört değer alırsa sırasıyla üst, sağ, alt ve sol kenarlık stilini belirler. Eğer tek değeri varsa tüm kenarlık değeri ataması yapılmış demektir. Eğer iki veya üç değer ataması yapılmış ise margin'de gördüğümüz sıralama söz konusudur.

Örnek 9-1

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
p.yeni {
border-style: solid;
}
</style>
</head>
<body>
<p class="yeni">Vestibulum felis. Maecenas pulvinar, ante convallis
pellentesque tristique, felis odio commodo metus, id euismod pede quam
sit amet ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>

```



Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

9.2.2. border-top-style, border-right-style, border-bottom-style, border-left-style

Yapısı: **border-[top,left,right,bottom]-style: <deger>**

Aldığı Değerler: **none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit**

Başlangıç değeri: **tanımlama yok**

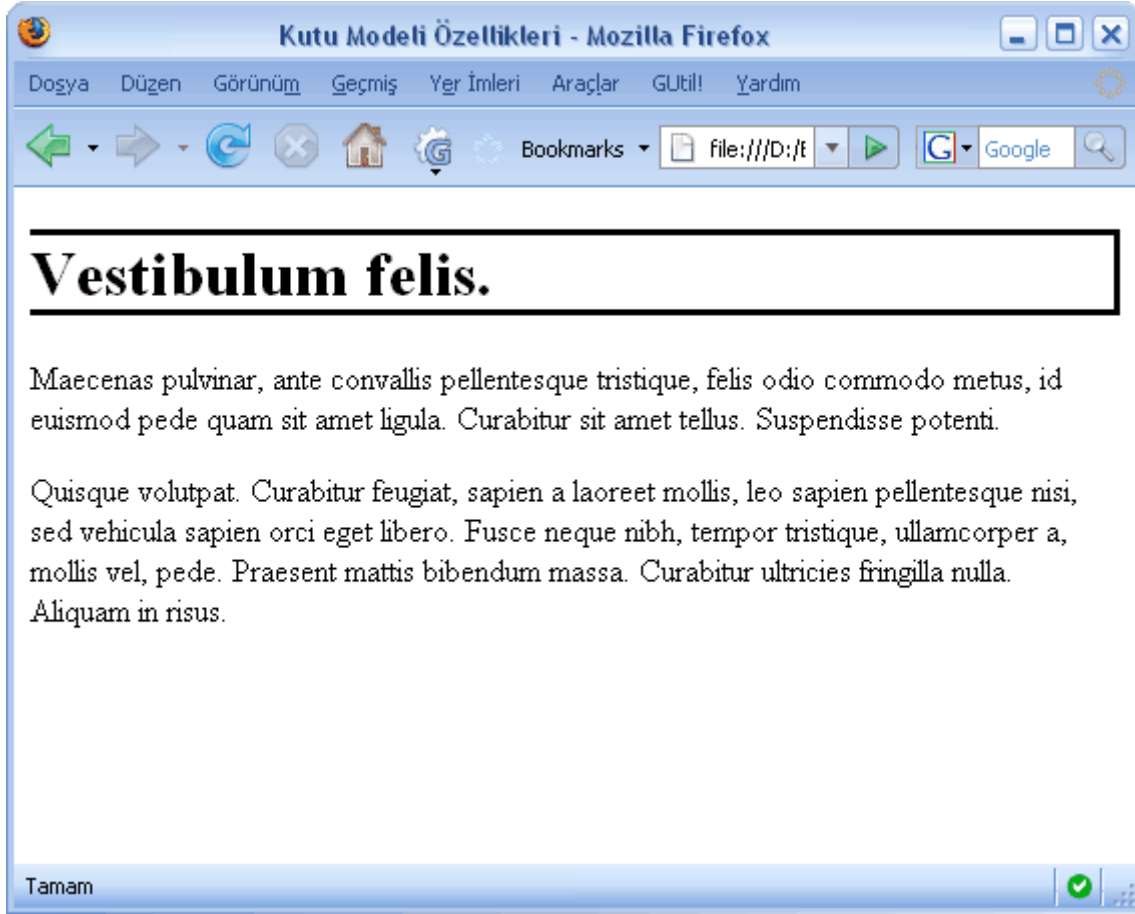
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Yok**

border-top-style, border-right-style, border-bottom-style, border-left-style border-style özelliğinin her kenara ayrı ayrı atamasını yapabilmek için kullanılır.

Örnek 9-2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
h1{
border-style: solid;
border-left-style: none;
}
</style>
</head>
<body>
<h1>Vestibulum felis.</h1>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus.</p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 6+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

9.2.3. Kenar çizgisi genişliği (border-width)

Yapısı: **border-width: <değer>**

Aldığı Değerler: [**thin** | **medium** | **thick** | **<uzunluk değeri >**]{1,4} | **inherit**

Başlangıç değeri: **tanımlama yok**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Yok**

İlk olarak biçemi belirledikten sonra kenarlık genişliğini belirlemek için **border-width** değeri kullanılır. **border-width** yüzde değeri alamaz.

thin [ince]: İnce kenar çizgisi.

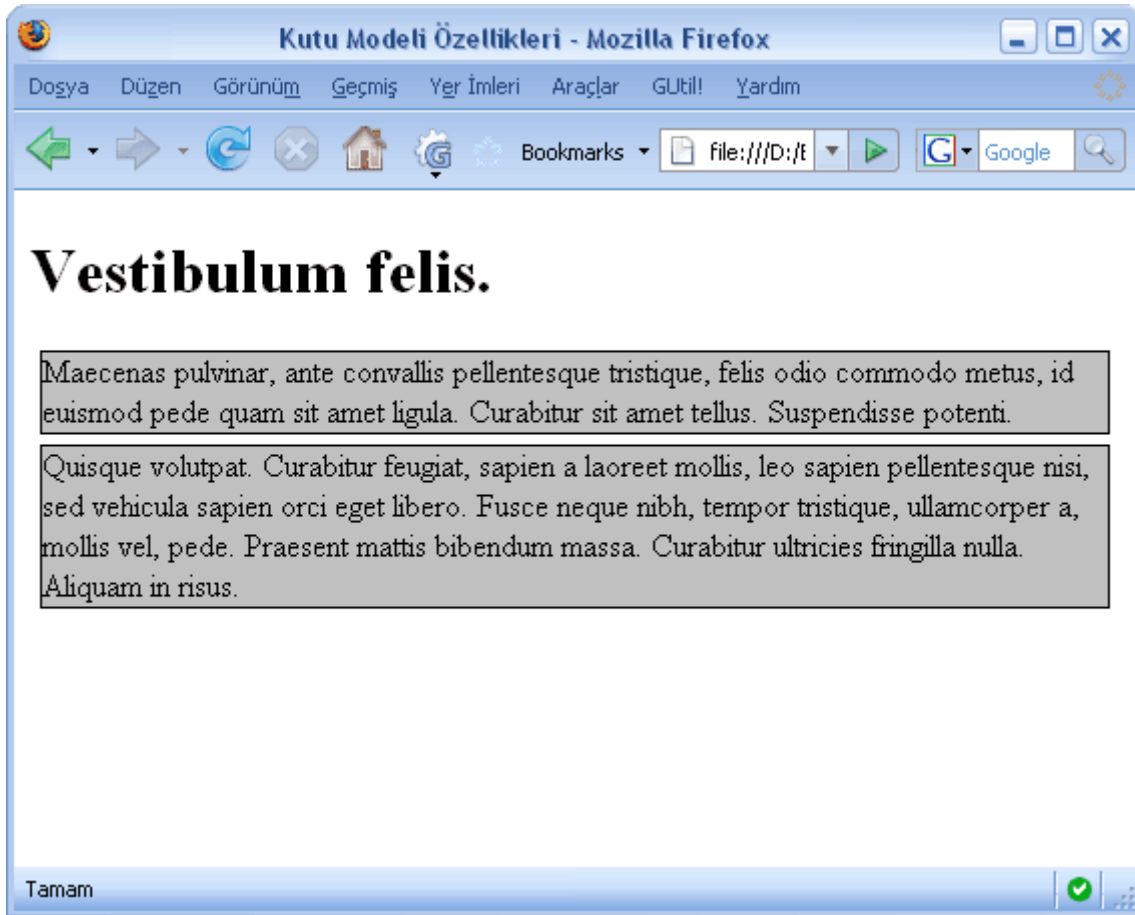
medium [orta]: Orta kalınlıkta kenar çizgisi.

thick [kalın]: Kalın kenar çizgisi.

Uzunluk: Kenar çizgisi kalınlığı doğrudan belirtilen (görelili olmayan/hesaplanmayan) bir değerdir ve negatif olamaz.

Örnek 9-3

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
p{
    margin: 5px;
    background-color: silver;
    border-style: solid;
    border-width: 1px;
}
</style>
</head>
<body>
<h1>Vestibulum felis.</h1>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

9.2.4. border-top-width, border-right-width, border-bottom-width, border-left-width

Yapısı: **border-[top,left,right,bottom]-width: <değer>**

Aldığı Değerler: **thin | medium | thick | <uzunluk değeri > | inherit**

Başlangıç değeri: **tanımlama yok**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Yok**

border-top-width, border-right-width, border-bottom-width, border-left-width border-width özelliğinin her kenara ayrı ayrı atamasını yapabilmek için kullanılır.

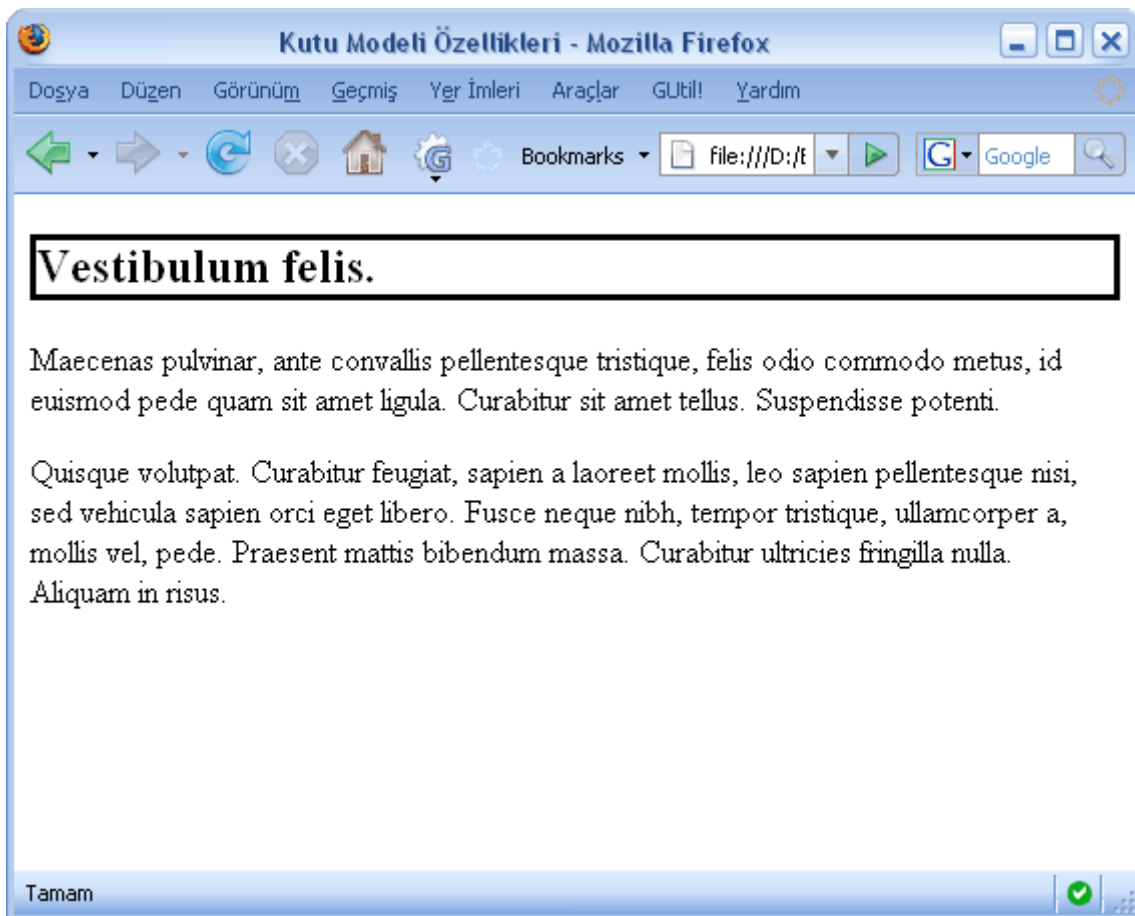
Örnek 9-4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
h2{
border-left-width: medium;
border-style: solid;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>

```



Tarayıcı Uyumu:

Internet Explorer 6+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

9.2.5. Kenar Çizgisi Rengi (border-color)

Yapısı: **border-color: <değer>**

Aldığı Değerler: [**<renk> | transparent**]{1,4} | **inherit**

Başlangıç değeri: **tanımlama yok**

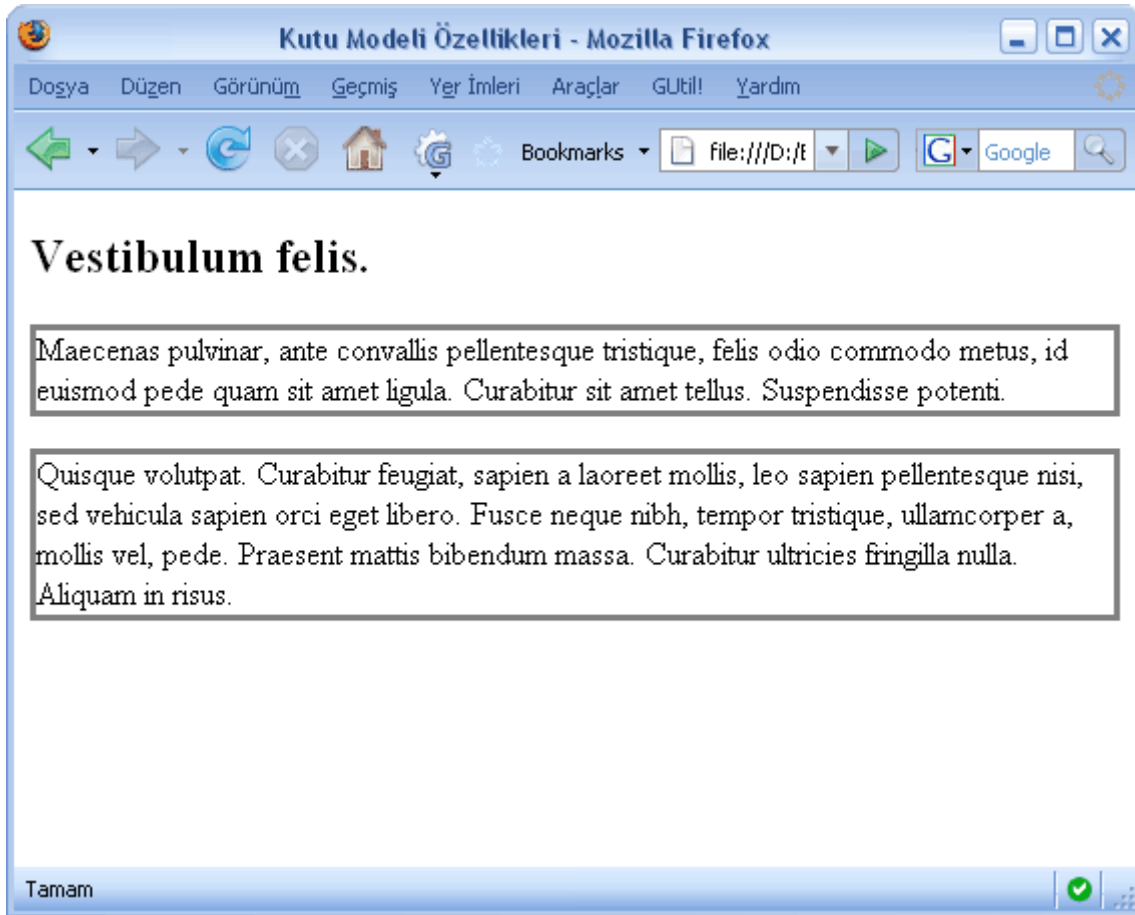
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

border-color özelliği kenarlık rengini belirler.

Örnek 9-5

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
p{
    border-style: solid;
    border-color: gray;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Tek değer tüm kenarlık renklerini belirler, her kenar için ayrı renk tanımlamasıda yapılabilir.

Tarayıcı Uyumu:

Internet Explorer 4+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

9.2.6. border-top-color, border-right-color, border-bottom-color, border-left-color

Yapısı: **border-[top,left,right,bottom]-color: <değer>**

Aldığı Değerler: [<renk> | transparent]{1,4} | inherit

Başlangıç değeri: **tanımlama yok**

Uygulanabilen elemanlar: **tüm elemanlar**

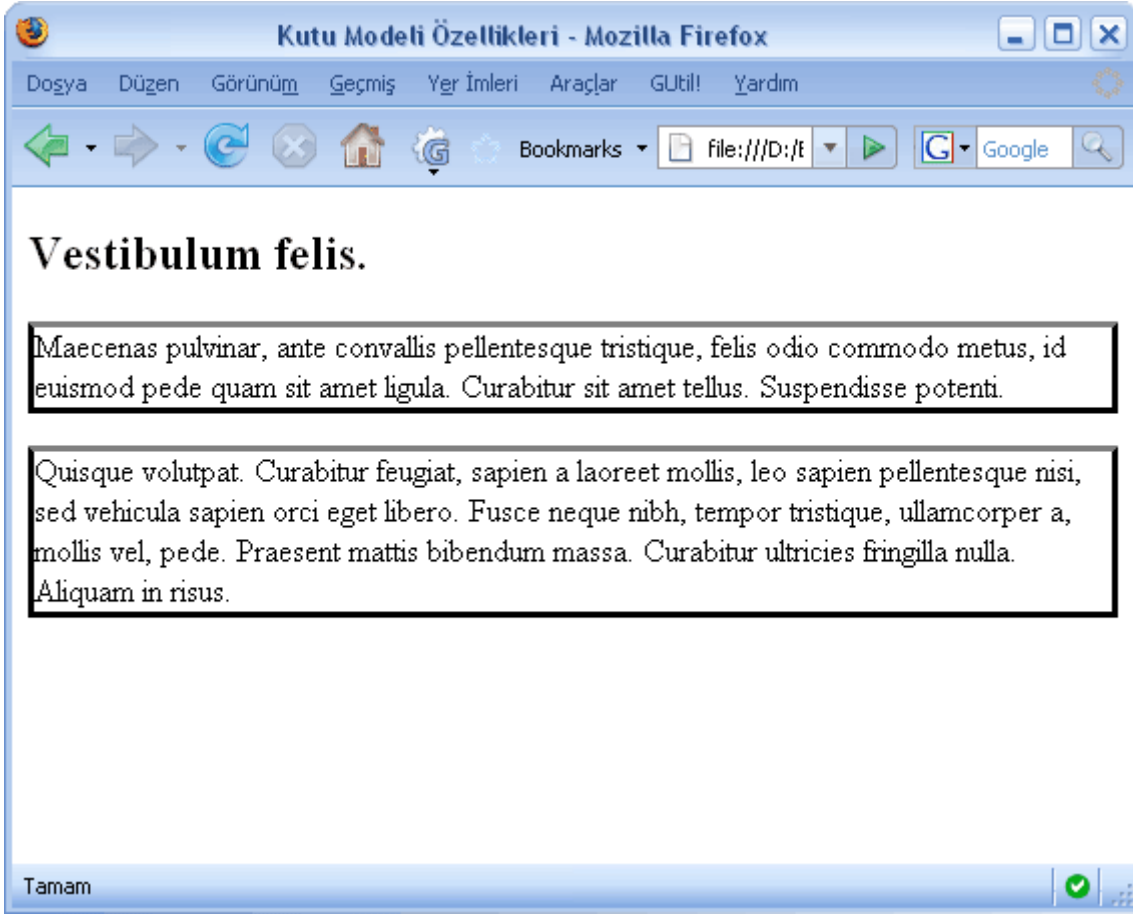
Kalıtıllık: **Yok**

border-color özelliğinin her kenara ayrı ayrı atamasını yapabilmek için kullanılır.

Örnek 9-6

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
p{
    border-style: solid;
    border-top-color: gray;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo
sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque
nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis
bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Tek değer tüm kenarlık renklerini belirler, her kenar için ayrı renk tanımlamasıda yapılabilir.

Tarayıcı Uyumu:

Internet Explorer 4+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

Bir kenarlığı yok etmek için kalınlık değeri veririz. Birde CSS2 ile birlikte gelen transparent özelliği vardır ki bu kenarlığı yok etmez sadece görünmez yapar.

9.2.7. Kenar Çizgisi (border)

Yapısı: **border: <değer>**

Aldığı Değerler: **<border-width> | <border-style> | <renk>**

Başlangıç değeri: **tanımlama yok**

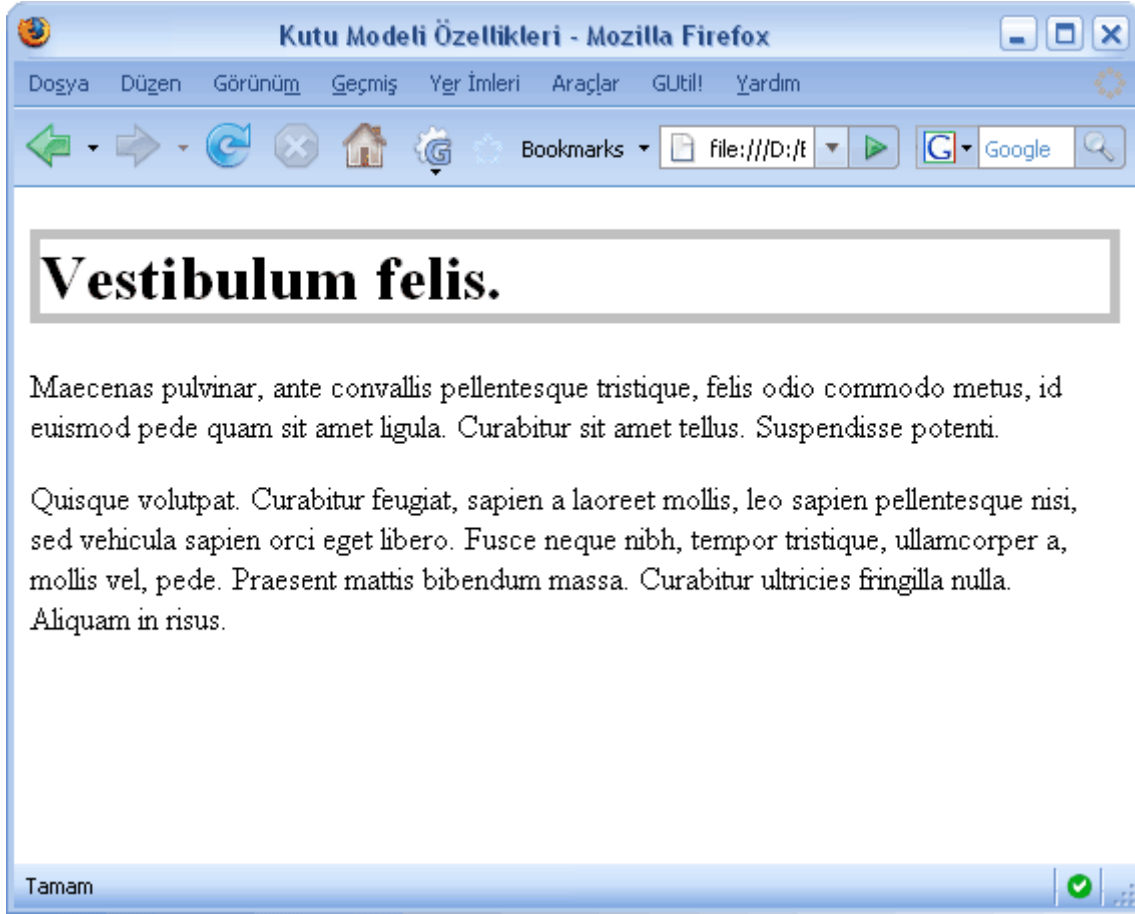
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Yok**

border yukarıda sıraladığımız özellikleri tek sefer de tanımlamak için kullanılır. Bir kısa yoldur.

Örnek 9-7

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
h1{
border: thick solid silver;
}
</style>
</head>
<body>
<h1>Vestibulum felis.</h1>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit amet
tellus. Suspendisse potenti.</p>
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo sapien
pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque nibh,
tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis bibendum
massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Bu özellikleri tek tek atamak isteseydik şöyle bir kod yazmamız gerekecekti.

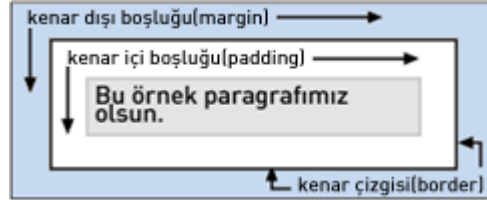
```
h1{
border-top: thick silver solid;
border-bottom: thick silver solid;
border-right: thick silver solid;
border-left: thick silver solid;
}
```

Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

9.3. Kenar Dış Boşluğu(Margin) Özellikleri

Kenar dış boşluğu(margin) özelliği bir elemanın etrafındaki boşluk olarak tanımlanır. Negatif değer alabilir. Tek tek özellikler(margin-top, margin-left vd.) atanabildiği gibi tek bir özellikle(margin) de tanımlama yapılabilir.



- Üst kenar dışı boşluğu (margin-top)
- Sağ kenar dışı boşluğu (margin-right)
- Alt kenar dışı boşluğu (margin-bottom)
- Sol kenar dışı boşluğu (margin-left)
- Kenar dışı boşluğu (margin)

9.3.1. Üst Kenar dışı boşluğu (margin-top)

Yapısı: **margin-top: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | auto**

Başlangıç değeri: **0**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Yok**

margin-top özelliği elemanın üst kenar dış boşluğunu belirli bir değer kadar veya yüzde olarak belirler. Negatif değer alabilir.

Örnek 9-8

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
body{
margin-top: 100px;
}
</style>
</head>
<body>
<h1>Vestibulum felis.</h1>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
```

```
<p>Quisque volutpat. Curabitur feugiat, sapien a laoreet mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero. Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede. Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla. Aliquam in risus. </p>
</body>
</html>
```



Internet Explorer 6'da body elementi veya kapsayıcı blok elemanlara genişlik tanımı yapıldığında dikeydeki kenar dış boşluğu(margin-top) yüzde değer kullanıldığında soruna nende oluyor.

Internet Explorer 7 ve alt sürümlerinde kenar dış boşluğu çökmesi sorunu yaşanmaktadır. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=362> adresine göz atın.

Tarayıcı Uyumu:

Internet Explorer 4+

Firefox 1+

Safari 1.3+

Opera 3.6+

CSS 1+

9.3.2. Sağ kenar dış boşluğu (margin-right)

Yapısı: **margin-right: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | auto | inherit**

Başlangıç değeri: **0**

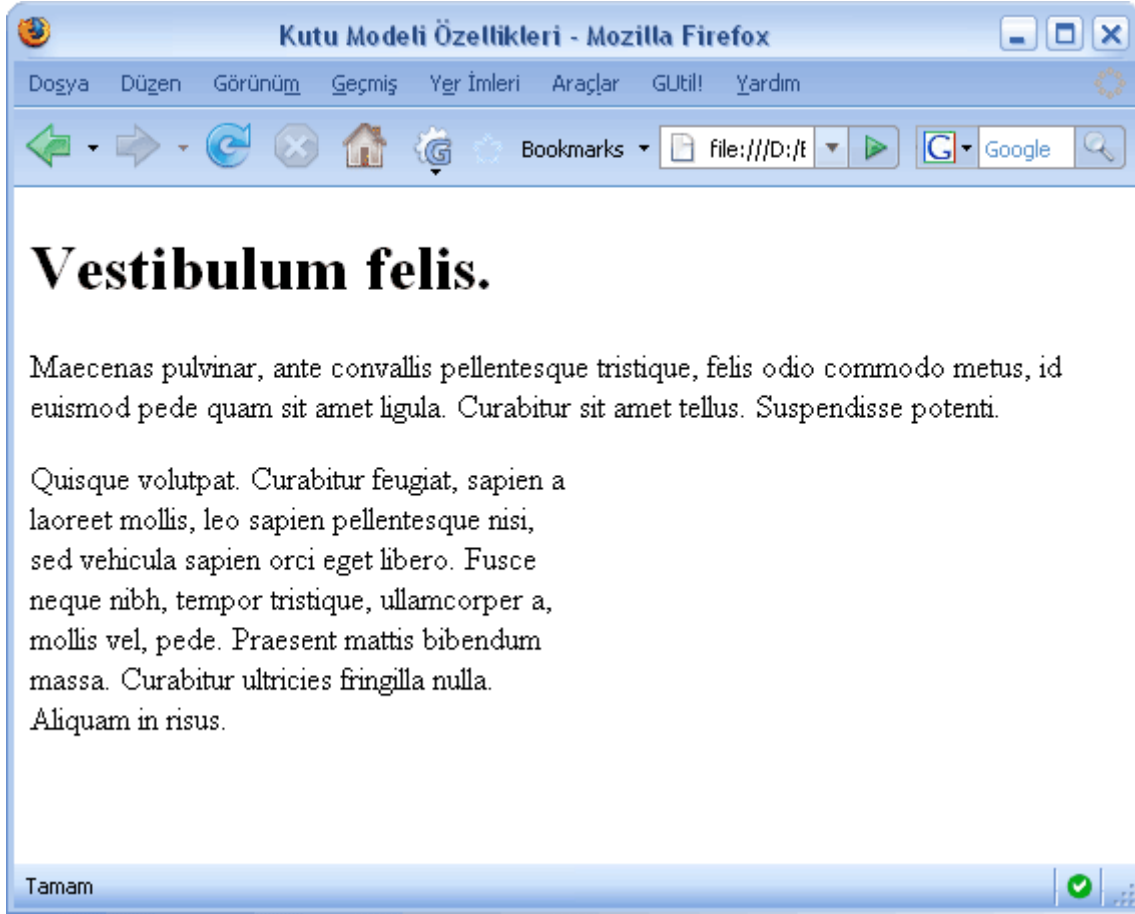
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

margin-right özelliği elemanın sağ kenar dış boşluğunu belli bir değer kadar veya yüzde olarak belirler. Negatif değer alabilir.

Örnek 9-9

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
p.diger {
    margin-right: 50%;
}
</style>
</head>
<body>
<h1>Vestibulum felis.</h1>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
</body>
</html>
```



Internet Explorer 5.5(IE6 garip mod dâhil) blok seviyeli elemanların ortalanması için kenar dış boşluğu sağ ve solu için verilen **auto** yöntemini desteklememektedir.

