

ETK229 Web Tasarımının Temelleri

Dersi

Ankara Üniversitesi Elmadağ Meslek Yüksekokulu

Öğretim Görevlisi : Murat Duman

Mail: mduman@ankara.edu.tr

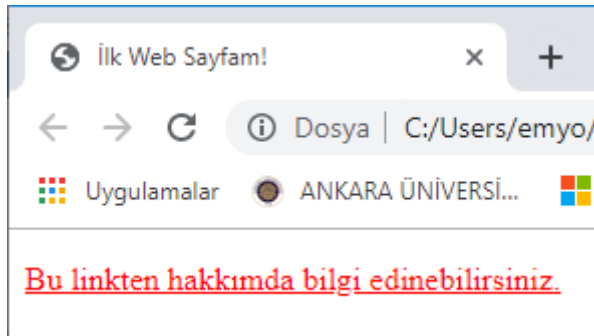
6. Hafta

Bölüm 9 : CSS (Cascading Style Sheets)

Örneğin; herhangi bir web sayfası içerisindeki bağlantı henüz tıklanmadığında mavi renkli iken eğer daha önceden tıklanmışsa tekrar tıklanmadan önce mor renklidir. Şimdi bu renklendirmeyi değiştirelim. Bunun için `style` tegini kullanacağız. `style` tegi `body` tegi ile işaretlenmiş bölge içerisinde herhangi bir yerde tanımlanabilir. Biz ilgili stili sadece tıkladığında yeni sayfaya yönlendirecek metin üzerinde kullanacağımız için `a` tegi içerisinde tanımlayacağız. `style` tegi kullanılırken stille ilgili özelliğe değer ataması yapılır. Örneğin; `style="color: red;"` kodu link vermede kullanılacak metnin rengini kırmızı yapar. İşlemin sonlandığını belirtmek için en sonda ";" kullanılır. İlgili kod parçacığı Şekil 9.1.'de verilmiş olup ilgili kod parçacığının çıktısı Şekil 9.2.'deki gibi olacaktır.

```
index.html x
1 <!doctype html>
2 <html>
3   <head>
4     <title>İlk Web Sayfam!</title>
5     <meta charset="utf-8"/>
6   </head>
7
8   <body>
9     <p><a href="hakkında/hakkında.html" style="color: red;">
10      Bu linkten hakkımda bilgi edinebilirsiniz.</a></p>
11   </body>
12 </html>
```

Şekil 9.1. İlgili kod



Şekil 9.2. İlgili ekran görüntüsü

Renk atamasında İngilizce renk isminden başka rgb kodu ile de renk ataması yapabilirsiniz. Örneğin; rgb(100,100,50). Ayrıyeten renk kodunu “#” sembolü ile birlikte hex olarak altı basamak olacak şekilde de girebilirsiniz. Örneğin; #e3e3e3.

Yine herhangi bir web sayfası içerisindeki bağlantı metnine bakıldığında metnin altını çizili olarak görürüz.

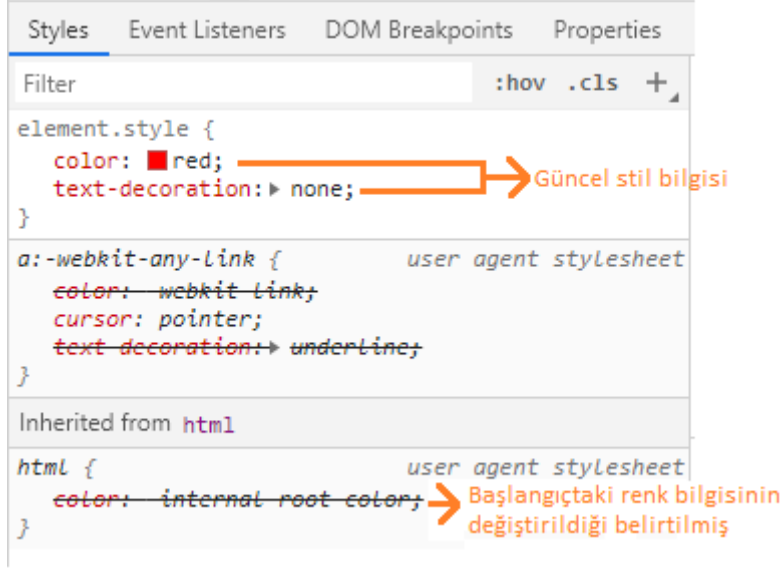
Peki metnin altını çizili olarak görmek istemiyorsak!

