

# İçindekiler

<b>1. Giriş</b>	<b>2</b>
<b>2. HTML</b>	<b>2</b>
2.1. Başlıklar	4
2.2. Paragraflar	5
2.3. Satır Sonları	5
2.4. Metin Biçimlendirme Tagları	6
2.5. Font Kullanımı	7
2.6. Resim Ekleme	7
2.7. Link Ekleme	7
2.8. Tablolar	8
2.9. Yatay Çizgi	12
2.10. LİSTELER	13
2.10.1. Sıralı listeler	13
2.10.2. Sırasız Listeler	15
2.11. Sayfa Özellikleri	16
2.12. Özel Karakterler	16
2.13. CSS	16
2.13.1. Metin Biçimlendirme	16
2.13.2. Linkler	18
2.14. Uygulama 1	19
2.15. Uygulama 2	20
2.16. Uygulama 3	21
2.17. ÇERÇEVELER	21
<b>3. CSS (STİL ŞABLON)</b>	<b>25</b>
3.1. STİL ŞABLON ÇEŞİTLERİ	26
3.1.1. Yerel Stil Şablonu	26
3.1.2. Global Stil Şablonu	26
3.1.3. Bağlantılı Stil Şablon	27
3.2. HTML ETİKETLERİ İLE CSS	28
3.2.1. Font Özellikleri	28
3.2.2. Text Özellikleri	30
3.2.3. Background Özellikleri	31
3.2.4. List Özellikleri	32
3.2.5. Position Özelliği	33
3.3. SEÇİCİLER (SELECTORS)	35
3.3.1. Class Selector (Sınıf Seçicisi)	36
3.3.2. id selector (id seçicisi)	37
3.4. CSS GENEL KULLANIM ŞEKİLLERİ	38

# 1. Giriş

- URL
- Google arama servisinde arama ipuçları
  - Normal arama
    - Normal arama
    - Birden fazla kelime kullanarak arama
    - Birden fazla kelimeyi “” işaretleri içinde kullanma
    - Site: parametresi
    - Domain: parametresi
    - Filetype: parametresi
  - Resim arama
- DNS kavramı
- İstemci – Sunucu kavramları
- Hosting, Free Hosting, Domain kavramları
- Web sitesi – web sayfası kavramları
- Web sayfası tasarımında dikkat edilmesi gereken noktalar
  - Sitenizin yükleme hızını dikkate alarak sayfada yer alan nesnelerin dosya boyutlarını ayarlayın,
  - Sitenizi ziyaret eden ziyaretçilerin farklı bilgisayar donanımı ve yazılımlarına sahip olduklarını dikkate alarak hazırladığınız web sayfalarını farklı bilgisayarlarda, tarayıcılarda ve ekran çözünürlüklerinde görüntüleyin,
  - Sayfalarınızı hazırlarken ziyaretçilerin görünüşü rahatsız edecek şekilde uyumsuz renkleri kullanmaktan kaçının,
  - Sitenizi oluşturan sayfaların tümünde aynı renk tonlarını ve yazı tiplerini kullanmaya çalışın,
  - Sayfanızın en iyi görüldüğü ekran çözünürlüğü ve tarayıcı ismini sayfanızın belirli bir bölümünde küçük bir not ile belirtebilirsiniz,
  - Sayfalarınızda çok fazla sayıda hareketli görüntüler (animasyonlar) kullanmaktan kaçının,
  - Tasarım tamamlandıktan sonra ve sayfalar yayınlanmadan önce sayfa üzerindeki tüm linklerin (köprü) çalıştığını kontrol edin.

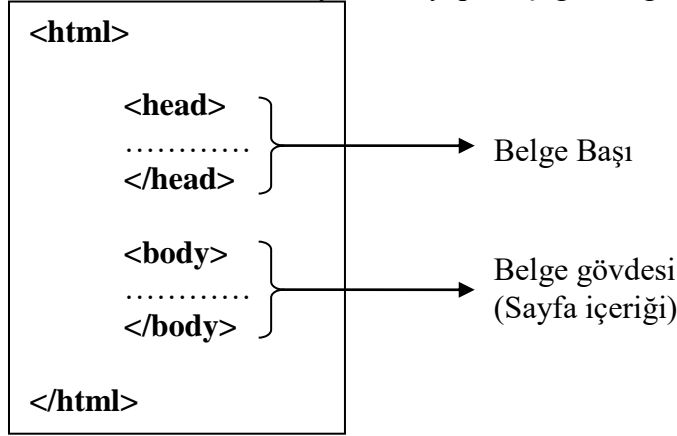
## 2. HTML

HTML bilmeden web sayfaları oluşturmak, otomatik pilot kullanmaya ve nasıl uçulacağını bilmemeye benzer, pek güvenli değildir.

Web sayfası oluşturmak için HTML bilmek gerekmebilse de, Web sayfanızda bir şeyler istediğiniz gibi çalışmazsa ya da benzersiz bir şeyler yapmak için sınırları zorluyorsanız, HTML bilmenin değeri ölçülemez. HTML bilmek, web sayfalarınız üzerinde tam denetimi garantiler.