Internet Explorer 6 ve alt sürümlerinde **float** uygulanmış elemanlara uygulanan sağ ve sol kenar boşluğu değerini iki kere uygular. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=131> adresine bakınız.

Tarayıcı Uyumu:

Internet Explorer 3+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

9.3.3. Alt kenar dış boşluğu (margin-bottom)

Yapısı: **margin-bottom: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | auto**

Başlangıç değeri: **0**

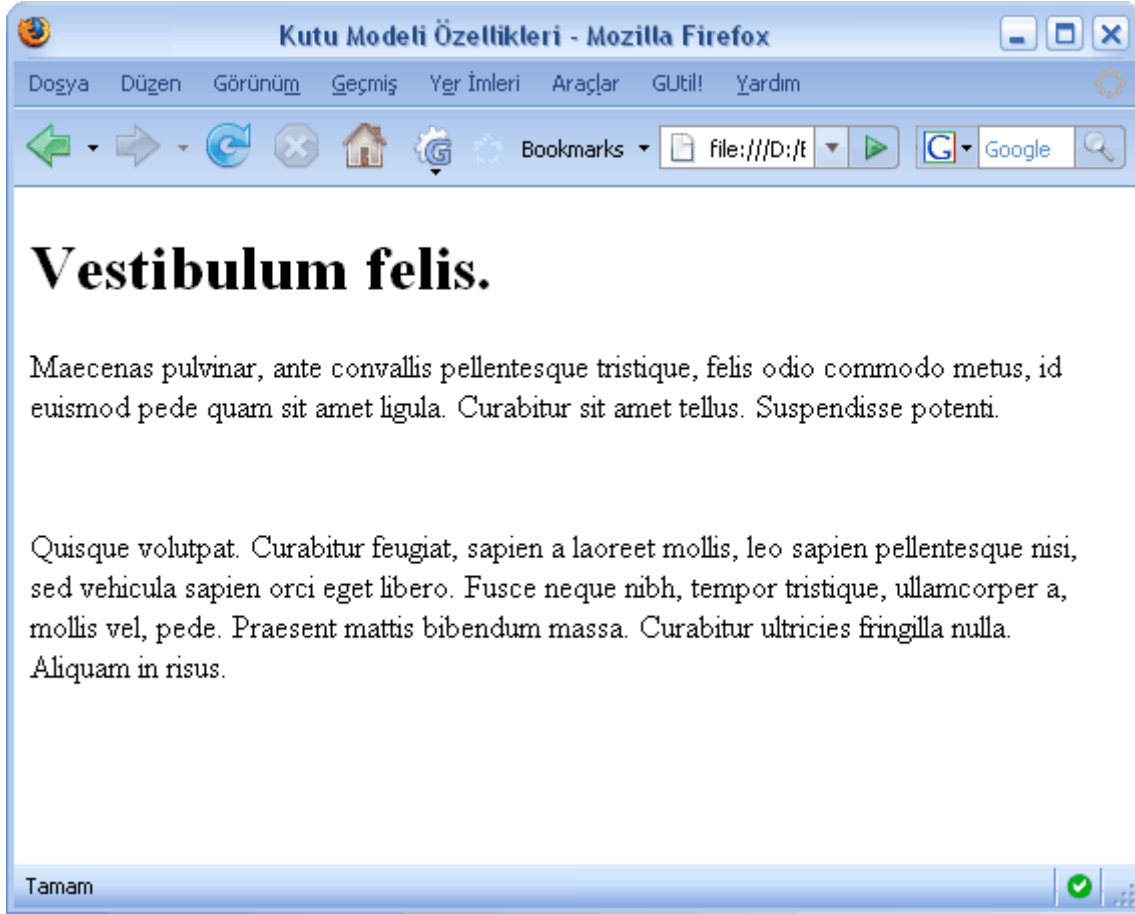
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

margin-bottom özelliği elemanın alt kenar dış boşluğunu belli bir değer kadar veya yüzde olarak belirler. Negatif değer alabilir.

Örnek 9-10

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
p {
margin-bottom: 50px;
}
</style>
</head>
<body>
<h1>Vestibulum felis.</h1>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id eismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
</body>
</html>
```



Internet Explorer 6'da body elementi veya kapsayıcı blok elemanlara genişlik tanımı yapıldığında dikeydeki kenar dış boşluğu(margin-top) yüzde değerlerinde sorun yaşanıyor.

Internet Explorer 7 ve alt sürümlerinde kenar dış boşluğu çökmesi sorunu yaşanmaktadır. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=362> adresine göz atın.

Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

9.3.4. Sol kenar dış boşluğu (margin-left)

Yapısı: **margin-left: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | auto**

Başlangıç değeri: **0**

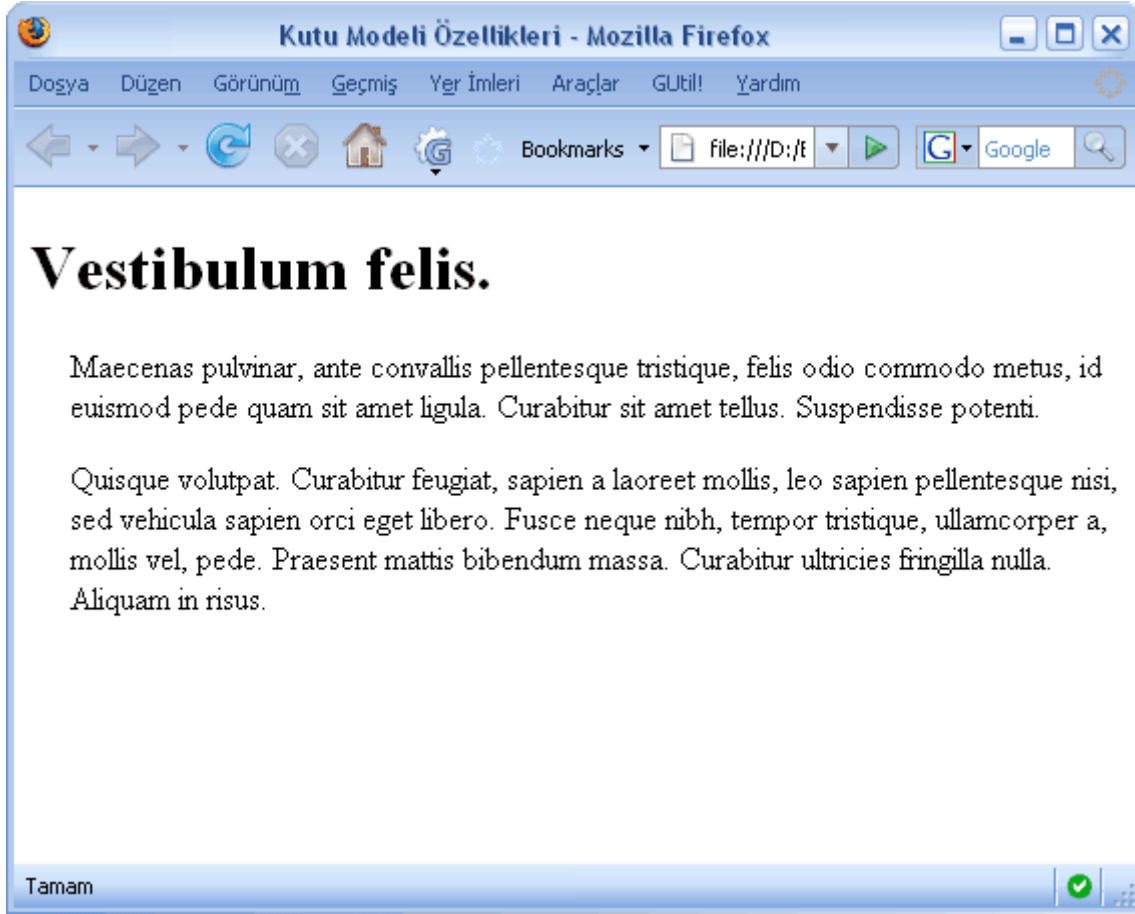
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

margin-left özelliği elemanın sol kenar dış boşluğunu belli bir değer kadar veya yüzde olarak belirler. Negatif değer alabilir.

Örnek 9-11

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
p {
margin-left: 20px;
}
</style>
</head>
<body>
<h1>Vestibulum felis.</h1>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id eismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
</body>
</html>
```



Internet Explorer 5.5(IE6 garip mod dâhil) blok seviyeli elemanların ortalanması için kenar dış boşluğu sağ ve solu için verilen **auto** yöntemini desteklememektedir.

Internet Explorer 6 ve alt sürümlerinde **float** uygulanmış elemanlara uygulanan sağ ve sol kenar boşluğu değerini iki kere uygular. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=131> adresine bakınız.

Tarayıcı Uyumu:

Internet Explorer 3+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

9.3.5. Kenar dış boşluğu (margin)

Yapısı: **margin: <deger>**

Aldığı Değerler: [**<percentage>** | **<length>** | **auto**]{1,4}

Başlangıç değeri: **Tanımsız**

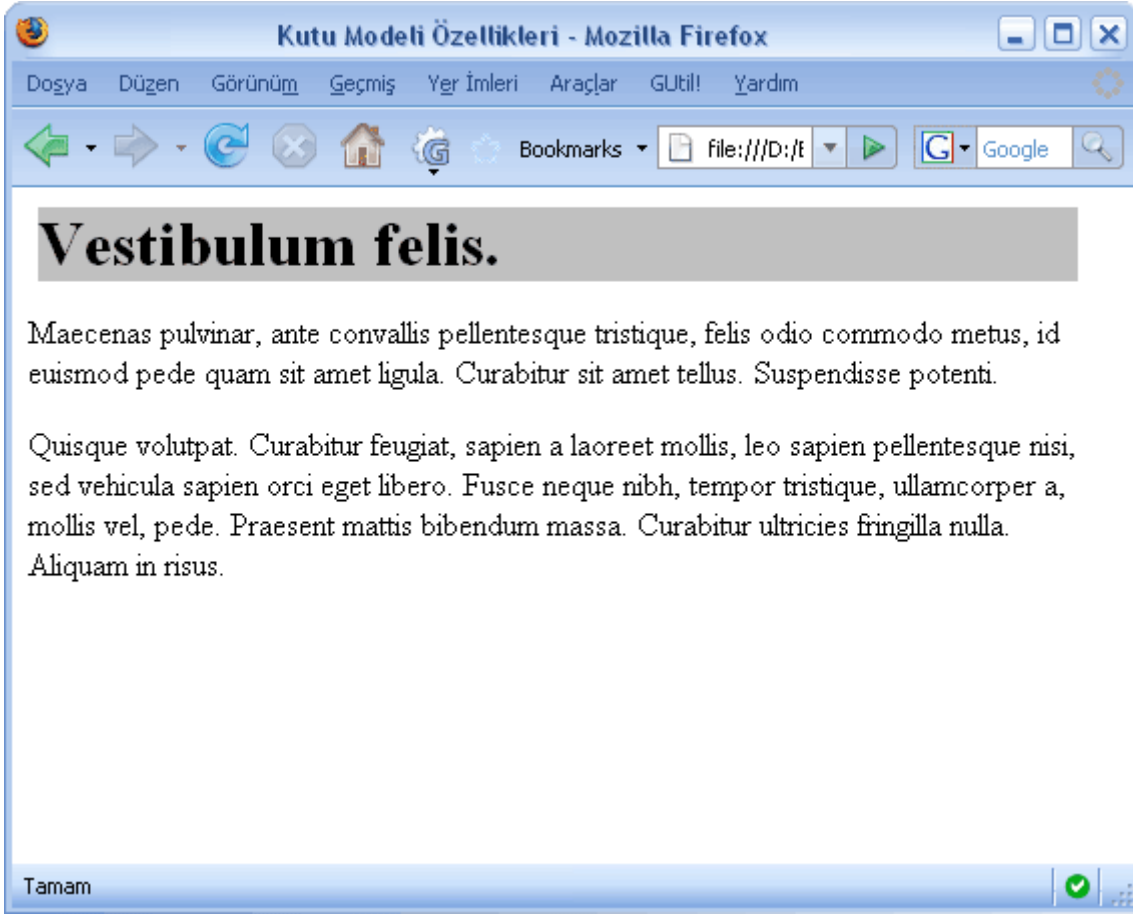
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

margin özelliği yukarıdaki özelliklerin tek bir özellik ile uygulanması için kullanılır. Bir kısa yoldur.

Örnek 9-12

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
h1 {
    margin: 0.25in;
    background-color: silver;
}
h1 {
    margin: 10px 20px 15px 5px;
}
</style>
</head>
<body>
<h1>Vestibulum felis.</h1>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
</body>
</html>
```



Yukarıda **h1** için margin değerleri sıralaması şöyledir:

margin: üst sağ alt sol(saat yönünde)

İkili ve üçlü kullanımda mevcuttur

```
h1{margin: 0.25em 0 0.5em;} /* esittir '0.25em 0 0.5em 0' */
h2 {margin: 0.15em 0.2em;} /* esittir '0.15em 0.2em 0.15em
0.2em' */
p {margin: 0.5em 10px;} /* esittir '0.5em 10px 0.5em 10px' */
p.close {margin: 0.1em;} /* esittir '0.1em 0.1em 0.1em 0.1em'
*/
```

Internet Explorer 6'da body elementi veya kapsayıcı blok elemanlara genişlik tanımı yapıldığında dikeydeki kenar dış boşluğu(margin-top) yüzde değerlerinde sorun yaşanıyor.

Internet Explorer 5.5(IE6 garip mod dâhil) blok seviyeli elemanların ortalanması için kenar dış boşluğu sağ ve solu için verilen auto yöntemini desteklememektedir.

Internet Explorer 7 ve alt sürümlerinde kenar dış boşluğu çökmesi sorunu yaşanmaktadır. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=362>

adresine göz atın.

Internet Explorer 6 ve alt sürümlerinde **float** uygulanmış elemanlara uygulanan sağ ve sol kenar boşluğu değerini iki kere uygular. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=131> adresine bakınız.

Tarayıcı Uyumu:

Internet Explorer 4+

Firefox 1+

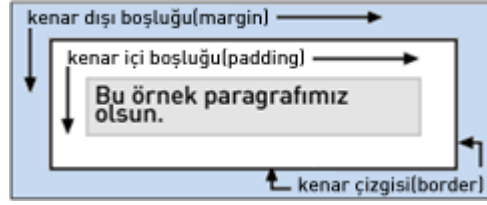
Safari 1.3+

Opera 3.6+

CSS 1+

9.4. Kenar iç boşluğu (Padding) özellikleri

Kenar içi boşluk(Padding) içerik alanı ile kenar çizgisi arasındaki alandır.



- Üst kenar içi boşluğu(padding-top)
- Sağ kenar içi boşluğu (padding-right)
- Alt kenar içi boşluğu (padding-bottom)
- Sol kenar içi boşluğu (padding-left)
- Kenar içi boşluğu (padding)

9.4.1. Üst kenar içi boşluğu(padding-top)

Yapısı: **padding-top: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | inherit**

Başlangıç değeri: **0**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Yok**

padding-top özelliği içerik alanı ile üst kenar çizgisi arasındaki mesafeyi belirler.

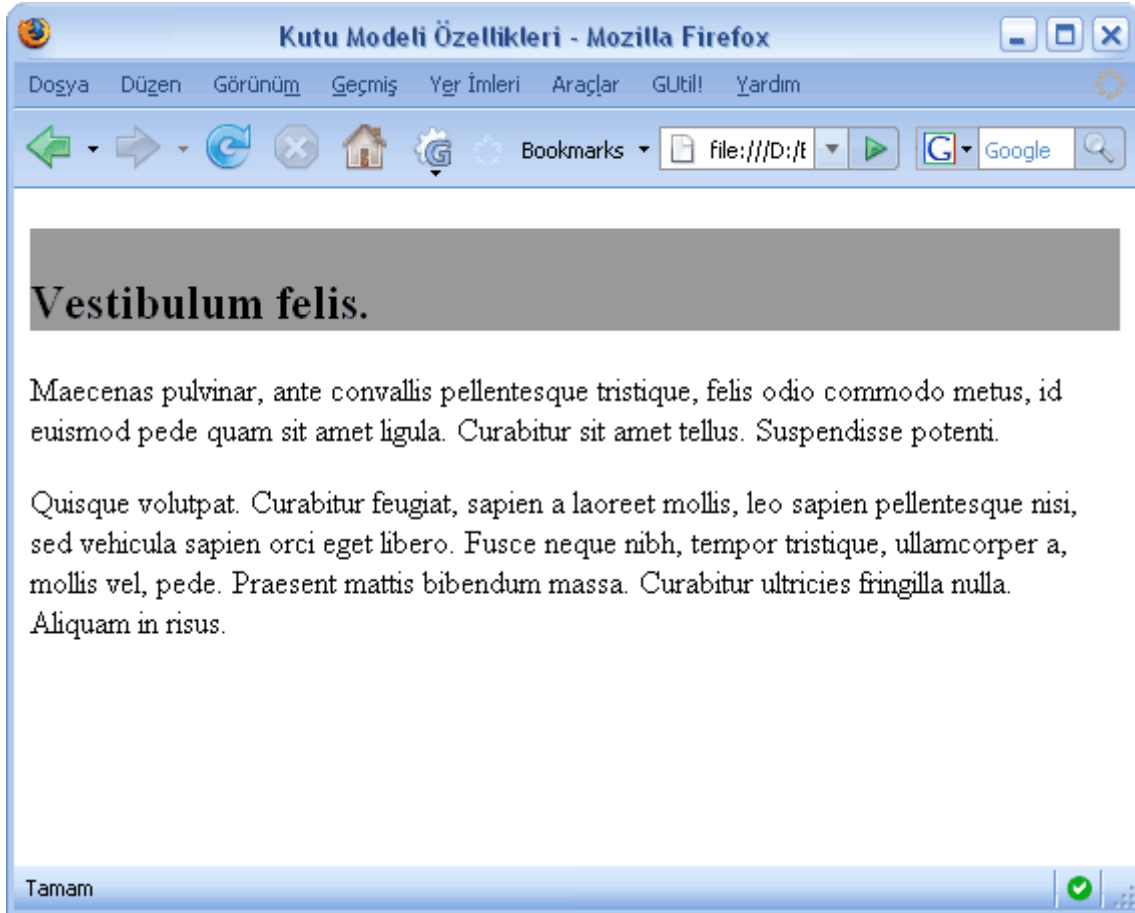
Örnek 9-13

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
h2 {
padding-top: 0.25in;
background-color:#999;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
```

```

mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
</body>
</html>

```



Internet Explorer 5.5 ve alt sürümlerinde(Internet Explorer 6 ve 7 garip mod dâhil) kenar içi boşluğu(padding) değerlerini içerik alanı genişliğinden sayar. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=13> adresini inceleyiniz.

Internet Explorer 6 ve alt sürümlerinde satırıçi elemanların dikey yükseklik değerini tam olarak göstermesi için position:relative değeri eklenir.

Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

9.4.2. Sağ kenar içi boşluğu(padding-right)

Yapısı: **padding-right: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | inherit**

Başlangıç değeri: **0**

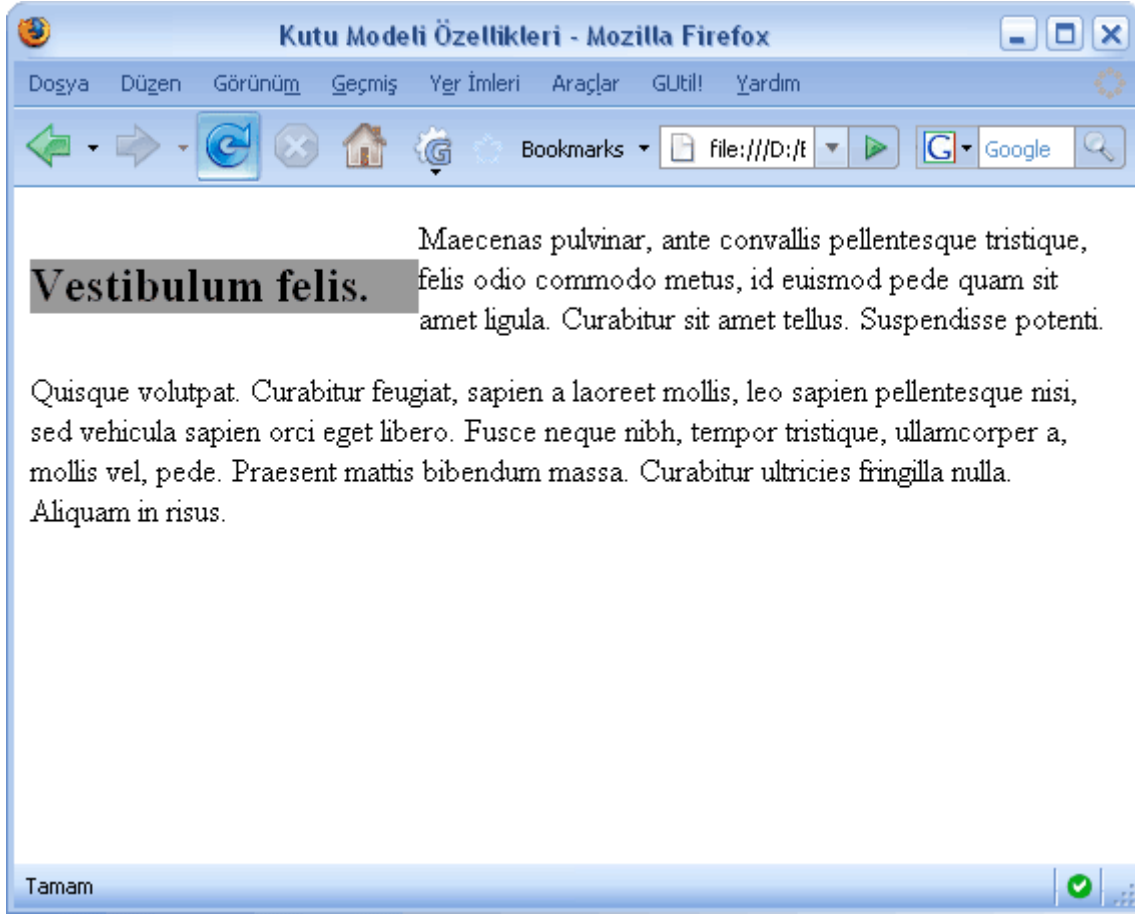
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

padding-right özelliği içerik alanı ile sağ kenar çizgisi arasındaki mesafeyi belirler.

Örnek 9-14

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
h2 {
padding-right: 0.25in;
background-color:#999;
float:left;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
</body>
</html>
```



Internet Explorer 5.5 ve alt sürümlerinde (Internet Explorer 6 ve 7 garip mod dâhil) kenar içi boşluğu(padding) değerlerini içerik alanı genişliğinden sayar. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=13> adresini inceleyiniz.

Firefox 2.0.0.7 ve önceki sürümlerinde iki satıra uzamış satır içi elemanlara uygulanan padding-right değeri sorun çıkarır. Önceki satır ve mevcut satırın her ikisinde padding uygulanır. Ayrıntılı bilgi için [tıklayınız.](#)

Tarayıcı Uyumu:

Internet Explorer 4+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

9.4.3. Alt kenar içi boşluğu (padding-bottom)

Yapısı: **padding-bottom: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | inherit**

Başlangıç değeri: **0**

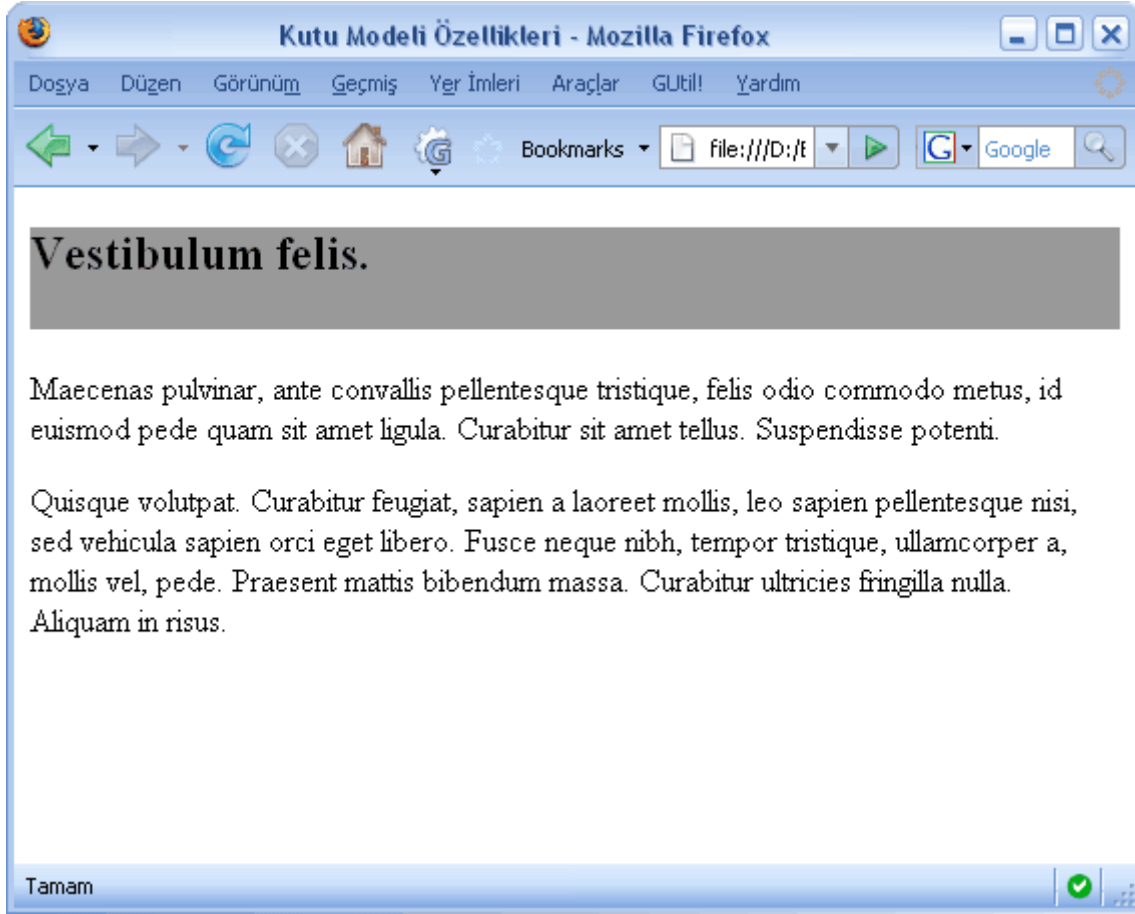
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Yok**

padding-bottom özelliği içerik alanı ile alt kenar çizgisi arasındaki mesafeyi belirler.

Örnek 9-15

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
h2 {
padding-bottom: 0.25in;
background-color: #999;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
</body>
</html>
```



Internet Explorer 5.5 ve alt sürümlerinde (Internet Explorer 6 ve 7 garip mod dâhil) kenar içi boşluğu (padding) değerlerini içerik alanı genişliğinden sayar. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=13> adresini inceleyiniz.