Bu işlem için **text-decoration** (metin dekorasyonu) özelliğini kullanacağız ve değer olarak metnin altını çizili istemediğimiz için “**none**” değerini atayacağız. İlgili kod parçacığı Şekil 9.3.’te verilmiştir.

```
8 <body>
9   <p><a href="hakkında/hakkında.html" style="color: red; text-decoration: none;">
10     Bu linkten hakkında bilgi edinebilirsiniz.</a></p>
11 </body>
12 <html>
```

Şekil 9.3. İlgili kod

İlgili linke sağ tıklayıp karşımıza çıkan menüden “inspect element” (elemanı incele) seçeneğini işaretlersek Şekil 9.4.’teki ekran görüntüsünü elde ederiz.



```
Styles | Event Listeners | DOM Breakpoints | Properties
Filter :hov .cls +
element.style {
  color: red;
  text-decoration: none;
}
a:-webkit-any-link { user agent stylesheet
  color: webkit-link;
  cursor: pointer;
  text-decoration: underline;
}
Inherited from html
html { user agent stylesheet
  color: internal root color;
}
```

Şekil 9.4. İlgili elemanın incelenmesi

İlgili eleman incelendiğinde Şekil 9.4.’te güncel renk ve metin bilgisine ait özellikler verilmiştir. Ayrıca yine aynı şekilde başlangıçtaki renk ve metin özelliklerinin değiştirildiği bilgisi elde edilebilmektedir.

Şekil 9.3.'e tekrar baktığımızda metnin stiline ait sadece iki özellik girdiğimiz halde kod parçacığının karmaşık hale gelmeye başladığını görürüz. Buradan stil özelliklerinin farklı bir alanda tanımlanmasının karmaşıklığı azaltacağı sonucunu çıkarabiliriz. Bu yüzden şimdi [stil.css](#) isimli yeni bir dosya oluşturalım. [index.html](#) dosyası içerisine tanımlamış olduğumuz renk bilgisi ve metin özellikleriyle ilgili bütün verileri (`color: red;text-decoration: none`) buraya aktaralım ve `style` kelimesini silelim. Ek olarak [index.html](#) dosyası içerisine bir başlık ekleyelim ve bu başlığın da yazı tipini değiştirelim. Stil dosyası içerisinde yapmak istediğimiz işleri maddeleyecek olursak:

- Bağlantı metninin rengini değiştirmek
- Bağlantı metninin altı çizili olmamasını sağlamak
- Başlığın fontunu değiştirmek (`h1` teği içerisinde `style` teği ile fontu değiştirmek yerine ilgili değişikliği CSS dosyası içerisinde yapacağız)

CSS dosyasında bu değişiklikleri yapabilmek için etkilenecek teği yazdıktan sonra süslü parantezler arasına teğin etkilenecek özelliklerini ve bu özelliklerin alacakları değerleri yazmak gerekmektedir.

index sayfasına dönecek olursak CSS dosyasına link verme işlemi **head** teği içerisinde yapılmaktadır. Bu link verme işlemi **link** teği ile yapılmakta ve **rel** (İngilizce bağlantı anlamına gelen “relation” kelimesinin kısaltması) anahtar kelimesi ile link verilen dosyaya hangi özelliğinden dolayı bağlantı verilmek istenildiğini belirtiriz. Burada ilgili dosyaya stil sayfası (style sheet) ya da stil dosyası olması sebebiyle bağlantı vermek istiyoruz. Bu yüzden ilgili kod satırı **link rel="stylesheet"** şeklinde olacaktır. Yine aynı kod satırında **href** teği ile CSS dosyasının dizini belirtilecektir.

