

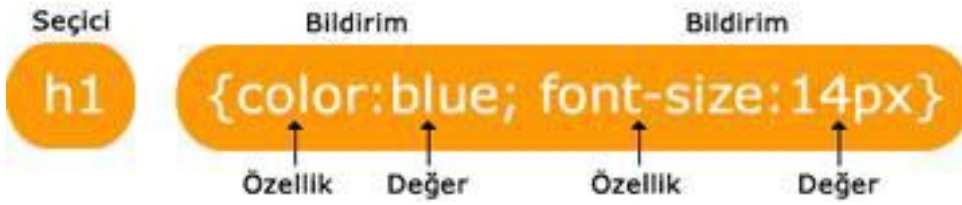
CSS

CSS, açılımı (**C**ascading **S**tyle **S**heet) olan ve dilimize **Stil Şablonları** olarak yerleşen bir işaretleme dilidir.

Stil Şablonu HTML öğelerine (yazı, paragraf, kenar çizgisi, resim, bağlantı...) stil vermek amacıyla kullanılır.

HTML/XHTML etiket dillerinin sayfa tasarımında yetersiz kalması nedeniyle “World Wide Web Consortium” (W3C, Dünya Çapında Ağ Birliği) tarafından oluşturulmuştur; içerik kısmı (HTML) ile biçimlendirme kısmını birbirinden ayırarak yüzlerce sayfayı tek bir dosya ile biçimlendirme olanağı tanımaktadır. Bu, web sayfalarımıza esneklik bize ise hız kazandırır.

1. CSS’nin Yapısı



CSS’nin söz dizimi yukarıda görüldüğü gibi “*Seçici*” ve “*Bildirim*” olarak iki ana bölümden oluşur. Bildirim ise kendi içinde *özellik* ve *değer* olarak iki kısma ayrılmaktadır.

1.1. Sınıf (Class) Seçicisi

Sınıf seçicileri birden fazla öğeye uygulanabilir ve birden fazla kullanılabilir. Örneğin *önemli* adında bir sınıf seçicisi oluşturalım, özellik olarak da *yazının renginin kırmızı ve kalın olmasını* sağlayalım;

```
.onemli {  
    color: #FF0000;  
    font-weight: bold;  
}
```

Sınıf seçici olarak belirlediğimiz ismin önüne “.” (nokta) koymamız gerekmektedir. Bu ismi herhangi bir ögenin **class** özniteliğine atayarak stili uygulamış oluruz. Örneğimiz için **.onemli** sınıf (class) seçicisi yukarıdaki gibi olacaktır. Bu stilimizi sayfamızda H1 ve p etiketlerine uygulayalım.

Kod:

```
<html>
<head>
<title>CSS</title>
<style type="text/css">
    .onemli {
        color:red;
        font-weight:bold;
    }
</style>
</head>
<body>
<p class="onemli">Artvin Çoruh Üniversitesi</p>
<h1 class="onemli">Artvin Meslek Yüksekokulu</h1>
</body>
</html>
```

Tasarım Görünümü:

Artvin Çoruh Üniversitesi

Artvin Meslek Yüksekokulu

1.2. ID Seçicisi

Sınıf (class) seçicilerini birden fazla öğeye atayarak kullanabiliyorduk; ID seçiciler ise **yalnız bir öğeye atanır ve bir kez kullanılırlar**. Örneğin, “haber_baslik” adında bir stil oluşturalım, özellik olarak da yazı renginin mavi ve ortalı olmasını sağlayalım;

```
#haber_baslik {
    text-align: center;
    color: blue;
}
```

ID seçicisi olarak belirlediğimiz ismin önüne “#” koymamız gerekmektedir. ID’ye verdiğimiz isim, sayfa içinde aynı isme sahip öğeye (etiket) uygulanmış olacaktır. Bu stilimizi p etiketine uygulayalım.