Internet Explorer 6 ve alt sürümlerinde satır içi elemanların dikey yükseklik değerini tam olarak göstermesi için position: relative değeri eklenir.

Tarayıcı Uyumu:

Internet Explorer 4+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

9.4.4. Sol kenar içi boşluğu (padding-left)

Yapısı: **padding-left: <değer>**

Aldığı Değerler: **<uzunluk değeri> | <yüzde> | inherit**

Başlangıç değeri: **0**

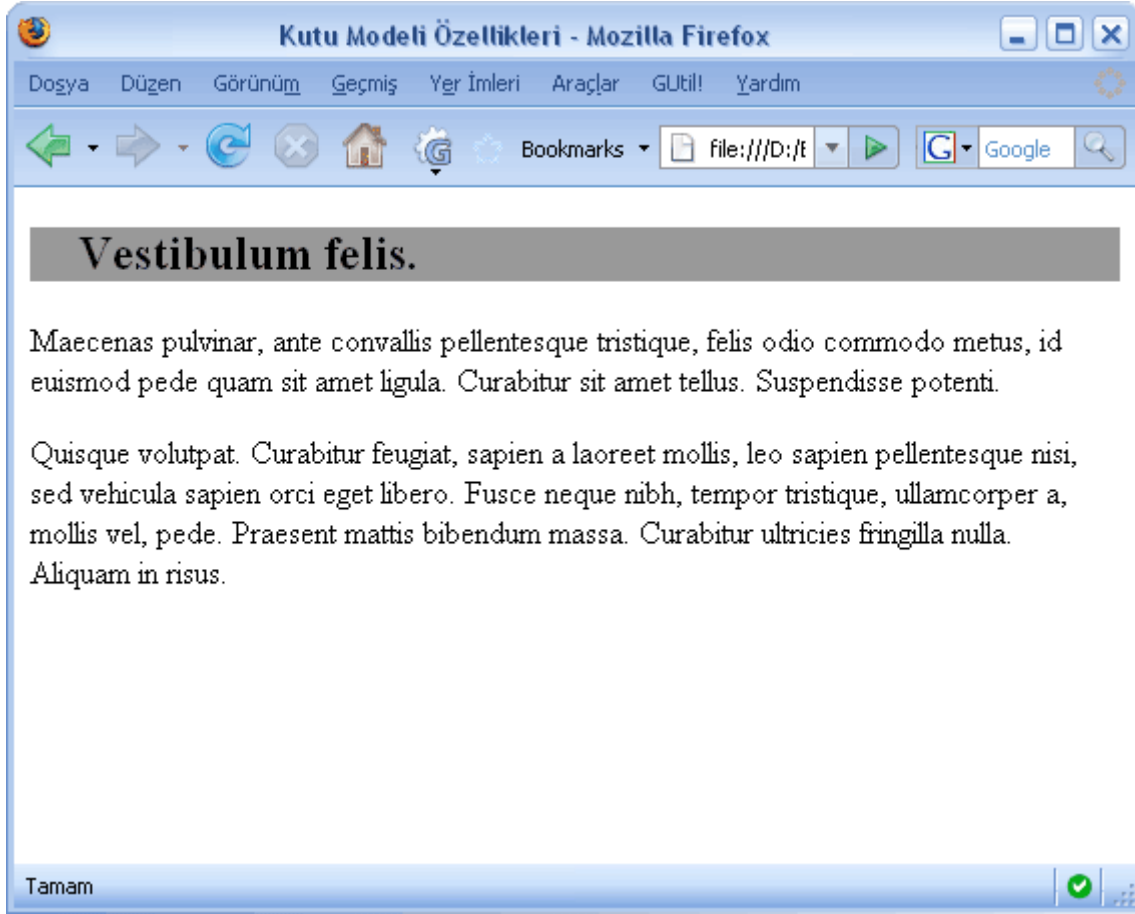
Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

padding-left özelliği içerik alanı ile sol kenar çizgisi arasındaki mesafeyi belirler.

Örnek 9-16

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
h2 {
padding-left: 0.25in;
background-color: #999;
}
</style>
</head>
<body>
<h2> Vestibulum felis.</h2>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus.</p>
</body>
</html>
```



Internet Explorer 5.5 ve alt sürümlerinde (Internet Explorer 6 ve 7 garip mod dâhil) kenar içi boşluğu (padding) değerlerini içerik alanı genişliğinden sayar. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=13> adresini inceleyiniz.

Tarayıcı Uyumu:

Internet Explorer 4+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

9.4.5. Kenar içi boşluğu (padding)

Yapısı: **padding: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde>{1,4} | inherit**

Başlangıç değeri: **0**

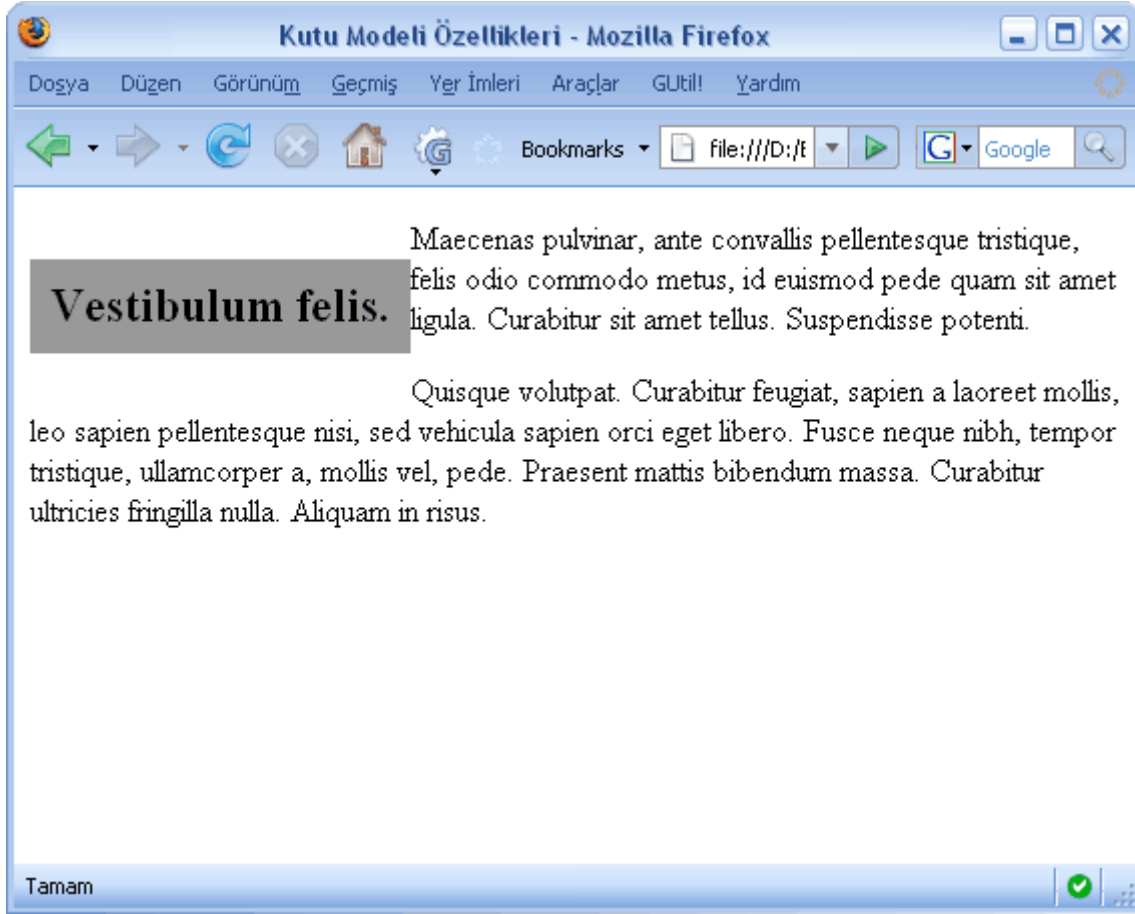
Uygulanabilen elemanlar: **Blok seviyeli ve yerleştirilen elemanlar**

Kalıtımsallık: **Yok**

padding özelliği padding-top, padding-right, padding-bottom ve padding-left değerlerini kısa yoldan atamak için kullanılır.

Örnek 9-17

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
h2 {
padding: 10px;
background-color: #999;
float:left;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
</body>
</html>
```



Kenar dışı boşluğu(margin)'ndaki gibi farklı kullanım şekilleri mevcuttur.

Internet Explorer 5.5 ve alt sürümlerinde(Internet Explorer 6 ve 7 garip mod dâhil) kenar içi boşluğu(padding) değerlerini içerik alanı genişliğinden sayar. Ayrıntılı bilgi için <http://www.fatihhayrioglu.com/?p=13> adresini inceleyiniz.

Internet Explorer 6 ve alt sürümlerinde satırıçi elemanların dikey yükseklik değerini tam olarak göstermesi için position:relative değeri eklenir.

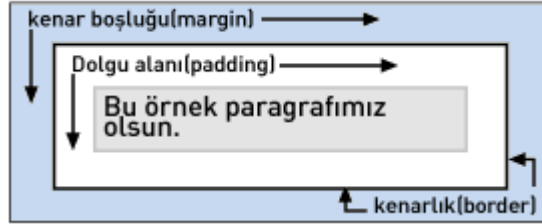
Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

9.5. Boyut Tanımlama Özellikleri

Kutu modeli özelliklerine devam ediyoruz. Sırasıyla aşağıdaki özellikleri inceleyeceğiz:

- Genişlik (Width)
- Yükseklik (Height)
- Asgari genişlik ve yükseklikler (min-width, min-height)
- Azami genişlik ve yükseklik (max-width, max-height)



9.5.1. Genişlik (width)

Yapısı: **width: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | auto | inherit**

Başlangıç değeri: **auto**

Uygulanabilen elemanlar: **Blok seviyeli ve yerleştirilen elemanlar**

Kalıtımsallık: **Yok**

Blok seviyeli ve yerleştirilen elemanların(örn: img, input, textarea vd.) tümü bir genişlik(width) değeri alır. Elemanların başlangıçtaki genişlik değeri **auto** yani kendi asıl genişliğidir. (Örneğin bir resmin genişliği gibi) Yüzde değeri ebeveyn elemanı kıstas olarak uygulanır. Negatif değeri almaz.

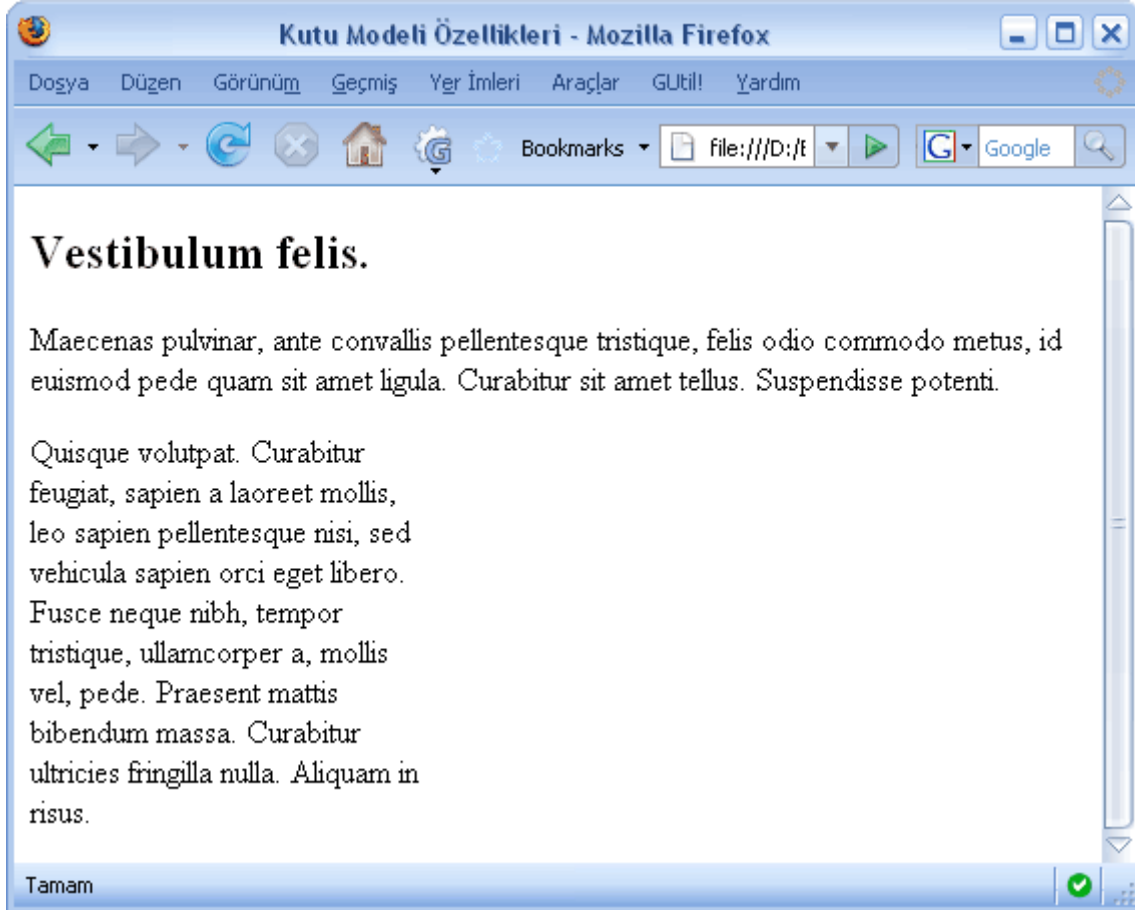
Örnek 9-18

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
p.diger {
width: 200px;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
```

```

commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
</body>
</html>

```



Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

9.5.2. Yükseklik (height)

Yapısı: **height: <değer>**

Aldığı Değerler: **<uzunluk değeri > | auto | inherit**

Başlangıç değeri: **auto**

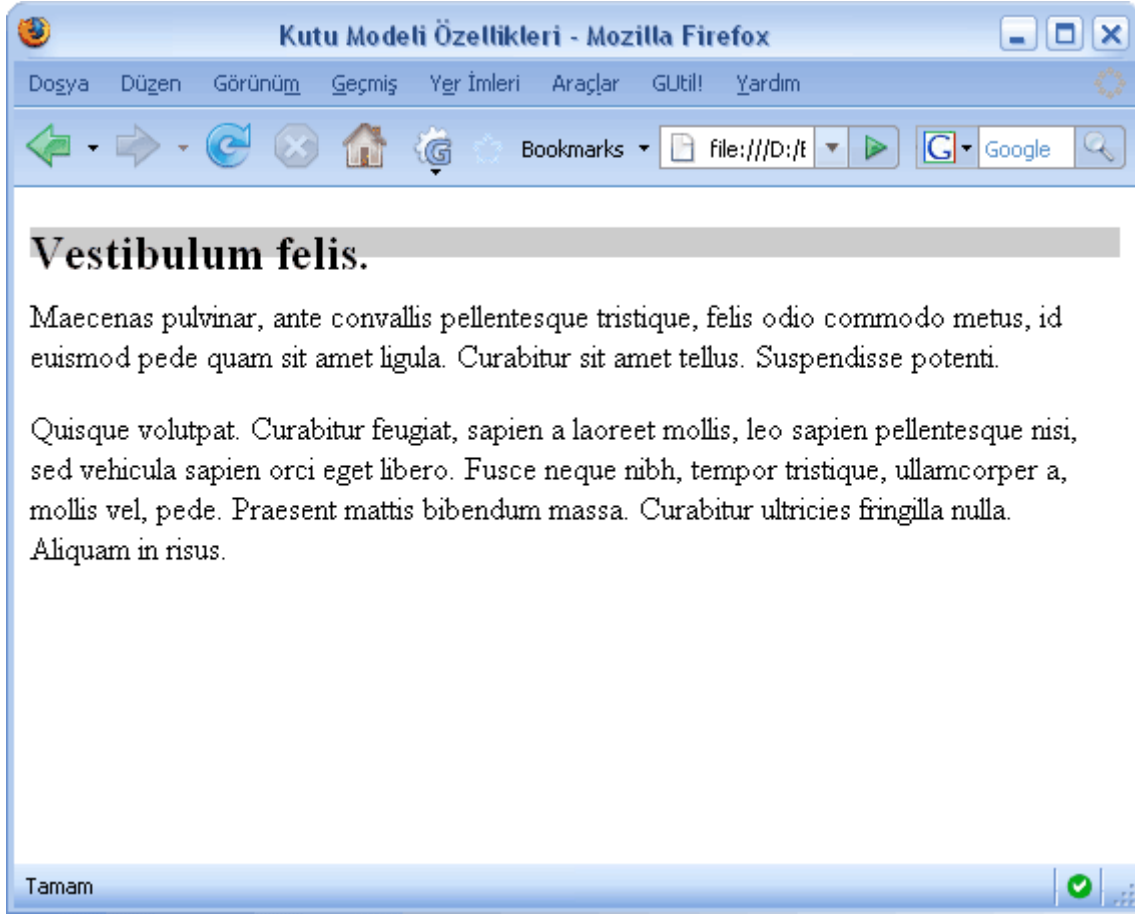
Uygulanabilen elemanlar: **Blok seviyeli ve yerleştirilen elemanlar**

Kalıtımsallık: **Yok**

Blok seviyeli ve yerleştirilen elemanların(örn: img, input, textarea vd.) tümü bir yükseklik(height) değeri alır. Elemanların başlangıçtaki yükseklik değeri **auto** yani kendi asıl yüksekliğidir. (Örneğin bir resmin yüksekliği gibi) Negatif değeri almaz.

Örnek 9-19

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
h2 {
    height: 15px;
    background-color: #CCCCCC;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 4+
 Firefox 1+
 Safari 1.3+
 Opera 3.6+
 CSS 1+

Bir kapsayıcı kutunun genişlik değerini azami ve asgari değerleri ile sınırlandırabiliriz. Bu özellikler CSS2 ile birlikte gelmiştir.

9.5.3. Asgari genişlik ve yükseklikler (min-width, min-height)

Yapısı: **min-width, min-height: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | inherit**

Başlangıç değeri: **0**

Uygulanabilen elemanlar: **yerleştirilmeyen satırıçi elemanlar ve tablo elemanlara**

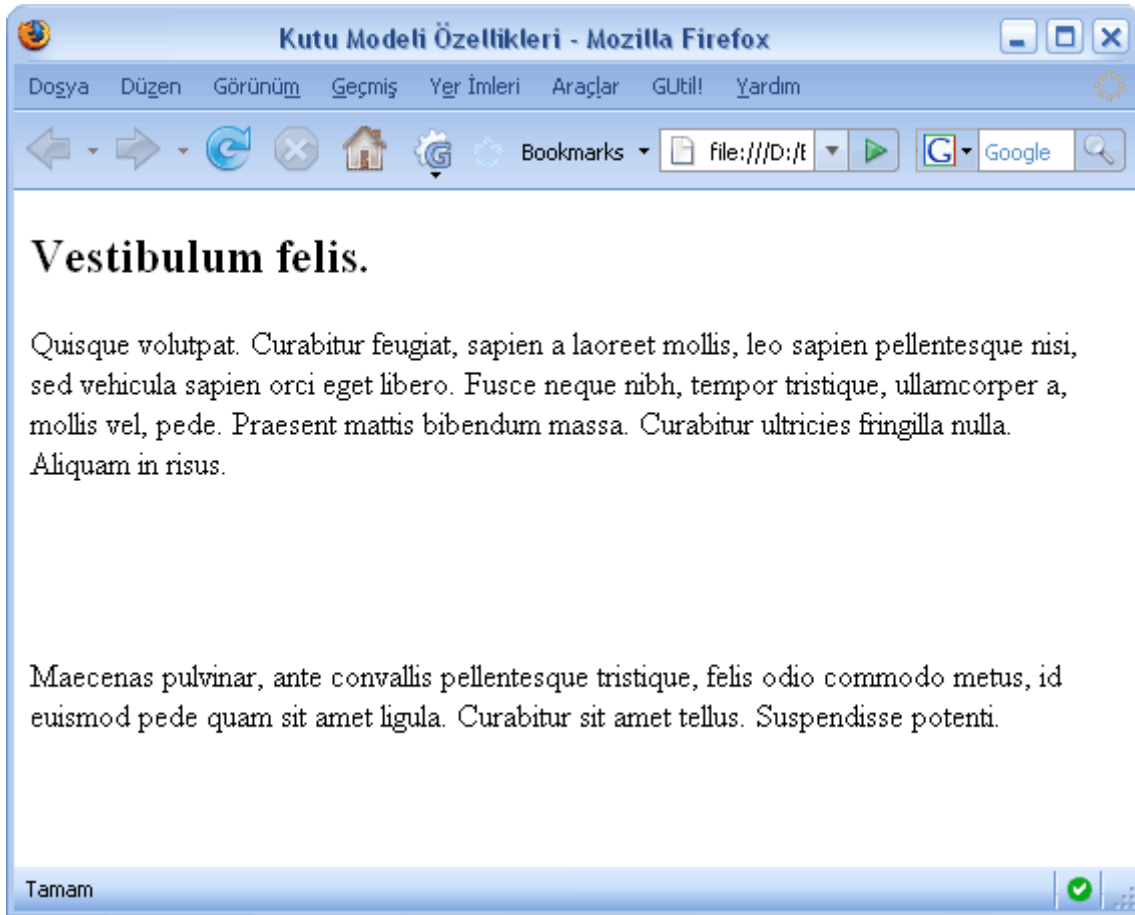
Kalıtıllık: **Yok**

min-width, min-height kapsayıcı kutunun asgari alacağı değerleri belirtmek için kullanılır.

Örnek 9-20

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
p.diger {
    min-height:150px;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 6(**desteklemiyor**), 7+
 Firefox 1+
 Safari 1.3+(kısmen), 2+
 Opera 6+
 CSS 2+

Aynı şekilde max-width ve max-height tanımlamalarında yapılır.

9.5.4. Azami genişlik ve yükseklik (max-width, max-height)

Yapısı: **max-width, max-height: <değer>**

Aldığı Değerler: **<uzunluk değeri > | <yüzde> | none | inherit**

Başlangıç değeri: **0**

Uygulanabilen elemanlar: **yerleştirilmeyen satırıçi elemanlar ve tablo elemanlara**

Kalıtıllık: **Yok**

max-width, max-height kapsayıcı kutunun azami alacağı değerleri belirtmek için kullanılır.

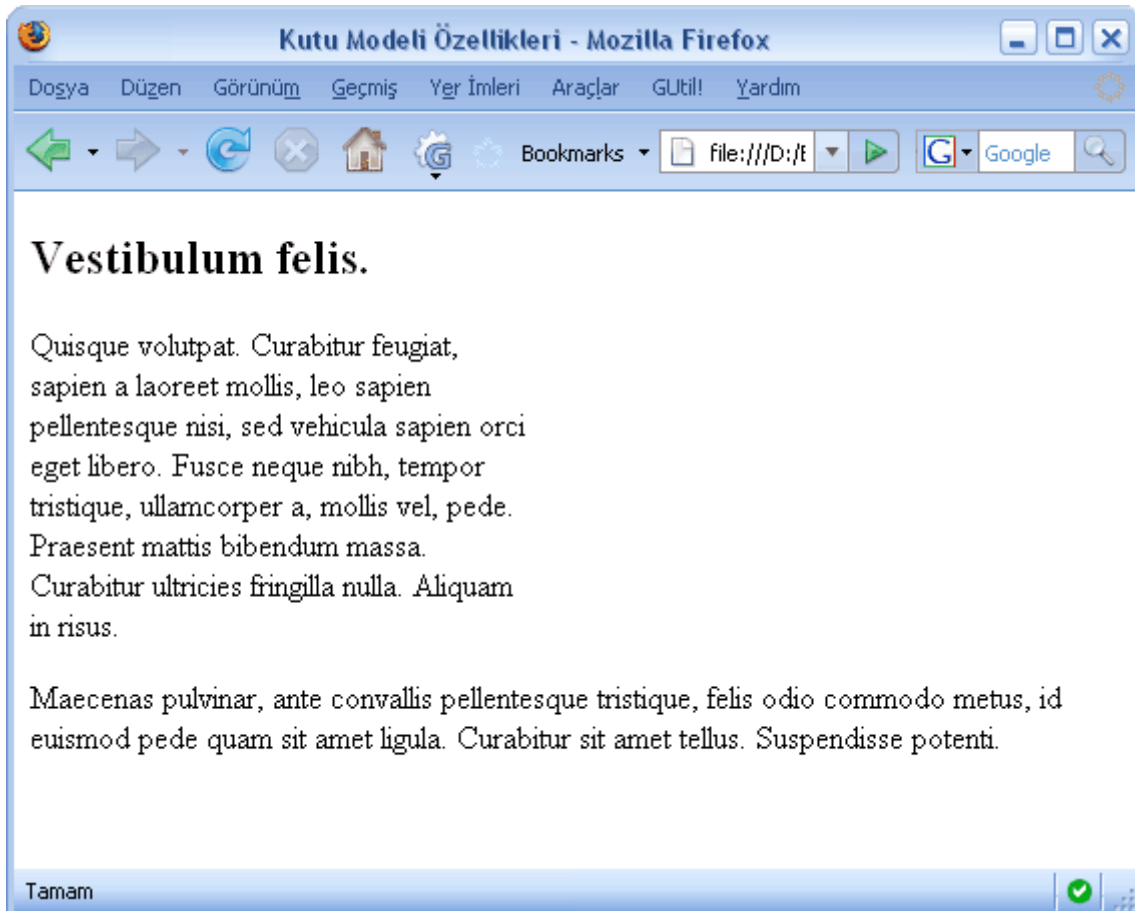
Örnek 9-21

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

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
p.diger {
    max-width: 250px;
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<p class="diger">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </p>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio commodo metus, id
eismod pede quam sit amet ligula. Curabitur sit amet tellus. Suspendisse potenti.</p>
</body>
</html>

```



Tarayıcı Uyumu:

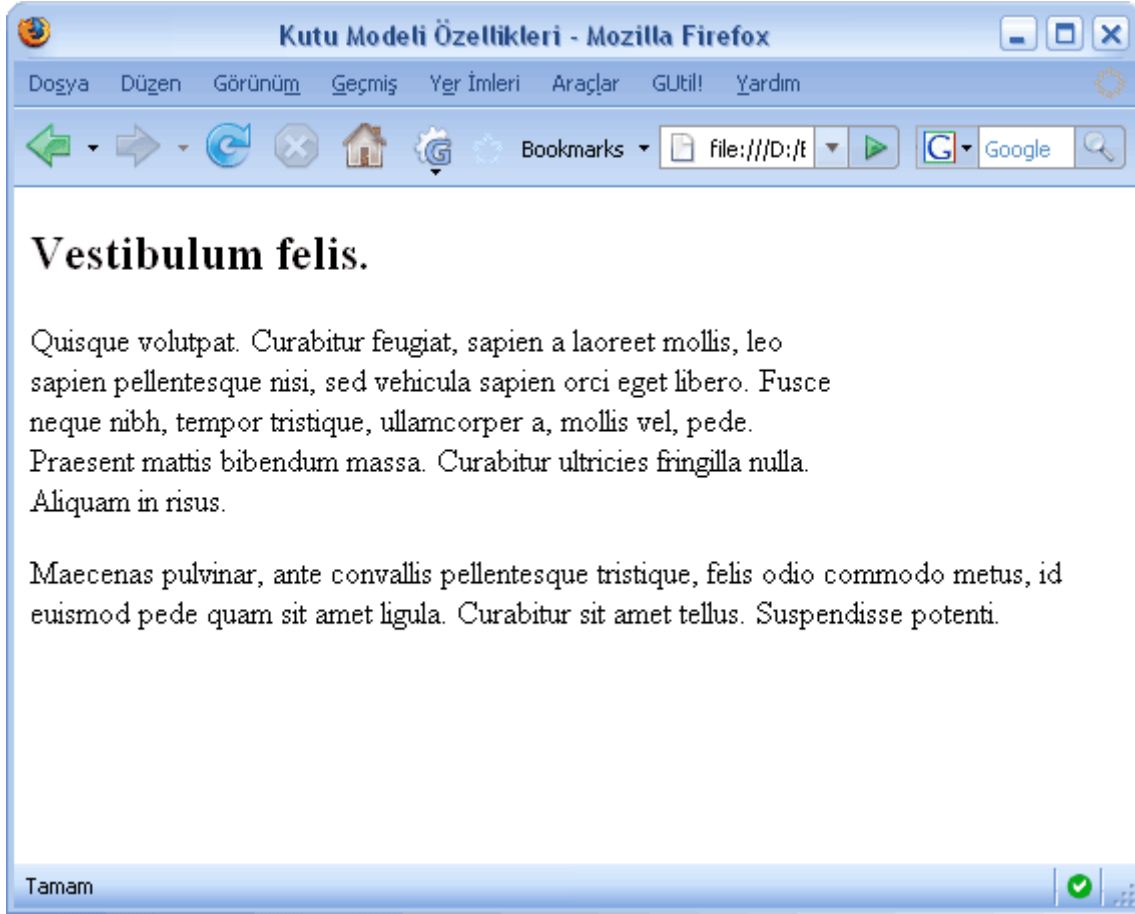
Internet Explorer 6(**desteklemiyor**), 7+
 Firefox 1+
 Safari 1.3+
 Opera 6+
 CSS 2+

Şöyle bir sonumuz vardır ki oda Internet Explorer(IE)'in 7. sürümünden sonra bu özellikleri desteklemeye başlamasıdır. Bu çok kullanışlı özellikleri bu nedenden dolayı kullanmamazlık yapmayacağız tabi.

