

CSS
(Cascading Style Sheets)

30 Ocak 2002

Giriş

CSS Nedir?

Sanaldoku programlamada önemli bir yere sahip CSS kavramı, 1995 yılında W3C (World Wide Web Consortium) tarafından geliştirilmiştir. O yıllarda çok hızlı bir şekilde gelişen HTML kullanımına çok büyük kolaylıklar getirmiştir. CSS ile HTML'de yapılması zahmet gerektiren işler, daha basit şekilde yapılabilir.

HTML imleri genel olarak dökümanın tarayıcı üzerinde nasıl görüneceğini belirler. Renkleri, yazı karakterlerini, resimleri ve daha bir çok veriyi tarayıcı üzerine istediğiniz şekilde yerleştirebilirsiniz. HTML'in ilk zamanlarında yetersiz olan buyrukları bu iş için giderek zenginleştirilmiştir. Sadece resim veya yazı değil, ses ve görüntü dosyalarını tarayıcı içerisinde görüntülemek mümkün olmuştur. Ancak 90'lı yıllarda gelişimini hızlandıran HTML beraberinde bazı uyumsuzluk hatalarını getirmiştir. Bunun üzerinde HTML'i belli standartlara oturtmak için W3C tarafından bazı çalışmalar başlatılmıştır. Buna göre W3C, "recommation" denilen tavsiye niteleği içeren dökümanlar oluşturmuştur. Bu dökümanlar <http://www.w3c.org> adresinde yayımlandıktan sonra tarayıcı üreticileri tarafından hayata geçirilmeye başlanmıştır. Böylece dünya çapında geçerli olan bir HTML standardı oluşmaya başlamıştır.

Bu bağlamda HTML kullanımını kolaylaştıran ve etkin bir şekilde kullanılmasını durumunda belli standartları yakalamayı sağlayan CSS gündeme gelmiştir. Bununla ilgili yine W3C tarafından bazı tavsiyeler yayımlanmıştır. 1996 yılında CSS1 oluşturulmuş, 1998 yılında ise CSS2, CSS1 üzerine oturtulmuştur.

Temel Özellikler

Zengin İçerik

HTML içerisinde kolaylıkla yapılamayacak şeyleri CSS kullanarak çok rahat bir şekilde yapabilirsiniz. Mesela, sanaldoku sayfanızda yer alan bütün H1

başlıklarının arka plan rengini, font büyüklüğünü ve rengini ayarlamak istiyorsunuz. Bunu klasik HTML kuralları ile yapmaya kalkarsanız bir tablo oluşturmak zorundasınız. Ayrıca bütün sayfalarda bu değişiklikleri yapmalısınız. Ama CSS kullanarak tek bir satırda bunu halledebilirsiniz.

```
H1 {color: black; background: yellow; font: italic}
```

Kullanma Kolaylığı

CSS genel itibariyle kullanımı kolay bir dildir. HTML'e göre daha kolay oluşturulabilecek bir yapıya sahiptir. HTML'de kullanılan < ve > işaretlerini CSS içerisinde kullanılmaz. Bu yüzden yazma kolaylığı sağlanır.

Esneklik

Yapılan bir değişiklik ile birden fazla web sayfası ile oynayabilirsiniz. Sadece bir adet dosya ile yüzlerce web sayfasının görünümünü değiştirebilirsiniz.

Cascading

Cascading kavramı, CSS ile yapılmış tanımlamaların ek bir iş yapmaya gerek kalmadan dökümanın bütün elemanlarını etkilemesini sağlar. Bu özellik ile

```
H1 {color: black; background: yellow; font: italic}
```

ile yaptığımız bir tanım, h1 türündeki bütün başlıkları etkiler.

Uygun Dosya Boyutları

CSS kullanarak HTML dökümanlarının boyutlarını makul seviyelere çekebilirsiniz.