Kod:

```
<html>
<head>
<title>CSS</title>
<style type="text/css">
    #haber_baslik {
        text-align:center;
        color:blue;
    }
</style>
</head>
<body>
<p id="haber_baslik">Bilgisayar Teknolojileri Bölümü</p>
<p>Bilgisayar Programcılığı Programı</p>
</body>
</html>
```

Tasarım Görünümü:



1.3. Etiket (Tag) Seçicisi

Etiket seçicileri aynı türden tüm öğelere (etiketlere), istediğimiz stili uygulamak için kullanılır. Örneğin sayfamızdaki tüm karakterlerin boyutunun 20px olmasını istiyorsak;

```
body { font-size: 20px; }
```

olarak yazmamız yeterlidir. Bir başka örnek olarak sayfamızdaki paragraflarda yazı büyüklüğünün **16px**, yazı tipinin **Arial** ve **kalın** olmasını sağlamamak için, stilimiz aşağıdaki şekilde oluşturulur.

```
p {
    font-family: Arial;
    font-size: 16px;
    font-weight: bold;
}
```

2. CSS Kullanım Alanları

Oluşturduğumuz stil şablonlarını sayfamıza nasıl uygulayacağımızı, nereye yerleştirmemiz gerektiğini belirleyen 3 yöntem vardır:

1. Yerel kullanım alanı
2. Global kullanım alanı
3. Bağlantılı kullanım alanı

2.1. Yerel Kullanım Alanı

Etiket içinde stil kullanımı olarak da adlandırılır. Bu kullanım şeklinde CSS direkt etiketin **“style”** özneliği kullanılarak uygulanır.

Örneğin, p etiketine, etiket içinde uygulanmak üzere yazı rengini kırmızı ve ortaya hizalı yapalım;

```
<p style="color:red; text-align:center;">Artvin MYO</p>
```

Yerel, bir başka deyişle etiket içi CSS kullanımı, tek tek her etiket içine yazılması gerektiği ve yönetiminin zor olması sebebiyle çok gerekmedikçe önerilmemektedir.

2.2. Global Kullanım Alanı

Dahili stil kullanımı olarak da adlandırılır. CSS kodumuz sayfamızın **<head>** kısmında **<style>...</style>** etiketi içinde yer alır. Kısım de olsa HTML ve CSS kodumuzu birbirinden ayırmış oluruz.

Örneğin, h3 etiketine uygulanmak üzere yazı rengini kırmızı, büyüklüğünü 12 pt ve sola hizalı yapalım;

```
<style type="text/css">
h3 {
    color: red;
    font-size: 12pt;
    text-align: left;
}
</style>
```

Global (dahili) olarak yerleştirdiğimiz CSS kodumuz sadece bu sayfayı etkileyecektir. Başka bir sayfada h3 etiketi normal görevini yapacaktır.

2.3. Bağlantılı Kullanım Alanı

Harici stil kullanımı olarak da adlandırılır. Stilimizi **.css** uzantılı stil şablonu dosyasına kaydederek, kullanmak istediğimiz sayfalarda **<link>** etiketi yardımıyla çağırırız. **HTML kısmı ile CSS kısmını birbirinden tamamen ayırmış oluruz.** Sayfalarımızın okunaklılığı ve yönetilebilirliği maksimum düzeye çıkacaktır. Bu nedenle en çok tercih edilen yöntemdir.

Örnek olarak, **style.css** adında harici bir stil şablonu (CSS) dosyası oluşturalım. Oluşturduğumuz dosyanın içine aşağıdaki satırları yazalım.

style.css dosyasının içeriği;

```
.baslik {  
    color: red;  
    text-align: right;  
    font-size: 16pt;  
}
```

Şimdi bu dosyayı sayfamıza **<link>** etiketini kullanarak bağlayalım;

```
<html>  
<head>  
<title>css</title>  
  <link href="style.css" rel="stylesheet" type="text/css">  
</head>  
<body>  
  <p class="baslik">Duyurular</p>  
</body>  
</html>
```

Dikkat etmemiz gereken nokta **css** dosyasını çağırdığımız sayfanın bulunduğu konumdan, **css** dosyasının bulunduğu konuma dosya yolunun tam olarak yazılmasıdır.