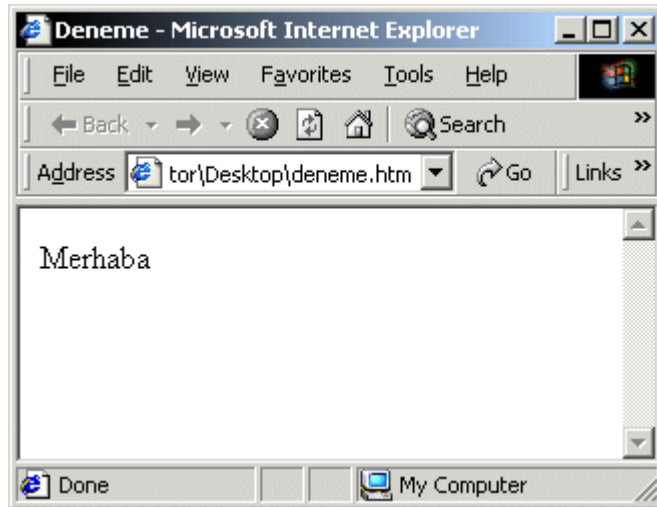
HTML (Hyper Text Markup Language) bir döküman formatlama dilidir. Web sayfaları temel olarak HTML taglarından (etiket) oluşur. Taglar ile web sayfasındaki yazılar biçimlendirilir, resim veya link eklenir. Bir web sayfasının kaynağına bakarsanız bu HTML taglarını görebilirsiniz. Browser bu tagları yorumlayarak web sayfasını görüntüler. Taglar < ile başlar ve > ile biter. HTML, büyük-küçük harf duyarlı bir dil değildir.

Temel olarak bir web sayfasının yapısı aşağıdaki gibidir.



Basit bir web sayfasının kodları aşağıdadır:

```
<html>
<head>
<title>Deneme</title>
</head>
<body>
<p>Merhaba</p>
</body>
</html>
```



Bu HTML kodlarını bir text editörü ile yazabilirsiniz. Bu iş için “not defteri” bile yeterlidir. HTML kodlarını yazdıktan sonra “htm” veya “html” uzantısı ile kaydedip, bir browser ile açarak sayfanızı görüntüleyebilirsiniz.

Her tag genellikle <title> </title> örneğinde olduğu gibi çifttir. Bütün taglar kapatılmalıdır. Ancak bunun bazı istisnaları vardır; örneğin <br> tagı gibi. Taglar açılma sırasına göre kapatılmalıdır. İlk açılan tag en son kapatılır. Tagları yazımında büyük harf veya küçük harf kuralı yoktur. Tagların tamamını büyük harfle yazabileceğiniz gibi küçük harflerle de yazabilirsiniz. Hatta büyük ve küçük harfleri karışık bir şekilde kullanarak taglar yazılabilir.

Web sayfalarında ilk tag <html> tagıdır. Bu tag browser’a HTML dökümanın başladığını bildirir. </html> tagı ise HTML dökümanın bittiğini belirtir.

<head> ve </head> arasına ise sayfayla ilgili bilgiler yazılır. Örneğin sayfanın başlığı burada belirtilir.

Sayfanın başlığı <title> ile </title> arasına yazılır. Sayfa başlığı browser penceresinin başlık çubuğunda görüntülenir.

<body> ile </body> arasına ise çeşitli biçimlendirme tagları yazılır.

## 2.1. Başlıklar

Başlık yazmak için <h1> den <h6> ya kadar olan taglar kullanılır. En büyük başlık için <h1>, en küçük başlık için ise <h6> kullanılır. Başlık taglarından sonra yazılan yazılar bir alt satırda görüntülenir.

```
<html>
<head>
<title>Deneme</title>
</head>
<body>
<h1>Başlık 1</h1>
<h2>Başlık 2</h2>
<h3>Başlık 3</h3>
<h4>Başlık 4</h4>
<h5>Başlık 5</h5>
<h6>Başlık 6</h6>
</body>
</html>
```

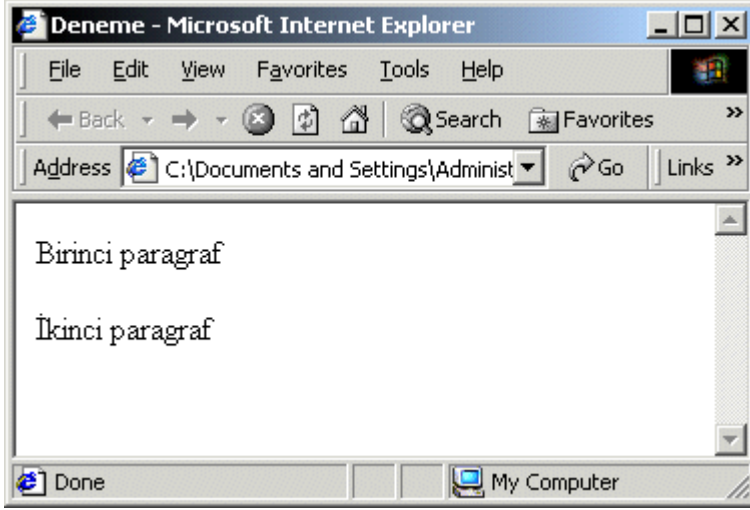
Yukarıdaki HTML kodları ile oluşturulan sayfanın browser'daki görünümü aşağıdaki gibidir.