CSS ve HTML

HTML dosyalarına CSS dosyalarını uygulayabilmek için CSS dosyasını HTML içerisinden çağırmanız gerekmektedir. Bu iş için <LINK>im'i kullanılır.

```
<LINK rel="stylesheet" type="text/css" href="stil.css">
```

Bu tür "stylesheet" dosyalarına "external style sheets" denir. "Stylesheet" dosyalarını ayrıca html dökümanı içerisine gömebilirsiniz. Ancak bu yöntem genellikle tercih edilmez. Çünkü CSS kullanmanın nedenlerinden biri olan esnekliği zedeler. Bu iş için <STYLE> adlı im kullanılır.

```
<STYLE type="text/css">
h1 {color: black}
</STYLE>
```

Bu yöntemi kullanırken bir şeye dikkat etmeniz gerekir. Eğer eski bir tarayıcı kullanıyor iseniz yukarıdaki satırlar tarayıcı ekranında olduğu gibi görünecektir. Çünkü `<STYLE>`im'i HTML'e yeni eklenen bir özelliktir. Bu handikapı çözmek için yukarıdaki kodu aşağıdaki şekliyle değiştirin.

```
<STYLE type="text/css"> <!--  
h1 {color: black}  
--> </STYLE>
```

CSS buyruklarını `<!--ve-->` imleri içerisine alarak eski tarayıcıların bu buyrukları görmemesini sağlarsınız.

CSS Yorumları

CSS içerisine yorum satırları eklemek için C dilindeki yapıya benzer şekilde `/**/` sembollerini kullanabilirsiniz.

```
h1 {color: black} /* Bu kısım dikkate alınmaz */
```


Genel Yapı

Temel Kurallar

Yapı

CSS içerisinde yapılan bütün tanımlar belli bir düzene göre yapılır.

```
h1 {color: black}
```

Yukarıdaki tanımda sol tarafta, kuralın uygulanacağı olan eleman bulunur. Buna “selector” denir. Süslü parantezler içerisine : işareti ile ayrılmış olarak bulunan kısım ise tanım bölümüdür. : işaretinin sol tarafı özelliği belirtir. Sağ tarafta ise değer belirtilir. Buna göre “h1” “selector”, color özellik, black ise değer olarak adlandırılabilir.

Basit Seçiciler

Bir seçici genellikle basit HTML imleridir. H1, BODY v.b. Bunların beraber bir XML dökümanını şekillendirmek için kullanılan CSS dökümanlarında XML içerisinde yer alan imler de bulunabilir.

```
kitap {color: yellow}
```

gibi. Burada kitapımı normal HTML imlerinde bir değildir.

Tanımlar

Biraz önce CSS tanımlarının uygulandığı imlerden bahsettik. Peki tanımlar nasıl yapılıyor. Bunun için : ile ayrılmış bir yapı kullanıyoruz. Sol taraf özelliği, sağ taraf ise değeri gösteriyor. Özelliği ifade etmek için önceden tanımlı bazı anahtar sözcükler kullanılır.

```
body {background: purple}
```

Değer olarak ise düzgün tanımlı bir veya birden fazla sözcük ardarda kullanabilirsiniz.

```
body {font: italic Verdana}
```

Eğer anahtar sözcük yanlış ise yazdığımız buyruk satırı boşa gider. Değer olarak yanlış bir şey yazarsanız ise, satır değil sadece ilgili kısım ihmal edilecektir. Hatta birden fazla değer atanmış ise yanlış olan değerden bir sonraki değer işlenecektir. Çoğu tarayıcı bu gibi hataları tolare edebiliyor. Bu kolaylık ilk başta güzel görünmesine rağmen, programcılara bazı kötü alışkanlıklar kazandırabilir.

Gruplama

Seçicileri Gruplama

CSS tanımları içerisinde yer alan seçiciler gruplar halinde verilebilir.

```
h1,h2,h3 {color: gray}
```

Yukarıdaki örnekte renk özelliği h1, h2 ve h3 imlerinin hepsine birden kazandırılmıştır.

Özellikleri Gruplama

Seçiciler gruplanabildikleri gibi özellikler de gruplar halinde tanımlanabilir.

```
p {color: blue;
background: grey;
font: Verdana; }
```

Herşeyi Gruplama

Bir önceki gruplamalar toplu bir şekilde yapılabilir.

```
h1, h2 {color: yellow;
font: Lucida;
background: blue;}
```


Sınıf ve ID Seçicileri

Sınıf Seçicileri

Sınıf (Class) tanımlarının yapıldığı HTML dökümanlarını CSS içerisinde şekillendirebilmek için kullanılır. Elimizde aşağıdaki gibi bir HTML dökümanı olsun.

```
<h1 class="onemli">Onemli Bir Baslik</h2>
```

Sınıfı “onemli” olan imleri CSS içerisinde tanımlayabilmek için aşağıdaki buyruğu kullanabiliriz.

```
.onemli {color: red}
```

Bu sayede HTML içerisinde herhangi bir im “onemli” olarak nitelendirilirse yazı rengi olarak kırmızı kullanılacaktır. Yukarıdaki tanımın yerine

```
h1.onemli {color: red}
```

buyruğunu kullansaydık sadece h1 imi bu kurala uyacaktır.