IE için bir çözüm vardır. **expression()** özelliği bu özellik sadece IE tarafından yorumlanır. Bu özellik dinamik özellik olarak tanımlanır. Yani bu kod normalde bir javascript kodudur.

Örnek 9-22

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style type="text/css">
#icerik{
  min-width: 200px;
  max-width: 400px;
  width: expression(document.body.clientWidth < 200?
"200px" : document.body.clientWidth > 400? "400px" : "auto");
}
</style>
</head>
<body>
<h2>Vestibulum felis.</h2>
<div id="icerik">Quisque volutpat. Curabitur feugiat, sapien a laoreet
mollis, leo sapien pellentesque nisi, sed vehicula sapien orci eget libero.
Fusce neque nibh, tempor tristique, ullamcorper a, mollis vel, pede.
Praesent mattis bibendum massa. Curabitur ultricies fringilla nulla.
Aliquam in risus. </div>
<p>Maecenas pulvinar, ante convallis pellentesque tristique, felis odio
commodo metus, id euismod pede quam sit amet ligula. Curabitur sit
amet tellus. Suspendisse potenti.</p>
</body>
</html>
```



Yukarıdaki tanımlamada css içerisinde javascript komutları kullanılmıştır. Bu tanımlamaları sadece IE anlayacaktır ve uygulayacaktır. Kodda eğer genişlik belirtilen maksimum değerden büyük ise atanan değeri alacak değilse auto değeri alacaktır.

Ayrıca min-height için daha kısa bir çözüm vardır.

```
#menu {
min-height:500px;
height:auto !important;
height:500px;
}
```

9.6. Görsel Sonuç Özellikleri

Kutu elemanların görünümüne çeşitli etkiler yapan özellikleri göreceğiz. CSS2 ile birlikte gelen bu özellikler aşağıda sıralanmıştır.

- **overflow** - taşma
- **clip** - kırpma
- **visibility** - görünürlük

9.6.1. Taşma (Overflow)

Yapısı: **overflow: <deger>**

Aldığı Değerler: **visible | hidden | scroll | auto | inherit**

Başlangıç değeri: **visible**

Uygulanabilen elemanlar: **Blok seviyesinden elemanlar ve yerleştirilen elemanlara**

Kalıtımsallık: **Yok**

Türkçeye taşmak(overflow) olarak çevirdiğimiz bu özellik sınırlarını belirlediğimiz kutu elemanının içeriğinin sınırları aşması durumunda nasıl davranacağını bildiren bir özelliktir. Özelliğe atanan değerleri tek tek inceleyelim

- inherit - kalıtımsallık üst elemanın değerini alabilir
- visible - kutu sınırları dışına taşan alanları göster
- hidden - kutu sınırları dışına taşan alanları gizle.
- scroll - kutu sınırlarını taşan kısmı kaydırma çubuğu ile ulaşmamızı sağlar.
- auto - kutu sınırlarını aşması durumunda scroll ile aynı etkiyi yapar.

Tarayıcı Uyumu:

Internet Explorer 6+

Firefox 1+

Safari 1.3+

Opera 7.5+

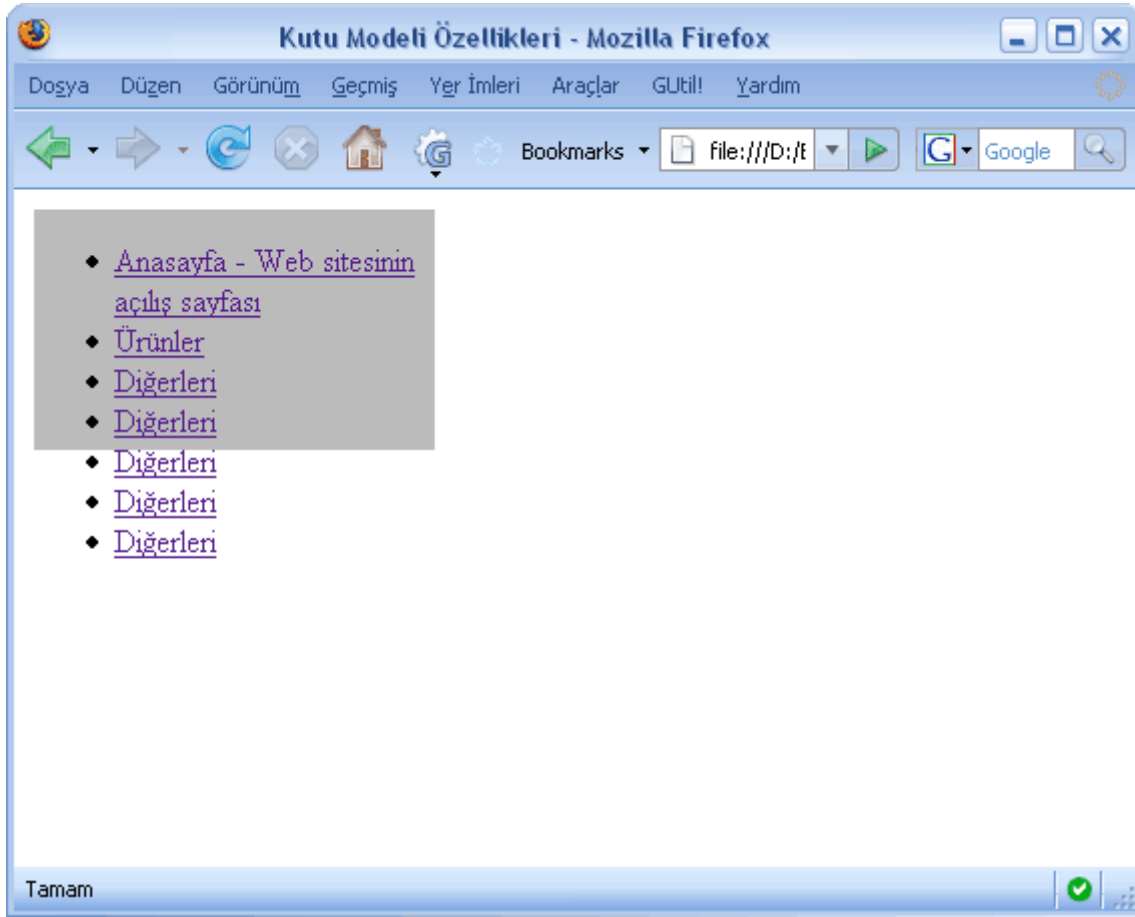
Safari 1.2

CSS 2+

overflow:visible Overflow'ın başlangıç değeri visible'dır. Bu değer taşma olan kısımların gösterilmesini sağlar.

Örnek 9-23

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style>
div#solMenu {
    position: absolute;
    top: 10px;
    left: 10px;
    width: 200px;
    height: 120px;
    background: #bbbbbb;
    overflow: visible;
}
</style>
</head>
<body>
<div id="solMenu">
<ul>
<li><a href="">Ana sayfa - Web sitesinin açılış sayfası</a></li>
<li><a href="">Ürünler</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
</ul>
</div>
</body>
</html>
```



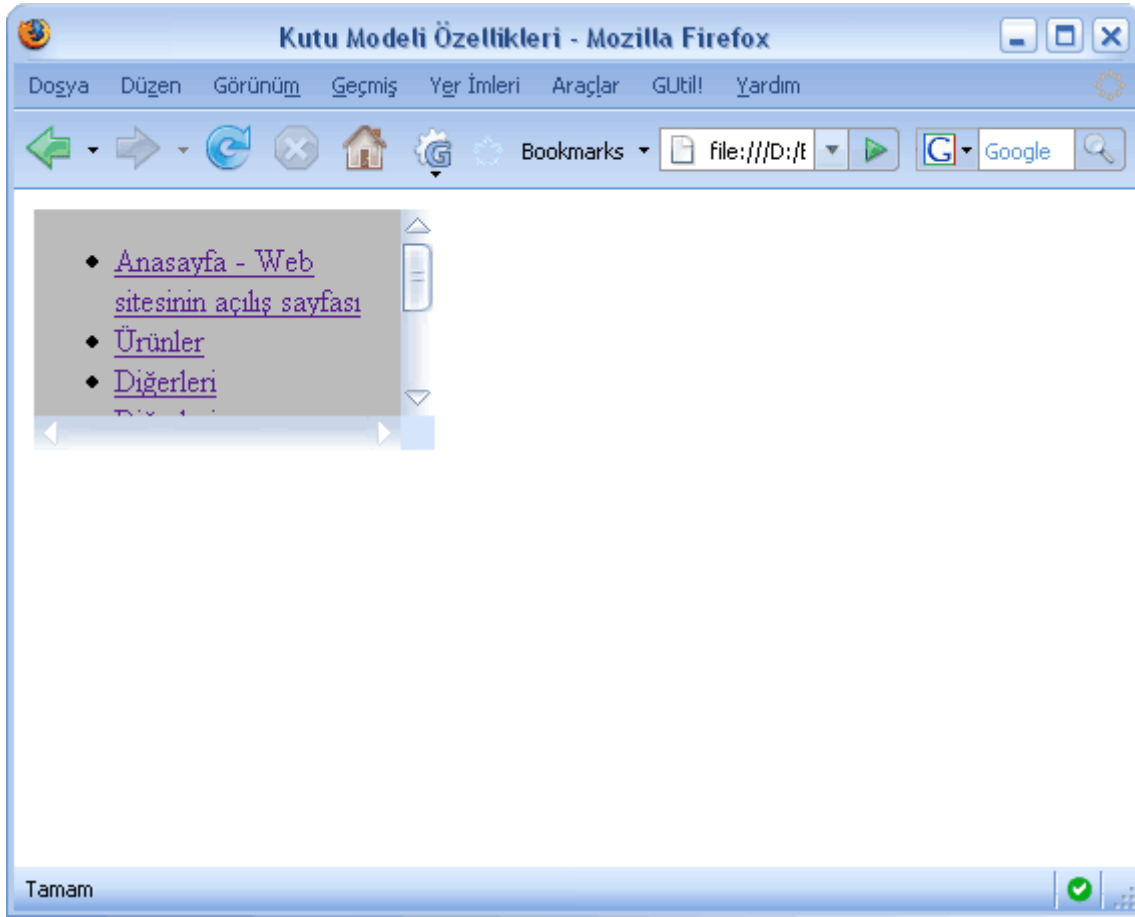
overflow özelliğinin değeri visible olduğu için içerik taşmasına rağmen taşan bölümler gösterilmiştir.

overflow: scroll eğer içeriğin taşmasını engellemek istiyorsak ama yine de taşan kısımlara ulaşılması için kaydırma çubuğu çıkması için kaydırma çubuğu(**scroll**) özelliği ekleriz.

Örnek 9-24

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style>
div#solMenu {
    position: absolute;
    top: 10px;
    left: 10px;
```

```
width: 200px;
height: 120px;
background: #bbbbbb;
overflow: scroll;
}
</style>
</head>
<body>
<div id="solMenu">
<ul>
<li><a href="">Ana sayfa - Web sitesinin açılış sayfası</a></li>
<li><a href="">Ürünler</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
</ul>
</div>
</body>
</html>
```

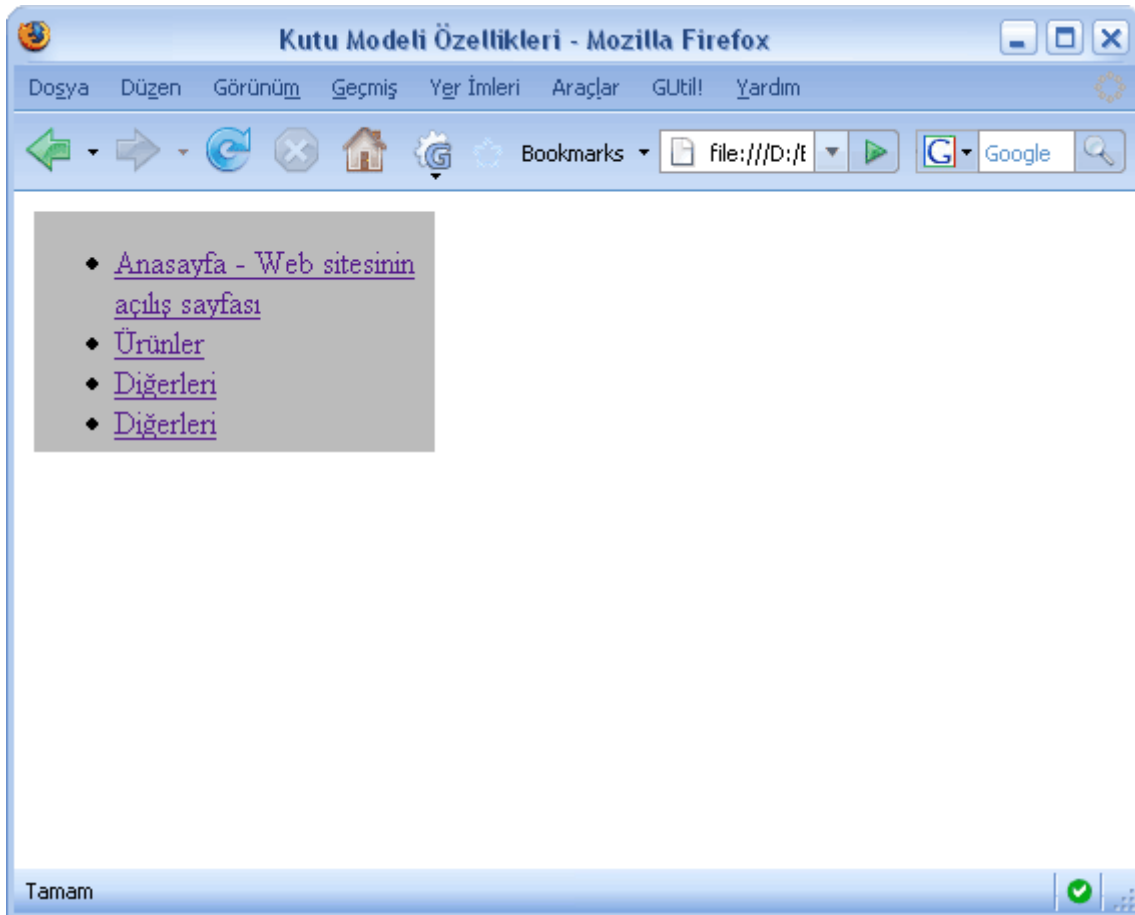


overflow: hidden eğer `overflow:hidden` değeri atarsak. Sınırladığımız kutunun dışına taşan kısımlar gösterilmeyecektir.

Örnek 9-25

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style>
div#solMenu {
    position: absolute;
    top: 10px;
    left: 10px;
    width: 200px;
    height: 120px;
    background: #bbb;
    overflow: hidden;
}
```

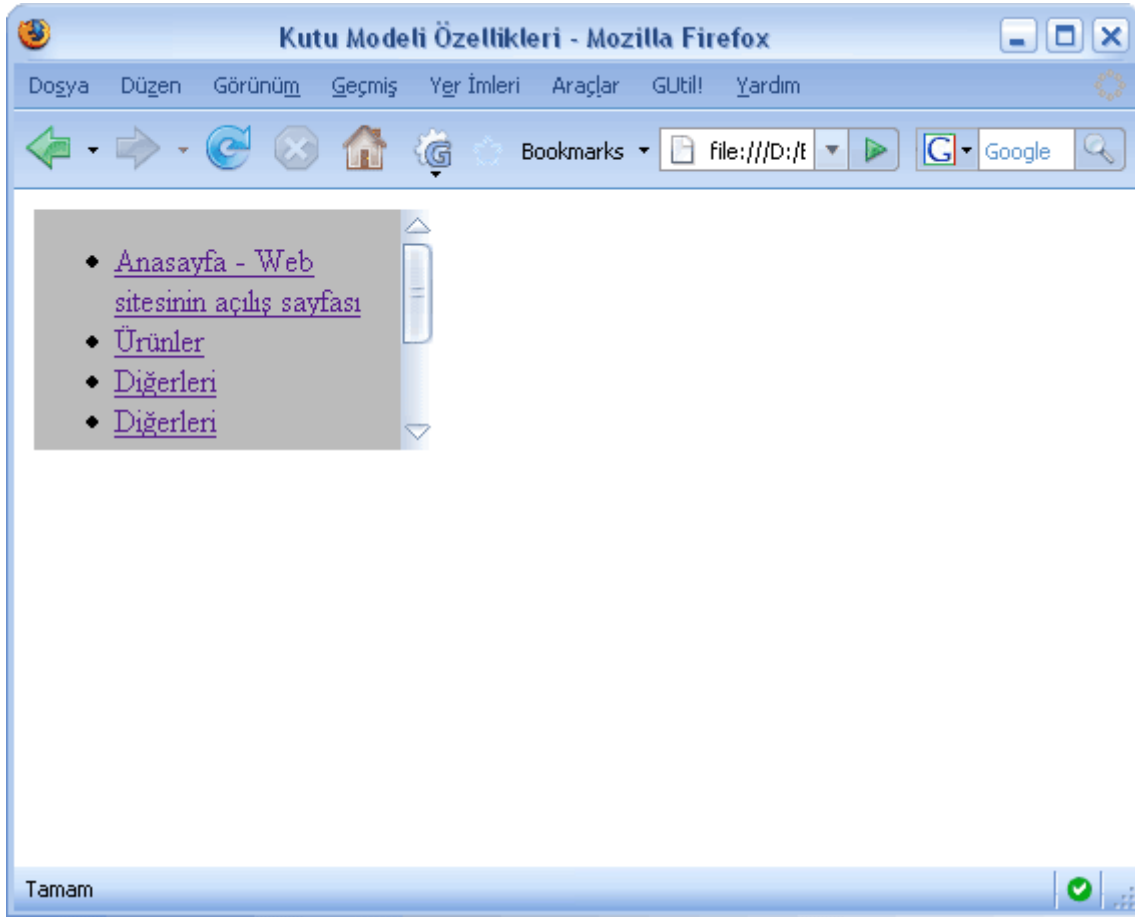
```
}  
</style>  
</head>  
<body>  
<div id="solMenu">  
<ul>  
<li><a href="">Ana sayfa - Web sitesinin açılış sayfası</a></li>  
<li><a href="">Ürünler</a></li>  
<li><a href="">Diğerleri</a></li>  
<li><a href="">Diğerleri</a></li>  
<li><a href="">Diğerleri</a></li>  
<li><a href="">Diğerleri</a></li>  
<li><a href="">Diğerleri</a></li>  
</ul>  
</div>  
</body>  
</html>
```



overflow: auto Son olarakta otomatik değeri vardır. Bu değeri atayınca kutu içeriğine göre otomatik işlem yapacaktır. Eğer taşma olmuyorsa kaydırma çubuğu çıkmayacaktır. Ama eğer içerik taşıyorsa kaydırma çubuğu çıkacaktır.

Örnek 9-26

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style>
div#solMenu {
    position: absolute;
    top: 10px;
    left: 10px;
    width: 200px;
    height: 120px;
    background: #bbbbbb;
    overflow:auto;
}
</style>
</head>
<body>
<div id="solMenu">
<ul>
<li><a href="">Ana sayfa - Web sitesinin açılış sayfası</a></li>
<li><a href="">Ürünler</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
<li><a href="">Diğerleri</a></li>
</ul>
</div>
</body>
</html>
```



Dikkat ederseniz overflow: scroll'dan farklı olarak sadece dikey kaydırma çubuğu çıkmıştır. Yatayda bir taşma olmadığı için yatay kaydırma çubuğu çıkmamıştır.

Safari'nin eski sürümlerinde sorun çıkarıyor Javascript ile bir çözüm var. <http://blog.deconcept.com/2005/03/25/safari-and-links-to-elements-in-overflow-auto-content>

Operadaki sorun için ise sadece min-height ve min-width özelliklerini kullanmak yeterli. <http://nontroppo.org/test/Op7/overflowbug.html>

Bunların dışında Internet Explorer'un tescillediği ve CSS3 standartlarında yer alan iki adet özellik daha vardır. overflow-x ve overflow-y

overflow-x ve overflow-y: Internet Explorer5x ve üzeri sürümlerden itibaren desteklediği bu özellikler aslında overflow gibi çalışır. Buradaki x ve y'dende anlaşıldığı gibi overflow özelliğini yatay ve dikey olarak ikiye ayıran özelliklerdir. overflow-x özelliği yatayda taşma olması durumunda

taşan kısımlara uygulanacak özellikleri(visible, scroll, hidden ve auto) atmamızı sağlar. overflow-y'de dikeyde aynı şeyi yapar. Firefox'unda 1.1 sürümünden itibaren bu özellikleri desteklemesi kullanılabilirliğini arttırmıştır. Tabi biraz daha beklemekte yarar var.

Tarayıcı Uyumu:

Internet Explorer 5+
Firefox 1+
Opera 7.5+
Safari 3
W3C's CSS Level 3+
CSS Profile 3.0

9.6.2. Kırpma (clip)

Yapısı: **clip: rect(<üst>,<sağ>,<alt>,<sol>)**

Aldığı Değerler: **rect(top, right, bottom, left) | auto | inherit**

Başlangıç değeri: **auto**

Uygulanabilen elemanlar: **mutlak konumlandırma uygulanmış elemanlar (CSS2'den itibaren Blok seviyesinden elemanlar ve yerleştirilen elemanlara)**

Kalıtıllık: **Yok**

Tarayıcı Uyumu:

Internet Explorer 5.5+(kısmen)
Firefox 1+
Safari 1.3+
Opera 3+
CSS 2+

clip, elemanın içeriğinin belli bir bölümünün gösterilmesi sağlar. Başlangıç değerinin **auto** atanmış olması herhangi bir kırpmanın olmayacağını gösterir.

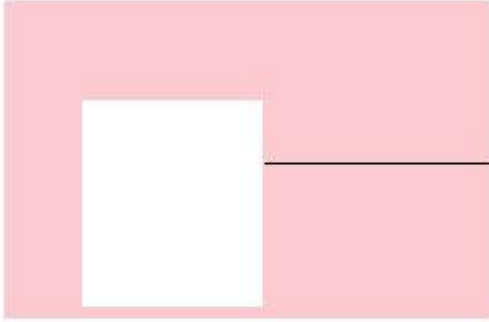
Resim düzenleme araçlarında yaptığımız kırpma işini CSS yardımıyla yapmamızı sağlayan bu özelliği Javascript ile birlikte kullanarak çok güzel uygulamalar yapabiliriz.

Clip'e atanan değerler **em** cinsinden verilerek esnek sonuçlar elde edilebilir. Ayrıca negatif değerlerde alabilir. Negatif değer atandığında eleman kutusunun dışına doğru kırpma yapacaktır.

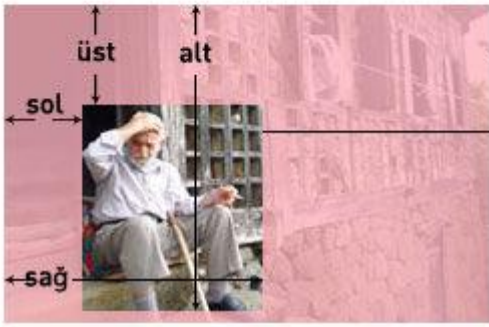
W3C standartları belirlerken konumlandırmalar arasında virgöl konmasını öneriyor. Ancak garip bir şekilde IE virgöl kullanılan örnekleri normal modda farklı yorumlamaktadır. Acayip modda ise doğru yorumlamaktadır. Ne yapalım bizde tüm tarayıcılara uygun olması için virgöl kullanamayacağız



Ahmet emicenin resmi



Uygulanan kırpma alanı



Element

Kırpılan Alan

Bir örnek yapalım.

CSS kodu:

Örnek 9-27

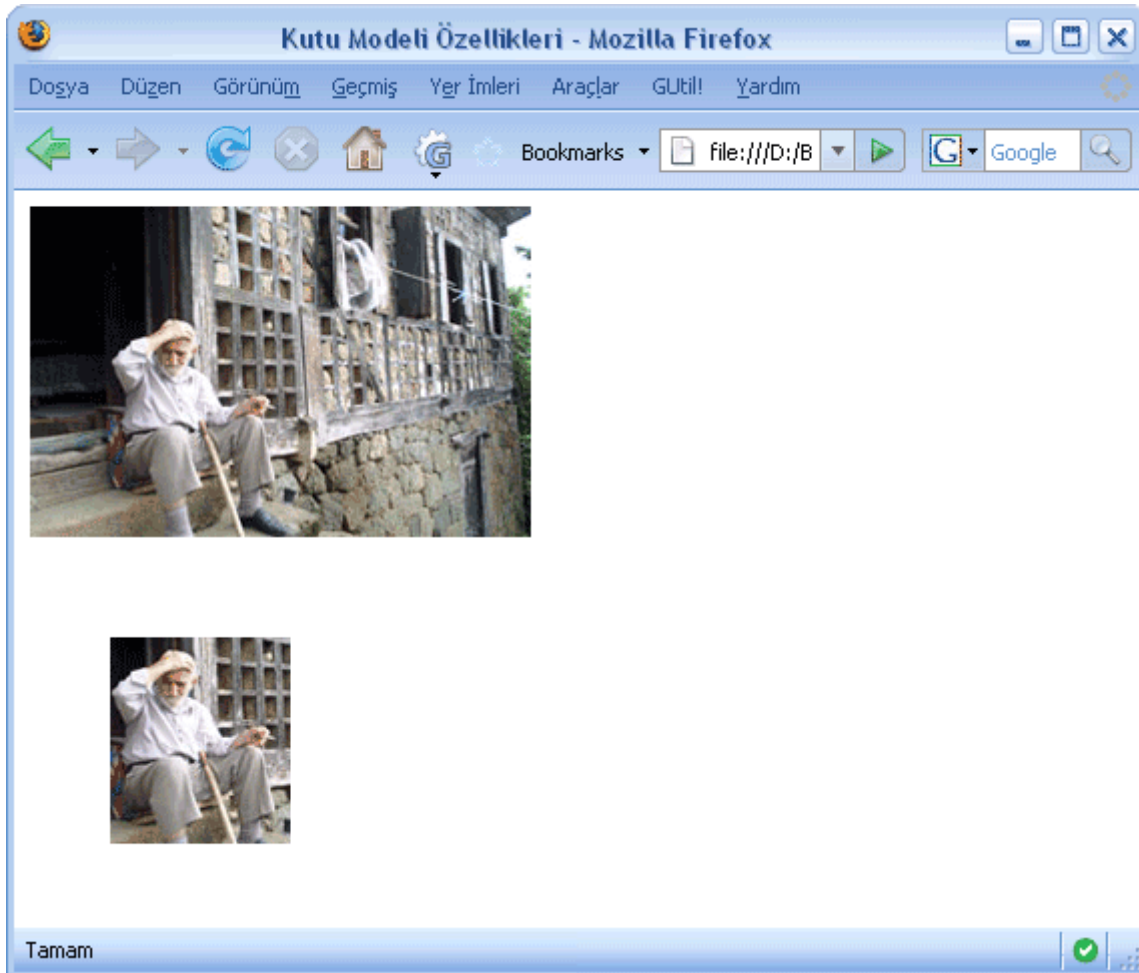
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style>
.kirpma{
  position: absolute;
  clip: rect(50px 130px 153px 40px);
}
```

```

</style>
</head>
<body>
<br />
<div class="kirpma">

</div>
</body>
</html>

```



9.6.3. Görünürlük (visibility)

Yapısı: **visibility: <değer>**

Aldığı Değerler: **visible | hidden | collapse | inherit**

Başlangıç değeri: **visible**

Uygulanabilen elemanlar: **tüm elemanlara**

Kalıtsallık: **Yok**

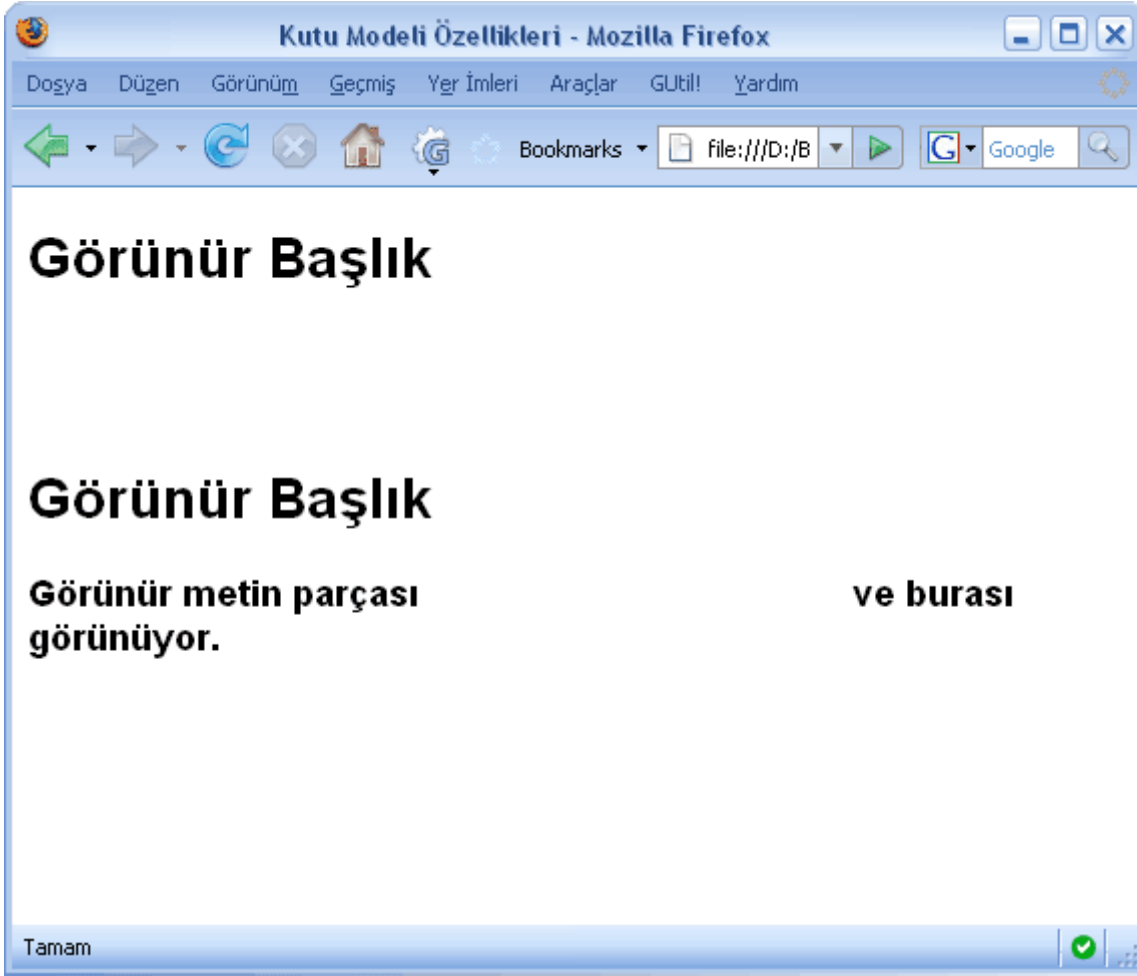
Tarayıcı Uyumu:

Internet Explorer 5.5+(kısmen)
 Firefox 1+
 Safari 1.3+(kısmen)
 Opera 3.6+
 CSS 2+

Görünürlük(visibility) özelliği nesnelerin görünümünü belirler. Değeri **visible** ise eleman görünür. Eğer değeri **hidden** ise eleman görünmezdir. Eğer bir elemanın görünürlük değeri hidden olarak atanmış ise eleman sayfa yapısındaki yeri korur ancak gözle görülmez.