## 2.2. Paragraflar

Paragraflar <p> tagı ile tanımlanır. Bu tag kullanıldığında paragraftan sonra boş bir satır meydana gelir. Örnek olarak:

```
<p>Birinci paragraf</p>
<p>İkinci paragraf</p>
```



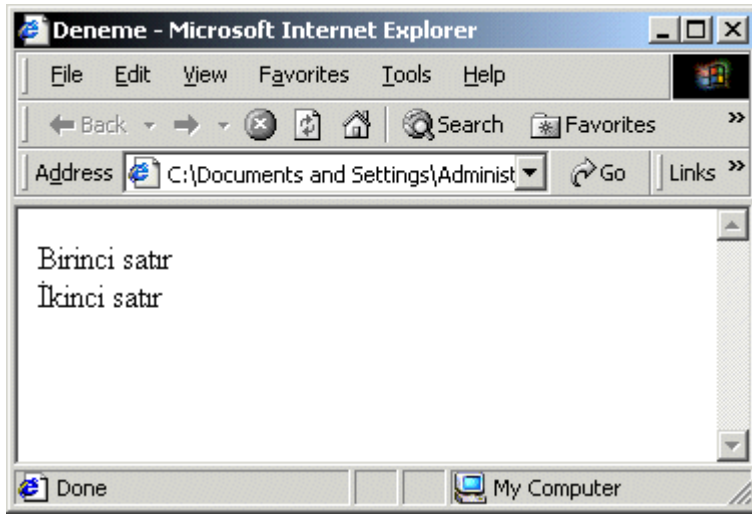
Metni hizalamak için “align” özelliği kullanılır. Sola yaslamak için “left”, sağa hizalamak için “right”, ortalamak için “center”, iki yana yaslamak için “justify” kullanılır. Örneğin:

```
<p align="center">Merhaba</p>
```

## 2.3. Satır Sonları

<p> tagı kullanıldığında paragraftan sonra yazılan yazılar arasında bir satır boşluk meydana gelir. Bunu istemiyorsanız <br> tagını kullanmanız gerekir. <br> tagından sonra yazılan yazılar arada boşluk olmadan bir alt satırda görüntülenir. Diğer tagların aksine <br> tagının kapatılması gerekmemektedir. Örnek olarak:

```
Birinci satır<br>
İkinci satır<br>
```



Yazıya bir alt satırdan devam etmek istiyorsanız mutlaka <p> veya <br> taglarından birini kullanmanız gerekir. Not defterinde satırın sonunda “enter” tuşuna basıp alt satırdan yazmaya devam etmeniz bile browser bunu göz önüne almayacaktır. Bu yazdıklarınızı yanyana görüntüleyecektir.

## 2.4. Metin Biçimlendirme Tagları

### Koyu Yazı

Koyu yazı <b> tagı ile tanımlanır. Örneğin:

<b>Koyu yazı</b>

### Eğik yazı

Eğik yazı <i> tagı ile tanımlanır. Örneğin:

<i>Eğik yazı</i>

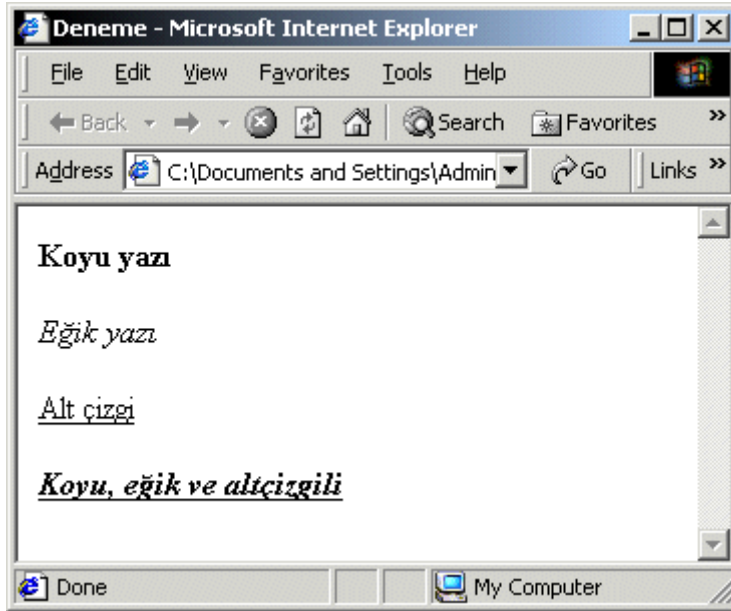
### Alt Çizgi

Altı çizgili yazı yazmak için <u> tagı kullanılır. Örneğin:

