
WEB TABANLI PROGRAMLAMA

BÖLÜM 5

CSS İLE HTML BİÇİMLENDİRME :
CSS TEMELLERİ VE HTML İLE
BİRLİKTE KULLANIMI

Prof. Dr. Turgay Tugay Bilgin

turgay.bilgin@btu.edu.tr



GENEL BAKIŞ...

5.1) CSS Seçiciler

5.2) Genel Seçiciler

5.3) Sınıf Seçiciler

5.4) Çoklu Stil Tanımlama

5.5) Seçicileri Gruplama

5.6) CSS Nitelikleri (Property)

5.7) CSS Kodlarının HTML İçinde Kullanım Şekilleri

5.7.1) Gömülü CSS (<style> Etiketini ile Kullanımı)

5.7.2) Satır İçi (Inline) CSS

5.7.3) Harici CSS Dosyaları

5.8) CSS Stilleri Etki Önceliği Sıralaması

5.9) CSS Açıklama Satırları

5.10) Çeşitli CSS Örnekleri

5.10.1) Arka Plan Rengi Değiştirme

5.10.2) Yazı Türü Değiştirme

5.10.3) Yazı Biçimi Değiştirme

5.10.4) Yazı Türü Koyuluğunu Değiştirme

5.10.5) Yazı Türü Büyüklüğü Değiştirme

5.10.6) Paragraf Girintisi Oluşturma

5.10.7) Metinlerde Gölge Oluşturma

5.10.8) Satır Yüksekliği Belirleme

5.10.9) Linklerin Rengini Belirleme

5.11) Özet

-
- Cascading Style Sheets (Basamaklı Stil Şablonları ya da Basamaklı Biçim Sayfaları), HTML dilinin yapabildiklerine ek olarak metin, tablo, form ve diğer html öğelerinin biçimlendirilmesi için fazladan olanaklar sunan bir web teknolojisidir. CSS, web sayfalarının sunumunu güzelleştirmek ve daha estetik hale getirmek amacıyla oluşturulmuş basit tasarım dilidir. *HTML ile yapılabilen birçok işlem CSS ile yapılabildiği gibi HTML ile yapılamayan birçok işlem ise CSS ile yapılabilir.*
 - CSS kullanmanın avantajları:
 - **CSS zaman kazandırır :** Yazılan CSS kodu birden fazla HTML etiketinde veya birden fazla HTML sayfasında kolayca kullanılabilir.
 - **Sayfalar daha hızlı yüklenir:** CSS ile oluşturulan tasarımlarda HTML gibi her bir etiket için ayrı ayrı biçimlendirme yapmaya gerek olmadığından yazılacak kod sayısı azalır.
 - **Bakım kolaylığı:** Birden fazla sayfadan oluşan web sitelerinde sitenin genelini etkileyecek tasarım değişiklikleri tek bir dosyadan yönetilebildiği için bakımı kolaydır.
 - **HTML diline göre daha fazla biçimlendirme olanağı:** CSS dilinde HTML'e göre çok daha detaylı biçimlendirme sağlayan komutlar mevcuttur.
 - **Web standartları ile uyumluluk:** Son yıllarda HTML diline ait biçimlendirme nitelikleri azaltılmakta, bunun yerine biçimlendirme komutları CSS diline kaydırılmaktadır. Bu nedenle güncel web standartları ile uyumluluk için CSS tabanlı tasarımlar çok daha avantajlı ve farklı web tarayıcılarda uyumluluğu daha yüksektir.

(5.1) CSS SEÇİCİLERİ

- CSS dili HTML dilinin özelliklerini genişletmek ve yeni özellikler eklemek adına geliştirildiği için, CSS dilinin tüm komutları belirli HTML etiketleri ile çalışacak şekilde tanımlanır. Bir veya birden fazla CSS komutunun hangi HTML etiketinin özelliklerini değiştireceğini bilmesi gerekir. Bu bildirimler “seçici” adı verilen terim ile yapılır. Seçiciler genellikle HTML etiketleridir. Seçici belirledikten sonra bu HTML etiketinin hangi niteliklerinin ne şekilde değiştirileceği belirlenmelidir. Bu işleme stil tanımlama denir ve aşağıdaki gibi gerçekleştirilir.

```
<style type="text/css">
Seçici {
    nitelik: değer
}
</style>
```

- Stil tanımlamaları mutlaka <style>...</style> etiketleri arasında yapılmalıdır. Bunun dışındaki bir bölümde yapılan tanımlamalar web tarayıcı tarafından HTML komutu gibi algılanacaktır.

Bir stil tanımında 3 temel terim bulunur:

- **seçici:** Seçici bir HTML etiketidir. CSS biçimlendirmesi sadece bu etikete uygulanır.
- **nitelik:** Seçicinin hangi niteliği değiştirilecekse o niteliği temsil eder. Buradaki nitelikler HTML nitelikleri değil CSS dilinin kendi nitelikleridir.
- **değer:** CSS niteliğinin değerini ifade eder. Her CSS niteliği bir değere sahiptir. Değer atama işlemi iki nokta üst üste operatörü ile gerçekleştirilir.

