

ETK229 Web Tasarımının Temelleri

Dersi

Ankara Üniversitesi Elmadağ Meslek Yüksekokulu

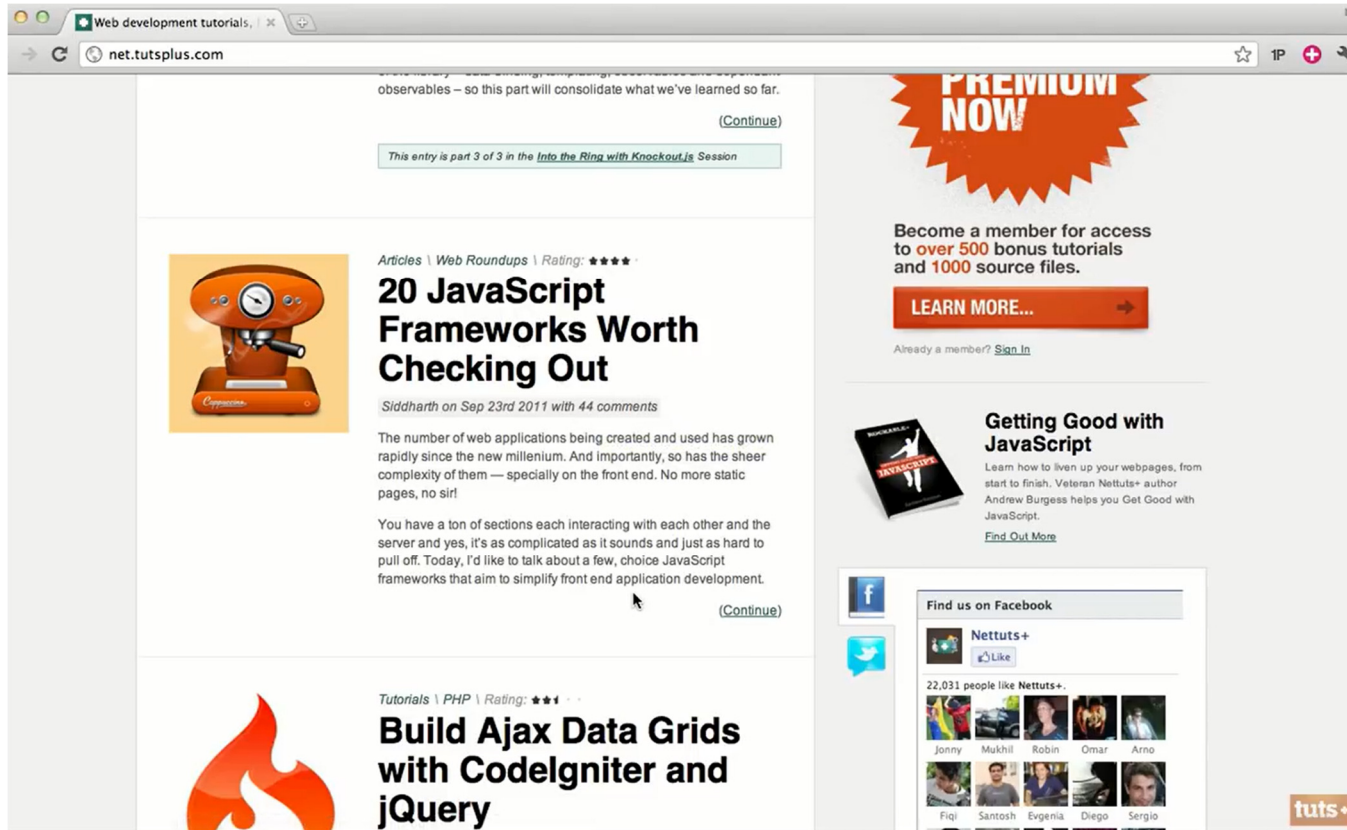
Öğretim Görevlisi : Murat Duman

Mail: mduman@ankara.edu.tr

14. Hafta

Bölüm 20 : Bölüm Oluşturma

<https://code.tutsplus.com/> sitesine ait ekran görüntüsü Şekil 20.1.'de verilmiştir.