ID Seçicileri

ID ile Sınıf birçok yönden birbirine benzerler ancak arada temel birkaç fark vardır. İlk olarak ID tanımlarında #işareti kullanılır. Ayrıca bir HTML dökümanında ID’ler tektir.

```
<h2 ID="onemli">Onemli Bir Baslik</h2>
```

```
#onemli {color: red}
```


Birimler

Renkler

Tanımlı Renkler

İsimleri ile çağrılacak ve yaygın tarayıcıların desteklediği 16 ana renk vardır. Bu 16 renk Windows işletim sisteminden geliyor. Windows'un ilk zamanlarında VGA monitörlerde kullanılabilen renkler bu 16 renk idi. Bu standart 16 renk haricinde bazı tarayıcı'ların tanıdığı renkler de mevcuttur. Mesela "orange" 16 renk içerisinde olmamasına rağmen bazı tarayıcı'larda kullanabilirsiniz.

RGB Kullanarak Renk Tanımı

RGB kelimesi Red, Green ve Blue kelimelerinin baş harflerinde oluşturulmuştur. Kırmızı, Yeşil ve Mavi renkleri karıştırarak bütün renkleri ifade etmek mümkündür gerçeğine dayanır. Bu sistemde hangi renkten ne kadar katkı yapılacağı belirlenir ve buna göre renk oluşturulur.

```
rgb(100%, 100%, 100%)
```

Yukarıdaki örnekte 3 ana renkten 100%'lük katkılar sağlanarak bir renk oluşturuluyor. Bu beyaz renge karşılık gelmektedir. Aşağıdaki örnekte gri rengin tonlarını nasıl elde edebileceğimizi görüyoruz.

```
<h1 class="bir">H1 class="bir" tipinde bir başlık</h1>  
<h1 class="iki">H1 class="iki" tipinde bir başlık</h1>  
<h1 class="uc">H1 class="uc" tipinde bir başlık</h1>  
<h1 class="dort">H1 class="dort" tipinde bir başlık</h1>  
<h1 class="bes">H1 class="bes" tipinde bir başlık</h1>  
<h1 class="alti">H1 class="alti" tipinde bir başlık</h1>
```

```

h1.bir {color: rgb(0%,0%,0%);}
h1.iki {color: rgb(20%,20%,20%);}
h1.uc {color: rgb(40%,40%,40%);}
h1.dort {color: rgb(60%,60%,60%);}
h1.bes {color: rgb(80%,80%,80%);}
h1.alti {color: rgb(100%,100%,100%);}

```

Sayılar Kullanarak Renk Tanımı

Yüzdeler ifade yerine sayı kullanarak da renkleri ifade edebiliriz. Bu sayılar 0 ile 255 rakamları arasında olmak zorundadır.

```
rgb(255,255,255)
```

Yukarıdaki örnekte beyaz renk tanımlanmıştır.

Hexadecimal Kullanarak Renk Tanımı

Bu yöntemde belirli bir renge karşılık gelen hexadecimal sayılar kullanılır.

```
h1 {color: #FFFFFF;}
```

Dikkat edilmesi gereken nokta sayıdan önce #simgesinin kullanılmasıdır.

Uzunluk Birimleri

Mutlak Uzunluk Birimleri

- *Inches (in)*
- *Santimetre (cm)*
- *Milimetre (mm)*
- *Points (pt)*

Bu uzunluk birimleri adı üzerinde mutlak uzunluklar belirlerler. Bu yüzden sanaldoku tasarımında çok kullanışlı değillerdir çünkü bilgisayardan bilgisayara bir çok özellik değişmektedir. Özellikler monitörlerin değişik ebatlarda ve özelliklerde olması bu birimlerin kullanılmasını zorlaştırmaktadır. 14 inch lik bir monitörde güzel uygun görünen bir resim 17 inch'lik monitörde kötü görünebilmektedir.