Dikkat ✍ *CSS ile HTML arasında fark varmı?*

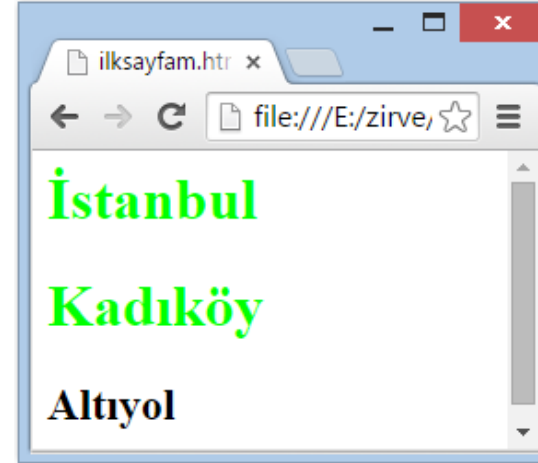
CSS dili HTML dilini tamamlayıcı bir dil olarak ortaya çıkmış olsa da iki dilin kuralları birbirinden tamamıyla farklıdır. Dolayısıyla CSS dilinde nitelikler, değerler ve tanımlamalar HTML dilindekinden tamamen farklı olduğu unutulmamalıdır.

Hatırlatma ✍ *CSS dilinde niteliklerin tanımlandığı satır sonları noktalı virgül ile bitirilmelidir.*

```
<html>
<head>
<style type="text/css" >
    h1 {
        color: #00FF00;
    }
</style>
</head>
<body>
<h1> İstanbul </h1>
<h1> Kadıköy</h1>
<h2> Altıyol </h2>
</body>
</html>
```

Kod 5.1. CSS Seçicileri

Yandaki örneği **cssornek.html** olarak kaydediniz. Örnekte görüleceği üzere stil tanımlama işlemi `<head> . . . </head>` bloğu içinde gerçekleştirilir.



Şekil 5.1 CSS ile stil tanımlama

Örnek çalıştırıldığında Şekil 5.1'deki görüntü elde edilir. Burada stil tanımlama sadece `h1` etiketine yapıldığından İstanbul ve Kadıköy ifadeleri yeşil, Altıyol ifadesi ise `h2` etiketi içinde olduğundan stil tanımını dışındadır ve varsayılan renk olan siyah renkte görülmektedir.

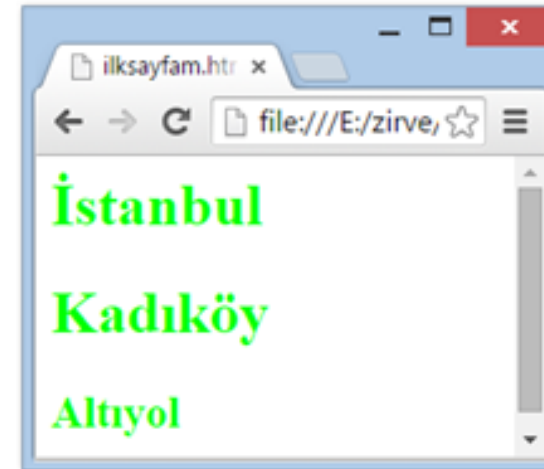
(5.2) GENEL SEÇİCİLER

- Seçicileri belirli bir HTML etiketi yerine tüm HTML etiketlerinde geçerli olacak şekilde oluşturabilirsiniz. Bu durumda seçici olarak “*” karakteri kullanılır.

```
<html>
<head>
<style type="text/css" >
    * {
        color: #00FF00;
    }
</style>
</head>
<body>
<h1> İstanbul </h1>
<h1> Kadıköy</h1>
<h2> Altıyol </h2>
</body>
</html>
```

Kod 5.2. Genel Seçiciler

- Örnek çalıştırıldığında Şekil 5.2’deki görüntü elde edilir. Burada stil tanımlama tüm HTML etiketlerine yapıldığından tüm etiketlerde yer alan metinler yeşil olmaktadır.



Şekil 5.2. CSS dilinde genel seçici tanımlama

(5.3) SINIF SEÇİCİLER

- Herhangi bir HTML etiketine stil tanımlamak yerine kendimizin isim verdiği stillere sınıf seçici adı verilir. Kendi verdiğimiz isime **sınıf** (class) adı verilir. Oluşturduğumuz sınıfı istediğimiz bir HTML etiketine uygulamak için etiketin **class** niteliğine sınıfımızın adını veririz.