2.4. CSS Birimleri

Stil şablonları (CSS) oluşturma esnasında nesnelerin boyutlarını, aralarındaki mesafeleri ve renklerini ayarlarken kullanabileceğimiz ölçü birimleri vardır.

2.4.1. Uzunluk Birimleri

Uzunluk birimleri **göreceli** (bağıl-relative) ve **kesin** (absolute) olmak üzere ikiye ayrılır.

2.4.1.1. Göreceli (Bağıl) Uzunluk Birimleri

em, **ex**, **px** ve **%** birimleri, göreceli ölçü birimleridir. Bu birimler diğer uzunluk birimlerine bağlı olarak davranış gösterir.

em: Tüm tarayıcıların varsayılan yazı büyüklüğü (body etiketinde tanımlı) **16px**’ dir. **16px**, **1em**’ e eşittir. O zaman **piksel/16** bize **em** değerini verecektir.

ex: Hangi yazı tipi seçilmişse o yazı tipindeki **küçük x** harfinin yüksekliğidir.

px: Ekran çözünürlüğüne göre değişen, ekranda bulunan noktalardan (piksel) bir tanesinin yüksekliğidir.

2.4.1.2. Kesin Uzunluk Birimleri

in: inç’i (inch) ifade etmektedir. **1 inç=2,54 cm**’dir.

cm: Santimetreyi ifade etmektedir.

mm: Milimetreyi ifade etmektedir.

pt: Point (punto), standart baskı birimidir. **Örneğin**: font-size:14 pt

pc: Pika, Bir inçin altıda birine eşit ölçü birimi. **1pc=12 pt**’dir.

2.4.1.3. Yüzde Birimleri

body { font-size: 100%; } → 16px

body { font-size: 75%; } → 12px

2.4.2. Renk Birimleri

Stil şablonlarının (CSS) temel taşlarından olan renk belirleme işlemi, renk ismi (İngilizce) verilerek veya RGB koduna dayalı olarak yapılmaktadır.

2.4.2.1. Renk İsimleri

Tüm tarayıcılarda geçerli olan 16 adet renk ismi belirlenmiştir. Bu renkler ve isimleri aşağıdaki tabloda listelenmiştir.

Renk Görünümü	Renk Adı	HEX değeri	RGB değeri
	White - Beyaz	#FFFFFF	255,255,255
	Yellow - Sarı	#FFFF00	255,255,0
	Fuchsia - Fuşya	#FF00FF	255,0,255
	Red - Kırmızı	#FF0000	255,0,0
	Silver - Gümüş	#C0C0C0	192,192,192
	Gray - Gri	#808080	128,128,128
	Olive - Zeytin yeşili	#808000	128,128,0
	Purple - Mor	#800080	128,0,128
	Maroon - Vişne Çürüğü	#800000	128,0,0
	Aqua - Turkuaz	#00FFFF	0,255,255
	Lime - Parlak yeşil	#00FF00	0,255,0
	Teal - Koyu Yeşil	#008080	0,128,128
	Green - Yeşil	#008000	0,128,0
	Blue - Mavi	#0000FF	0,0,255
	Navy - Yeşil	#000080	0,0,128
	Black - Siyah	#000000	0,0,0

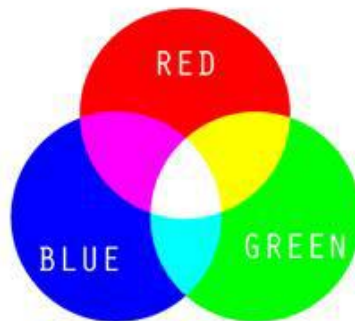
2.4.2.2. RGB Kodu ile Renk Belirleme

RGB kodu **Red** (kırmızı), **Green** (yeşil) ve **Blue** (mavi) olmak üzere 3 rengi esas alan kodlama sistemidir. Bu sisteme göre tüm renkler bu üç temel rengin karışımından oluşur.