<u>Alt çizgi</u>

Koyu, eğik ve alt çizgili bir metin yazmak için bu üç tagın kullanımı ise:

<b><i><u>Koyu, eğik ve altçizgili</u></i></b>



## 2.5. Font Kullanımı

HTML’de fontlar <font> tagı ile tanımlanır.

```
<font face="Verdana" size="2" color="#008080">Deneme</font>
```

Kullanılacak fontun ismi “face” özelliği ile tanımlanır. Font büyüklüğü “size” özelliği ile tanımlanır. Renk ise “color” özelliği ile tanımlanır. Renk heksadesimal kodlarla olduğu gibi rengin ingilizce karşılığını yazarak da tanımlanabilir. Heksadesimal kodlamada # karakterinden sonra sırasıyla kırmızı, yeşil ve mavi renklerin değerleri tanımlanır. Renk bu üç temel rengin karışımından elde edilir. Örneğin kırmızı renk için color="#ff0000" veya color="red" kullanılabilir.

## 2.6. Resim Ekleme

Web sayfasına resim eklemek için <img> tagı kullanılır.

```

```

Resim adı “src” özelliği ile tanımlanır.

Align özelliği resmi hizalamak için kullanılır. Sola yaslamak için “left”, sağa yaslamak için “right” ve ortalamak için “center” kullanılır.

“width” resmin genişliğini, “height” yüksekliğini tanımlar.

“alt” özelliği ile fare imleci resim üzerine geldiğinde görünecek metin belirlenir.

“border” ise resmin çerçeve kalınlığını belirlemek için kullanılır.

## 2.7. Link Ekleme

Herhangi bir html sayfaya, resme veya herhangi bir dosyaya link vermek için <a> tagı kullanılır. Örneğin Süleyman Demirel Üniversitesi’ne link vermek için:

```
<a href="http://www.sdu.edu.tr">Süleyman Demirel Üniversitesi</a>
```

“href” ile link verilecek url, sayfa veya dosya adı tanımlanır. Bir HTML sayfaya link verilecekse <a href="sayfa.html"> şeklinde kullanılır. Eğer dosya bir üst dizindeyse <a href="../sayfa.html"> şeklinde kullanılır.

Link vermek için resim kullanılacaksa:

```
<a href="http://www.sdu.edu.tr">

</a>
```

Linkin yeni sayfada açılması için ise şöyle bir kod yazılabilir:

```
<a href="http://www.sdu.edu.tr" target="_blank">Süleyman Demirel Üniversitesi</a>
```

Bir sayfa içinde farklı kısımlara link vermek için ilk önce sayfadaki link verilecek yerler tanımlanmalıdır. Örneğin sayfamızda içindekiler bölümüne link verelim. İçindekilerin başladığı yere

```
<a name="icindekiler">İçindekiler</a>
```

yazılır. Buraya link vermek için ise

```
<a href="#icindekiler">İçindekiler</a>
```

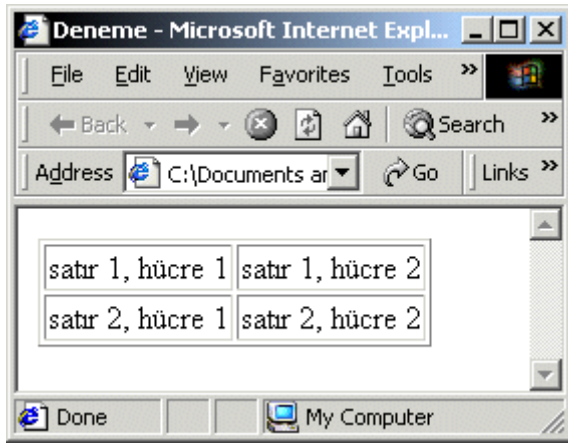
kullanılır.

## 2.8. Tablolar

Tablolar <table> tagı ile oluşturulur. Tablo oluşturulurken önce tablo satırları <tr> tagı ile oluşturulur. Sonra oluşturulan satır <td> tagı ile hücrelere bölünür. Tablo içine yazılacak bilgiler <td> ve </td> tagları arasına yazılır.

```
<table border="1">
  <tr>
    <td>satır 1, hücre 1</td>
    <td>satır 1, hücre 2</td>
  </tr>
  <tr>
    <td>satır 2, hücre 1</td>
    <td>satır 2, hücre 2</td>
  </tr>
</table>
```

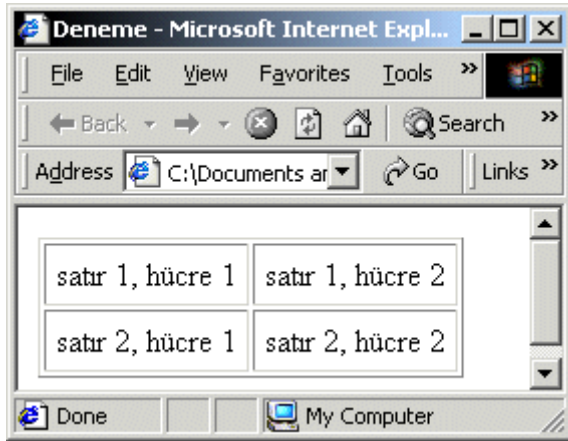




“border” özelliği tablonun çerçevesinin kalınlığını belirler. 0 değeri verilirse tablo çerçevesi gözükmeyecektir.