Dikkat ✍ *Sınıf seçicilerin tanımlanması*

Sınıf seçiciler tanımlanırken **mutlaka** nokta (.) karakteri ile başlamalıdır. Fakat kullanılırken nokta olmadan kullanılır.


```
<html>

<head>

<style type="text/css" >

    .yesil {
        color: #00FF00;
    }

</style>

</head><body>

<h1 class="yesil"> İstanbul </h1>

<h1> Kadıköy</h1>

<h2> Altıyol </h2>

</body>

</html>
```

Kod 5.3. Sınıf Seçiciler

Örnekte “.yesil” adlı bir sınıf oluşturulmuştur. Bu sınıf İstanbul yazısı oluşturan h1 etiketine class niteliği aracılığıyla uygulanmıştır.

Dikkat ⚡ *Sınıf seçicilerin kullanımı*

Örnekte class=".yesil" değil class="yesil" şeklinde kullanıldığında dikkat ediniz.

Örnek çalıştırıldığında sadece İstanbul ifadesinin yeşil renkte diğer metinlerin ise siyah renkte görüntülediği görülecektir.

(5.4) ÇOKLU STİL TANIMLAMA

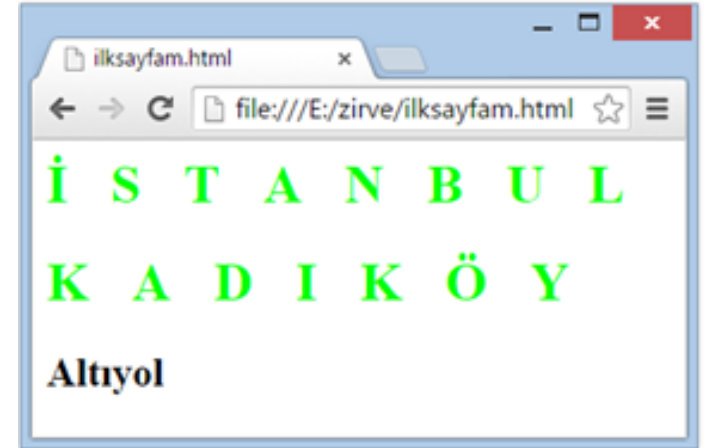
- Herhangi bir HTML elemanı için birden fazla stil komutu tanımlayabilirsiniz. Bunun için her bir satır süslü parantez bloğu içinde ve *satır sonlarında **noktalı virgül*** bulunmalıdır.

Kod 5.4. Çoklu Stil tanımlama

```
<html><head>
<style type="text/css" >
  h1 {
    color: #00FF00;
    font-weight: bold;
    letter-spacing: .8em;
    text-transform: uppercase;
  }
</style>
</head><body>
<h1> İstanbul </h1>
<h1> Kadıköy</h1>
<h2> Altıyol </h2>
</body></html>
```

- Örnekteki tüm satırlar tek satırda yanyana da yazılabilir fakat okunabilirlik açısından her birini ayrı satırda yazmak tercih edilmelidir. Burada tanımlanan stil komutlarından:
 - color: rengi belirler,
 - font-weight: yazı tipinin kalın, normal veya eğik olmasını sağlar,
 - letter-spacing: Harfler arası boşluğu ayarlar,
 - text-transform: metni büyük veya küçük harfe çevirir.

-
- Örnek çalıştırıldığında Şekil 5.3'deki görüntü elde edilir. HTML kodunda İstanbul, Kadıköy ve Altıyol ifadelerinin sadece ilk harfi büyük durumdayken çıktıda tümü büyük görünmekte ve harfler arası aralık normalden çok daha fazla olarak görüntülenmektedir.



Şekil 5.3 CSS dilinde çoklu stil tanımlama

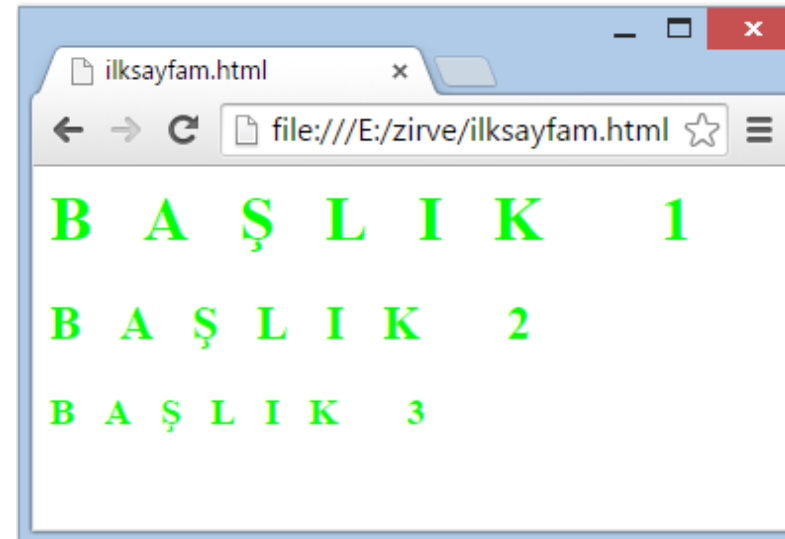
(5.5) SEÇİCİLERİ GRUPLAMA

```
<html><head>
<style type="text/css" >
  h1,h2,h3 {
    color: #00FF00;
    font-weight: bold;
    letter-spacing: .8em;
    text-transform: uppercase;
  }
</style>
</head><body>
<h1> İstanbul </h1>
<h1> Kadıköy</h1>
<h2> Altıyol </h2>
</body></html>
```

Kod 5.5. Seçicileri Gruplama

Birden çok seçici aralarına virgül koyularak aynı stil bloğunu paylaşabilirler. Bu durumda, aşağıdaki örnekte olduğu gibi stil bloğu hem <h1> hem <h2> hemde <h3> etiketlerine uygulanacaktır.