Tüm renklerin karışımları **beyazı** verir. (**Beyaz renk kodu: #FFFFFF**)

Tüm kanalları kapatırsak **siyah** rengi elde ederiz. (**Siyah renk kodu: #000000**)

Kırmızı (#FF0000) – Yeşil (#00FF00) – Mavi (#0000FF)



2.4.2.2.1. Kısa RGB Kodu ile Renk Belirleme

Uzun RGB kodu ile renk belirleme işinin kısaltılmış halidir. #000 ile #FFF arasında renk değerleri alabilir.

Örneğin, { background-color: #f00 } arka plan rengini kırmızı yapacaktır.

2.4.2.2. Tamsayı Değer ile Renk Belirleme

Örneğin, RGB (0,0,0) → Siyah, RGB(255,255,255) → Beyaz
RGB(255,0,0) → Kırmızı

Sonuç olarak,

color: red;
color: #F00;
color: #FF0000;
color: rgb(255,0,0);
color: rgb(100%, 0%, 0%);

Yandaki satırların hepsi,
Kırmızı rengi ifade eder.

NOT:

Stil şablonlarımızı yazarken açıklama satırı eklemek için /* */ karakterleri kullanılır.

3. CSS Özellikleri

- | | |
|---------------------------|------------------------|
| ▪ Zemin Özellikleri | ▪ Liste Özellikleri |
| ▪ Font Özellikleri | ▪ Pozisyon Özellikleri |
| ▪ Kutu Modeli Özellikleri | ▪ Tablo Özellikleri |

3.1. Zemin Özellikleri

Öğelere zemin (background) rengi vermek veya zemin resmi eklemek ile onların nasıl davranış göstereceğini kapsar. Bu özellikler;

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

h1 { background-color: #3070E7; } → h1 etiketi için zemin rengini #3070E7 yapacaktır.

body { background-image:url('bg.jpg'); } → "bg.jpg" isimli görseli zemin resmi olarak sayfaya ekler.

3.1.1. Zemin Resmi Tekrarı

"background-repeat" özelliği zemine eklenen görselin tekrarlanma biçimini kontrol etmek için kullanılır. Dört ayrı değer alır:

- **background-repeat:repeat;** (Görselin hem yatay olarak hem de dikey olarak tekrar etmesini istiyorsak bu özelliği kullanırız. Bu özellik yazılmasa bile CSS varsayılan olarak bunu yapacaktır.)
- **background-repeat:no-repeat;**
- **background-repeat:repeat-x;** (Görsel yatay olarak tekrar edecektir.)
- **background-repeat:repeat-y;** (Görsel dikey olarak tekrar edecektir.)

```
body {  
    background-image:url('bg.jpg');  
    background-repeat:repeat-x;  
}
```

3.1.2. Zemin Resmi Pozisyonu

"background-position" özelliği görselin sayfadaki pozisyonunu belirler.

- Dikey hizalamada → üst (**top**), orta (**center**) ve alt (**bottom**)
- Yatay hizalamada → sol (**left**), orta (**center**) ve sağ (**right**)

- **background-position: top left;**
- **background-position: top center;**
- **background-position: top right;**
- **background-position: center left;**
- **background-position: center center;**
- **background-position: center right;**
- **background-position: bottom left;**
- **background-position: bottom center;**
- **background-position: bottom right;**
- **background-position: x% y%;**
- **background-position: xpos ypos;**

```
body {  
    background-image: url('bg.jpg');  
    background-repeat: no-repeat;  
    background-position: center center;  
}
```

```
body { background:url('bg.jpg') no-repeat center center; }
```

3.1.3. Zemin Resmi İliştirme

"**background-attachment**" özelliği zemine yerleştirdiğimiz görselin kaydırma çubuğu ile birlikte hareket edip etmeyeceğinin kontrolünü sağlar.