Örnek 9-28

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style>
  body {
    font-family: sans-serif;
    font-weight: bold;
    font-size: 14px;
  }
</style>
</head>
<body>
<h1 style="visibility: visible">Görünür Başlık</h1>
<h1 style="visibility: hidden; background-color: #f00; border:1px
solid #000; margin:20px 0; padding-top:10px;">Görünmez
Başlık</h1>
<h1 style="visibility: visible">Görünür Başlık</h1>
<p style="font-size: large">Görünür metin parçası<b style="visibility:
hidden">görünmez metin parçası</b> ve burası görünüyor.</p>
</body>
</html>
```



Örnekte gördüğümüz gibi eleman görünmediği halde sayfadaki yerini korumaktadır. Ayrıca gizli nesneye atanan kenarlık ve Artalan rengide görünmemektedir. Ancak elemanın konumu etkileyen kenar dışı boşluğu(margin) ve kenar içi boşluğu(padding) değeri görünmese de sayfayı etkilemektedir.

Ata elemanı görünmez olan bir çocuk elemana eğer görünür özelliği atanmış ise ata eleman görünmez ancak çocuk eleman görünür. Bu ters bir durum olarak görünür ancak bu bize avantaj sağladığı durumlarda yok değildir. Örnek yaparsak

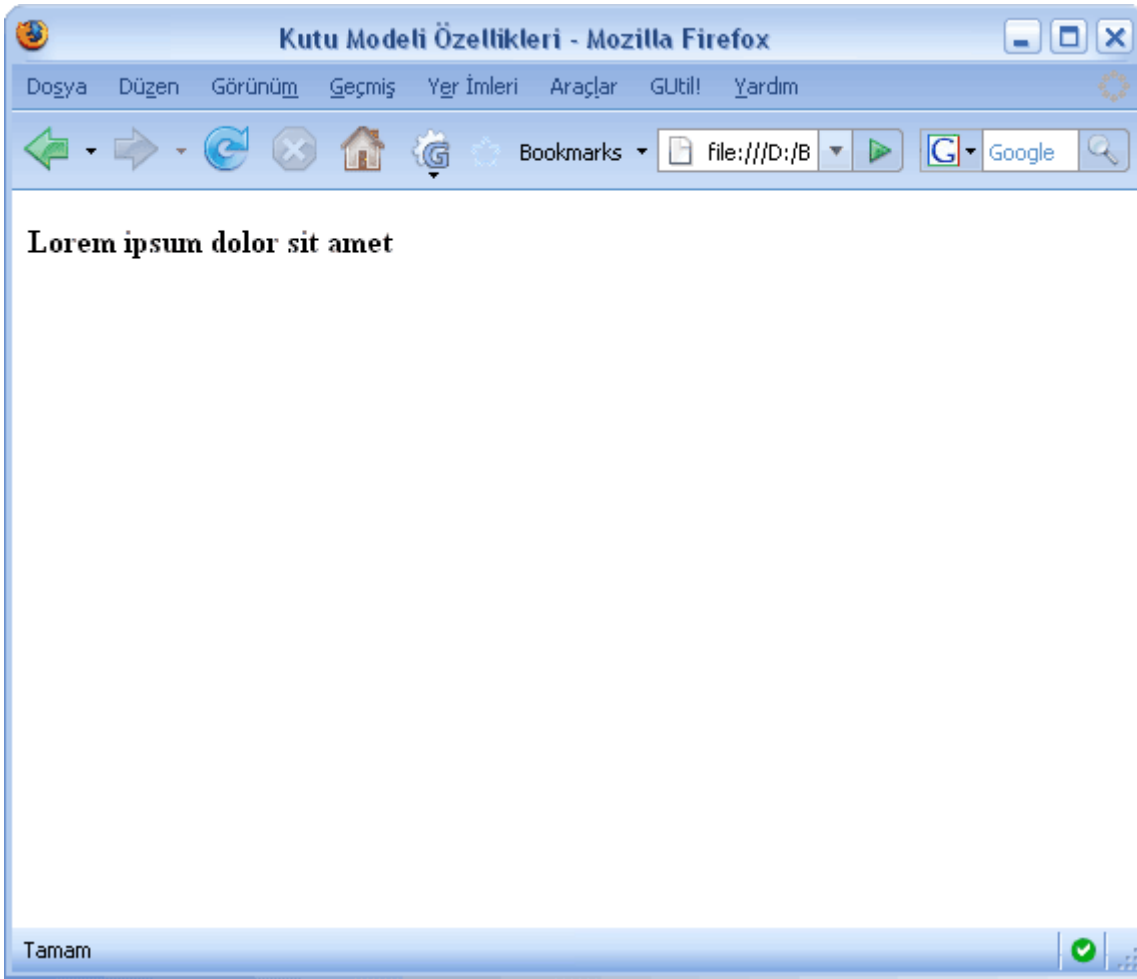
CSS kodu

Örnek 9-29

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

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Kutu Modeli Özellikleri</title>
<style>
p.gorunmez {
    visibility: hidden;
}

p.gorunmez strong {
    visibility: visible;
}
</style>
</head>
<body>
<p class="gorunmez"> <strong>Lorem ipsum dolor sit amet</strong>,
consectetur adipiscing elit. Cras sem. In sagittis mauris eu turpis. Mauris
ut lorem nec pede consequat suscipit. </p>
</body>
</html>
```



visibility: collapse özelliği vardır. collapse tablo elemanlar için hidden etkisi yapar. Ancak burada bir fark var gizlenen elemanlar yerini korumaz. Ayrıca bu özelliği tüm tarayıcılar desteklemiyor, bunun yerine display:none özelliği kullanılır.

10. Yüzen kutu ve Konumlandırma Özellikleri

10.1. Yüzen kutu özellikleri

CSS ile web sitesi kodlamanın en önemli özelliklerinden iki tanesidir float ve clear. Float bir elemanı Normal Akışın dışına çıkarmak için kullanılan bir yöntemdir. Bir biri ardına sıralanmış elemanlara float uygulandığında elemanlar eğer yeterli alan varsa yan yana dizilirler. Float Dinamik Göreceli Konumlandırma gibidir ve bu amaçla kullanılabilir.

Clear özelliği ile float uygulanmış elemandan sonraki elemanlara float etkisini durdurmamız mümkündür. Örneğin iki paragrafınız olsun ve bunlardan sadece ilkini float uygulanmış elemanın yanına koymak istiyorsunuz, bu durumda ikinci paragrafa clear özelliği atadığınızda bu eleman float uygulanmış elemanın altında konumlanacaktır.

10.1.1. float özelliği

Yapısı: **float: <değer>**

Aldığı Değerler: **left | right | none | inherit**

Başlangıç değeri: **none**

Uygulanabilen elemanlar: **tüm elemanlar**

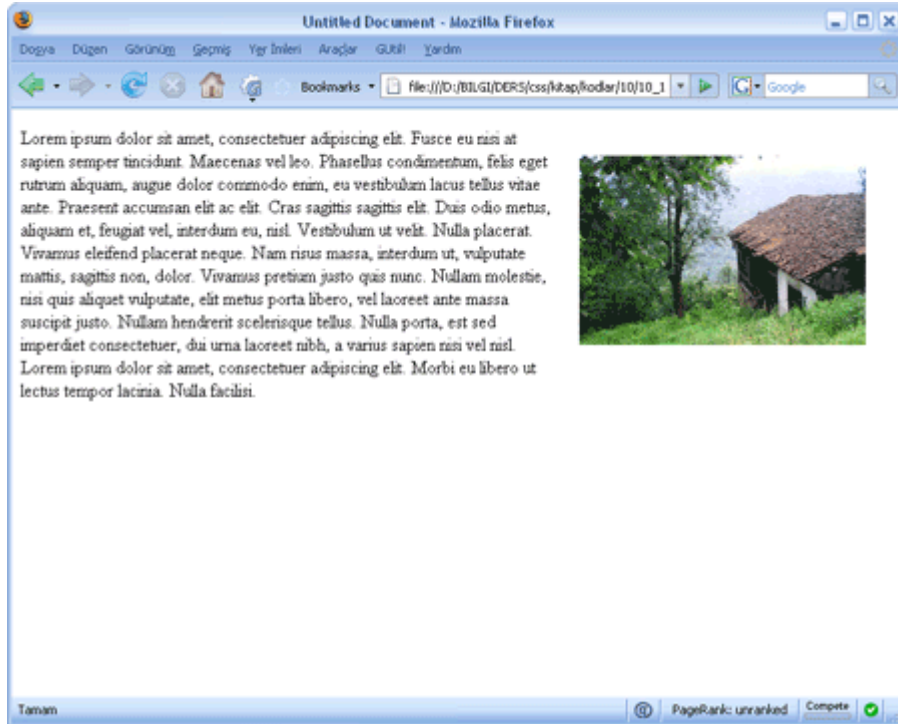
Kalıtımsallık: **Yok**

float özelliği bir elemanı(img, table, div vd.) konumunu belirlemek(sağa veya sola) için kullanılır. Siz bir elemana **float:left** değeri vererseniz diğer elemanlar o elemanın sağından akar. Bu HTML3,2 deki resme(**img**) uygulanan **align="left"** ve **align="right"** ile özdeş bir özelliktir. Ancak CSS 1 tüm elemanlara **float** özelliği vermeyi desteklemiyordu. HTML 3,2 sadece **img** ve **table** için bu özelliği kullanmamıza izin veriyordu. **float** özelliği CSS ile tablosuz web sitesi oluşturma metodunun en önemli öğelerinden biridir.

Örnek 10-1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
```

```
/>
<title>Untitled Document</title>
<style>
p img {
float: right;
margin: 25px;
}
</style>
</head>
<body>
<p>Lorem ipsum dolor sit
amet, consectetur adipiscing elit. Fusce eu nisi at sapien semper
tincidunt. Maecenas vel leo. Phasellus condimentum, felis eget rutrum
aliquam, augue dolor commodo enim, eu vestibulum lacus tellus vitae
ante. Praesent accumsan elit ac elit. Cras sagittis sagittis elit. Duis odio
metus, aliquam et, feugiat vel, interdum eu, nisl. Vestibulum ut velit.
Nulla placerat. Vivamus eleifend placerat neque. Nam risus massa,
interdum ut, vulputate mattis, sagittis non, dolor. Vivamus pretium justo
quis nunc. Nullam molestie, nisi quis aliquet vulputate, elit metus porta
libero, vel laoreet ante massa suscipit justo. Nullam hendrerit scelerisque
tellus. Nulla porta, est sed imperdiet consectetur, dui urna laoreet nibh,
a varius sapien nisi vel nisl. Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Morbi eu libero ut lectus tempor lacinia. Nulla facilisi.</p>
</body>
</html>
```

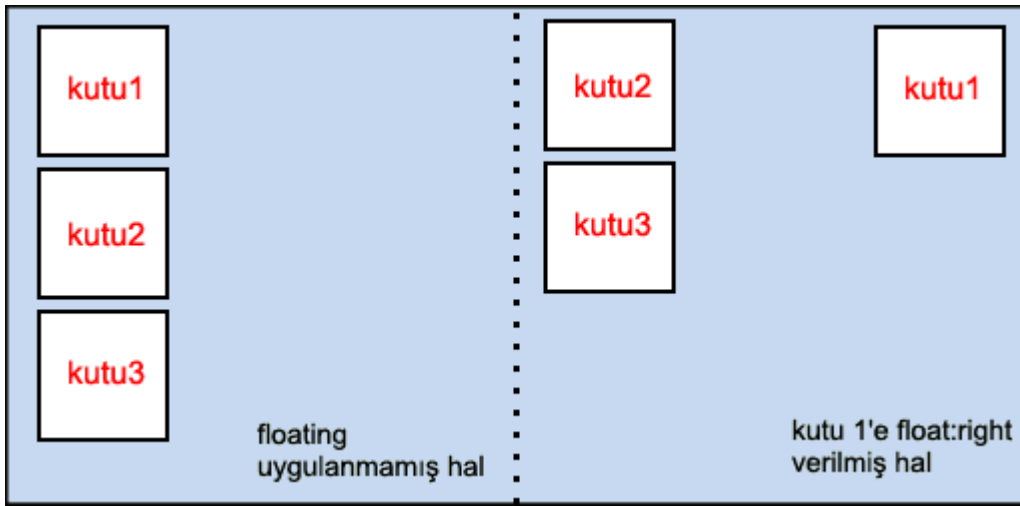


Tarayıcı Uyumu:

Internet Explorer 4+
Firefox 1+
Safari 1.3+
Opera 3.6+
CSS 1+

Bir elemana float uygulandığında, float uygulanan kutu kapsayıcı kutunun veya diğer float kutusunun dış kenarının sağına veya soluna kaydırılabilir. Float uygulanan kutu, dokümanın normal akışının dışına çıkacaktır. Dokümandaki normal kutular float elemanı burada yokmuş gibi normal akışına devam edeceklerdir.

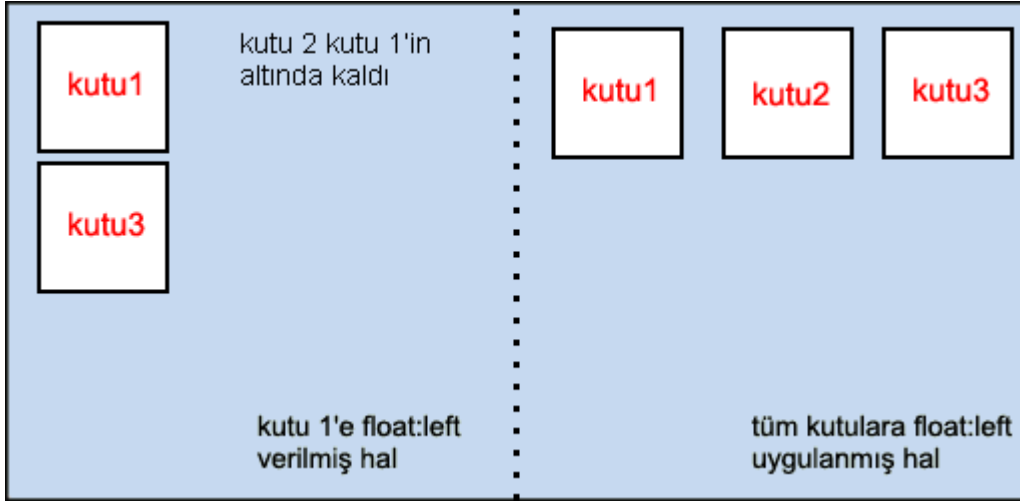
Aşağıdaki şekilde görüldüğü gibi başlangıçta normal akış içinde bulunan 1.kutu **float:right** özelliği ile sağ tarafa kaymış ve normal akıştaki elemanların akışı 1.kutu yokmuş gibi devam etmiştir.



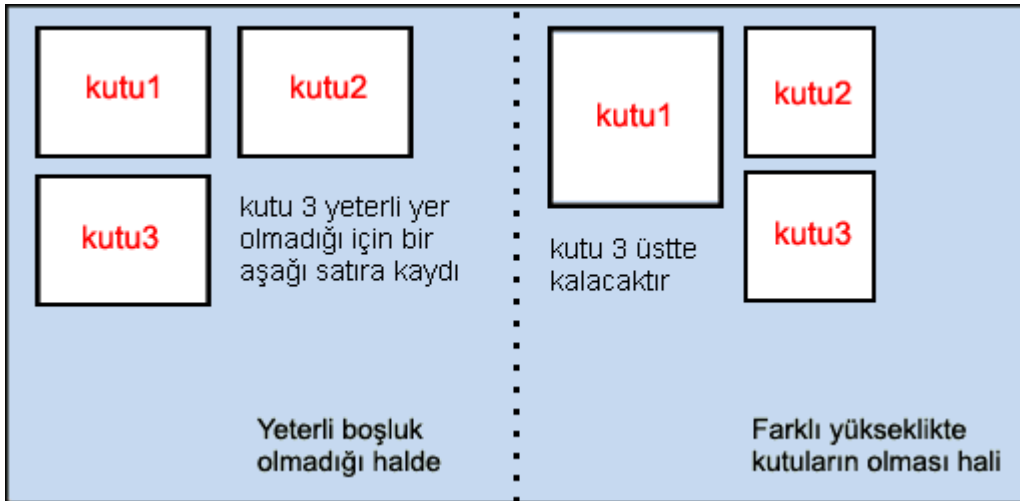
Aşağıdaki şekilde **float:left** tanımlaması yapılarak 1.kutu kapsayıcı kutunun en soluna dayanmıştır. Diğer iki kutu ise normal akışına devam etmektedir.

```
p {float:left; width:200px; margin:0;}  
img {float:left; margin:0 4px 4px 0;}
```

Burada her iki elemanada float uygulanarak metnin resmi sarmasına son verilmiştir. Bu çoklu kolon oluşturma tekniğinin basit bir örneğidir. Float uygulanan her eleman satır boyunca yan yana dizilecektir.



Eğer kapsayıcı kutu genişliği her üçünün sığacağı kadar değil ise sondaki kutu kendiliğinden aşağı kayacaktır. Eğer kutular farklı yükseklikte ise yüksekliği düşük olan kutudan sonra gelen kutu bunun altında devam edecektir. Bu durumu düzeltmek için clear özelliği kullanılır bunu aşağıda göreceğiz.



10.1.2. clear özelliği

Yapısı: **clear: <değer>**

Aldığı Değerler: **none | left | right | both**

Başlangıç değeri: **tanımsız**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Yok**

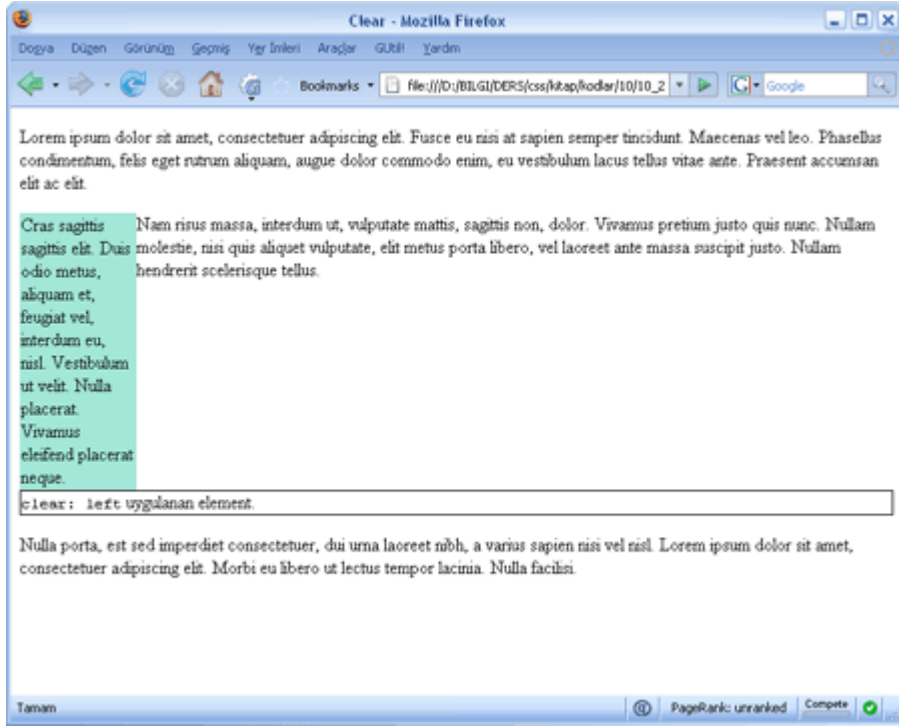
Resim ve metin elemanları diğer elemanlara göre floating eleman olarak tanımlanır. **clear** özelliği **floating** uygulanmayan elemana köşe tanımı yapar. **left** değeri uygulandığı elemanı floating elemanın bir alt soluna atar, **right** özelliği benzer şekilde sağa atar, **none** özelliği elemanın başlangıç değerini geri döndürür ve **both** değerinde iki floating elemanın aşağısına atar. **clear** özelliği CSS ile tablosuz web sitesi oluşturma metodunun en önemli öğelerinden biridir.

Örnek 10-2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Clear</title>
<style type="text/css">
#sol {
width: 100px;
float: left;
background: #A2E7D7;
padding: 1px;
margin:0;
}

#temizle {
clear: left;
border: 1px solid #000;
}

</style>
</head>
<body>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce eu
nisi at sapien semper tincidunt. Maecenas vel leo. Phasellus
condimentum, felis eget rutrum aliquam, augue dolor commodo enim, eu
vestibulum lacus tellus vitae ante. Praesent accumsan elit ac elit. </p>
<p id="sol">Cras sagittis sagittis elit. Duis odio metus, aliquam et,
feugiat vel, interdum eu, nisl. Vestibulum ut velit. Nulla placerat. Vivamus
eleifend placerat neque. </p>
<p>Nam risus massa, interdum ut, vulputate mattis, sagittis non, dolor.
Vivamus pretium justo quis nunc. Nullam molestie, nisi quis aliquet
vulputate, elit metus porta libero, vel laoreet ante massa suscipit justo.
Nullam hendrerit scelerisque tellus. </p>
<p id="temizle"><code>clear: left</code> uygulanan eleman.</p>
<p>Nulla porta, est sed imperdiet consectetur, dui urna laoreet nibh, a
varius sapien nisi vel nisl. Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Morbi eu libero ut lectus tempor lacinia. Nulla facilisi.</p>
</body>
</html>
```



Tarayıcı Uyumu:

Internet Explorer 3+
Firefox 1+
Safari 1.3+
Opera 4+
CSS 1+

10.2. Konumlandırma Özellikleri

CSS ile web sayfası kodlamanın üç önemli konusu **kutu modeli**, **konumlandırma(positioning)** ve **yüzen kutu özelliği** konularıdır.

Burada CSS ile konumlandırma hakkında bilgi vermeye çalışacağız.

Konumlandırma özellikleri HTML elemanlarının sayfa içerisindeki konumu belirlememize yarayan özelliklerdir.

CSS'de üç temel konumlandırma şekli vardır: **normal akış**(position:static ve position:relative), **float** ve **mutlak konumlar**(position:absolute). Tüm elemanlar başlangıçta **normal akış** şeklinde konumlandırılmıştır.

Normal Akış

Eğer sayfalardaki elemanlara herhangi bir konumlandırma yapılmamış ise veya float değeri verilmemiş ise sayfaya normal akış konumlandırması uygulanır. Sayfada ilk eleman en üstte son elemanda en sonda olacak şekilde sıra ile aşağı doğru sıralanacaktır.

Satırıcı kutular yatay olarak tasarlanır. Yatay boşluk kenar dış boşluğu(margin), kenar iç boşluğu(padding) ve kenar çizgisi ile hesaplanır. Bu nedenle dikey kenar dış boşluğu(margin), kenar iç boşluğu(padding) ve kenar çizgisi değerleri etkisizdir.



10.2.1. Konumlandırma özelliği (position)

Yapısı: **position: <değer>**

Aldığı Değerler: **static | relative | absolute | fixed | inherit**

Başlangıç değeri: **static**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtımsallık: **Yok**

Tarayıcı Uyumu:

Internet Explorer 4+(kısmen, IE4 den sonra absolute, relative ve static değerlerini destekliyor. IE7 ile birlikte fixed'i da destekliyor)

Firefox 1+

Safari 1.3+

Opera 5+

CSS 2+

10.2.2. Üst, sağ, alt ve sol (top, right, bottom, left)

Yukarıda tanımladığımız konumlandırma özelliklerinden relative, absolute ve fixed değerini alan elemanlar üst, sağ, alt ve sol özelliklerine bağlı olarak konumlandırılırlar.

Yapısı: **top, right, bottom, left : <değer>**

Aldığı Değerler: **<uzunluk> | <yüzde> | auto | inherit**

Başlangıç değeri: **auto**

Uygulanabilen elemanlar: **postion uygulanan elemanlara**

Kalıtımsallık: **Yok**

Tarayıcı Uyumu:

Internet Explorer 4+(kısmen yüzde ve uzunluk değerlerini , IE6 dan sonra inherit hariç hepsini)

Firefox 1+

Safari 1.3+

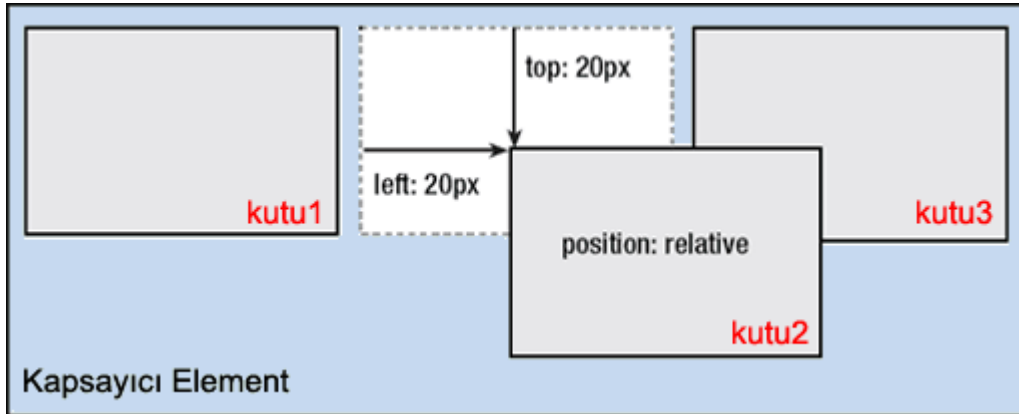
Opera 5+

CSS 2+

Göreceli Konumlandırma(position:relative)

Göreceli konumlandırma anlaşılması kolay bir konudur. Bir elemana göreceli bir değer verdiğinizde o bulunduğu yeri dikkate alarak verilen değerleri uygulayacaktır. Bir elemanın(göreceli) başlangıç noktasını dikey ve yatay olarak değiştirebiliriz. Eğer biz elemanın üst(top) değerine 20px tanımlarsak eleman bulunduğu yerden 20px aşağıya kayacaktır, eğer elemanın sol(left) değerini 20px olarak tanımlarsak elemanımız soldan 20px sağa doğru kayacaktır.