Yukarıdaki kod çalıştırıldığında Şekil 5.4'deki çıktı elde edilir.



Şekil 5.4 Gruplanmış seçiciler örneği

(5.6) CSS NİTELİKLERİ (PROPERTY)

- Şimdiye kadar öğrendiklerimizin hepsi seçiciler ile ilgilidir. Seçiciler tanımlandıktan sonra süslü parantez ile başlayıp süslü kapatma parantezi ile biten bölümde ise CSS nitelikleri ve bunların aldığı değerler yazılır. Nitelikler ve değerler arasında iki nokta üst üste işareti bulunur ve her bir nitelik-değer çiftinden sonra noktalı virgül kullanılır.
- Nitelikler color, font-weight, letter-spacing gibi ingilizce terimlerden oluşurlar. CSS'de kullanılabilecek 100'den fazla nitelik bulunmaktadır, bunların her biri tek tek anlatılmayacak, sadece sık kullanılanlar örnekler içinde yer alacaktır. CSS'de kullanılan tüm niteliklerin listesi ve alabildikleri değerler W3C konsorsiyum web sitesinde aşağıdaki URL'de bulunmaktadır.

<https://www.w3schools.com/cssref/>

-
- CSS kodlarınızı yazarken Adobe Dreamweaver gibi bir editör kullanıyorsanız CSS komut tamamlama ve yeni CSS oluşturma sihirbazı gibi yardımcılarını kullanarak daha kolay CSS oluşturabilirsiniz.

(5.7) CSS KODLARININ HTML İÇİNDE KULLANIM ŞEKİLLERİ

- HTML içinde CSS etiketlerini 3 farklı şekilde kullanabilirsiniz. Bunların her biri aşağıda açıklanmıştır.

5.7.1 Gömülü CSS (<style> Etiketi ile Kullanım) : Şimdiye kadar yapılan tüm örnekler Gömülü CSS olarak oluşturulmuştur. Gömülü CSS durumunda CSS kodları <style>...</style> etiketleri arasına yazılır. Genel kullanımı aşağıdaki gibidir.

```
<html>
<head>
<style type="text/css" >
    CSS kodları buraya yazılır.
</style>
</head></html>
```

Kod 5.6. <style> Etiketi ile Kullanım

5.7.2. Satır İçi (Inline) CSS : Bu tür kullanımda stil tanımlaması seçilen HTML elemanının *style* niteliği içerisinde **çift tırnaklar arasında** yapılır. Oluşturacağımız biçimlendirmenin sadece 1 HTML elemanından tek seferlik kullanılmasını istiyorsanız bu yöntemi tercih edebilirsiniz.

```
<html>

<body>

<h1 style="color:#0000FF;">Başlık 1</h1>

<h2> Başlık 2 </h2>

<h3> Başlık 3 </h3>

</body>

</html>
```

Kod 5.7. Satır İçi CSS



Şekil 5.5 Satır içi CSS örneği

- Yukarıdaki kod çalıştırıldığında Şekil 5.5'deki çıktı elde edilir.

5.7.3 Harici CSS Dosyaları : CSS kodlarınızı birden fazla HTML dosyasında tekrar yazmadan kullanmak isterseniz yukarıdaki 2 yöntemin ikisi de işinize yaramayacaktır. Bu durumda harici CSS dosyaları kullanılır.

- Harici CSS dosyaları tıpkı HTML dosyalarında olduğu gibi düz metin dosyalardır. Tek farkları dosya uzantılarının **.css** olmasıdır. ***Bu dosyaların içinde yalnızca CSS kodu bulunmalıdır, HTML kodu bulunmamalıdır.***
- `<link>` adlı HTML etiketi aracılığıyla css uzantılı dosyalar HTML dosyalar ile ilişkilendirilir. Bu durumda web tarayıcı stil komutlarını “.css” uzantılı dosyadan okuyarak HTML dosya üzerinde uygular.
- `<link>` etiketinin kullanımı örnekte görülmektedir. Örnekte görüleceği üzere kod içerisinde hiçbir CSS seçicisi veya niteliği yoktur. Tüm CSS kodları **stildosyam.css** isimli dosyada bulunmaktadır.