- Görselin kaydırma çubukları ile birlikte hareket etmesini istiyorsak değer olarak "**scroll**" kullanırız. Varsayılan (default) değer budur.
- Görselin kaydırma çubukları ile birlikte hareket etmemesini, sabit kalmasını istiyorsak değer olarak "**fixed**" kullanırız.

```
body {  
    background-image: url('bg.jpg');  
    background-repeat: no-repeat;  
    background-attachment: scroll;  
}
```

3.2. Font Özellikleri

Font özellikleri web sayfamızdaki metinlerin font ailesini, boyutunu, kalınlık ayarlarını, büyük-küçük harf olmasını ve stilini değiştirmek için kullanılır. Bu özellikler;

- font-family
- font-size
- font-weight
- font-style
- line-height
- font

```
p { font-family: Verdana, Arial, Helvetica, sans-serif; }
```

```
p { font-size: 20px; }
```

```
p { font-size: 1.25em; }
```

```
p { font-weight: bold | bolder | normal | lighter; }
```

```
p { font-style: normal | italic; }
```

```
p { line-height: 15px; } // Satırlar arası mesafeyi 15 px yapar.
```

```
font: bold 12px/15px Verdana, sans-serif; // Kısaltma işlemi yapılabilir.
```

3.3. Metin Özellikleri

Bu özellikler;

- color
- text-align
- text-decoration
- text-transform
- text-indent
- line-height
- letter-spacing
- word-spacing
- vertical-align

p { color: #449BDB; }

p { text-align: left | center | right | justify; }

a { text-decoration: none | overline | underline | blink; }

p { text-transform: uppercase | lowercase | capitalize; }

p { text-indent: 40px; } // Her paragraf 40px içerden başlar.

p { line-height: 15px; } // Satırlar arası mesafeyi 15 px yapar.

p { letter-spacing: 4px; } // Metnin karakterleri arasındaki boşluğu 4 px yapar.

p { word-spacing: 10px; } // Kelimeler arası mesafeyi 10 px yapar.

3.4. Kutu Modeli Özellikleri

Kutu modeli 3 ana öğeden oluşur. Bunlar;

- **margin** (kenar dışı boşluğu)
- **border** (kenar kalınlığı)
- **padding** (kenar içi boşluğu)



3.4.1. Kenar Dışı Boşluğu Özellikleri

margin öğesinin alt elemanları;

- **margin-top** (üst kenar dışı boşluğu)
- **margin-right** (sağ kenar dışı boşluğu)
- **margin-bottom** (alt kenar dışı boşluğu)
- **margin-left** (sol kenar dışı boşluğu)
- **margin** (kenar dışı boşlukları) (Kısaltma amacıyla kullanılır.)

Kısaltma işlemi:

`margin-top: 10px;`
`margin-right: 5px;`
`margin-bottom: 20px;`
`margin-left: 15px;`

} Kısaltılmış hali
`margin: 10px 5px 20px 15px;`

margin: 25px 50px 75px 100px; (üst 25px, sağ 50px, alt 75px, sol 100px olur.)