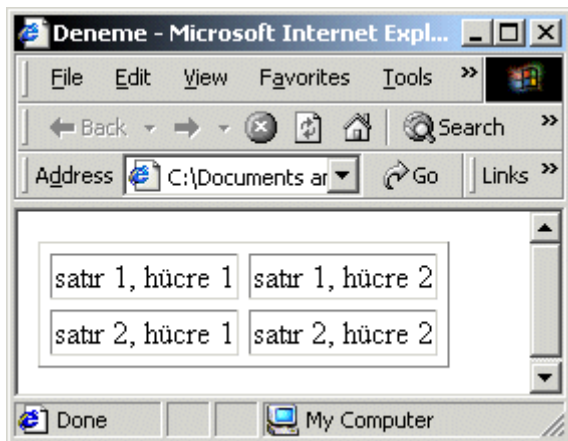
“cellpadding” hücre kenarları ile yazılar arasındaki boşluğu belirler. Kullanımı:

```
<table border="1" cellpadding="5">
```



“cellspacing” hücreler arasındaki boşluğu belirler. Kullanımı:

```
<table border="1" cellspacing="5">
```



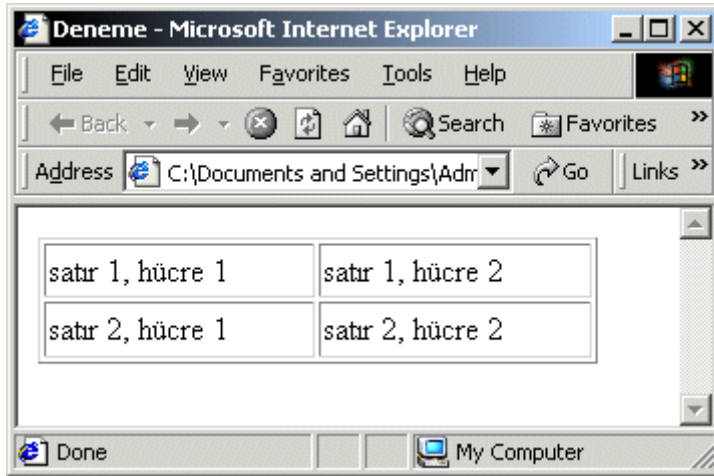
Tablonun boyutları hücre içindeki metinlerin büyüklüğüne göre browser tarafından belirlenir.

Tablonun boyutları sabit tutulmak isteniyorsa “width” ve “height” özellikleri kullanılır.

“width” ve “height” değerleri yüzde ve piksel olmak üzere iki şekilde belirlenir. Eğer yüzde olarak tanımlanırsa tablo boyutu browser penceresinin boyutuna göre değişir. Piksel olarak

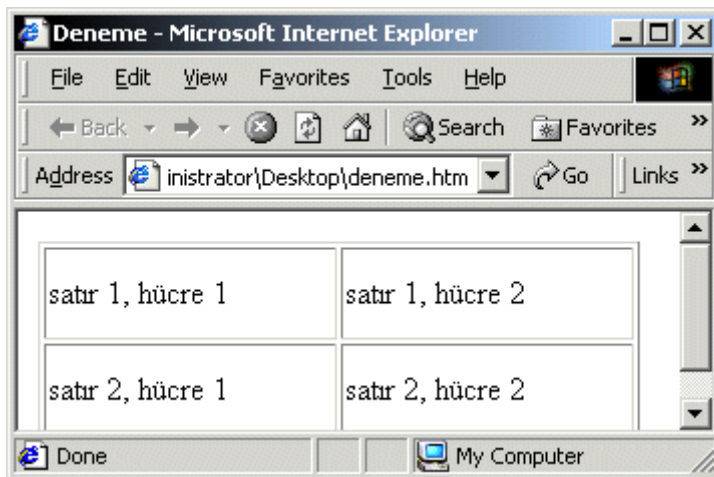
tanımlanırsa tablo boyutu sabit olur. Eğer tablo boyutu pencere boyutundan büyükse tablonun bir kısmı gözükmeyecektir. Yüzde olarak tanımlanırsa:

```
<table border="1" width="90%" height="80%">
```



Piksel olarak tanımlanırsa:

```
<table border="1" width="250" height="100">
```



Hücrelerin boyutlarını ayarlamak için ise “width” ve “height” <td> tagıyla kullanılmalıdır.

Tablo içindeki metinlerin hizalanması için “align” özelliği kullanılır.