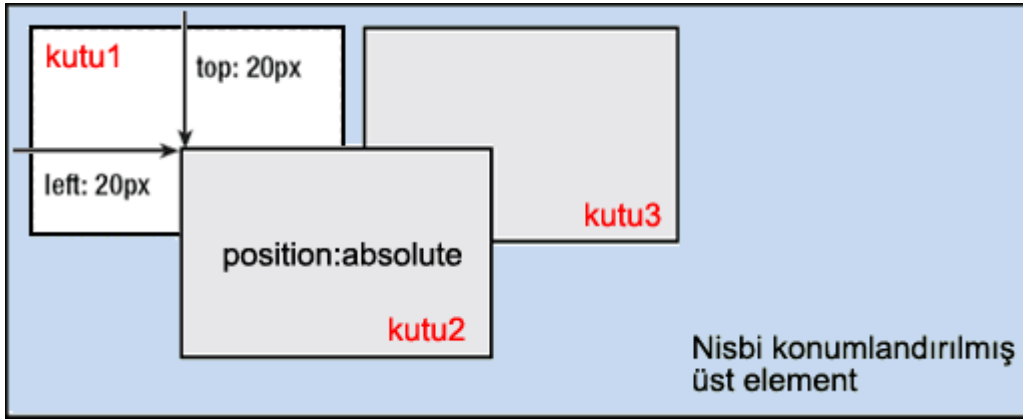
```
p#FarkliParagraf{  
position: relative;  
left: 20px;  
top: 20px;  
}
```



Göreceli konumlandırma kullanılarak yeri belirlenen bir elemanın başlangıçtaki konumundaki boşluğu ve diğer elemanlarla arasındaki mesafeyi koruyacaktır.

Mutlak(Absolute) Konumlandırma

Göreceli konumlandırma Normal Akışın bir parçasıdır. Aksine Mutlak Konumlandırmada uygulanan eleman bulunduğu yere boşluk bırakmayacaktır. Diğer elemanlar normal akıştaki yerlerini bu elemanı yok sayarak devam edeceklerdir. Aşağıdaki şekilde görüldüğü gibi



Mutlak konumlandırılmış elemanın konumu en yakındaki üst elemanın konumuna bağlıdır. Eğer bir ata elemanın konumu belirlenmemişse elemanımız ana bloğa göre konumlanacaktır. Mutlak konumlandırma HTML elemanı ve sayfa çözünürlüğünü ile ilişkilidir.

Göreceli olarak konumlandırılmış bir elemanın içindeki Mutlak Konumlandırılmış bir eleman içerik bloğunun üst, sağ, sol ve altından hizalandırılabilir. Bu kullanım bize birçok uygulamada yardımcı olacaktır ve bu kullanım ile çok güzel sonuçlar elde edeceğiz.

İnsanların konumlandırma ile ilgili genel problemleri hangi konumlandırma seçeneğinin ne işe yaradığını unutmalarıdır. Göreceli Konumlandırma sayfadaki elemanın başlangıç durumuna göre göreceli bir durumda olduğunu, oysa Mutlak Konumlandırmanın en yakın ata elemana veya içerik bloğuna göre göreceli değer içerdiği.

Mutlak Konumlandırma ile yerleştirilmiş eleman sayfada elemanı normal akışın dışına çıkarır. Bu eleman diğer elemanların üzerine binecektir. Bu üstte kalma altta kalma işinide **z-index** özelliği ile hallederiz. **z-index** değeri büyük olan eleman üstte küçük olan eleman ise altta kalacaktır.

Örneğin geniş bir blok içindeki bir paragraftaki yazıların sağ ve altta olmasını istiyorsak. Dıştaki içerik kutusunu göreceli konumlandırıp içindeki paragrafı mutlak konumlandırarak bu işi halledebiliriz.

```
#adresler {  
width: 700px;  
height: 100px;  
position: relative;  
}
```

```
#adresler .tel {  
position: absolute;  
right: 10px;  
bottom: 10px;  
text-align: right;
```

```
}  
.....  
<div id="adresler">  
<p class="tel">Tel: 0845 838 6163</p>  
</div>
```

Not:Bu kullanım gelişmiş tarayıcılar için geçerlidir. IE 5.5 ve IE 6(Windows)'da bu sorun çıkarır. Sorunu engellemek için yükseklik ve genişlik tanımı yapılmalıdır.

Mutlak konumlandırma sayfa planlaması yaparken çok kullanılır. Özellikle göreceli konumlandırılmış bir ata eleman varsa. Tüm siteyi sadece Mutlak Konumlandırma ile planlayabilmemiz mümkündür. Bunun için tüm elemanların belirli boyutlandırma işlemleri yapılması şarttır tabii bu elemanların üst üste binmesini engellemek için. Çünkü Mutlak Konumlandırılmış bir eleman doküman akışı dışına çıkacaktır ve normal akıştaki eleman kutularına etki etmeyecektir. Eğer siz Mutlak Konumlandırıdığınız elemanı genişletmek isterseniz örneğin yazıtipi boyutunu arttırdığınızda elemanı kapsayan kutu normale göre genişlese de etrafındaki elemanların akıştaki yeri değişmeyecektir.

Sabit Konumlandırma (position:fixed)

Sabit konumlandırma Mutlak Konumlandırmanın bir alt kategorisidir. Sabit konumlandırılmış elemanın sayfadaki görünümü çok farklıdır. Bunu özellikle web sayfasında bir elemanın konumunu sabitlemek için kullanabiliriz. Sabit konumlandırma ile yerleştirilmiş eleman belirtilen yerde asılı kalacaktır. Web tarayıcısının kaydırma çubuğu aşağı yukarı kaydırılması ile Sabit Konumlandırma ile yerleştirilmiş elemanın yeri değişmeyecektir ve sabit kalacaktır. Ne yazık ki IE 6 bu özelliği desteklemiyor(düzeltilmesini göreceğiz). IE 7 strict yorumlayıcısı kullanılıyorsa destekliyor diğer hallerde desteklemiyor.

Sabit konumlandırmayı sayfanın belli bir alanını sabitleyip diğer alanları bu kısmın yanından, altından kaydırmak için kullanırız genelde.

Aslında bunun için CSS bize **position:fixed** kodunu sunmaktadır. Ancak bu kodu IE6 ve önceki sürümlerde desteklememesi bize alternatif çözümler aramaya sevk ediyor.

İlk metodumuz normal konumlandırma özelliğinin fixed uygulanmış şeklidir.

Örnek 10-3

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<style type="text/css">
body {
margin:0;
padding:0;
}
div#ustKisim {
position:fixed;
background: gray;
color: yellow;
height:50px;
width:100%;
top:0;
left:0;
}
div#icerik {
margin:75px 0 0 0;
}
</style>
</head>
<body>
<div id="ustKisim">Lorem ipsum dolor sit amet</div>
<div id="icerik">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam nec
erat. Fusce magna massa, nonummy eget, aliquam tempus, imperdiet
vel, odio. Sed mauris. Quisque sodales urna vitae lorem. Vestibulum
aliquet, odio aliquam convallis lobortis, turpis metus semper ligula, ut
vulputate sem justo eu quam. Sed feugiat, lectus sit amet porttitor
auctor, nulla odio volutpat nisl, sit amet egestas augue diam at urna. Sed
justo arcu, luctus eget, porta in, auctor sit amet, massa. Integer
interdum. Nunc condimentum scelerisque enim. Pellentesque orci.
Maecenas at neque. Sed non diam ac turpis pretium dictum. Phasellus
ligula felis, aliquet nec, tristique non, posuere a, magna. Duis eu quam
quis tortor vehicula sodales. Donec et mi eget augue bibendum interdum.
Proin dapibus odio.</p>
....
</div>
</body>
</html>
```

Başta belirttiğimiz gibi buradaki en büyük sorunumuz IE6 ve daha eski sürümlerde bu kodun çalışmamasıdır. Bundan sonraki metot buna çözüm aramak içindir.

IE6'da çalışan position:fixed metodu

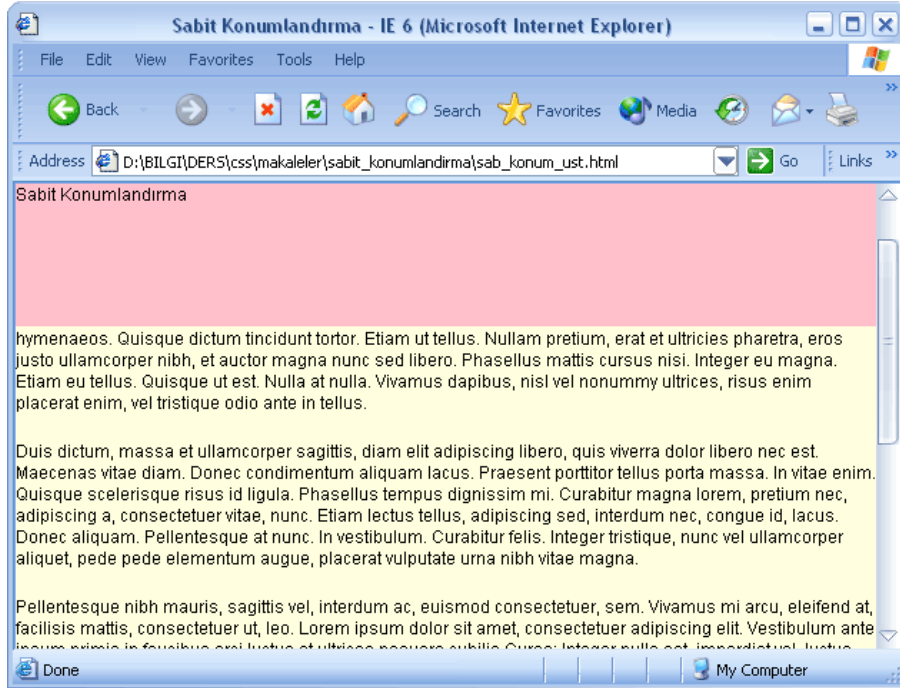
Bu metot IE'un css içinde **expression()** komutunu yardımı ile javascript kodu işletmemize izin vermesi prensibine dayanır. Bizde bu metotta bu

komut yardımı ile **position:fixed** sorununu çözeceğiz. **expression()** yardımı ile daha önce IE'nin min-width ve min-height sorununu çözdüğümüzü hatırlayın.

Örnek 10-4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>Sabit Konumlandırma</title>
<style>
body {
background: #fffcc;
font: 12px sans-serif;
margin: 0;
padding: 0;
}
div#ustAlan {
width: 100%;
height: 100px;
background: pink;
position: fixed;
top: 0;
left: 0;
z-index:100;
}
#icerik{
position:absolute;
top:105px;
}
</style>
<!--[if lt IE 7]>
<style>
body {
background: #fffcc url('http://') fixed;
}
div#ustAlan {
position: absolute;
top: expression(eval(documentElement.scrollTop));
}
</style>
<![endif]-->
</head>
<body>
<div id="ustAlan">Sabit Konumlandırma</div>
<div id="icerik">
```

```
<p>Lorem ipsum dolor sit amet ....</p>
</div>
</body>
</html>
```



Yukarıdaki kodu iki kısımda inceleyebiliriz. Birinci kısım ilk örnekte olduğu gibi **position:fixed** kullanılan IE7, FF, Opera ve Safari için yazılan kısımdır. İkinci kısım IE7'den önceki sürümleri belirten şartlı koşul içerisindeki kodlamadır. Bu sürümlerde **position:fixed** desteklenmediği için **position:absolute** değeri atanmış ve üst değerimizi(top) javascript'in **documentElement.scrollTop** değeri yardımı ile belirlenip sabitlemiştir.

Ayrıca alt alanı sabitlemek için

```
bottom: auto;
top: expression((documentElement.scrollTop +
documentElement.clientHeight this.clientHeight));
```

Kodlaması kullanılır.

Sabitlemek istediğimiz nesneyi üstten 10 piksel aşağıda sabitlemek için

```
top: expression(eval(documentElement.scrollTop) + 10)
```

Şeklinde kodlarız. Bu kodlama tekniği kullanılarak birçok farklı şablon elde edilebilir.

Bu yöntemin bazı sorunları vardır, bunlara değinmeden geçmeyelim.

- Bu metot IE5'de ve IE7, IE6'da garip modda çalışmaz.
- Bu metot javascript'in hizmet dışı bırakıldığı tarayıcılarda çalışmaz.
- **background: url('http://') fixed;** anlamsız kodu kullanılmaktadır. Bu kod kaydırma çubuğu hareket ettikçe sabitlenen nesnenin kıpramaması içindir.

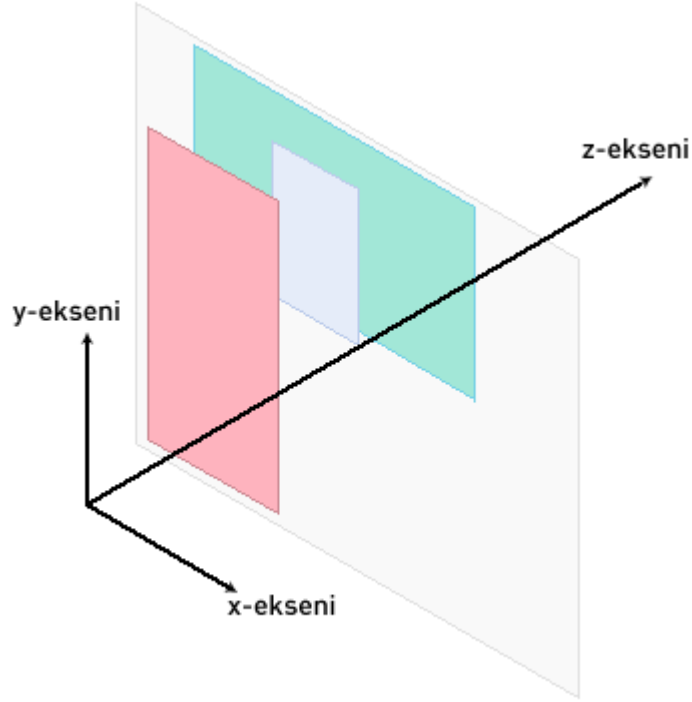
Bunun dışında birçok alternatif yöntem vardır. Benim deneyip en son kullanmaya karar verdiğim metot bu olduğu için sizlere bu kodu gösterdim.

10.3. z-index özelliği

z-index, konumlandırma elemanları uygulanmış katmanların z-eksenindeki konumu belirlemek için kullanılır.

z-index tüm elemanlara uygulanamaz sadece konumlandırma değeri atanmış elemanlara uygulanır. Bunun anlamı eğer bir elemanı aynı z-ekseni üzerinde üste veya alt sıraya atmak istiyorsanız ilk olarak konumlandırma değeri atamalısınız.

z-index değeri dokümanımızdaki elemanların istiflenme sıralarını belirler. z-index elemanları z-ekseninde nerede duracağınız belirler, üstte veya altta. z-indeks değeri küçük olan elemanlar altta büyük olan elemanlar ise üstte görünecektir. Sitenizi kodlarken birçok yerde z-index ihtiyaç duymayız çünkü normal sayfa akışında elemanlar yan yana durduğu için böyle bir özelliğe ihtiyaç yoktur, ne zamanki elemanlara konumlandırma değeri tanımladığımızda elemanlar bir biri üzerine bindiğinde z-index değerine ihtiyaç duyarız. z-index mutlak, göreceli ve sabit konumlandırma atandığı veya elemana eksi değer verilerek konumu değiştirildiğinde uygulanır.



Yapısı: **z-index: <değer>**

Aldığı Değerler: **<sayısal değer > | auto | inherit**

Başlangıç değeri: **auto**

Uygulanabilen elemanlar: **konumlandırılmış elemanlara**

Kalıtımsallık: **Yok**

Tarayıcı Uyumu:

Internet Explorer 4+(kısmen, IE4 den sonra auto ve sayısal değerleri destekliyor)

Firefox 1+

Safari 1.3+

Opera 5+

CSS 2+

Konumlandırma değeri relative, absolute ve fixed uygulanmış katmanların görünürlüğünü z-index ile ayarlayabiliriz.

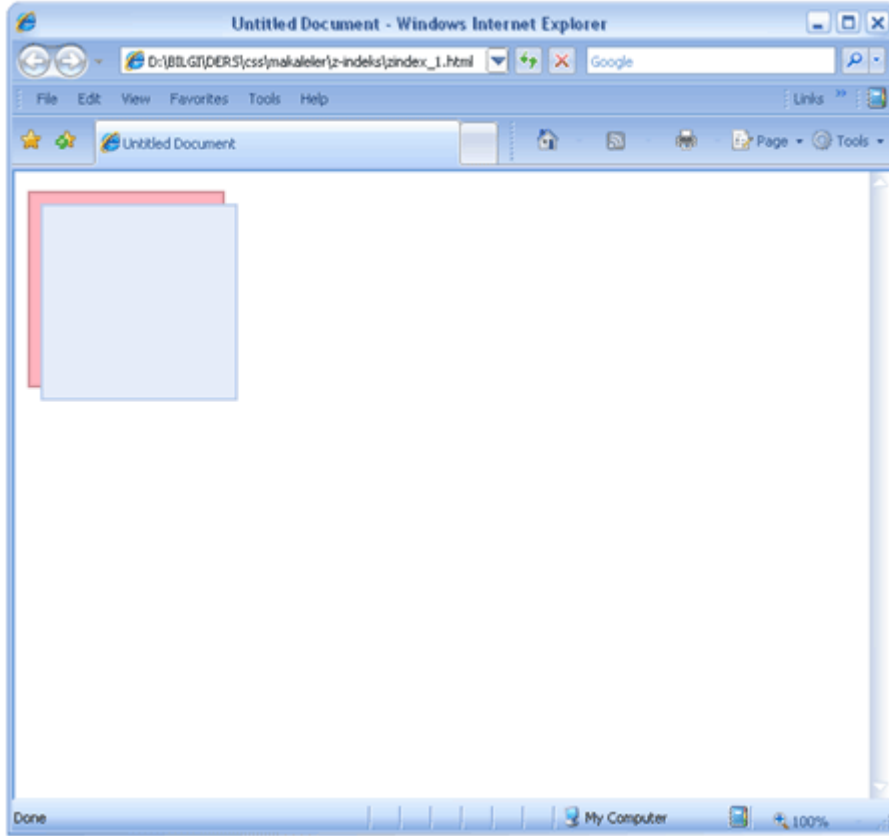
```
div{  
  position:absolute;  
  width:150px;  
  height:150px;  
}  
div.bir {  
  background: #FEB3BE;  
  border:2px solid #CC8B94;  
  top: 0;
```

```
left: 0;
}
div.iki {
background: #E5ECF9;
border: 2px solid #BCCCEB;
top: 10px;
left: 10px;
}
```

Yukarıda örnekte görüldüğü gibi katmanlar üst üste sıralanmıştır. Birbirinden 10px üst ve 10px soldan mesafe bırakılmıştır. Üstte kalan katman alttakileri gizlemiştir. Tüm katmanların z-index değeri atanmamıştır bu nedenle başlangıç değeri olan z-index:auto değerini almışlardır.

```
<body>
<div class='kapsul'>
<div class='bir'></div>
<div class='iki'></div>
</div>
</body>
```

z-index değeri otomatik olduğu için her katman Html'deki sırasına göre yerleşmiştir. İlk başta yazılmış olanlar altta sonrakiler üstte olacak şekilde sıralanmıştır.

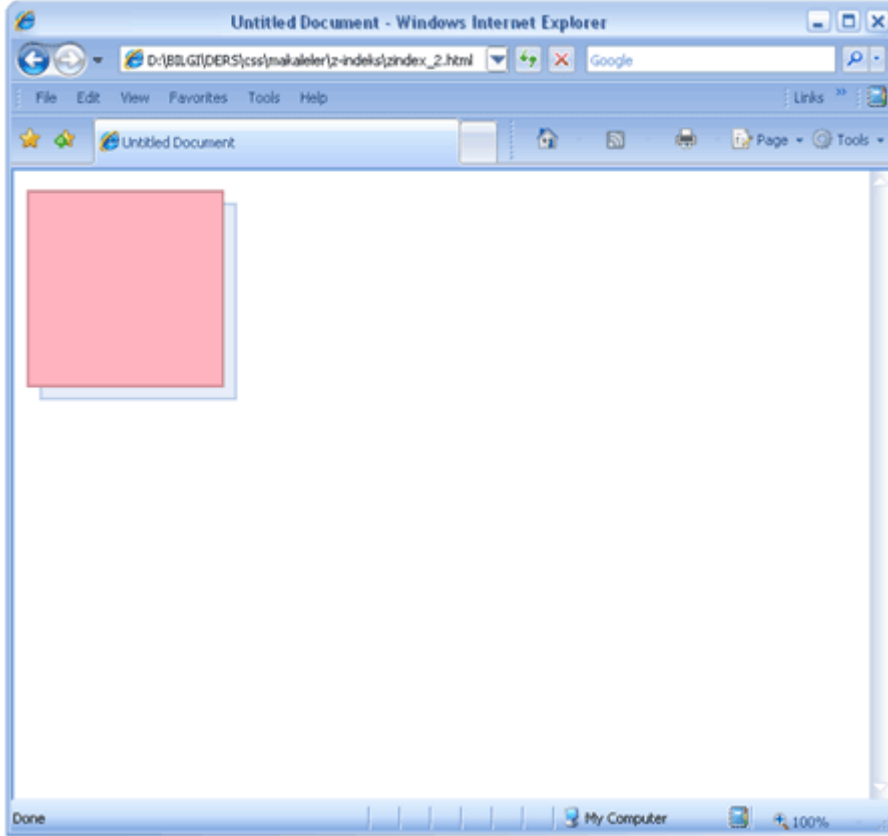


Eğer katmanları istediğimiz sıra ile göstermek istiyor isek her katmana sırasına göre sayısal değer atamalıyız. Yüksek değer alan katmanlar üstte düşük değer alan katmanlar ise altta kalacaktır. Buna göre istediğimiz görüntüyü elde etmek için z-index değerleri vermeliyiz.

Yukarıda yaptığımız örnekte alttaki kırmızı katmanı üste çıkarmak için z-index değerini 2 versek. Mavi katmanın z-index değerini 1 verirsek katman görünümünü tersine dönecektir.

```
div{  
  position:absolute;  
  width:150px;  
  height:150px;  
}  
div.bir {  
  background: #FEB3BE;  
  border:2px solid #CC8B94;  
  top: 0;  
  left: 0;  
  z-index:2  
}  
div.iki {  
  background: #E5ECF9;  
  border:2px solid #BCCCEB;  
  top: 10px;
```

```
left: 10px;  
z-index:1  
}
```



İç içe girmiş katmanlarda z-index davranışları farklıdır. İç içe geçmiş katmanlardaki z-index:auto değeri gibi davranır ve sayısal atamaları dikkate almaz.

```
<body>  
<div class='kapsul'>  
<div class='bir'><div class='iki'></div></div>  
</div>  
</body>
```

z-index değeri yüksek olmasına rağmen **“.bir”** sınıfını alan katmana altta kalacaktır. Çünkü iç içe geçmiş elemanlarda z-index'e atanan sayısal değerler geçersizdir. Sıralama z-index:auto ya göre yapılır.

Sonuç olarak z-index özelliğinin çok kullanışlı ancak sorunları çok bir özellik olduğunu düşünüyorum. Birçok makalede istediğim manada detaylı bir z-index anlatımı bulamadığımı itiraf etmeliyim. Ancak kodlama yaparken aklımızda olması gereken kuralları kavrarsak z-index özelliğini daha bilinçli kullanacağımız kesin:

- z-index değeri; aynı z-ekseninde bulunan(kapsayan elemanlar hariç) katmanların altta mı üstte mi kalacağını belirlemek için kullanılır.
- z-index sadece position değeri **relative**, **absolute** ve **fixed** olan nesnelere uygulanır.
- Saydamlık değeri(opacity) 1'den küçük verilen nesnelere z-index kullanımı daha kolay anlaşılır.

Konumlandırma uygulanmış elemanlarda z-index etkileri

- Mevcut sıralama durumunda elemanın sırasını belirler
- Elemanın kendi kısmındaki durumun belirler

z-index eksi değerlerinde Firefox sorun çıkarıyor. Firefox 3'da sorun giderilmiş.

Eğer z-index değeri atanmamış ise z-eksenindeki arkadan öne doğru sıralama aşağıdaki gibidir:

- Normal akıştaki kutular koddaki sıraya göre sıralanır
- float uygulanmış kutular
- Konumlandırma uygulanmış elemanlarda ise kodlamadaki sıraya göre

10.4. display özelliği

Yapısı: **display: <değer>**

Aldığı Değerler: **none | inline | block | inline-block | list-item | run-in | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | inherit**

Başlangıç değeri: **inline**

Uygulanabilen elemanlar: **tüm elemanlar**

Kalıtıllık: **Yok**

Bu özellik bir elemanı oluşturan kutu tipini belirler.

Bu özelliğin diğer özellikler ile bir çok etkileşimli durumları vardır, bunlara ileride değineceğiz.

HTML elemanlarını başlangıçta blok seviyeli ve satır içi seviyeli olarak tanımlıdır. Ancak biz bu elemanların başlangıç değerlerini **display** özelliği ile değiştirebiliriz. Yani biz bir satır içi seviyedeki elemanı örneğin bir linki(<a>....) blok seviyeden bir elemana dönüştürebiliriz. Aynı şekilde bir blok seviyedeki elemanı satır içi seviyeden bir elemana dönüştürebiliriz.

block: Bu değer bir elemanın bir blok kutusu üretmek için kullanılır.

inline: Bu değer bir elemanın bir veya daha fazla sayıda satır içi kutu üretmek için kullanılır.

list-item: Bu değer bir elemanın (HTML'deki gibi) bir blok kutu ve bir liste öğesi satır içi kutusu üretmek için kullanılır.

none: Bu değer bir elemanın biçimleme yapısında hiçbir kutu üretmemesine sebep olur (yani, elemanın yerleşime hiçbir etkisi olmaz; başka bir deyişle, gösterilmez). Ayrıca alt elemanlar da görünmez ve bu davranış alt elemanların display nitelikleriyle etkisiz kılınamaz.

table, inline-table, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell ve table-caption: Bu değerler bir elemanın bir tablo elemanı gibi davranmasına sebep olur.

run-in, inline-block, ve inline-table özelliklerini tüm tarayıcılar desteklememektedir.

```
p {  
  display: inline;  
}  
em {
```

```
display: block;  
}
```

Internet Explorer 7+:

- inline-table, run-in, table, table-caption, table-cell, table-column, table-column-group, table-row, and table-row-group bu değerleri desteklemiyor.
- HTML elemanlarından thead ve tfoot için sadece table-footer-group and table-header-group özelliklerini destekliyor.
- Başlangıç değeri satırıçi seviyesinden olanlar için inline-block değerini destekliyor.
- Li elamanına atanan list-item özelliğini bir blok seviyeli elman gibi yorumluyor
- inherit değerini desteklemiyor

Firefox 2.0+ ve Opera 9.2+:

* colgroup elemanı için sadece table-column-group değerini destekliyor.

* col elemanı için sadece table-column özelliğini destekliyor.

Firefox 2.0 inline-block, inline-table, veya run-in özelliklerini desteklemiyor. Bu özellikler Firefox 3 versiyona eklenmiş.

Tarayıcı Uyumu:

Internet Explorer 5.5+(kısmen)

Firefox1+ (kısmen)

Safari1.3+

Opera 9.2+

CSS 2+

11. Tablo Özellikleri

W3C tablo ve tablo elemanlarının görünümünü düzenlemek için tablo özelliklerini ekleme gereği duydu. Tablolar diğer HTML elemanlarından daha farklı kendine has elemanlardır ve farklı özellikleri vardır. Burada CSS2 ile birlikte gelen yeni özellikleri inceleyeceğiz.

CSS2 ile birlikte gelen bu özelliklerden birçoğumuzun haberi olmayabilir. Ancak bizlere yardımcı olacak bu özelliklere göz atmak güzel. Bu özellikleri uygulamamızda en önemli etken tabii ki IE'nin bu özelliklerin bir kısmını desteklememesidir. Bu tabii ki IE'ye puan kaybettiriyor ve Microsoft'da bunun farkına geçte olsa vardığı söyleniyor ve IE8 ile birlikte tam CSS desteğini bizlere sunacakmış, tabii o zamana kadar Firefox IE'yi silip süpürmezse.