Şekil 20.1. İlgili ekran görüntüsü

İlgili ekran görüntüsü incelendiğinde her bir gönderi için bir makale kısmı olduğu ve ilgili kısmın bütün gönderiler için aynı olduğu görülmektedir. Örneğin; hepsinde ilgili bir resim, bir başlık, gönderinin ne zaman ve kim tarafından yapıldığı, yorum sayısı, kategoriyle ilgili medya bilgisi (örneğin; gönderinin reytingi), iki paragraf, devam (Continue) kısmı vs. bulunmaktadır. Her biri kendine özgü ve siteden bağımsız olarak değerlendirilebilecek gönderiler birer makale olarak değerlendirilip [<article>](#) teği içerisine alınabilir. Sonrasında ilgili resim eklenebilir. İlgili gönderiye ait medya bilgisini liste şeklinde görüntületebiliriz. Bu liste elemanlarının her biri bir linke sahiptir. Reyting kısmını kodlamayacağız çünkü; bu kısım wordpress üzerine kurulu ilgili web sitesinin kullandığı bir plug-in tarafından gerçekleştirilmiştir. Şimdi yazar hakkındaki bilgileri gireceğiz. Bu kısmın orijinal web sayfasında nasıl gerçekleştirildiğini görmek için ilgili kısım üzerine sağ tıklayarak “incele” diyebiliriz. Burada yazar ismi ve yorumlar üzerinde link mevcuttur. Yorumlar kısmı gerçek bir uygulamada dinamik olarak oluşturulmaktadır; fakat burada sadece stil açısından kodlama yapılacaktır. Metinsel kısmı “metin” class isimli div içerisine yerleştirelim. Resme de linke ekleyelim.

css dosyasına geçmeden önce **margin** komutuna ait farklı kullanımlara bakalım:

margin: auto; içerik ortalanır

margin: 20px; bütün yönlerden (üst, alt, sağ, sol) 20 piksel mesafe bırakılır

margin: 20px auto; üst ve alt için 20 piksel mesafe ve sağ ile sol içinde otomatik mesafe bırakılır

margin-top: 20px; üstten 20 piksel mesafe

margin-bottom: 20px; alttan 20 piksel mesafe

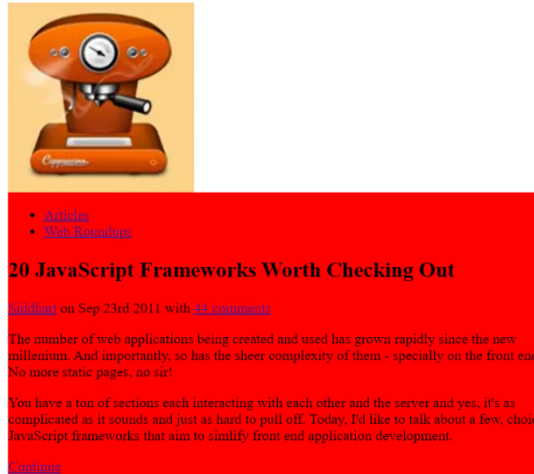
margin: 1px 2px 3px 4px; sırasıyla üstten, sağ, alt ve soldan mesafeler

margin: 10px 20px; sırasıyla üstten ve alttan 10 piksel, soldan ve sağdan 20 piksel

margin: 1px 2px 3px; 1 piksel üstten, sağdan 2 piksel, 3 piksel alttan ve sol mesafe belirtilmediği için sol için de sağa girilen mesafe yani 2 piksel geçerli olur

Aynı kullanım **padding** anahtar kelimesi için de geçerlidir.

Dikkat edilirse Şekil 20.1.'de sol taraftaki resmin altı boştur ve resim haricindeki bütün içerik sağ taraftadır. Bu nedenle sağ taraftaki bölümün tamamını sadece stil içerikli değişiklikler yapabilmek için “sag_taraf” class isimli div içerisine alalım. Bu divi tamamen konumlandırma (positioning) amaçlı kullandık. Resim için float: left; ve sag_taraf class isimli div için float: right; kod satırları girildiği halde elde edilen görüntü Şekil 20.2.'de görüldüğü gibidir. sag_taraf class isimli div için renklendirme yapıldığında bunun nedeninin div'in varsayılan görüntülenme şeklinin blok olmasından kaynaklandığı anlaşılır.