İlgili kod parçacığı Şekil 9.5.'te verilmiş olup ilgili kod parçacığının çıktısı Şekil 9.6.'daki gibi olacaktır. İlgili şekilde CSS dosyası incelendiğinde üç adet font bilgisi girildiği görülmektedir. Bu kodlama, web sayfasını görüntüleyen kişinin bilgisayarının ilk fontu görüntüleyememesi durumunda sırasıyla hangi fontları kullanılabileceği bilgisini vermektedir.



The image shows a code editor with two tabs: 'index.html' and 'stil.css'. The 'index.html' tab is active and displays the following code:

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>ilk Web Sayfam!</title>
5     <meta charset="utf-8"/>
6     <link rel="stylesheet" href="stil.css">
7   </head>
8
9   <body>
10    <h1>Web Sayfam</h1>
11    <p><a href="hakkında/hakkında.html">
12      Bu linkten hakkımda bilgi edinebilirsiniz.</a></p>
13  </body>
14 </html>
15
```

The 'stil.css' tab is also active and displays the following code:

```
1 a{
2   color: red;text-decoration: none;
3 }
4
5 h1{
6   font-family: helvetica; arial; sans-serif;
7 }
```

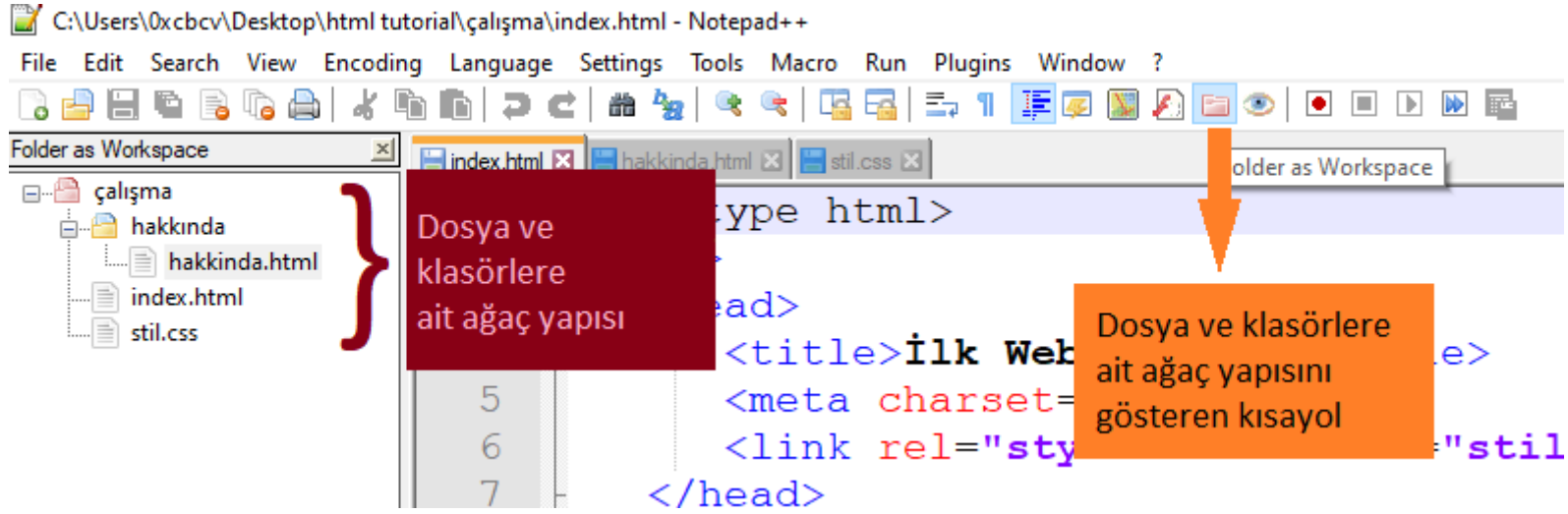
Şekil 9.5. İlgili kod

Web Sayfam

[Bu linkten hakkımda bilgi edinebilirsiniz.](#)

Şekil 9.6. İlgili ekran görüntüsü

Şekil 9.7.'de dosya ve klasörlerin ağaç yapısında görüntülemesini sağlayan kısayol verilmiştir. Ekranın sol tarafındaki ilgili ağaç yapısı incelendiğinde sadece birkaç tane olmasına rağmen dosyaların şimdiden