Bağlı Uzunluk Birimleri

Bu kavramı bir örnekler açıklayalım.

```
h1 {font-size: 24px;}  
h1 {margin-left: 1em}
```

Burada ilk önce h1 için 24 piksellik bir font büyüklüğü öngördük. Sonra soldan içeriye doğru 1em’lik girmesini sağladık. em birimi önceden tanımlanmış font’un büyüklüğüne eşit olacak şekilde belirlenmiştir. Bu durumda h1 soldan içeriye 24 piksel kaydırılacaktır.

Fakat bu aşamada ince bir nokta devreye giriyor. Aşağıdaki örneği inceleyelim.

```
<p>Bu paragraf içerisinde <small>küçük</small>  
bir kelime var</p>
```

```
small {font-size: 0.8em}
```

Normalde small imi içerisinde yer alan “küçük” kelimesi içerisinde bulunduğu cümleden %20 oranında küçük yazılacaktır. tarayıcılar bu işlemi gerçekleştirirken fontun etki ettiği cümledeki “x” harfini referans alırlar. Ne yazık ki her font için “x” harfinin büyüklüğü değişik olacaktır. Bu sebeple “küçük” kelimesi her font için farklı büyüklüklerde gösterilecektir.

Metin İşlemleri

Hizalama

text-indent

Yaygın olarak paragrafların ilk cümlelerin soldan içeriye kaydırılması için kullanılan bir özelliktir.

```
p {text-indent: 1in}
```

text-align

4 tane değeri olabilir; left, right, center ve justify. Kolaylıkla anlaşılacağı şekilde left sola, right sağa yaslama işlemi yapar. center ise yazıyı ortalar. justify ise yazıyı sağ ve sol marjnlere uyacak şekilde ayarlar.

```
h1,h2,h3,h4,h5 {text-align: center}
```

buyruğu ile bütün başlıkların ortalanmasını sağlayabilirsiniz.

white-space

Metnin içerisinde geçen başlıkların tarayıcı üzerinde nasıl görüntüleneceğini belirler. Üç farklı değeri olabilir; pre, nowrap, normal. Normal seçeneği extra boşlukların ihmal edilmesini sağlar. Pre seçeneği boşlukların olduğu gibi görüntülenmesini sağlar. Nowrap ise metnin satır atlamasını engeller. Satır atlamak için farklı buyruklar kullanmak zorunda kalırız.

```
<p style="white-space: nowrap;">Bu paragrafın içerisinde  
"nowrap" seçeneği kullanılmıştır. nowrap metnin boylu boyunca  
devam etmesini sağlar.  
<br>Satırı atlamak için br buyruğunu kullanmak zorundasınız.</p>
```

Yukarıdaki buyruğun tarayıcıda (tarayıcı) elde edeceğimiz görüntüsü aşağıdaki gibi olacaktır.

Bu paragrafın içerisinde "nowrap" seçeği kullanılmıştır. nowrap metnin boylu boyunca devam etmesini sağlar. Satrı atlamak için br buyruğunu kullanmak zorundasınız.

word-spacing

Bu özellik kelimeler arasındaki boşlukları belirlemeye yarar. Örnek verelim:

```
<p style="word-spacing: normal;">Bu paragrafta kelimeler  
arasındaki  
boşluk "normal" olarak tanımlanmıştır.</p>
```

```
<p style="word-spacing: 0.5em;">Bu paragrafta kelimeler  
arasındaki  
boşluk "0.5em" olarak belirlenmiştir.</p>
```

İlk örnekte kelimeler arası boşluk normal olarak bırakılacak ama ikinci örnekte boşluk yüzde 50 kadar artırılmıştır.

text-transform

Dört seçeneği vardır; uppercase, lowercase, capitalize ve none. Uppercase bütün harfleri büyük harfe, lowercase bütün harfleri küçük harflere, capitalize kelimelerin ilk harflerini büyük harflere çevirir. none ise hiçbir şey yapmaz.