Şekil 20.2. İlgili ekran görüntüsü

Bu görüntüyü düzeltebilmek için div'in kapladığı alanın genişliğini resmin sol tarafına sığabilecek şekilde ayarlamaktır. Bu nedenle "sag_taraf" class isimli div için genişlik olarak 350 piksel girilebilir.

Başlığın font büyüklüğünü öğrenmek için sırasıyla sağ tıkla inspect (incele) ve sonrasında "computed" kısmından ilgili veri alınabilir. Her bir satır arasındaki boşluğu ayarlamak için **line-height** özelliği kullanılır.

Ayrıca medya class isimli div içerisindeki liste elemanlarının aynı satırda görüntülenmesi gerekmektedir. Bunun için css kodunda ilgili alana **display: inline;** kod satırı girilmelidir.

Şekil 20.1.'den görüleceği üzere her linkten sonra bir ayraç "\ " (divider) vardır.

```
.medya a:after{  
    content: " \005C";  
}
```

kod satırlarıyla medya class ismiyle tanımlanan sıralı olmayan listede bulunan liste elemanlarından sonra (after anahtar kelimesi yardımıyla) “\” işareti konulur. “\005C”; kodu css’te “\” karakterinin karşılığıdır. Bu kod yerine “\\” kullanılarak ta “\” karakteri oluşturulabilir. İlgili kelimeyle “\” karakteri arasında yeteri kadar boşluk bırakabilmek için padding anahtar kelimesi kullanılır. Şimdiye kadar yaptığımız değişikliklerle elde ettiğimiz web sayfası Şekil 20.3.’te verilmiştir.



[Articles](#) \ [Web Roundups](#) \

20 JavaScript Frameworks Worth Checking Out

[Siddhart](#) on Sep 23rd 2011 with [44 comments](#)

The number of web applications being created and used has grown rapidly since the new millenium. And importantly, so has the sheer complexity of them - specially on the front end. No more static pages, no sir!

You have a ton of sections each interacting with each other and the server and yes, it's as complicated as it sounds and just as hard to pull off. Today, I'd like to talk about a few, choice JavaScript frameworks that aim to simplify front end application development.

[Continue](#)

Şekil 20.3. İlgili ekran görüntüsü

Bu şekilde dikkat edilirse son liste elemanı olan “Web Roundups” kelimesinden sonra “\” karakteri gelmiştir. Halbuki Şekil 20.1.’de verilen orijinal web sayfasında son liste elemanından sonra herhangi bir karakter gelmediği görülmektedir. O halde bu son liste elemanını ortadan kaldırmamız gerekmektedir. Bunun için

```
.medya li:last-child:after{  
    content: 'none;  
}
```

kod satırlarını yazabiliriz. Bu arada liste elemanları ile başlık arasında belirli bir mesafe kalmıştır, bunu da ortadan kaldırmak için

```
.medya{  
    padding-left: 0px;  
}
```

kod satırlarını yazabiliriz.

Başığa ait web sayfasının üst kısmına kadar olan varsayılan (default) boşluğu gidermek için

```
article h2{  
    margin-top: 0px;  
}
```

kod satırlarını yazabiliriz. Kod satırına **article h2** şeklinde başladık; çünkü bu işlemde bütün h2 elemanlarının değil, sadece **article** teğindeki h2 elemanının etkilenmesini istiyoruz.

Gönderene ait bilgilerin bulunduğu kısım küçük bir kutu içerisinde bulunmaktadır. Bunun için ilgili paragrafı "gonderen_info" class ismiyle tanımlayalım:

```
<p class="gonderen_info">
```

ve css dosyasında bu class ismini kullanarak kutuyu oluşturalım. Bu kutuyu oluşturmada id değil class kullandık; çünkü aynı işlem web sayfasındaki bütün gönderiler için tekrarlanmaktadır.

İlgili kutu için

```
.gonderen_info{  
    background: #f3f3f3;  
    color: #999;  
    padding: 3px;  
}
```

kod satırlarını yazabiliriz.

Şekil 20.1.'de linklerin renkleri yeşil değildir. Bu değişikliği yapabilmek için

```
.gonderen_info a{  
    color: #999;  
}
```

kod satırlarını yazabiliriz.

Burada şunu da belirtmemiz gerekir ki ccs'te aşağıdaki kod satırları yukarıdaki kod satırlarını etkiler. Ancak yukarıdaki kod satırının ağırlığı daha fazlaysa bu gerçekleşmez.