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="stildosyam.css" />
</head>
<body>
<h1>Başlık 1</h1>
<h2>Başlık 2</h2>
<p>Bu bir örnek paragraftır.</p>
</body>
</html>
```

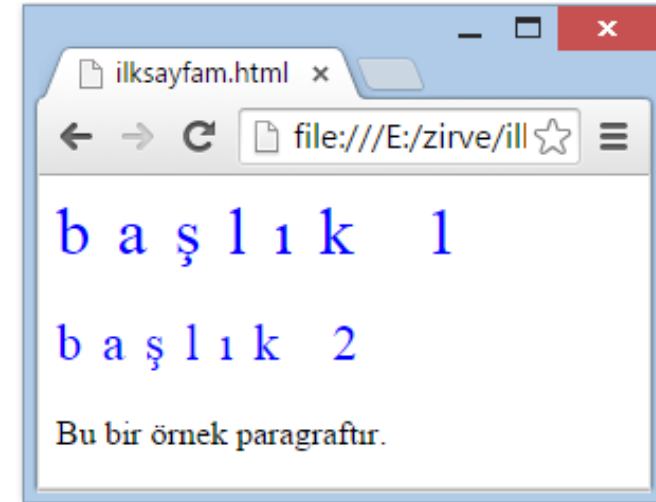
Kod 5.8. Harici CSS Dosyaları örneği

- Önceki slaytta verilen kodları **cssornek.html** olarak kaydediniz. Kod 5.7'deki `<link>` etiketiyle eklenen **stildosyam.css** dosyasının içeriği ise şöyledir:

```
h1, h2, h3 {  
  color: #0000FF;  
  font-weight: normal;  
  letter-spacing: .4em;  
  text-transform:  
  lowercase;  
}
```

Kod 5.9. stildosyam.css dosyasının yapısı

- HTML belge web tarayıcıda çalıştığında Şekil 5.6'daki görüntü elde edilir.



Şekil 5.6 Harici CSS örneği

- Harici CSS dosyası kullanımı bölüm 5.7'de verilen 3 tür kullanımdan en yaygın olanıdır. Çünkü Bu şekilde kullanım **birden fazla HTML dosyasının yalnızca tek bir CSS dosya ile kontrol edilmesine olanak sağlar**. Böylece birçok sayfa içeren tasarımlarda tasarımın genelini etkileyecek değişiklikler yalnızca tek bir sayfa üzerinde yapılacak değişiklik ile kontrol altına alınabilir.

(5.8) CSS STİLLERİ ETKİ ÖNCELİĞİ SIRALAMASI

- CSS kodları HTML dosya içerisinde 3 farklı konumda yer alabildiğini ifade ettik. Bu 3 konumda aynı etiket için farklı kurallar tanımlanmış ise bunlardan hangisi geçerli olacaktır? İşte bu durumda öncelik sırası kavramı işin içine girer. CSS stillerinde tanımlandığı konuma göre öncelik sıralaması şöyledir:
 1. Satır içi CSS tanımları en yüksek önceliğe sahiptir. Aynı etikete farklı tanımlar yapıldığı durumda web tarayıcı “satır içi CSS” içinde ne var ise onu ilk olarak çalıştırır.
 2. Satır içi CSS tanımlanmamışsa `<style>...</style>` etiketleri arasında tanımlanmış olan kodun önceliği vardır.
 3. Yukarıdaki iki tanım da yapılmamışsa Harici CSS dosya içindeki tanımlar dikkate alınarak web tarayıcı tarafından çalıştırılır. Diğer bir ifadeyle en düşük öncelik harici CSS dosyalarındadır.

(5.9) CSS YORUM SATIRLARI

- CSS dilinin kurallarının geçerli olduğu bölgelerde HTML yorum satırları kullanılamaz. CSS dilinde yorum satırları C/C++ dillerindeki çok satırlı yorum işareti olan `/* */` ile aynıdır.

```
/* bu bir harici css dosyadır */  
h1, h2, h3 {  
  color: #0000FF;  
  font-weight: normal;  
  letter-spacing: .4em;  
  text-transform: lowercase;  
/* harici css dosya sonu */  
}
```

Kod 5.10. CSS Yorum Satırları

(5.10) ÇEŞİTLİ CSS ÖRNEKLERİ

- Bu bölümde CSS etiketleri ile yapılabilecek biçimlendirmelere örnekler verilmiştir.

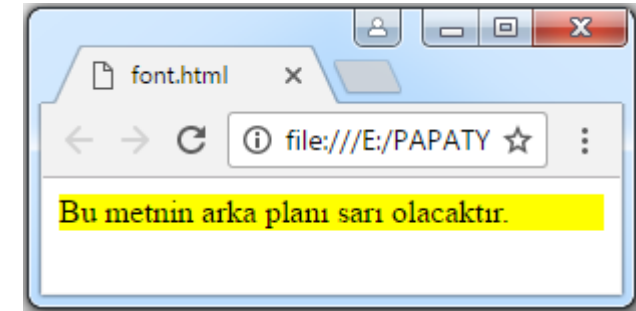
(5.10.1) Arka plan rengi değiştirme

Kod 5.11. Arka plan rengi değiştirme