Fontlar

Font Aileleri

- Serif
- Sans Serif
- Monospace
- Cursive
- Fantasy

Yukarıdaki isimler genel font ailelerini tanımlarlar. Serif font ailesindeki fontlarda karakterlerin kenarlarında küçük süslü çıkıntılar vardır. Sans Serif fontlarda bu çıkıntılar yoktur.

```
body {font-family: Sans Serif;}
```

Font isimlerini ise aşağıdaki gibi belirtebilirsiniz.

```
body {font-family: Sans Serif, Verdana;}
```

Font kullanımına güzel bir örnek verelim. Diyelim ki sayfaların altına imzımızı atmak istiyoruz. Bunun için bir font belirleyebiliriz.

```
p.imza {font-family: Author99, ScriptTM, cursive;  
text-align: left;}
```

Bu tanımlı yaptıktan sonra HTML içerisinde

```
<p class="imza">Yazar ismi</p>
```

buyruğu verildiğinde “Yazar ismi” tümcesi belirtilen fontta ve sola bitişik olarak yazılacaktır.

font-size

Fontun büyüklüğünü belirtir. xx-small, x-small, small, medium, large, x-large,xx-large seçenekleri vardır.

```
<p style="font-size: xx-small;">Bu paragraf xx-small boyutunda </p>  
<p style="font-size: x-small;">Bu paragraf x-small boyutunda </p>  
<p style="font-size: small;">Bu paragraf small boyutunda </p>  
<p style="font-size: medium;">Bu paragraf medium boyutunda </p>  
<p style="font-size: large;">Bu paragraf large boyutunda </p>  
<p style="font-size: x-large;">Bu paragraf x-large boyutunda </p>  
<p style="font-size: xx-large;">Bu paragraf xx-large boyutunda </p>
```

Renkler ve Arkaplanlar

Renkler

Renk tanımlama işlemini daha önce anlatmıştık. Bir örnekle hatırlayalım.

```
body {color:black;}
A:link {color: #808080;}
A:visited {color: silver;}
A:active {color: #333333;}
```

Metinlerin renkleri ayarlanabildiği gibi metni çevreleyen çizgilerin de renkleri CSS iler belirlenebilir.

```
p.cerceve {color: purple; border-style: solid;
border-color: black;}
```

Renk tanımında dikkat edilecek önemli bir nokta vardır. Bir eleman için renk tanımı yapıldığında o elemanın içerisinde yer alan diğer elemanlarda o renk ile gösterilecektir. Bu özelliğe “inheritance” denir. Mesela

```
body {color: red;}
```

tanımı yapıldığında aksi belirtilmediği sürece HTML dökümanı içerisinde yer alan bütün metinlerin kırmızı olmasını zorlarsınız. <body> iminin doğal alt üyeleri olan <p><h1>gibi imler <body>iminin özelliklerini miras alırlar.

Arkaplan Rengi

Bütün elemanlar için geçerli olmasada çoğu HTML imi için bir arkaplan rengi belirlenebilir.

```
<pre class="alinti">Bu yazi
pre.alinti {background-color: green;}
CSS buyruęu ile biçimlendirilmiştir.</pre>
```

Yukarıdaki örnekte `<pre>`imi ve "alinti"sınıfı ile tanımlanan her metnin arkaplan rengini yeşil olarak tanımlamış bulunmaktayız.

Arkaplan Resmi

Bu özellikte arkaplan rengi gibi bütün elemanlar için uygulanamaz. Bir örnek verelim.

```
body {background-image: url(meg.jpg)}
```

Burada `url`anahtar kelimesi resimlerin yerini belirtmek için kullanılır. Resimlerin yerel olarak makinede bulunması gerekmez. Tam URL (Uniform Resource Locator) kullanarak resim belirtebilirsiniz.

```
body {background-image: url(http://www.w3c.org/background.jpg)}
```

Arkaplan resimleri sadece `<body>`imi için kullanılmak zorunda değildir. Bazı HTML imleri için de kullanabilirsiniz.

```
<p style="background-image: url(gozler.jpg);
background-color: black;">Bu
paragrafın arkaplanı olarak bir resim kullanılmıştır.
Bunun için
class="alinti">style="background-image: url(gozler.jpg);
buyruęu kullanılmıştır.</p>
```