- Başlık Konumu Özelliği (table-layout)
- Tablo Planı özelliği (caption-side)
- Hücre Boşluklarını ayarlama (border-collapse)
- Hücre Dışı boşluklarını ayarlama (border-spacing)
- Boş hücre ayarları (empty-cells)

11.1. Başlık Konumu Özelliği (table-layout)

Yapısı: **table-layout: <değer>**

Aldığı Değerler: **auto | fixed | kalıtsallık**

Başlangıç değeri: **auto**

Uygulanabilen elemanlar: **display: table ve display:inline-table olarak belirlenen elemanlara**

Kalıtsallık: **Var**

Normalde tablolar içeriğindeki bilgiye göre genişliğini arttırır. Bu bize bir esnek kazandırır ancak bazen tablo genişliğimizin sabit kalmasını isteriz. table-layout:fixed değeri tablo genişliğimizi sabitlememizi sağlar. table-layout:auto ise tablo genişliğini içeriğe göre arttırır.

Örnek 11-1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>table-layout örneği</title>
<style>
table.otomatik {
table-layout: auto;
```

```

width: 350px;
border-collapse: collapse;
}
table.sabit {
table-layout: fixed;
width: 350px;
border-collapse: collapse;
}
td, th {
border: 1px solid;
}
</style>
</head>
<body>
<table cellpadding="0" cellspacing="0" class="otomatik">
<caption>Limitli ADSL Fiyatları</caption>
<tr>
<th>Hızı*(Kbps) </th>
<th>Kota(GB) </th>
<th>Bağlantı Ücreti** </th>
<th>Aylık Ücret </th>
<th>Limitin Üzerindeki Her MB için Ücret*** </th>
</tr>
<tr>
<td>1024/256'e kadar</td>
<td>4</td>
<td rowspan="3">29 YTL</td>
<td>29 YTL</td>
<td>0,010 YTL</td>
</tr>
<tr>
<td>1024/256'e kadar</td>
<td>6</td>
<td>39 YTL</td>
<td>0,009 YTL</td>
</tr>
<tr>
<td>2048/512'e kadar</td>
<td>6</td>
<td>49YTL</td>
<td>0,009 YTL</td>
</tr>
</table>
&nbsp;
<table cellpadding="0" cellspacing="0" class="sabit">
<caption>Limitli ADSL Fiyatları</caption>
<tr>
<th>Hızı*(Kbps) </th>

```

```

<th>Kota(GB) </th>
<th>Bağlantı Ücreti** </th>
<th>Aylık Ücret </th>
<th>Limitin Üzerindeki Her MB için Ücret*** </th>
</tr>
<tr>
<td>1024/256'e kadar</td>
<td>4</td>
<td rowspan="3">29 YTL</td>
<td>29 YTL</td>
<td>0,010 YTL</td>
</tr>
<tr>
<td>1024/256'e kadar</td>
<td>6</td>
<td>39 YTL</td>
<td>0,009 YTL</td>
</tr>
<tr>
<td>2048/512'e kadar</td>
<td>6</td>
<td>49YTL</td>
<td>0,009 YTL</td>
</tr>
</table>
</body>
</html>

```

table-layout örneği - Mozilla Firefox

Dosya Düzen Görünüm Geçmiş Yer İmleri Araçlar GÜBİL Yardım

file:///D:/BILGI/DERS/c

Google

Limitli ADSL Fiyatları

Hız*(Kbps)	Kota(GB)	Bağlantı Ücreti**	Aylık Ücret	Limitin Üzerindeki Her MB için Ücret***
1024/256'e kadar	4	29 YTL	29 YTL	0,010 YTL
1024/256'e kadar	6		39 YTL	0,009 YTL
2048/512'e kadar	6		49 YTL	0,009 YTL

Limitli ADSL Fiyatları

Hız* Kbps	Kota(GB)	Bağlantı Ücreti**	Aylık Ücret	Limitin Üzerindeki Her MB için Ücret***
1024/256'e kadar	4	29 YTL	29 YTL	0,010 YTL
1024/256'e kadar	6		39 YTL	0,009 YTL
2048/512'e kadar	6		49 YTL	0,009 YTL

Tamam

PageRank: unranked

Compete

Tarayıcı Uyumu:

Internet Explorer 5+
Firefox 1+
Safari 1.3+
Opera 7.5+
CSS 2+

CSS2 ile birlikte left ve right değerleri de vardı ancak kullanışsızlığı nedeniyle CSS2.1'de kaldırıldı.

11.2. Tablo planı özelliği (caption-side)

Yapısı: **caption-side: <değer>**

Aldığı Değerler: **top | bottom | inherit**

Başlangıç değeri: **top**

Uygulanabilen elemanlar: **display: table-caption** olarak belirlenen elemanlara

Kalıtıllık: **Yok**

caption-side: Tablonun başlığının(<caption>) nerede(alтта mı üstte mi) olacağını belirler.

The screenshot shows a Mozilla Firefox browser window displaying a page titled 'Limitli ADSL Fiyatları'. The page contains two identical tables. The top table is captioned 'Limitli ADSL Fiyatları' and the bottom table is also captioned 'Limitli ADSL Fiyatları'. The tables contain the following data:

Hız* (Kbps)	Kota (GB)	Bağlantı Ücreti**	Aylık Ücret	Limitin Üzerindeki Her MB için Ücret***
1024/256'e kadar		29 YTL	29 YTL	0,010 YTL
1024/256'e kadar	6	29 YTL	39 YTL	0,009 YTL
2048/512'e kadar	6	49 YTL	49 YTL	0,009 YTL

Bu özelliği IE(ie5+/mac destekliyor) desteklemiyor. Tüm tarayıcıların desteklediği şekilde kod yazmak için <caption> etiketinin **align** özelliği kullanılmalıdır. Örnek:

<caption align="bottom"> ve <caption align="top">

Tarayıcı Uyumu:

Internet Explorer (**desteklemiyor**)

Firefox 1+

Safari 1.3+

Opera 7.5+

CSS 2+

11.3. Hücre Boşluklarını ayarlama (border-collapse)

Yapısı: **border-collapse: <değer>**

Aldığı Değerler: **collapse | separate | kalıtsallık**

Başlangıç değeri: **separate**

Uygulanabilen elemanlar: **display:table ve display:table-caption**

olarak belirlenen elemanlaraKalıtsallık: **Var**

border-collapse, tablo hücrelerinin etrafındaki kenarlık ile olan mesafeyi azaltmamızı veya açmamızı sağlar. **separate** ataması yapılmış ise hücreler arasına bir kaç piksellik mesafe konur, HTML içinden bu boşlukları <table> etiketinin cellspacing özelliği ile kaldırmak istesek bile tarayıcılar bu seferde iki kenarlığı bir bir üstü gösterecektir. Eğer **collapse** değeri verilirse iki hücre arasındaki mesafe kalkacaktır ve tek kenarlık görünecektir.

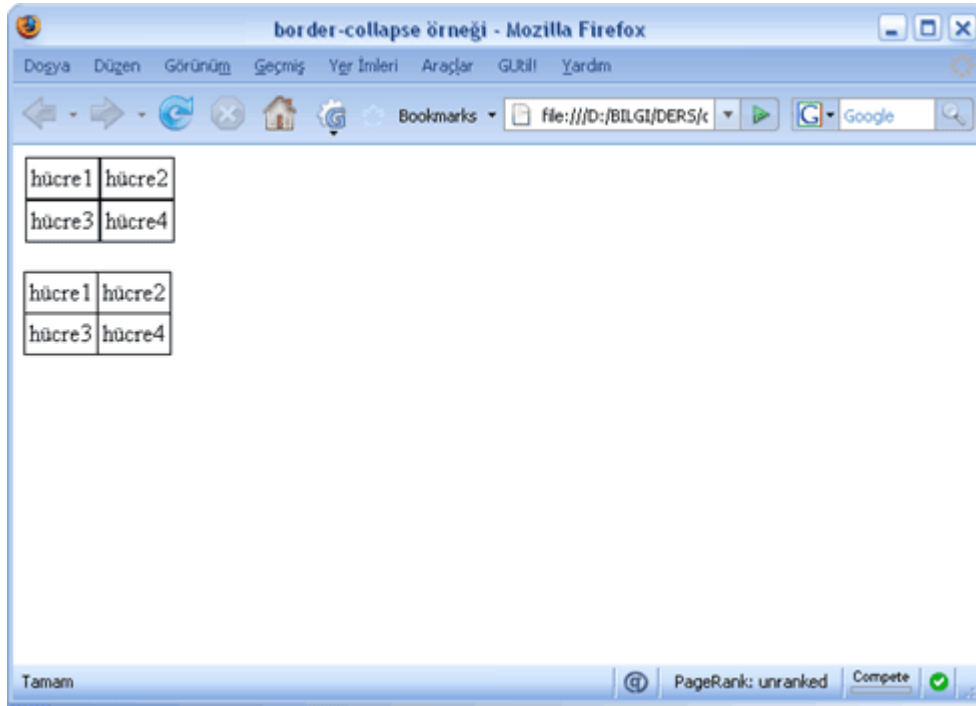
Örnek 11-2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>border-collapse örneği</title>
<style>
table.ayri {
border-collapse: separate;
}
table.birlesik {
border-collapse: collapse;
}
td {
border: 1px solid black;
padding: 3px;
}
</style>
</head>
<body>
<table cellspacing="0" class="ayri">
<tr>
<td>hücre1</td>
<td>hücre2</td>
</tr>
<tr>
<td>hücre3</td>
<td>hücre4</td>
</tr>
</table>
&nbsp;
<table border="0" cellspacing="0" class="birlesik">
<tr>
<td>hücre1</td>
<td>hücre2</td>
```

```

</tr>
<tr>
<td>hücre3</td>
<td>hücre4</td>
</tr>
</table>
</body>
</html>

```



Tarayıcı Uyumu:

Internet Explorer 5.5
 Firefox 1+
 Safari 1.3+
 Opera 7.5+
 CSS 2+

11.4. Hücre dışı boşluklarını ayarlama (border-spacing)

Yapısı: **border-spacing: <değer>**

Aldığı Değerler: **<uzunluk> <uzunluk>? | inherit**

Başlangıç değeri: **0**

Uygulanabilen elemanlar: **display:table** ve **display:table-caption** olarak belirlenen elemanlara

Kalıtıllık: **Var**

border-spacing, normal uzunluk değerleri alır ve **border-collapse:separate** ile birlikte kullanılır. Eğer tek değer verilirse bu tüm

kenarlar için geçerlidir, iki değer atanırsa ilki yatayda boşluk için ikincisi dikey de boşluk vermek için tanımlanmıştır.

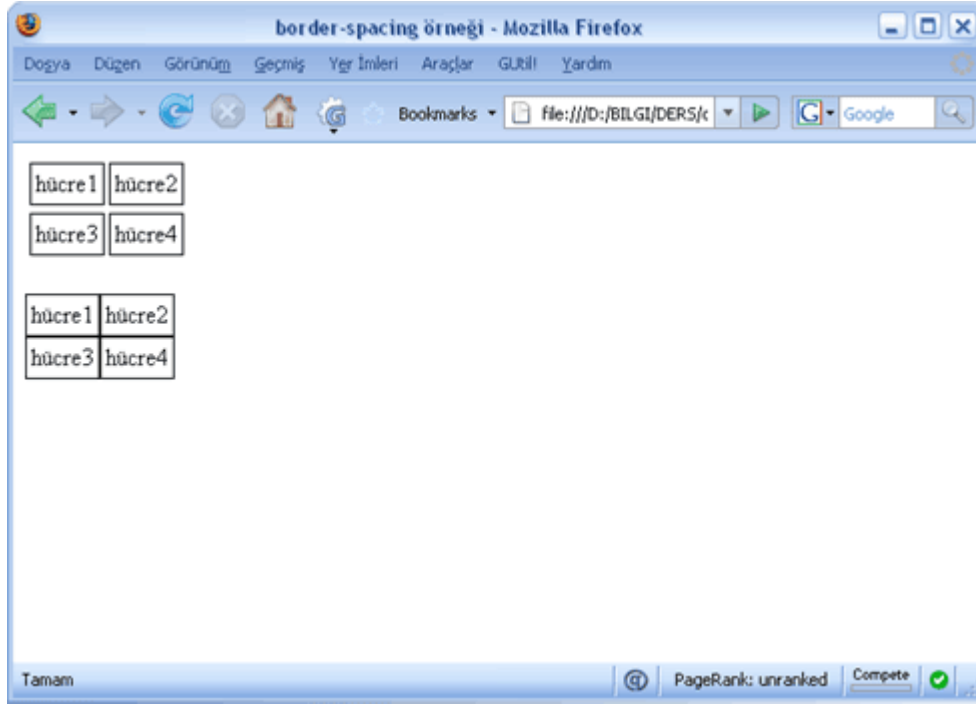
Örnek 11-3

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>border-spacing örneği</title>
<style>
table.ayri {
border-collapse: separate;
border-spacing: 3px 5px;
}
table.birlesik {
border-collapse: separate;
}
td {
border: 1px solid black;
padding: 3px;
}
</style>
</head>
<body>
<table cellspacing="0" class="ayri">
<tr>
<td>hücre1</td>
<td>hücre2</td>
</tr>
<tr>
<td>hücre3</td>
<td>hücre4</td>
</tr>
</table>
&nbsp;
<table border="0" cellspacing="0" class="birlesik">
<tr>
<td>hücre1</td>
<td>hücre2</td>
</tr>
<tr>
<td>hücre3</td>
<td>hücre4</td>
</tr>
```

```

</table>
</body>
</html>

```



Tarayıcı Uyumu:

Internet Explorer (**desteklemiyor**)
 Firefox 1+
 Safari 1.3+
 Opera 7.5+
 CSS 2+

11.5. Boş hücre ayarları (empty-cells)

Yapısı: **empty-cells: <değer>**

Aldığı Değerler: **show | hide | kalıtsallık**

Başlangıç değeri: **show**

Uygulanabilen elemanlar: **display:table-cell olarak belirlenen elemanlara**

Kalıtsallık: **Var**

Bir tabloya bazı değerler atadığımızda(örneğin kenarlık değeri) boş olan hücreler bu değeri görmez, bu durumu düzeltmek için boş olan hücreler için boş karakter() koyarız genel. **empty-cells** özelliği ile boş kalan hücrelere nasıl davranması gerektiğini bildirebiliriz. Göster(**show**) değeri ile hücrenin içeriği olan hücreler ile aynı özellikleri almasını sağlayabiliriz. Gizle(**hide**) değeri ile de hücrenin yokmuş gibi davranmasını sağlarız.

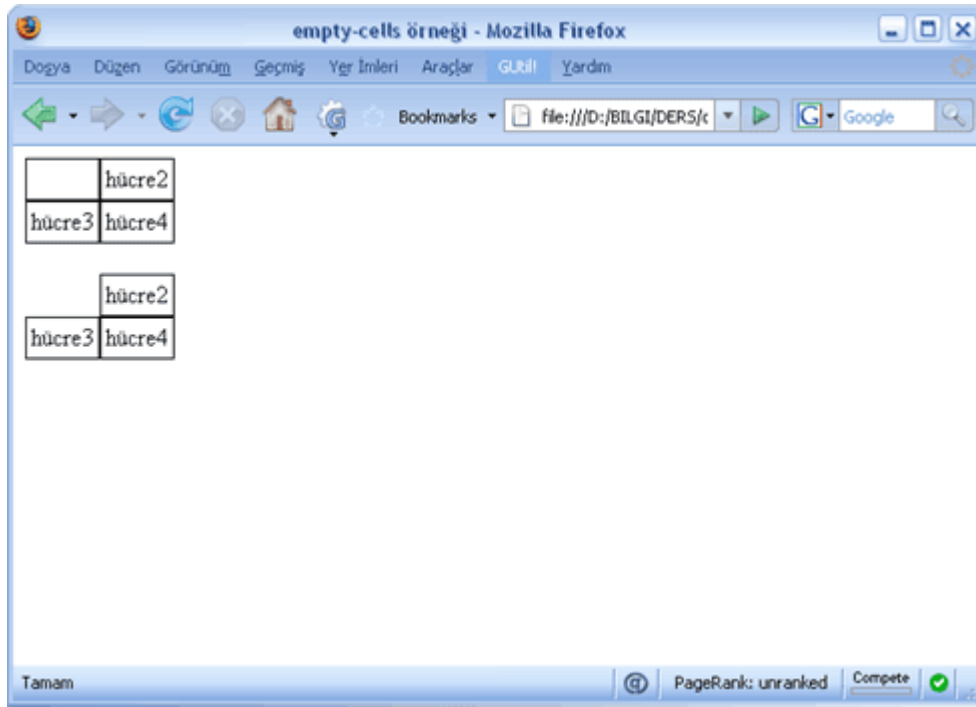
Bu özellik Internet Explorer tarafından desteklenmemektedir.

Örnek 11-4

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
<title>empty-cells örneği</title>
<style>
table.goster {
empty-cells:show
}
table.gizle {
empty-cells:hide
}
td {
border: 1px solid black;
padding: 3px;
}
</style>
</head>
<body>
<table cellspacing="0" class="goster">
<tr>
<td></td>
<td>hücre2</td>
</tr>
<tr>
<td>hücre3</td>
<td>hücre4</td>
</tr>
</table>
&nbsp;
<table border="0" cellspacing="0" class="gizle">
<tr>
<td></td>
<td>hücre2</td>
</tr>
<tr>
<td>hücre3</td>
<td>hücre4</td>
</tr>
</table>
</body>
</html>

```



Tarayıcı Uyumu:

Internet Explorer (**desteklemiyor**)

Firefox 1+

Safari 1.3+

Opera 7.5+

CSS 2+

12. Kullanıcı Arayüzü Özellikleri

Kullanıcı arayüzü özelliği tarayıcıdaki kullanıcı arayüzündeki nesnelere değiştirme izni verir.

CSS gittikçe etkinliğini arttırmaktadır. Kullanıcı arayüzü özellikleri ile tarayıcının odaklanma özelliklerine dokümanın yazıtipi ve rengini değiştirmek için ve fare imlecini şekliyle değiştirmemize olanak sağlar.

Firefox bunu daha da ileri giderek tarayıcının birçok özelliğini değiştirmemize olanak sağlar. CSS'e benzeyen XUL dili ile tarayıcının üst menüsüne, kenar tablalarına, mesaj kutularına, durum çubuğuna ve diğer tarayıcı elemanlarını değiştirmemizi sağlar.

12.1. Fare imleci özellikleri tanımlama (cursor)









Yapısı: **cursor: <değer>**









Aldığı Değerler: [**<uri>** ,]* [**auto** | **crosshair** | **default** | **pointer** | **move** | **e-resize** | **ne-resize** | **nw-resize** | **n-resize** | **se-resize** | **sw-resize** | **s-resize** | **w-resize** | **text** | **wait** | **help** | **progress**]

Başlangıç değeri: **auto**

Uygulanabilen elemanlar: **tüm elemanlara**

Kalıtımsallık: **Var**

	crosshair	İnce artı imleci
	default	İçeriğe bakmaksızın platformun başlangıç imlecini gösterir.
	pointer	Linklerde çıkan el işareti imlecini gösterir.
	move	Taşıma imlecini gösterir.
	e-resize	Batı yönü(east) imlecini gösterir.
	ne-resize	Kuzey doğu yönü(north east) taşıma imlecini gösterir.
	nw-resize	Kuzey batı yönü(north west) imlecini gösterir.
	n-resize	Kuzey yönü(North) imlecini gösterir.

	se-resize	Güney doğu yönü(south east) imlecini gösterir.
	sw-resize	Güney batı yönü(south west) imleci
	s-resize	Güney yönü(South) imlecini gösterir.
	w-resize	Batı yönü(west) imlecini gösterir.
	text	Metin imlecini gösterir.
	wait	Bekleme imlecini gösterir. Genelde kum saati şeklindedir.
	help	Yardım imlecini gösterir
	auto	Mevcut içeriğe göre tarayıcının başlangıçta tanımladığı imlecini gösterir.
	progress	Yükleniyor imlecini gösterir. Bu özellik için farklı tarayıcılar farklı şekiller çıkarır.

URI özelliği için örnek:

```
a.external {  
  cursor: url(disarilink.cur), pointer  
}
```

IE 5.5 ve alt sürümlerinde pointer değerini desteklemez. Onun yerine standartlarda olmayan hand özelliği destekler.

IE inherit özelliği desteklemiyor.

Opera 9.2 sürümü ve öncesinde URI değerini desteklemez.

Safari 3 sürümü ve öncesinde URI değerini desteklemez.

CSS3 ile birlikte yeni

Tarayıcı Uyumu:

Internet Explorer 5.5+
Firefox 1+
Opera 9.2+(kısmen)
Safari 1.3+(kısmen)
CSS 2+

12.2. Dış hat çizgisi özellikleri(outline)

CSS2 ile birlikte gelen kutu dışı şekillendirme özellikleri kenar çizgisine(border) benzer bir özelliktir. Fakat kenar çizgisi özelliğinden iki

önemli farkı vardır. İlki outline özelliği kenar çizgisi gibi normal akışı etkilemez. Bir elemana 10px outline değeri atandığında elemanın etrafındaki elemanların üzerine 10px kadar taşar. İkinci farklılık outline kenar çizgisi gibi dikdörtgen şeklinde değildir. Oval şekilde değildir.

Outline özelliği kullanıcının o ana nereye odaklandığını belirlemek için kullanılır. Tarayıcı arayüzünde kullanıcının nereye odaklandığını bilmek bize birçok avantajlar sağlayacaktır.

12.2.1 Dış hat çizgisi biçimi (outline-style)

Dış hattın biçimini belirlemek için kullanırız.

Yapısı: **outline-style: <deger>**

Aldığı Değerler: **none | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit**

Başlangıç değeri: **none**

Uygulanabilen elemanlar: **tüm elemanlara**

Kalıtıllık: **Yok**

Dış hat biçimi none değeri haricindeki değerler için görüntülenir.

none: Dış hat görünmesin anlamındadır.

dotted: Dış hat belirli aralıklarla dizilmiş noktalardan oluşur.

dashed: Dış hat belirli aralıklarla dizilmiş çizgi parçalarından(tire) oluşur.

solid: Dış hat bir çizgi olara görünür

double: iki çizgi şeklinde görünür. İki çizginin kalınlıkları ile aralarındaki boşluğun genişliğinin toplamı outline-width değerine eşittir.

groove: Dış hat tuvale oyulmuş gibi görünür.

ridge: groove'un zıddı: Dış hat çizgisi kabartma gibi görünür.

inset: Dış hat çizgisinin kutunun tamamını tuvale gömülü gibi görünmesine sebep olur.

outset: inset'in zıddı: Dış hat çizgisi kutunun tamamının kabartma gibi görünmesine sebep olur.

Tarayıcı Uyumu:

Internet Explorer (**desteklemiyor**)

Firefox 1.5+

Opera 9.2+
Safari 1.3+
CSS 2+

12.2.2. Dış hat çizgisi genişliği (outline-width)

Atandığı elemanın dışına çizilen hattın kalınlığını belirlemek için kullanırız.

Yapısı: **outline-width: <değer>**
Aldığı Değerler: **thin | medium | thick | length | inherit**
Başlangıç değeri: **Medium**
Uygulanabilen elemanlar: **tüm elemanlara**
Kalıtsallık: **Yok**

Dış hat çizgisi genişliği border-width gibi CSS uzunluk değerlerini (px, pt, em) ve anahtar kelimeler(thin <= medium <= thick) alır, ancak yüzde değer ve eksi değer almaz.

Tarayıcı Uyumu:

Internet Explorer (**desteklemiyor**)
Firefox 1.5+
Opera 9.2+
Safari 1.3+
CSS 2+

12.2.3. Dış hat çizgisi rengi (outline-color)

Atandığı elemanın dışına çizilen hattın rengini belirlemek için kullanırız.

Yapısı: **outline-color: <değer>**
Aldığı Değerler: **color | invert | inherit**
Başlangıç değeri: **invert**
Uygulanabilen elemanlar: **tüm elemanlara**
Kalıtsallık: **Yok**

CSS'in desteklediği renk değerlerini kullanabiliriz, hatta invert değerini de kullanabiliriz. Invert, rengi tersine çevirir.

Tarayıcı Uyumu:

Internet Explorer (**desteklemiyor**)
Firefox 1.5+

Opera 9.2+
Safari 1.3+
CSS 2+

12.2.4. Dış hat çizgisi (outline)

Dış hat çizgisi(outline) özelliği yukarıda gördüğümüz özelliklerin tek seferde kullanılmasını sağlayan bir kısa yoldur.

Yapısı: **outline: <değer>**

Aldığı Değerler: [**<outline-color> || <outline-style> || <outline-width>] | inherit**

Başlangıç değeri: **tanımlı değildir**

Uygulanabilen elemanlar: **tüm elemanlara**

Kalıtıllık: **Yok**

Kenar çizgisi(border) özelliğine benzer bir yapısı vardır. Ancak ayrıldıkları kısım kenar çizgisinde(border) her kenar için bir atama(border-right, border-bottom, vd.) yapılmasına karşın dış hat çizgisinde(outline) böyle bir özellik yoktur. Dış hat çizgisinin(outline) tüm kenarları aynı özellikleri taşır.

Dış hat çizgileri(outline) normal akıştaki boşlukları tamamlamaz. (taşma yapmaz.) diğer elemanların üzerini örter. Dış hat çizgisinin(outline) önemli özelliklerinden biri etrafını çevreleyen elemanlar üzerinde(üzerine örtmek dışında) bir etki yapmamasıdır, bu sayfa planlamadaki problemlerimizi çözmemiz için önemlidir. Problem olan bir elemana dış hat çizgisi(outline) ataması yaptığımızda tam olarak sorunun ne olduğunu göreceğiz ve elemanın tam yerini öğreneceğiz. Bu kenar çizgisi(border) uygulamaktan farklıdır, dış hat çizgisi(outline) uygulandığında dokümandaki normal akış üzerine bir etki yapmayacaktır.

Dış hat çizgileri(outline) satırıçi elemanların etrafına yerleşir. İki satıra kaymış satırıçi elemanlarda dış hat çizgisi dikdörtgen şeklinde olmayacaktır. Bu da kenar çizgisi(border) ile arasındaki farklardan biridir.

Dış hat çizgileri elemana odaklanıldığında görsel bir çıkıntı gibi görünmesini sağlayabilir.

Tarayıcı Uyumu:

Internet Explorer (**desteklemiyor**)

Firefox 1.5+

Opera 9.2+

Safari 1.3+

12.3. Sistem yazıtipi ve rengi özellikleri

Sistem yazıtipi ve rengi özellikleri ile biz kullanıcının bilgisayarında kurulu sistemin(Windows, Linux, mac vd.) yazıtipi ve renk özelliklerini kullanmamızı sağlayan özelliklerdir. Ancak burada şöyle bir sorun var ki CSS3'de bu özelliklerin kullanımına karşı çıkmıştır. Şimdilik son durum budur ama gelecekte ne olur onu bilemeyiz.