Bir satırdaki tüm hücrelerdeki metinlerin hizalanması için <tr> tagıyla kullanılır. Örneğin:

```
<tr align="center"><td></td><td></td></tr>
```

şeklinde kullanılabilir. Sadece bir hücredeki metinlerin hizalanması için <td> tagıyla kullanılır. Örneğin:

```
<tr><td align="center"></td><td></td></tr>
```

Tablonun arkaplan rengini değiştirmek için “bgcolor” özelliği kullanılır. Tablonun tamamının arkaplan rengini değiştirmek için:

```
<table border="1" bgcolor="#66CCFF">
```

Sadece bir hücrenin değiştirmek için ise

```
<td bgcolor="#66CCFF">
```

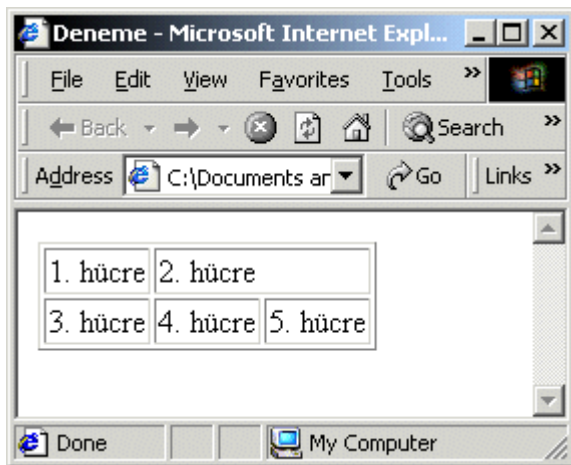
Arkaplan olarak resim kullanmak istenirse “background” özelliği kullanılır.

```
<table border="1" background="resim.gif">
```

Hücre yüksekliğinin diğer hücrelerden daha fazla olması için “rowspan”, genişliğinin daha fazla olması için “colspan” kullanılır.

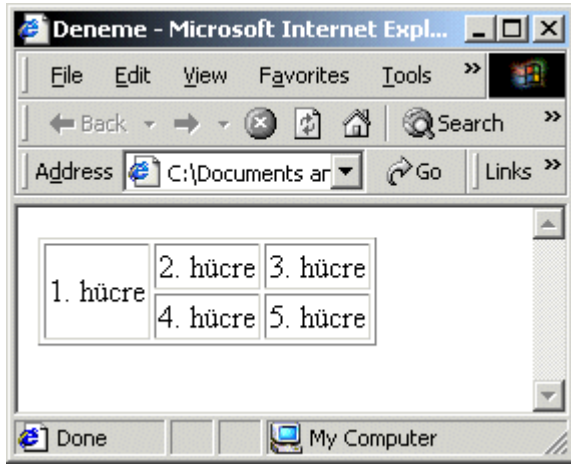
### “colspan” kullanımı

```
<table border="1">
  <tr>
    <td>1. hücre</td>
    <td colspan="2">2. hücre</td>
  </tr>
  <tr>
    <td>3. hücre</td>
    <td>4. hücre</td>
    <td>5. hücre</td>
  </tr>
</table>
```



### “rowspan” kullanımı

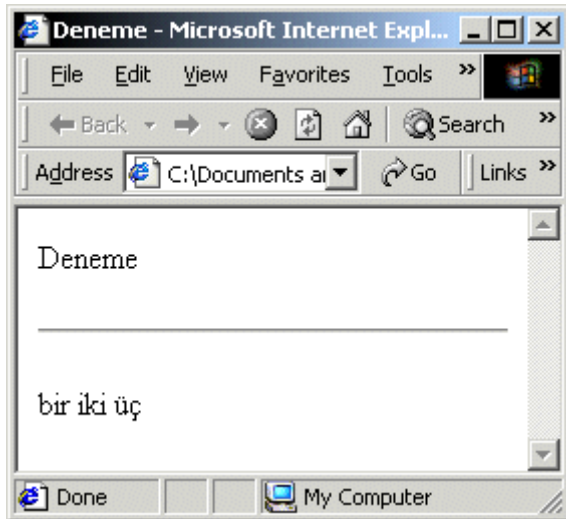
```
<table border="1">
  <tr>
    <td rowspan="2">1. hücre</td>
    <td>2. hücre</td>
    <td>3. hücre</td>
  </tr>
  <tr>
    <td>4. hücre</td>
    <td>5. hücre</td>
  </tr>
</table>
```



## 2.9. Yatay Çizgi

Yatay çizgi oluşturmak için `<hr>` tagi kullanılır. Bu tagı kapatmaya gerek yoktur. Örneğin:

```
<html>
<head>
<title>Deneme</title>
</head>
<body>
    <p>Deneme</p>
    <hr>
    <p>bir iki üç</p>
</body>
</html>
```



## 2.10. LİSTELER

HTML bize üç tip liste hazırlama imkânı veriyor. Bunlar;

Sıralı listeler (ordered list)

Sırasız listeler (unordered list)

Tanımlama listeleri (definition list)

Sıralı listeler rakam veya harf ya da her ikisini içice kullanarak liste oluşturmamızı, Sırasız listeler rakam/harf yerine madde imleri koyarak liste oluşturmamızı sağlar. Tanımlama listeleri ise bir listeden çok kalabalık metinlerde okumayı kolaylaştırmaya yardımcı olabilecek bir araçtır.