Güzel arkaplan resimleri oluşturmak için CSS'nin sağladığı birkaç özellikten bahsetmekte yarar var. Diyelim ki sanaldoku sayfanızın sol tarafında dikey olarak yerleştirilmiş bir süs istiyorsunuz. bunun için normalde yapmanız gereken şey yüksekliği az ama genişliği fazla olan bir resim dosyayı oluşturmak, sonra da bu resim dosyayı `<body>`imi içerisinde arkaplan resmi olarak göstermenizdir. Sanaldoku tarayıcıları bur resmi dikey olarak uzatacakları için resminiz güzel duracaktır. Ancak oluşturacağınız resmin sadece sol tarafında bir şeyler yer alacak geri kalan büyük kısım boşluktan oluşacaktır.

bu etkiyi yaratmak için CSS'nin öngördüğü metod çok basit. Sol tarafa koymayı istediğiniz resmi normal boyutları ile ele almak ve resmi dikey doğrultuda tekrarlamasını sağlamak. Bunu yapmak için `background-repeat` özelliği kullanılır. 4 değer alabilir; `repeat`, `repeat-x`, `repeat-y` ve `none`. Tahmin ettiğiniz gibi `repeat` değeri resmin dikey ve yatay doğrultularda tekrarlanmasını sağlar. `repeat-x` yatay doğrultuda, `repeat-y` ise dikey doğrultuda aynı işi yapar. `none` ise resmi olduğu gibi bırakır. Bir örnek verelim.

```
body    {background-image: url(button.png);  
         background-repeat: repeat;  
         font-family: Sans Serif, Garamond;}
```

Yukarıda anlatılan özelliklere ek olarak `background-attachment` özelliği bahsedilebilir. İki değeri vardır; `scroll` ve `fixed`. `scroll` seçeneği arkaplanın metinler kaydırıldığında onlarla beraber hareket etmesini sağlar. `fixed` ise arkaplanın sabit kalmasını sağlar.

```
body    {background-image: url(button.png);  
         background-repeat: repeat;  
background-attachment: fixed;  
         font-family: Sans Serif, Garamond;}
```


Örnek

sample.css

```
body {background-image: url(button.png);
      background-repeat: repeat-y;
      background-attachment: fixed;
      font-family: Sans Serif, Garamond;}
```

```
// Tanimli Renkler
p.aqua {color: aqua}
p.black {color: black}
p.blue {color: blue}
p.fuchsia {color: fuchsia}
p.gray {color: gray}
p.green {color: green}
p.lime {color: lime}
p.maroon {color: maroon}
p.navy {color: navy}
p.olive {color: olive}
p.purple {color: purple}
p.red {color: red}
p.silver {color: silver}
p.teal {color: teal}
p.white {color: white}
p.yellow {color: yellow}
```

```
h1.bir {color: rgb(0%,0%,0%);}
h1.iki {color: rgb(20%,20%,20%);}
h1.uc {color: rgb(40%,40%,40%);}
h1.dort {color: rgb(60%,60%,60%);}
h1.bes {color: rgb(80%,80%,80%);}
```

```
h1.alti {color: rgb(100%,100%,100%);}
```

```
p.sol {text-indent: -30px;}
p.sola {text-align: left;}
p.saga {text-align: right;}
p.ortala {text-align: center;}
p.ayarla {text-align: justify;}
```

```
p.imza {font-family: Author99, ScriptTM, cursive;
text-align: left;}
```

```
p.cerceve {color: purple; border-style: solid; border-color: black;}
```

```
pre.alinti {background-color: green;}
```