Ek – A Tüm CSS özellikleri

Tarayıcı Uyumu: IE: Internet Explorer, F: Firefox

W3C: "W3C"ün özelliği eklediği CSS sürümü

Artalan (Background) özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
background	Tüm artalan özelliklerinin bir kısaltmasıdır.	<i>background-color</i> <i>background-image</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-position</i>	4	1	1
background-attachment	Artalan resminin sayfanın diğer elemanlarına göre sabitlenmesini veya kaymasını belirler	scroll fixed	4	1	1
background-color	Elemanlara artalan rengini tanımlar	<i>color-rgb</i> <i>color-hex</i> <i>color-name</i> transparent	4	1	1
background-image	Elemanlara artalan resmi tanımlar	url(URL) none	4	1	1
background-position	Artalan resminin konumunu başlangıç noktalarını belirler	top left top center top right center left center center center right bottom left bottom center bottom right <i>x% y%</i> <i>xpos ypos</i>	4	1	1
background-repeat	Artalan resminin yenileme durumunu tanımlar	repeat repeat-x repeat-y	4	1	1

		no-repeat			
--	--	-----------	--	--	--

Kenar çizgisi (Border) özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
Border	Tüm kenar çizgisi özelliklerinin kısaltmasıdır	<i>border-width</i> <i>border-style</i> <i>border-color</i>	4	1	1
border-bottom	Alt kenar çizgisi özelliklerinin kısaltmasıdır	<i>border-bottom-width</i> <i>border-style</i> <i>border-color</i>	4	1	1
border-bottom-color	Alt kenar çizgisi rengini tanımlar	<i>border-color</i>	4	1	2
border-bottom-style	Alt kenar çizgisinin biçimini tanımlar	<i>border-style</i>	4	1	2
border-bottom-width	Alt kenar çizgisinin kalınlığını tanımlar	thin medium thick <i>length</i>	4	1	1
border-color	Tüm kenar çizgilerinin rengini tanımlar. Bir veya birden fazla kenara tanımlama yapılabilir.	<i>color</i>	4	1	1
border-left	Sol kenar çizgisi özelliklerinin kısaltmasıdır	<i>border-left-width</i> <i>border-style</i> <i>border-color</i>	4	1	1
border-left-color	Sol kenar çizgisi rengini tanımlar	<i>border-color</i>	4	1	2
border-left-style	Sol kenar çizgisinin biçimini tanımlar	<i>border-style</i>	4	1	2
border-left-width	Sol kenar çizgisinin kalınlığını tanımlar	thin medium thick <i>length</i>	4	1	1
border-right	Sağ kenar çizgisi özelliklerinin kısaltmasıdır	<i>border-right-width</i> <i>border-style</i> <i>border-color</i>	4	1	1
border-right-color	Sağ kenar çizgisi rengini tanımlar	<i>border-color</i>	4	1	2

border-right-style	Sol kenar çizgisinin biçimini tanımlar	<i>border-style</i>	4	1	2
border-right-width	Sol kenar çizgisinin kalınlığını tanımlar	thin medium thick <i>length</i>	4	1	1
border-style	Tüm kenar çizgilerinin biçimini tanımlar. Bir veya birden fazla kenara tanımlama yapılabilir.	none hidden dotted dashed solid double groove ridge inset outset	4	1	1
border-top	Üst kenar çizgisi özelliklerinin kısaltmasıdır	<i>border-top-width</i> <i>border-style</i> <i>border-color</i>	4	1	1
border-top-color	Üst kenar çizgisi rengini tanımlar	<i>border-color</i>	4	1	2
border-top-style	Üst kenar çizgisinin biçimini tanımlar	<i>border-style</i>	4	1	2
border-top-width	Üst kenar çizgisinin kalınlığını tanımlar	thin medium thick <i>length</i>	4	1	1
border-width	Tüm kenar çizgilerinin kalınlıklarını tanımlar. Bir veya birden fazla kenara tanımlama yapılabilir.	thin medium thick <i>length</i>	4	1	1

Tasnif özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
clear	Floating uygulanmayan elemanın köşe tanımını yapar.	left right both none	4	1	1
cursor	Fare imlecinin nasıl	<i>url</i>	4	1	2

	görüneceğini belirler.	auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help			
display	Elamanın nasıl görüneceğinin belirler.	none inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption	4	1	1
float	Elemanın(resim veya metin) diğer elemanlara göre konumunu belirler.	left right none	4	1	1
position	Elemanın konumunu belirler.	static relative absolute fixed	4	1	2
visibility	Elemanın görünür veya gizli olacağını belirler.	visible hidden collapse	4	1	2

Boyut tanımlama özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
height	Eleman yüksekliğini tanımlar.	auto <i>length</i> %	4	1	1
line-height	Satırlar arası mesafeyi tanımlar.	normal <i>number</i> <i>length</i> %	4	1	1
max-height	Eleman azami yüksekliğini tanımlar.	none <i>length</i> %	-	1	2
max-width	Eleman azami genişliğini tanımlar.	none <i>length</i> %	-	1	2
min-height	Eleman asgari yüksekliğini tanımlar.	<i>length</i> %	-	1	2
min-width	Eleman asgari genişliğini tanımlar.	<i>length</i> %	-	1	2
width	Eleman genişliğini tanımlar.	auto % <i>length</i>	4	1	1

Yazıtipi (Font) özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
font	Tüm yazıtipi özelliklerinin bir kısa yoludur.	<i>font-style</i> <i>font-variant</i> <i>font-weight</i> <i>font-size/line-height</i> <i>font-family</i> caption icon menu message-box small-caption status-bar	4	1	1
font-family	Yazıtipi ailesini veya ailelerini ve ait olduğu grubunu belirler	<i>family-name</i> <i>generic-family</i>	3	1	1

font-size	Yazıtipi boyunu belirler	xx-small x-small small medium large x-large xx-large smaller larger <i>length</i> %	3	1	1
font-style	Yazıtipi biçimini belirler	normal italic oblique	4	1	1
font-variant	Metnin küçük-büyük harf ile gösterilmesini belirler.	normal small-caps	4	1	1
font-weight	Yazıtipi kalınlığını belirler	normal bold bolder lighter 100 200 300 400 500 600 700 800 900	4	1	1

İçerik yönetim özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
content	:before ve :after sözde sınıfları ile kullanılan içerik kısmını yönetir.	<i>string</i> <i>url</i> <i>counter(name)</i> <i>counter(name, list-style-type)</i> <i>counters(name, string)</i> <i>counters(name, string, list-style-type)</i>		1	2

		attr(X) open-quote close-quote no-open-quote no-close-quote			
counter-increment	Seçicinin her durumu için sayaç değerini düzenler	none <i>identifier number</i>			2
counter-reset	Seçicinin her durumu için sayaç değerini belirler	none <i>identifier number</i>			2
quotes	Tırnak işaretlerini belirler.	none <i>string string</i>	-	1	2

Liste (List) özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
list-style	Liste özelliklerinin bir kısa yoludur.	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i>	4	1	1
list-style-image	list-style işaretinin yerine resim koymak için kullanılır.	none <i>url</i>	4	1	1
list-style-position	list-item işaretinin yerini belirler.	inside outside	4	1	1
list-style-type	list-item işaretinin tipini belirler	none disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-alpha upper-alpha lower-greek lower-latin upper-latin hebrew armenian georgian cjk-ideographic hiragana	4	1	1

		katakana hiragana-iroha katakana-iroha			
marker-offset		auto <i>length</i>		1	2

Kenar dış boşluğu (Margin) özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
Margin	Kenar dış boşluğu özelliklerinin kısaltmasıdır	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>	4	1	1
margin-bottom	Elemanın alt kenar dış boşluğunu tanımlar	auto <i>length</i> %	4	1	1
margin-left	Elemanın sol kenar dış boşluğunu tanımlar	auto <i>length</i> %	3	1	1
margin-right	Elemanın sağ kenar dış boşluğunu tanımlar	auto <i>length</i> %	3	1	1
margin-top	Elemanın üst kenar dış boşluğunu tanımlar	auto <i>length</i> %	3	1	1

Dış hat çizgisi (Outlines) özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
Outline	Dış hat çizgileri özelliklerinin kısaltmasıdır	<i>outline-color</i> <i>outline-style</i> <i>outline-width</i>	-	1.5	2
outline-color	Elemanı saran dış hat çizgilerinin rengini tanımlar	<i>color</i> invert	-	1.5	2
outline-style	Elemanı saran dış hat çizgilerinin biçimini tanımlar	none dotted dashed solid double groove ridge	-	1.5	2

		inset outset			
outline-width	Elemanı saran dış hat çizgilerinin kalınlığını tanımlar	thin medium thick <i>length</i>	-	1.5	2

Kenar iç boşluğu (Padding) özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
Padding	Kenar iç boşluğu özelliklerinin kısaltmasıdır	<i>padding-top</i> <i>padding-right</i> <i>padding-bottom</i> <i>padding-left</i>	4	1	1
padding-bottom	Elemanın alt kenar iç boşluğunu tanımlar	<i>length</i> %	4	1	1
padding-left	Elemanın sol kenar iç boşluğunu tanımlar	<i>length</i> %	4	1	1
padding-right	Elemanın sağ kenar iç boşluğunu tanımlar	<i>length</i> %	4	1	1
padding-top	Elemanın üst kenar iç boşluğunu tanımlar	<i>length</i> %	4	1	1

Konumlandırma (Positioning) özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
bottom	Bir elemanın üst elemanının altına göre ne kadar aşağıda ya da yukarıda olacağını belirler.	auto % <i>length</i>	5	1	2
clip	Elemanın sınırlarını belirlemek için kullanılır.	<i>shape</i> auto	4	1	2
left	Bir elemanın üst elemanın soluna göre ne kadar solda ya da sağda olacağını belirler.	auto % <i>length</i>	4	1	2
overflow	Eleman içeriğinin taşma durumunda	visible hidden	4	1	2

	nasıl davranacağını belirler.	scroll auto			
position	Elemanın konumunu belirler.	static relative absolute fixed	4	1	2
right	Bir elemanın üst elemanın sağına göre ne kadar solda ya da sağda olacağını belirler.	auto % <i>length</i>	5	1	2
top	Bir elemanın üst elemanın üstüne göre ne kadar aşağıda ya da yukarıda olacağını belirler.	auto % <i>length</i>	4	1	2
vertical-align	Elemanın dikey hizalamasını tanımlar.	baseline sub super top text-top middle bottom text-bottom <i>length</i> %	4	1	1
z-index	Elemanın z-eksenindeki yerini tanımlar	auto <i>number</i>	4	1	2

Tablo (Table) özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
border-collapse	Tablo hücrelerinin etrafındaki kenarlık ile olan mesafeyi azaltmamızı veya açmamızı sağlar.	collapse separate	5	1	2
border-spacing	Tablo hücrelerinin kenar çizgileri arasındaki mesafeyi ayarlar	<i>length length</i>	5	1	2

caption-side	Tablo başlığının tarafını belirler.	top bottom left right	5	1	2
empty-cells	Tablodaki boş hücrelerin gösterilip gösterilmeyeceğini tanımlar	show hide	5	1	2
table-layout	Tablo hücrelerinin, satırlarını ve kolonlarını gösterimini ayarlar	auto fixed	5	1	2

Metin (Text) özellikleri

Özellik	Açıklaması	Değerleri	IE	F	W3C
color	Metinlerin renklerini tanımlar.	<i>color</i>	3	1	1
direction	Metin yönünü tanımlar	ltr rtl	6	1	2
line-height	Satırlar arası mesafeyi tanımlar	normal <i>number</i> <i>length</i> %	4	1	1
letter-spacing	Karakterler arası boşluğu arttırmak veya azaltmamızı sağlar	normal <i>length</i>	4	1	1
text-align	Eleman içindeki metinleri hizalamamızı sağlar	left right center justify	4	1	1
text-decoration	Metinleri süslememizi sağlar	none underline overline line-through blink	4	1	1
text-indent	Eleman içindeki metnin ilk satırının girintisini tanımlar	<i>length</i> %	4	1	1
text-shadow	Metinlere gölge vermeye yarar	none <i>color</i>			

		<i>length</i>			
text-transform	Metni büyük-küçük harf çevirmek için kullanılır.	none capitalize uppercase lowercase	4	1	1
unicode-bidi	Çok yönlülük algoritmasını ayarlamamızı sağlar	normal embed bidi-override	5		2
white-space	Elemanların boşluklarının nasıl işlem göreceğini belirler.	normal pre nowrap	5	1	1
word-spacing	Sözcükler arasındaki boşluk değerini belirler.	normal <i>length</i>	6	1	1

Sözde sınıflar (Pseudo-classes)

Sözde sınıf	Amacı	IE	F	W3C
:active	Aktif elemanlara stil tanımlamamızı sağlar.	4	1	1
:focus	Odaklandığımız elemanlara stil tanımlamamızı sağlar.	-	1.5	2
:hover	Fare imlecinin üzerine gittiği elemanlara stil tanımlamamızı sağlar.	4	1	1
:link	Ziyaret etmediğimiz bağlantılara stil tanımlamamızı sağlar.	3	1	1
:visited	Ziyaret ettiğimiz bağlantılara stil tanımlamamızı sağlar.	3	1	1
:first-child	Belirtilen elemanın ilk Çocuk Elemanına stil atamak için kullanılır.	-	1	2
:lang	İçerikteki bir elemanı farklı bir dilde yazmamızı sağlar.	-	1	2

Sözde elemanlar (Pseudo-elements)

Sözde-eleman	Amacı	IE	F	W3C
:first-letter	Metnin ilk harfine stil tanımlamamızı sağlar.	5	1	1
:first-line	Metnin ilk satırına stil tanımlamamızı sağlar.	5	1	1

:before	Eleman öncesine bazı metinleri eklememizi sağlar.		1.5	2
:after	Eleman sonrasına bazı metinleri eklememizi sağlar.		1.5	2

Ek – B Kaynakça

- <http://www.w3schools.com/css/default.asp>
- <http://www.belgeler.org/recs/css2/index.html#index-toc>
- <http://reference.sitepoint.com/css>
- <http://www.cheapostdir.com/reference/css21/cover.html#minitoc>
- <http://www.blooberry.com/>
- <http://www.456bereastreet.com/>
- <http://css.maxdesign.com.au/>
- <http://westciv.com/>
- <http://www.w3.org/TR/CSS21/visufx.html>
- <http://www.seifi.org/css/creating-thumbnails-using-the-css-clip-property.html>
- http://www.ibloomstudios.com/articles/misunderstood_css_clip/
- http://www.w3schools.com/css/pr_pos_clip.asp
- <http://www.w3.org/TR/REC-CSS2/tables.html>
- http://www.w3schools.com/css/css_table.asp
- <http://www.blooberry.com/indexdot/css/propindex/table.htm>
- <http://www.simplebits.com/>
- <http://www.devarticles.com>
- <http://www.w3.org/TR/REC-CSS1#inheritance>
- <http://www.adobe.com/>
- <http://www.creativepro.com/>
- <http://meyerweb.com/>
- <http://css-discuss.incutio.com/?page=BoxModelHack>
- <http://www.createwebmagic.com/css101/lesson2/?inc=part3>
- [CSS Hacks and Filters : Making Cascading Stylesheets Work](#)
- [CSS Mastery: Advanced Web Standards Solutions](#)
- [Muhammed Sevim](#)
- [Web Standards Solutions: The Markup and Style Handbook](#)
- <http://home.tampabay.rr.com/bmerkey/examples/fake-position-fixed.html>
- <http://tagsoup.com/cookbook/css/fixed/>
- <http://annevankesteren.nl/test/examples/ie/position-fixed.html>
- <http://annevankesteren.nl/2005/01/position-fixed-in-ie>
- <http://annevankesteren.nl/2004/07/fixed-positioning>
- <http://www.howtocreate.co.uk/fixedPosition.html>
- <http://www.cssplay.co.uk/layouts/fixed.html>
- <http://www.cssplay.co.uk/layouts/body2.html>
- <http://www.cssplay.co.uk/layouts/fixit.html>
- http://www.gunlaug.no/contents/wd_additions_15.html
- <http://www.gunesintamicinde.com/ie-6-ve-position-fixed-problemi/>
- <http://www.search-this.com/2007/08/15/give-me-some-zzzzs/>
- <http://www.brainjar.com/css/positioning/default5.asp>
- <http://joshuaink2006.johnoxton.co.uk/blog/82/flash-content-and-z-index>

- <http://www.blooberry.com/indexdot/css/properties/position/zindex.htm>
- http://www.westciv.com/style_master/academy/css_tutorial/properties/page_layout.html#z-index
- http://developer.mozilla.org/en/docs/Understanding_CSS_z-index
- <http://css-discuss.incutio.com/?page=OverlappingAndZIndex>
- <http://csscreator.com/blog/z-index>
- <http://www.fatihhayrioglu.com/?p=151>
- <http://www.ahmetblog.net/>
- <http://www.richard-york.com/books.html>

Ek – C Sözlük

A

Absolute: Mutlak
Align: Hizalama
Attribute Selectors: Özellik Seçicileri

B

background: artalan –images: artalan resmi, -color: artalan rengi
background-attachment: Artalan resim bağılılığı
background-color: Artalan rengi
background-image: Artalan resmi
background-position: Artalan resim konumu
background-repeat: Artalan resim yenileme
Belge Nesne Modeli: DOM
Block-level: Blok seviyesi
border: Kenar çizgisi
border-color: Kenar çizgisi rengi
border-collapse: Hücre Boşluklarını ayarlama
boder-style: Kenar çizgisi biçemi
border-spacing: Hücre Dışı boşluklarını ayarlama
border-width: Kenar çizgisi genişliği
bottom: alt
Box model: Kutu modeli
Button: Düğme

C

Cache: önbellek
Cascading Style Sheets: Stil Şablonları
caption-side: Başlık Konumu Özelliği
Class Selector: Sınıf Seçici
Check box: onay kutusu
clip: Kırpma
Color: Renk
cursor: Fare imleci

D

Dash: tire

Declaration Block: Bildirim Bloğu
Default: Başlangıç değeri
Descendant Selectors: Torun Seçicileri
Direct Child Selectors: Çocuk Seçicileri
direction: Metin yazma yönü
dotted: noktalı

E

empty-cells: Boş hücre ayarları

F

Floating: yüzen
font: Yazıtipi
font-family: Yazıtipi ailesi
font-size: Yazıtipi boyutu
font-style: Yazıtipi stili
font-variant: Yazıtipi farklılıkları
font-weight: Yazıtipi kalınlığı

G

Grouping: Gruplama

H

Height: Yükseklik

I

Id Selector: Tekil Seçici
Italic: Eğik
Image: Resim
Inheritance: Kalıtsallık
Inline: Satır içi

L

left: sol
letter-spacing: Harfler arası mesafe
line-height: Satırlar arası yükseklik

Link: Bağ
list: Liste
list-style: Liste stili
list-style-image: Liste imleyici resmi
list-style-position: Liste imleyici konumu
list-style-type: Liste imleyici tipi

M

margin: Kenar dış boşluğu
margin-bottom: Alt kenar dış boşluğu
margin-left: Sol kenar dış boşluğu
margin-right: Sağ kenar dış boşluğu
margin-top: Üst kenar dış boşluğu
max-height: Azami yükseklik
max-width: Azami genişlik
min-height: Asgari yükseklik
min-width: Asgari genişlik
Mouse: Fare
Multiple Classes: Çoklu sınıflar

N

NameSpace: Aduzayı
Next Sibling Selector: Bitişik Kardeş Seçiciler
Nonreplaced element: Yerleştirilmeyen elemanlar
Normal flow: Normal akış

O

Outline: Dış hat çizgisi
overflow: Taşma

P

padding: Kenar iç boşluğu
padding-top: Üst kenar içi boşluğu
padding-right: Sağ kenar içi boşluğu
padding-bottom: Alt kenar içi boşluğu
padding-left: Sol kenar içi boşluğu
Positioning: Konumlandırma
position:absolute: Mutlak konumlandırma
position:fixed: Sabit Konumlandırma

position:relative: Göreceli konumlandırma
Print: yazdırmak
Printer: yazıcı
Print out: çıktı
Property: Özellik
Pseudo-Classes: Sözde sınıflar
Pseudo-Elements: Sözde elemanlar

R

Relative: Göreceli
Replaced element: Yerleştirilen elemanlar
right: sağ
Root element: Kök Eleman

S

Script: Betik
Scrollbar: Kaydırma çubuğu
Selector: Seçiciler
Simple Attribute Selection: Basit Özellik Seçicisi
Space: boşluk
Specificity: Etkinlik
System fonts: Sistem yazıtipi
System colors: Sistem renkleri

T

table: Tablo
table-layout: Tablo Planı özelliği
Text: Metin
text-align: Metin hizalama
text-decoration: Metin süsleme
text-indent: Metin Girintileme
text-shadow: Metin gölgeleme
text-transform: Harf büyüklüğü
top: üst

U

Universal Selector: Evrensel Seçiciler
User Interface: Kullanıcı arayüzü

V

Value: Değer

vertical-align: Dikey hizalama

visibility: Görünürlük

W

Web coder: web kodlamacısı

white-space: Elemanlar arası boşluk

Width: Genişlik

word-spacing: Kelimeler arası mesafe

Z

z-axis: z-ekseni

İndex Tablosu

#

#RGB, 16
#RRGGBB, 16

:

:active, 47
:focus, 46
:hover, 47
:lang sözde sınıfı, 51
:link, 44
:visited, 45

@

@import, 6

A

after, 56
alt, 189
Alt kenar dış boşluğu (margin-bottom), 138
Alt kenar içi boşluğu (padding-bottom), 149
Anahtar sözcükler, 14
Aptana, 4
armenian, 112
Artalan (background), 109
Artalan (Background), 99
Artalan rengi (background-color), 99
Artalan resim bağılılığı (background-attachment), 105
Artalan resim yenileme (background-repeat), 103
Artalan resmi (background-image), 101
Artalan resmi konumu (background-position), 106
Asgari genişlik, 159
auto, 216
Azami genişlik, 161

B

Basamaklı Yapı, 59
Basit Özellik Seçicisi, 36
Başlık Konumu Özelliği (table-layout), 204
BEdit, 4
before, 56
Biçimlendirme kısmı, 1
Bildirim Bloğu(Declaration Block)., 10
Bitişik Kardeş Seçiciler(Next Sibling Selector), 31
blink, 93
block, 202
Blok seviyesinden elemanlar(Block-level element), 120

border-bottom-color, 130
border-bottom-style, 123
border-bottom-width, 127
border-collapse, 208
border-left-color, 130
border-left-style, 123
border-left-width, 127
border-right-color, 130
border-right-style, 123
border-right-width, 127
border-spacing, 210
border-top-color, 130
border-top-style, 123
border-top-width, 127
Boş hücre ayarları (empty-cells), 212
Boşlukların korunması (white-space), 94
bottom, 189
Boyut Tanımlama, 156

C

capitalize, 89
cjk-ideographic, 112
clear, 186
clip, 174
cm, 15
CoffeeCup HTML Editor, 4
Crimson Editor, 4
crosshair, 215
CSS Yorumları, 17
CSS'in Avantajları, 1
CSS3.0, 1
CSSEdit, 4
cursor, 215

Ç

Çocuk Seçicileri(Direct Child Selectors), 29
Çok yönlülük algoritması (unicode-bidi), 97
Çoklu sınıflar, 21

D

decimal, 111
decimal-leading-zero, 111
default, 215
Değer(Value)., 10
Dış hat çizgisi (outline), 219
Dış hat çizgisi biçemi (outline-style), 217
Dış hat çizgisi genişliği (outline-width), 218
Dış hat çizgisi özellikleri(outline), 217
Dış hat çizgisi rengi (outline-color), 219
Dikey hizalama (vertical-align), 81
Dikkate Değer Özellik Seçicisi, 41
Dinamik kısım, 1
display, 202
Dizgeler(Strings), 17
Doctype, 7

DOCTYPE Geçişi(DOCTYPE switching), 8

E

eğik(italiktik), 72
 em, 15
 empty-cells, 212
 e-resize, 216
 e-texteditor, 4
 Etkinlikleri(Specificity), 59
 Evrensel Seçiciler(Universal Selector), 33
 ex, 15
 expression(), 163

F

Fare imleci özellikleri tanımlama (cursor), 215
 first:child:, 49
 first-letter (ilk harf), 54
 first-line (ilk satır), 55
 float, 182
 float:left, 185
 float:right, 184

G

garip mod, 7
 Genişlik (width), 156
 georgian, 112
 Göreceli Konumlandırma(position:relative), 189
 Göreceli Uzunluklar, 15
 Görsel Sonuç Özellikleri, 166
 Görünürlük (visibility), 177
 Graplama, 12

H

Harf Büyüklüğü (text-transform), 89
 Harfler arası mesafe (letter-spacing), 87
 Harici CSS Dosyası, 5
 hebrew, 112
 height, 157
 help, 216
 hiragana, 112
 hiragana-iroha, 112
 Hücre Boşluklarını ayarlama (border-collapse), 208
 Hücre dışı boşluklarını ayarlama (border-spacing), 210

İ

İçerik Alanı, 121
 İçerik kısmı, 1
 İlk Çocuk Elemanını seçmek, 49
 in, 15
 inherit (kalıtsallık)değeri, 63
 inline, 202

K

Kalıtsallık, 59
 Kalıtsallık(Inheritance), 61
 katakana, 112
 katakana-iroha, 112
 Kenar Çizgisi (border), 132
 Kenar çizgisi biçemi (border-style), 121
 Kenar çizgisi genişliği (border-width), 125
 Kenar Çizgisi Rengi (border-color), 129
 Kenar dış boşluğu (margin), 142
 Kenar Dış Boşluğu(Margin), 135
 Kenar iç boşluğu (Padding), 146
 Kenar içi boşluğu (padding), 153
 Kenarlık Çizgisi(Border), 121
 Kesin Uzunluklar, 15
 Kırpma (clip), 174
 Kısmi Özellik Değeri Seçicisi, 39
 Kod içinde (In-line), 5
 Konumlandırma, 182
 Konumlandırma özelliği (position), 188
 Konumlandırma Özellikleri, 188
 Kök Eleman(Root element), 120
 Kullanıcı Arayüzü, 215
 Kutu Modeli, 119

L

left, 189
 Liste biçemi (list-style), 117
 Liste imleyici konumu (list-style-position), 115
 Liste imleyici resmi (list-style-image), 114
 Liste imleyici tipi (list-style-type), 111
 Listeleme, 111
 list-item, 202
 lower-alpha, 112
 lowercase, 89
 lower-greek, 112
 lower-latin, 112
 lower-roman, 111

M

Macromedia Dreamweaver, 4
 max-height, 161
 max-width, 161
 medium [orta], 125
 Metin Girintileme (text-indent), 78
 Metin gölgelendirme (text-shadow), 93
 Metin hizalama (text-align), 80
 Metin yazma yönü(direction), 96
 Metin(Text), 78
 Microsoft FrontPage, 4
 Microsoft® Expression, 4
 min-height, 159
 min-width, 159
 mm, 15
 move, 216
 Mutlak(Absolute) Konumlandırma, 190

N

ne-resize, 216
 none, 202
 no-repeat, 103
 Normal Akış, 188
 Normal akış(Normal flow), 120
 NotePad, 3
 Notepad 2, 4
 Notepad++, 4
 nowrap, 94
 n-resize, 216
 nw-resize, 216

O

outline, 217, 219
 outline-color, 219
 outline-style, 217
 outline-width, 218
 overflow, 166
 overflow-x, 173
 overflow-y, 173

Ö

Özellik Değeri Seçicisi, 37
 Özellik Seçicileri(Attribute Selectors), 35
 Özellik(Property), 10

P

padding, 153
 padding-bottom, 150
 padding-left, 152
 pc, 15
 position: absolute, 191
 position: relative, 190
 pre, 94
 progress, 216
 pt, 15
 px, 15

R

Renk (color), 65
 Renk Değerleri, 16
 Renk İsimleri, 16
 repeat, 103
 repeat-x, 103
 repeat-y, 103
 RGB Renk Değerleri, 16
 rgb(R%,G%,B%), 16
 RGBA (RGB with Alpha channel), 16
 right, 189

S

Sabit Konumlandırma (position:fixed), 192
 sağ, 189
 Sağ kenar dış boşluğu (margin-right), 137
 Sağ kenar içi boşluğu(padding-right), 148

Satırın altını, üstünü, ortasını çizme ve yanıp
 sönme (text-decoration), 91
 Satır içi seviyeden elemanlar(Inline element),
 120
 Satırlar arası yükseklik (line-height), 83
 Seçiciler, 19
 Seçiciler(Selector), 10
 se-resize, 216
 Sınıf Seçicisi(Class Selector), 19
 Sistem yazıtipi, 220
 small-caps, 74
 sol, 189
 Sol kenar dış boşluğu (margin-left), 140
 Sol kenar içi boşluğu (padding-left), 151
 Sosyal yazıtipi aileleri, 67
 Sözcükler arası mesafe (word-spacing), 85
 Sözde Dinamik Sınıf Seçicileri, 46
 Sözde eleman seçicileri, 54
 Sözde Sınıf Seçicileri, 44
 Sözde sınıfları birleştirmek, 52
 Sözde(Pseudo) elemanı, 44
 Sözde(Pseudo) sınıf, 44
 s-resize, 216
 Standart mod, 7
 Stil Şablonları, 1
 Style Master, 4
 sw-resize, 216

T

Tablo, 204
 Tablo planı özelliği (caption-side), 207
 Taşma (Overflow), 166
 Tekil Seçiciler(Id Selector), 23
 text, 216
 TextEdit, 3
 thick [kalın], 126
 thin [ince], 125
 top, 189
 TopStyle, 4
 Torun Seçicileri (Descendant Selectors), 28

U

upper-alpha, 112
 uppercase, 89
 upper-latin, 112
 upper-roman, 111
 URL Değerleri, 16
 Uzunluk Değerleri, 15

Ü

Üst, 189
 Üst Kenar dış boşluğu (margin-top), 135
 Üst kenar içi boşluğu(padding-top), 146

V

visibility, 177
 visiblity: collapse, 181

W

wait, 216
width, 156
WordPad, 3
w-resize, 216

Y

Yazıtipi (font), 75
Yazıtipi ailesi (font-family), 67
Yazıtipi biçemi (font-style), 72
Yazıtipi boyutu (font-size), 68

Yazıtipi farklılıkları (font-variant), 74
Yazıtipi kalınlığı (font-weight), 71
Yazıtipi(Font), 65
Yerleştirilen eleman(Replaced element), 120
Yerleştirilmeyen elemanlar(Nonreplaced element), 120
Yükseklik (height), 157
Yüzde Değerleri, 15
Yüzen kutu, 182

Z

z-index, 196