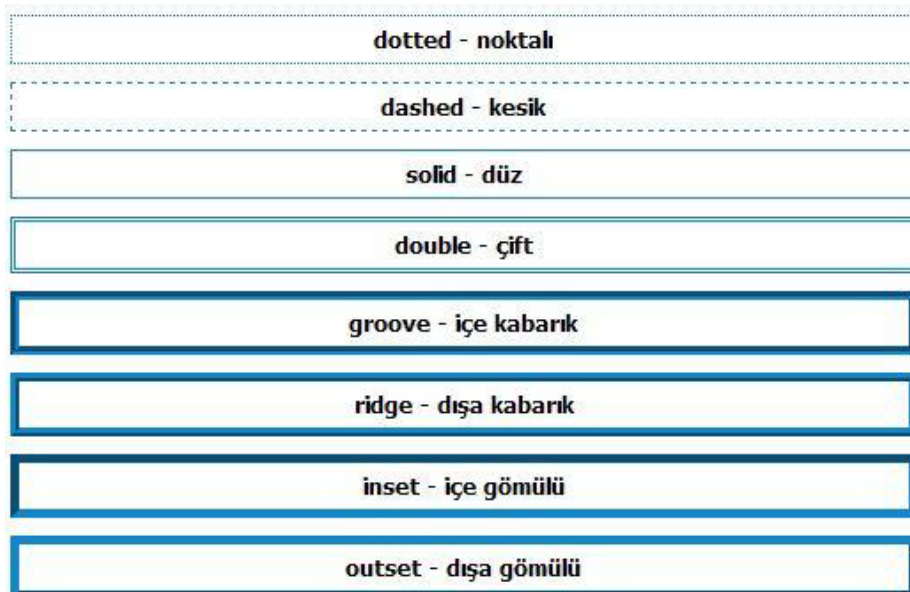
margin: 25px 50px; (üst ve alt 25px, sağ ve sol 50px olur.)

margin: 25px; (Tüm kenar dışı boşlukları 25px olur.)

3.4.2. Kenar Çizgisi Özellikleri

Bu özellikler şunlardır;

- **border-style**
 - (border-top-style | border-right-style | border-bottom-style | border-left-style)
- **border-width**
 - (border-top-width | border-right-width | border-bottom-width | border-left-width)
- **border-color**
 - (border-top-color | border-right-color | border-bottom-color | border-left-color)



```
.kenarlik1 { border-style: dotted; }
```

```
.border1 {  
    border-top-style: solid;  
    border-bottom-style: solid;  
    border-left-style: double;  
    border-right-style: double;  
}
```

Kısaltma:

```
.border1 {  
    border-style: solid double solid double;  
}
```

border-width: 2px 4px 5px 8px; (üst 2px, sağ 4px, alt 5px, sol 8px olur.)

border-width: 2px 5px; (üst ve alt 2px, sağ ve sol 5px olur.)

border-width: 2px; (Tüm kenar çizgisi genişlikleri 2px olur.)

border-color: blue; // Kenarların hepsi mavi renk.

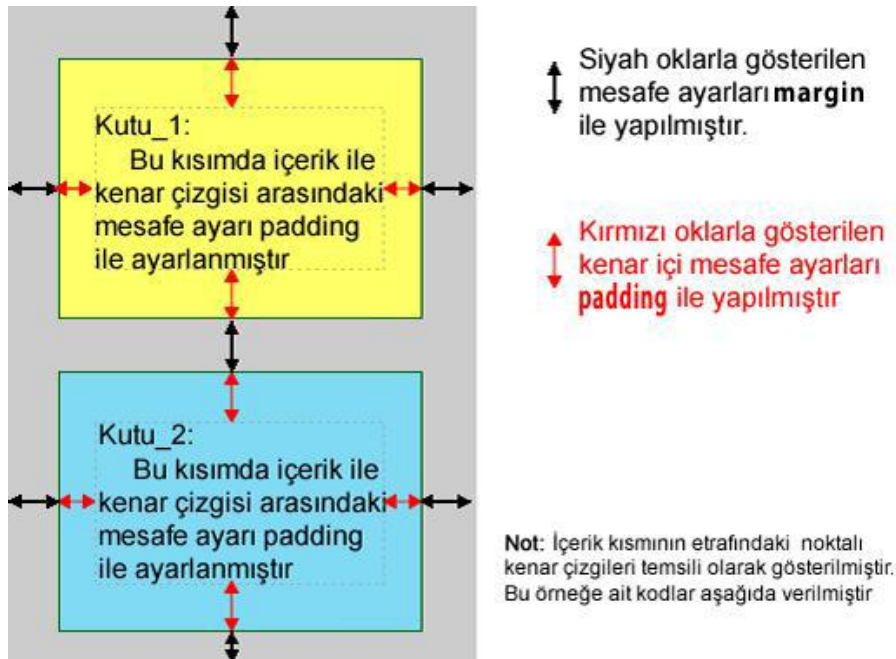
border-top-color: #F60; | **border-left-color:** #060; ...