### 2.10.1. Sıralı listeler

Liste içine alınacak metinler **<ol>...</ol>** etiketleri arasına alınarak yazılır. Bu etiketler listenin başladığını ve bittiğini belirtir. Listenin maddelerinin başına ise **<li>** (**list item**) etiketini getiriyoruz. Bu etikette tıpkı **<br>** etiketi gibi sonlandırılmıyor. **<ol>** etiketine parametreler ekleyebiliyoruz. Bunlarla listemizin rakamla mı harfle mi başlayacağını (**type**) yada hangi rakam/harfle başlayacağını (**start**) belirtebiliyoruz. Compact parametresi ise listenin mümkün olan minimum satır aralığına sahip olmasını sağlıyor.

Bundan sonraki örneklerimizde sayfa kodunun yalnız body (**gövde**) bölümünü vereceğiz. Kodun geri kalan kısımlarını kendi sayfanızda tam olarak yazmayı unutmayın.

```
<ol type="1">
  <li>Kimya
  <ol type="a">
    <li>Inorganik
    <li>Analitik
  </ol>
  <li>Fizik
  <ol type="a">
    <li>Dinamik
    <li>Statik
```

```
</ol>
li>Matematik
<ol type="a">
  <li>Sayılar
  <li>Diğer
<ol type="i">
  <li>Türev
li>Integral
</ol>
</ol>
</ol>
```

1. Kimya
  - a. Organik
  - b. İnorganik
  - c. Analitik
2. Fizik
  - a. Dinamik
  - b. Statik
3. Matematik
  - a. Sayılar
  - b. Diğer
    - i. Türev
    - ii. Integral

Listeleri buradaki örnekte olduğu gibi iç içe hazırlamak ta mümkün. Dikkat edeceğimiz nokta, ise <ol> etiketi ile başlayıp liste maddelerinin her birisinin başına <li> etiketini getirmek ve listelemeyi bitirmek istediğimiz yerde </ol> etiketini yazmak. Liste içinde yeni bir liste oluşturmak istediğimizde listelenecek maddeden sonra tekrar <ol> etiketini yazıyoruz ve bahsedilen kuralları aynen uyguluyoruz. **Type** parametresinde kullanabileceğimiz değerler şunlar olabilir; sayılar,harfler (küçük/büyük)ve romen rakamları (i,ii,iii gibi)

### 2.10.2. Sırasız Listeler

Bu tip listede de mantık aynı. Fark, listeleme yaparken maddelerin başına harf, rakam gibi unsurlar yerine küçük yuvarlaklar, kareler kullanabilmemiz. **<ol>** etiketi yerine **<ul>** etiketini kullanıyoruz, maddeler için kullandığımız **<li>** etiketi burada da geçerli. **<ol>** için parametreler ise şöyle; type için disc (**içi dolu daire**), circle (**içi boş daire**), square (**içi dolu kare**). Compact parametresi Sırasız listelerde de kullanılabiliyor.

```
<ul type="disc">
<li>Kimya
<ul type="square">
<li>Inorganik
<li>Analitik
</ul>
<li>Fizik
<ul type="square">
<li>Dinamik
<li>Statik
</ul>
<li>Matematik
<ul type="square">
<li>Sayılar
<li>Diğer
<ul type="circle">
<li>Türev
<li>Integral
</ul>
</ul>
</ul>
</ul>
```

```
Kimya
  Organik
  Inorganik
  Analitik
Fizik
  Dinamik
  Statik
Matematik
  Sayılar
  Diğer
    Türev
    Integral
```

## 2.11. Sayfa Özellikleri

### Sayfa Zemin Rengi ve Resmi

Sayfanın zemin rengini değiştirmek için <body> tagının “bgcolor”, zemin resmi eklemek için “background”, özellikleri kullanılır. Örneğin:

```
<body bgcolor="#FFFFFF" background="resim.gif">
```

### Sayfa Kenar Boşlukları

Sayfanın kenar boşlukları <body> tagının margin özellikleri ile ayarlanır. Sol kenardan boşluk “leftmargin”, üstboşluk ise “topmargin” özelliği ile ayarlanır. Örneğin:

```
<body topmargin="10" leftmargin="10">
```

## 2.12. Özel Karakterler

Bazı karakterleri yazmak için kodlarını kullanmak gerekir. HTML kodları içerisinde ne kadar boşluk kullanılırsa kullanılsın sadece bir tane boşluk görüntülenir. Sayfa içerisinde ard arda gelen boşluk yazmak için “&nbsp;” kullanmak gerekir.