```
<html>
<p style="background-
color:yellow;">
Bu metnin arka planı sarı
olacaktır.
</p>
</html>
```

- Örnekte “yellow” yerine HTML ünitesinde öğrendiğimiz ingilizce renk adları veya onaltılık sayı sisteminde rengin kod numarası yazılabilir.

- Yandaki örneği **ornek.html** olarak kaydedin. Örneği çalıştırdığınızda Şekil 5.7’deki görüntü elde edilir.



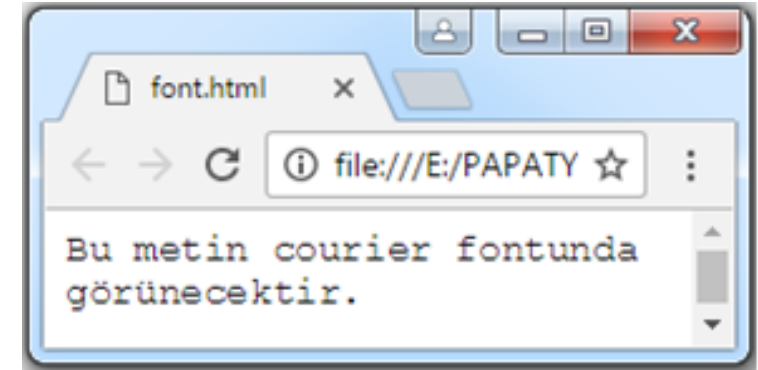
Şekil 5.7 Arka plan rengi örneği

(5.10.2) YAZI TİPİ DEĞİŞTİRME

Kod 5.12. Yazı Tipi Değiştirme

```
<html>
<p style="font-family:courier;">
Bu metin courier fontunda görünecektir.
</p>
</html>
```

- Yukarıdaki örneği ornek.html olarak kaydedin. Örneği çalıştırdığınızda Şekil 5.8'deki görüntü elde edilir.
- Bu örnekte “courier” yerine işletim sisteminizde bulunan herhangi bir font adı yazabilirsiniz.



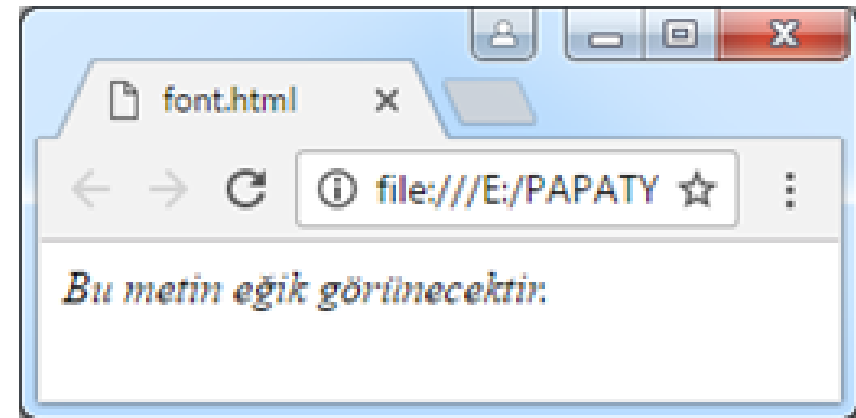
Şekil 5.8 Yazı tipi değiştirme örneği

(5.10.3) YAZI BİÇİMİ DEĞİŞTİRME

Kod 5.13. Yazı Biçimi Değiştirme

```
<html>
<p style="font-style:italic;">
Bu metin eğik görünecektir.
</p>
</html>
```

- Yandaki örneği **ornek.html** olarak kaydedin. Örneği çalıştırdığınızda Şekil 5.9'daki görüntü elde edilir



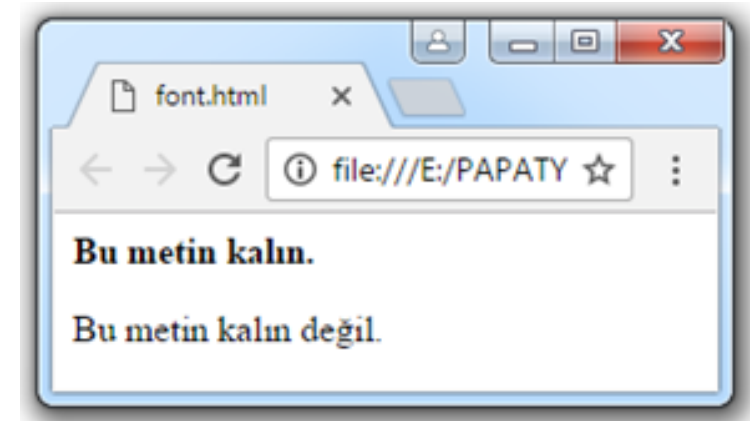
Şekil 5.9 Yazı biçimi değiştirme örneği

(5.10.4) YAZI TİPİ KALINLIĞI DEĞİŞTİRME

Kod 5.14. Yazı Tipi Kalınlığı Değiştirme