sample.html

```
<html>
<head >
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
<title>CSS Örnek Sayfası</title>
<link rel="stylesheet" type="text/css" href="sample.css">
</head>

<body>

<h1 class="bir">H1 class="bir" tipinde bir başlık</h1>
<h1 class="iki">H1 class="iki" tipinde bir başlık</h1>
<h1 class="uc">H1 class="uc" tipinde bir başlık</h1>
<h1 class="dort">H1 class="dort" tipinde bir başlık</h1>
<h1 class="bes">H1 class="bes" tipinde bir başlık</h1>
<h1 class="alti">H1 class="alti" tipinde bir başlık</h1>
<p class="aqua">AQUA </p>
<p class="black">BLACK </p>
<p class="blue">BLUE </p>
<p class="fuchsia">FUCHSIA </p>
<p class="gray">GRAY </p>
<p class="green">GREEN </p>
<p class="lime">LIME </p>
<p class="maroon">MAROON </p>
<p class="navy">NAVY </p>
```



```
<p class="olive">OLIVE </p>
<p class="purple">PURPLE </p>
<p class="red">RED </p>
<p class="silver">SILVER </p>
<p class="teal">TEAL </p>
<p class="white">WHITE </p>
<p class="yellow">YELLOW </p>
```

```
<p class="sol">Bu paragrafın ilk cümlesi sol tarafa
doğru 2 inch kadar itilmiştir. Yazının rengi neden mavi? Bunu da
siz bulun. </p>
```

```
<p class="sola">Bu paragraf sola doğru yaslanmıştır.
Bu paragraf sola doğru yaslanmıştır.
Bu paragraf sola doğru yaslanmıştır.
Bu paragraf sola doğru yaslanmıştır.</p>
```

```
<p class="saga">Bu paragraf saga doğru yaslanmıştır.
Bu paragraf saga doğru yaslanmıştır.
Bu paragraf saga doğru yaslanmıştır.
Bu paragraf saga doğru yaslanmıştır.</p>
```

```
<p class="ortala">Bu paragraf ortalanmıştır.
Bu paragraf ortalanmıştır.
Bu paragraf ortalanmıştır.
Bu paragraf ortalanmıştır.</p>
```

```
<p class="ayarla">Bu paragraf sağa ve sola doğru ayarlanmıştır.
Bu paragraf sağa ve sola doğru ayarlanmıştır.
Bu paragraf sağa ve sola doğru ayarlanmıştır.
Bu paragraf sağa ve sola doğru ayarlanmıştır.</p>
```

```
<p style="white-space: nowrap;">Bu paragrafın içerisinde "nowrap" seçeği
kullanılmıştır. nowrap metnin boylu boyunca devam etmesini sağlar.
<br>Satır atlamak için br buyruğunu kullanmak zorundasınız.</p>
```

```
<p style="word-spacing: normal;">Bu paragrafta kelimeler arasındaki
boşluk "normal" olarak tanımlanmıştır.</p>
```

```
<p style="word-spacing: 0.5em;">Bu paragrafta kelimeler arasındaki  
boşluk "0.5em" olarak belirlenmiştir.</p>
```

```
<p style="font-family: Sans Serif;"> Bu paragraf Sans Serif  
font ailesi ile yazılmıştır. </p>
```

```
<p style="font-size: xx-small;">Bu paragraf xx-small boyutunda yazıldı</p>  
<p style="font-size: x-small;">Bu paragraf x-small boyutunda yazıldı</p>  
<p style="font-size: small;">Bu paragraf small boyutunda yazıldı</p>  
<p style="font-size: medium;">Bu paragraf medium boyutunda yazıldı</p>  
<p style="font-size: large;">Bu paragraf large boyutunda yazıldı</p>  
<p style="font-size: x-large;">Bu paragraf x-large boyutunda yazıldı</p>  
<p style="font-size: xx-large;">Bu paragraf xx-large boyutunda yazıldı</p>
```

```
<p class="cerceve">Bu paragraf  
p.cerceve {color: purple; border-style: solid; border-color: black;}  
CSS buyruğu ile biçimlendirilmiştir. </p>
```

```
<p class="imza">Yazar ismi</p>
```

```
<pre class="alinti">Bu yazı  
pre.alinti {background-color: green;}  
CSS buyruğu ile biçimlendirilmiştir.</pre>
```

```
<p style="background-image: url(gozler.jpg); background-color: black;">Bu  
paragrafın arkaplanı olarak bir resim kullanılmıştır.  
Bunun için  
class="alinti">style="background-image: url(gozler.jpg);  
buyruğu kullanılmıştır.</p>
```

```
<p style="background-image: url(button2.png);  
background-repeat: repeat;  
background-attachment: fixed;">  
style="background-image: url(button.png); background-repeat: repeat;  
style="background-image: url(button.png); background-repeat: repeat;  
style="background-image: url(button.png); background-repeat: repeat;  
</p>
```

```
</body>  
</html>
```


Kaynakça

- [1] Cascading Style Sheets, The Definitive Guide, Eric A. Meyer , O'Reilly Assoc., 1999