Peki tam olarak ne demek istiyoruz! Örneğin;

```
<p class="gonderen_info"><a href="#">Siddhart</a> on Sep 23rd 2011 with  
    <a href="#">44 comments</a></p>
```

```
a{  
    color: green;  
}  
...  
a{  
    color: blue;  
}
```

Yukarıdaki kod satırları çalıştırıldığında link metni mavi renkli olarak görüntülenir. Çünkü link rengini değiştirmede en son kod etkindir. Ancak; ilgili bileşeni etkileyecek kod ilgili bileşeni ne kadar spesifik tanımlarsa o kadar ağırlığı artar ve css dosyasında daha yukarılarda kodlansa bile sonuç itibariyle bileşeni etkileyebilir. Örneğin;

```
<p class="gonderen_info"><a href="#">Siddhart</a> on Sep 23rd 2011 with  
    <a href="#">44 comments</a></p>
```

```
.gonderen_info a{  
    color: green;  
}  
...  
a{  
    color: blue;  
}
```

Yukarıdaki kod satırları çalıştırıldığında link metni yeşil olarak görüntülenir.

Linklerin altındaki çizgiyi kaldırmak için **text-decoration: none;** kod satırını kullanacağız. Ayrıca linklerin üzerinde gezinirken imleç resmi değişmektedir ve metin rengi siyah olmaktadır. Buraya kadar yazdığımız kodu çalıştırdığımızda elde ettiğimiz ekran görüntüsü Şekil 20.4.'te verilmiştir.



[Articles](#) \ [Web Roundups](#)

20 JavaScript Frameworks Worth Checking Out

Siddhart on Sep 23rd 2011 with 44 comments

The number of web applications being created and used has grown rapidly since the new millenium. And importantly, so has the sheer complexity of them - specially on the front end. No more static pages, no sirl

You have a ton of sections each interacting with each other and the server and yes, it's as complicated as it sounds and just as hard to pull off. Today, I'd like to talk about a few, choice JavaScript frameworks that aim to simlify front end application development.

[Continue](#)

Şekil 20.4. İlgili ekran görüntüsü

Şekil 20.4. incelendiğinde linkleri barındıran kutunun sağ tarafında fazladan boşluk bulunduğunu görürüz. Bu boşluğu kutuyu sola yaslayarak kaldırabiliriz.

Bu sefer de float işleminden kaynaklı Şekil 20.5.'te verilen taşma gerçekleşecektir.



[Articles](#) \ [Web Roundups](#)

20 JavaScript Frameworks Worth Checking Out

The
Siddhart on Sep 23rd 2011 with 44 comments

number of web applications being created and used has grown rapidly since the new millenium. And importantly, so has the sheer complexity of them - specially on the front end. No more static pages, no sir!

You have a ton of sections each interacting with each other and the server and yes, it's as complicated as it sounds and just as hard to pull off. Today, I'd like to talk about a few, choice JavaScript frameworks that aim to simlify front end application development.

[Continue](#)

Şekil 20.5. İlgili ekran görüntüsü

Bu taşma metin isimli class'la tanımlanan bölüme `clear: both;` komutu satırı girilerek ortadan kaldırılabilir. Çünkü taşma bu kısımda gerçekleşmiştir.

Şimdi de başlıkla linkler arasındaki varsayılan boşluğu kaldıracacağız.

Bunun için

```
article h2{ margin-top: 0px; margin-bottom: 0px }
```

komut satırını ya da aynı anlama gelen

```
article h2{ margin: 0px }
```

komut satırını kullanacağız.

Şekil 20.1.'de en üstteki metinle resmin üst kısmının aynı hizada olduğunu görürüz.

Ancak; bu durum Şekil 20.5.'te yoktur. İlgili düzenlemeyi yapabilmek için medya class ismi ile tanımlanan sıralı olmayan liste için css dosyasında

```
margin: 0px 0px 6px;
```

komut satırını gireceğiz.

Başlık kısmında harfler arasındaki mesafeyi ayarlamak için

```
letter-spacing: -1px;
```

komut satırını kullanacağız.

Sonrasında linklerin bulunduğu kısım ile kutunun geri kalanındaki metnin font büyüklüğünü ve paragraflardaki ana metnin font büyüklüğünü; link metninin rengi ile paragraflardaki ana metnin rengini eşitleyeceğiz.