border-color: #F60 #060 #CCC #0F0;

3.4.3. Kenar İçi Boşluğu Özellikleri

padding öğesinin alt elemanları;

- **padding-top** (üst kenar içi boşluğu)
- **padding-right** (sağ kenar içi boşluğu)
- **padding-bottom** (alt kenar içi boşluğu)
- **padding-left** (sol kenar içi boşluğu)
- **padding** (kenar içi boşlukları) (Kısaltma amacıyla kullanılır.)



```

.kutu1 {
    border: 1px solid #060;
    padding: 10px 10px 10px 10px;
    margin: 20px 10px 10px 20px;
    background-color: #FF6;
}
.kutu2 {
    border: 2px dotted #060;
    padding: 5px 5px 5px 5px;
    margin: 20px 10px 10px 20px;
    background-color: #81DAF3;
}
body {
    font-size: 14px;
    font-family: Helvetica;
    background-color: #EEE;
}

```

```

<html>
<head>
<title>css</title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div class="kutu1">
    Artvin Çoruh Üniversitesi, Artvin MYO, Bilgisayar<br>
    Bilgisayar Teknolojileri Bölümü, Bilgisayar Programcılığı
</div>

<div class="kutu2">
    Artvin Çoruh Üniversitesi, Artvin MYO, Bilgisayar<br>
    Bilgisayar Teknolojileri Bölümü, Bilgisayar Programcılığı
</div>
</body>
</html>

```

Artvin Çoruh Üniversitesi, Artvin MYO, Bilgisayar
Bilgisayar Teknolojileri Bölümü, Bilgisayar Programcılığı

Artvin Çoruh Üniversitesi, Artvin MYO, Bilgisayar
Bilgisayar Teknolojileri Bölümü, Bilgisayar Programcılığı

3.5. Liste Özellikleri

list-style-type:

- disk (içi dolu yuvarlak)
- circle (çember)
- square (kare)
- decimal (1, 2, 3, ...)
- lower-roman (i, ii, iii,...)
- upper-roman (I, II, III,...)
- lower-alpha (a, b, c,...)
- upper-alpha (A, B, C,...)

list-style-image: Liste biçiminin resim olmasını sağlar.

list-style-image: url("resim.gif");

```

li {
    list-style-type: decimal;
}

```

3.6. Pozisyon Özellikleri

position:

- **absolute** (Katmanın yerinin kesin olarak belirlenmek istendiğinde kullanılır.)
- **relative** (Katmanın yerinin görelî (diğer öğelere göre değışebilen) olarak belirlenmek istendiğinde kullanılır.)
- **absolute** (Katmanın yerinin sabit olarak belirlenmek istendiğinde kullanılır.)
- **top** (Katmanın üst kısımdan kaç piksel aşağıda olması gerektiğini belirler.)
- **left** (Katmanın sol kısımdan kaç piksel aşağıda olması gerektiğini belirler.)
- **width** (Katmanın genişliğinin kaç piksel olacağını belirler.)
- **height** (Katmanın boyunun kaç piksel olacağını belirler.)
- **auto** (Otomatik olarak belirlenir.)
- **scroll** (Kaydırma çubukları ekler.)
- **visibility** (Katmanın görünebilirlik ayarı yapar.)
 - **visible** (Görünür hale getirir.)
 - **hidden** (Gizler.)

div {

position: absolute;
top: 20px; left: 10px;
width: 300px; height: 200px;

}