```
<html>
<p style="font-weight:bold;">Bu metin kalın.</p>
<p> Bu metin kalın değil.</p>
</html>
```

- Yukarıdaki örneği **ornek.html** olarak kaydedin. Örneği çalıştırdığınızda Şekil 5.10'daki görüntü elde edilir.



Şekil 5.10 Yazı tipi kalınlığı değiştirme örneği

(5.10.5) YAZI TİPİ BOYUTU DEĞİŞTİRME

Kod 5.15. Yazı Tipi Boyutu Değiştirme

```
<html>
<p style="font-size:30px;">Bu yazı 30 pixeldir.</p>
<p style="font-size:small;">Bu yazı küçüktür.</p>
<p style="font-size:large;">Bu yazı büyüktür.</p>
</html>
```

- Yukarıdaki örneği **ornek.html** olarak kaydedin. Örneği çalıştırdığınızda Şekil 5.11'daki görüntü elde edilir.
- Örnekte font-size niteliğine small, large gibi İngilizce deyimler değer olarak verilebildiği gibi font büyüklüğü piksel (**px**) olarak sayısal değerler de alabilir.



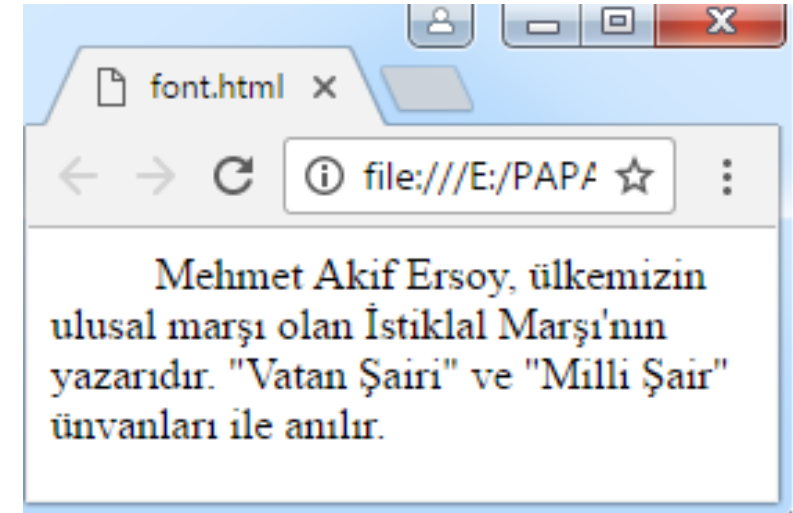
Şekil 5.11 Yazı tipi boyutu değiştirme örneği

(5.10.6) PARAGRAF GİRİNTİSİ OLUŞTURMA

Kod 5.16. Paragraf Girintisi Oluşturma

```
<html>
<p style="text-indent:1cm;">
Mehmet Akif Ersoy, ülkemizin ulusal
marşı olan İstiklal Marşı'nın yazarıdır.
"Vatan Şairi" ve "Milli Şair" ünvanları
ile anılır.
</p>
</html>
```

- Yukarıdaki örneği **ornek.html** olarak kaydedin. Örneği çalıştırdığınızda Şekil 5.12'deki görüntü elde edilir.
- Örnekte text-indent niteliği girinti ölçüsünü "cm" cinsinden sayısal değerler olarak alır.



Şekil 5.12 Paragraf girintisi oluşturma örneği

(5.10.7) METİNLERDE GÖLGE OLUŞTURMA

Kod 5.17. Metinlerde Gölge Oluşturma

```
<html>
<p style="text-shadow:4px 4px 4px blue;">
Mehmet Akif Ersoy, ülkemizin ulusal
marşı olan İstiklal Marşı'nın yazarıdır.
"Vatan Şairi" ve "Milli Şair" ünvanları
ile anılır.
</p>
</html>
```

- Örnekte text-shadow niteliğinde ilk parametre olan 4px yatay ekseninde gölgeyi öteleme miktarı, ikinci parametre olan 4px dikey ekseninde gölgeyi öteleme miktarı ve üçüncü parametre gölgenin netliğini belirten ölçü, son parametre ise gölge rengidir. Değerler ile oynayarak farklı tarzlarda gölgeli metinler elde edebilirsiniz.

- Yandaki örneği **ornek.html** olarak kaydedin. Örneği çalıştırdığınızda Şekil 5.13'daki görüntü elde edilir.



Şekil 5.13 Metinde gölge oluşturma örneği

(5.10.8) SATIR YÜKSEKLİĞİ BELİRLEME

Kod 5.18. Satır Yüksekliği Belirleme