Paragraflardaki ana metnin satır yüksekliklerini

`line-height: 21px;`

komut satırı ile ayarlayacağız.

Şimdi en alttaki “Continue” kelimesini sayfanın sağ tarafına çekmek için kelimenin bulunduğu link kısmını “devam” class ismiyle tanımlayacağız:

```
<a href="#" class="devam">Continue</a>
```

Sonrasında css dosyasına

```
.devam{
```

```
    float: right;
```

```
}
```

kod satırlarını gireceğiz.

Orijinal sayfada “Continue” kelimesi parantez içerisindedir. Bunu css dosyasına

```
.devam:after{  
    content: ')'  
}
```

```
.devam:before{  
    content: '('  
}
```

kod satırlarını girerek gerçekleştirebiliriz.

Ayrıca link metinlerinin üzerine gelindiğinde yazı renginin siyah olmasını istiyorsak:

```
a:hover{  
    color: black;  
}
```

kod satırlarını kullanabiliriz.

Web sayfasının son hali Şekil 20.6.'da, index.html sayfasına ait kod Şekil 20.7.'de ve css sayfasına ait kod Şekil 20.8.'de verilmiştir.



[Articles](#) \ [Web Roundups](#)

20 JavaScript Frameworks Worth Checking Out

Siddhart on Sep 23rd 2011 with 44 comments

The number of web applications being created and used has grown rapidly since the new millenium. And importantly, so has the sheer complexity of them - specially on the front end. No more static pages, no sir!

You have a ton of sections each interacting with each other and the server and yes, it's as complicated as it sounds and just as hard to pull off. Today, I'd like to talk about a few, choice JavaScript frameworks that aim to simplify front end application development.

[\(Continue\)](#)

Şekil 20.6. İlgili ekran görüntüsü

```

6 <body>
7 <article>
8   <a href="#">
9     
10    </a>
11    <div class=sag_taraf>
12      <ul class="medya">
13        <li><a href="#">Articles</a></li>
14        <li><a href="#">Web Roundups</a></li>
15      </ul>
16      <h2>20 JavaScript Frameworks Worth Checking Out</h2>
17      <p class="gonderen_info"><a href="#">Siddhart</a> on Sep 23rd 2011 with
18      <a href="#">44 comments</a></p>
19      <div class="metin">
20        <p>The number of web applications being created and used has
21        grown rapidly since the new millenium. And importantly, so
22        has the sheer complexity of them - specially on the front end.
23        No more static pages, no sir!</p>
24        <p>You have a ton of sections each interacting with each other and
25        the server and yes, it's as complicated as it sounds and just as
26        hard to pull off. Today, I'd like to talk about a few, choice
27        JavaScript frameworks that aim to simlify front end application
28        development.</p>
29      </div>
30      <a class="devam" href="#">Continue</a>
31    </div>
32 </article>
33 </body>
34 </html>

```

Şekil 20.7. İlgili ekran görüntüsü

```

body{
  width: 600px;
  margin: 20px auto;
  font-family: sans-serif;
}

h2{
  font-size: 30px;
  line-height: 30px;
  letter-spacing: -1px;
}

a{
  color: green;
  font-size: 14px;
}

a:hover{
  color: black;
}

p{
  font-size: 14px;
  line-height: 21px;
}

article img{
  float: left;
  margin-right: 30px;
}

```

```

article h2{
  margin: 0px;
}

.devam{
  float: right;
}

.devam:after {
  content: ') '
}

.devam:before {
  content: '('
}

.medya{
  padding-left: 0px;
  margin: 0px 0px 6px;
}

.medya a, .gonderen_info a{
  font-style: italic;
  font-size: 14px;
}

.medya li{
  display: inline;
}

```

```

.medya li:after{
  content: "\\\";
  padding: 0 3px;
  color: #999;
}

.medya li:last-child:after{
  content: none;
}

.sag_taraf{
  float: right;
  width: 350px;
}

.gonderen_info{
  background: #f3f3f3;
  color: #999;
  padding: 3px;
  float: left;
  font-size: 14px;
}

.gonderen_info a{
  color: #999;
  text-decoration: none;
}

.metin{
  clear: both;
  color: #575757;
}

```

Şekil 20.8. İlgili ekran görüntüsü