karmaşık bir yapı oluşturmaya başladığı görülür. O yüzden resimleri, CSS dosyalarını, JavaScript dosyalarını kendi klasörlerinde tutmak iyi bir pratik olacaktır. Örneğin; stil diye bir klasör oluşturalım ve stil.css dosyasını bu klasörün içerisine atalım. Bununla beraber ilgili kod satırında dosya konumu olarak `href="stil/stil.css"` değişikliğini yapmamız gerekecektir.



Şekil 9.7. İlgili ekran görüntüsü

Bölüm 10 : Resim Dosyaları İle Çalışmak

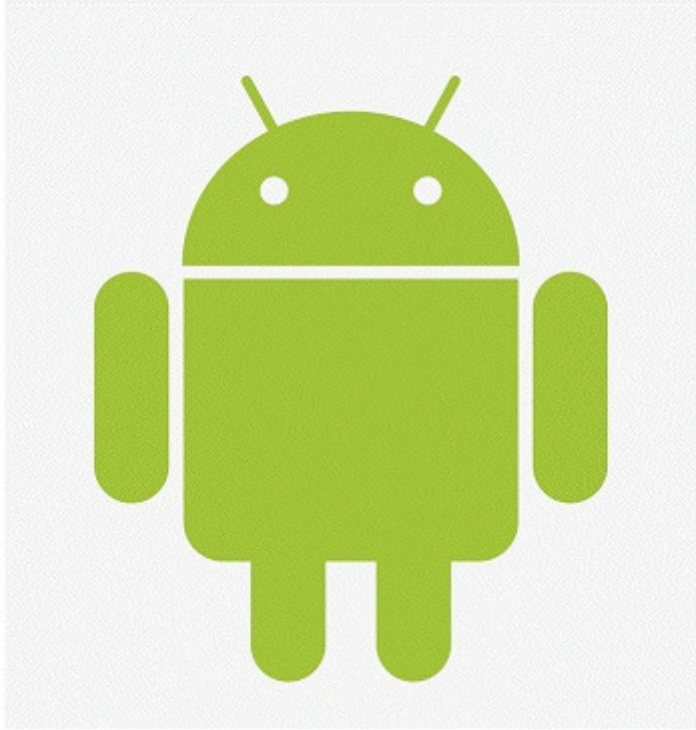
Oluşturacağımız “resim” klasörünün içerisine “android.jpg” isimli resim dosyasını kaydedelim. <body> tegi içerisinde herhangi bir yere

 satırını yazalım tegi resim dosyası ile işlem yapabilmemizi sağlarken resmin kaynağını **src** (source; kaynak anlamına gelen kelimesinin kısaltması) anahtar kelimesi ile belirtiyoruz. **alt** (alternative; alternatif anlamına gelen kelimesinin kısaltması) anahtar kelimesi ile resmin görüntülenememesi durumunda yerine görüntülenecek alternatif metin verilmektedir.

Ayrıca internette istediğiniz herhangi bir resim üzerine sağ tıklayarak “resim adresini kopyala” diyebilir ve bu adresi doğrudan kendi sayfanızdaki resim kaynağı kısmına yapıştırarak o resmi kendi sayfanızda görüntüleyebilirsiniz. Fakat bu iyi bir pratik değildir. İlgili ekran görüntüsü Şekil 10.1. de verilmiştir.

Web Sayfam

Bu linkten hakkımda bilgi edinebilirsiniz.



Şekil 10.1. İlgili ekran görüntüsü