```
<html>  
<p style="line-height:50px;">  
Mehmet Akif Ersoy, ülkemizin ulusal  
marşı olan İstiklal Marşı'nın yazarıdır.  
"Vatan Şairi" ve "Milli Şair" ünvanları  
ile anılır.  
</p>  
</html>
```

- Örnekte line-height niteliği yükseklik ölçüsünü piksel (px) cinsinden sayısal değerler olarak alır

- Yandaki örneği **ornek.html** olarak kaydedin. Örneği çalıştırdığınızda Şekil 5.14'daki görüntü elde edilir.



Şekil 5.14 Satır yüksekliği belirleme örneği 28

(5.10.9) LİNKLERİN RENKLERİNİ BELİRLEME

- `<a> ... ` etiketi ile yapılan HTML linklerinin renklerini CSS ile belirlemek için birtakım özel stil komutları bulunmaktadır. Bunlar:
 - **a:link** Ziyaret edilmemiş linklerin rengini belirtir.
 - **a:visited** Ziyaret edilmiş linklerin rengini belirtir.
 - **a:hover** Kullanıcı fare ile linkin üzerine (tıklamadan) geldiğinde linkin hangi renkte olacağını belirtir.
 - **a:active** Kullanıcının linki tıklaması sırasında linkin hangi renkte olacağını belirtir.

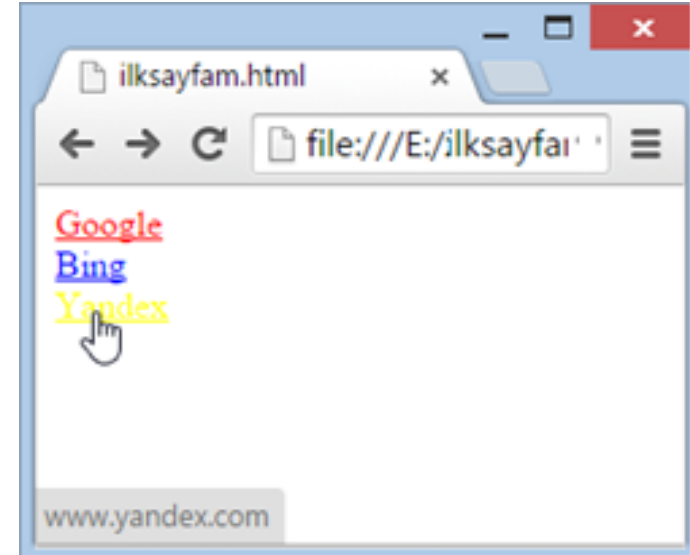
Dikkat  *Link komutların kullanılma sırası önemlidir*

a:hover komutu **a:link** ve **a:visited** komutlarından sonra gelmelidir. **a:active** komutu da **a:hover** komutundan sonra gelmelidir.

Kod 5.19. Linklerin Renklerini Belirleme

```
<html>
<head>
<style type="text/css">
a:link {color:red};
a:visited {color:blue};
a:hover {color:yellow};
a:active {color:green};
</style>
</head>
<body>
<a href="http:// www.google.com">Google</a><br />
<a href="http:// www.bing.com">Bing</a><br />
<a href="http:// www.yandex.com">Yandex</a><br />
</body>
</html>
```

- Yukarıdaki örneği **link.html** olarak kaydedin. Örneği çalıştırdığınızda Şekil 5.15'deki görüntü elde edilir.



Şekil 5.15 Linklerin renklerini belirleme

-
- Örnekte görüldüğü üzere Yandex linki üzerine gelindiğinde sarı renk oluşmakta, bing linki de daha önce ziyaret edilmiş olduğundan mavi, google linki ise hiç ziyaret edilmediğinden kırmızı olarak görüntülenmiştir.

(5.11) ÖZET

- Cascading Style Sheets (CSS), HTML dilinin yapabildiklerine ek olarak metin, tablo, form ve diğer html öğelerinin biçimlendirilmesi için fazladan olanaklar sunan bir web teknolojisidir. CSS, web sayfalarının sunumunu güzelleştirmek ve daha estetik hale getirmek amacıyla oluşturulmuş basit tasarım dilidir. Bir veya birden fazla CSS komutunun hangi HTML etiketinin özelliklerini değiştireceğini bilmesi gerekir. Bu bildirimler “seçici” adı verilen terim ile yapılır. Stil tanımlamaları mutlaka `<style>...</style>` etiketleri arasında yapılmalıdır.
- CSS nitelikleri color, font-weight, letter-spacing gibi ingilizce terimlerden oluşurlar. CSS’de kullanılabilecek 100’den fazla nitelik bulunmaktadır. CSS’de kullanılan tüm niteliklerin listesi ve alabildikleri değerler W3C konsorsiyum web sitesinde bulunmaktadır.
- CSS kodlarını birden fazla HTML dosyasında her seferinde tekrar tanımlamadan kullanmak gerektiğinde harici CSS dosyaları tercih edilir. Harici CSS dosyaları tıpkı HTML dosyalarında olduğu gibi düz metin dosyalardır. Tek farkları dosya uzantılarının ".css" olmasıdır.