Sembol	Kod	Örnek
Copyright	&copy;	©
Registered trademark	&reg;	®
Trademark	&#8482;	™
Küçük	&lt;	<
Büyük	&gt;	>
Ampersand	&amp;	&
Boşluk	&nbsp;	
Tırnak	&quot;	"

## 2.13. CSS

CSS (Cascading Style Sheets) HTML’nin bazı eksik yönlerini kapatmak amacıyla geliştirilmiştir.

Stiller <head> </head> tagları arasında tanımlanır. CSS’deki özelliklerin değerleri yazılırken tırnak işareti kullanılmaz. Örneğin sayfanın özelliklerini CSS ile değiştirmek için:

```
<style type="text/css">
  body { color: black; background: white; }
</style>
```

Burada “color” metnin rengini, “background” ise arkaplan rengini belirler.

### 2.13.1. Metin Biçimlendirme

CSS ile metin biçimlendirmeyi bir örnekle açıklayalım:

```
<style type="text/css">
  p { font-family: verdana, Garamond; font-size:10pt; color:green;}
</style>
```



Yukarıdaki örnekteki stil ile paragrafların biçimi değiştirilmiştir. <p> tagı kullanıldığında bu stiller geçerli olur. Bu sayede her paragrafta metnin biçimini belirtmeye gerek kalmaz, <p> tagını kullanmak yeterlidir. Font türü “font-family” özelliği ile tanımlanır. İlk olarak tanımlanan font kullanılır, bu font kullanıcının bilgisayarında yoksa ikinci font kullanılır. Font büyüklüğü “font-size”, rengi ise “color” özelliği ile tanımlanır.

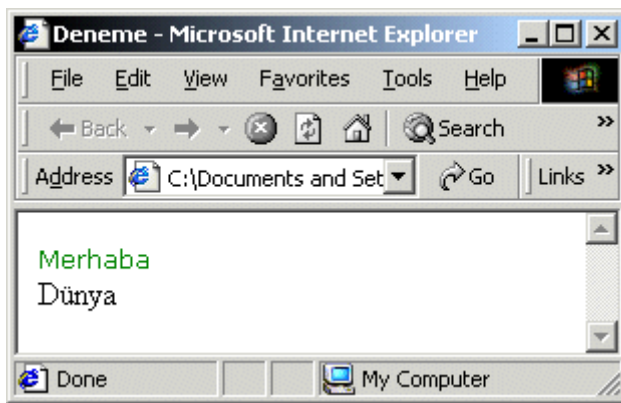
Kalın yazı için “font-weight: bold”, eğik yazı için “font-style: italic” kullanılır.

Eğer her paragrafta aynı stili kullanmak istemiyorsanız aşağıdaki gibi kullanmalısınız.

```
<html>
<head>
<title>Deneme</title>
<style type="text/css">
    p.yazi { font-family: verdana, Garamond; font-size:10pt; color:green;}
</style>
</head>
<body>
    <p class="yazi">Merhaba</p>
</body>
</html>
```

<p> tagında olduğu gibi stiller diğer taglarla da kullanılabilir. Eğer standart taglar kullanılmadan stiller kullanılmak isteniyorsa:

```
<html>
<head>
<title>Deneme</title>
<style type="text/css">
    div.yazi { font-family: verdana, Garamond; font-size:10pt; color:green;}
</style>
</head>
<body>
    <div class="yazi">Merhaba</div> Dünya
</body>
</html>
```



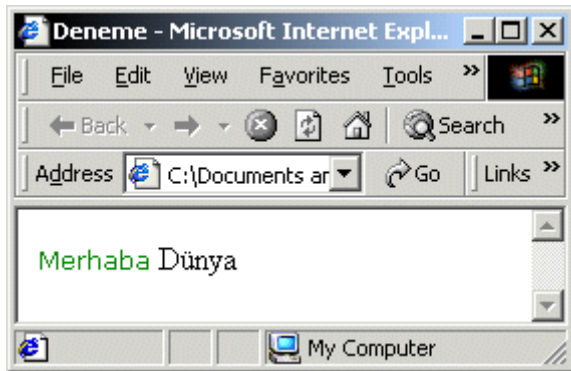
Resimde görüldüğü gibi </div> tagından sonra yazılanlar bir alt satırdan devam eder. Bu durum istenmiyorsa “span” kullanılmalıdır. Örneğin:

```
<html>
<head>
<title>Deneme</title>
```

```

<style type="text/css">
    span.yazi { font-family: verdana, Garamond; font-size:10pt; color:green;}
</style>
</head>
<body>
    <span class="yazi">Merhaba</span> Dünya
</body>
</html>

```



### 2.13.2. Linkler

Linklerin çeşitli özellikleri de CSS ile değiştirilebilir. Örneğin:

```

<html>
<head>
<title>Deneme</title>
<style type="text/css">
    a:link { text-decoration:none; color: rgb(0, 0, 0) }
    a:visited { text-decoration:none; color: rgb(153, 0, 153) }
    a:active { text-decoration:none; color: rgb(255, 0, 102) }
    a:hover { text-decoration:underline; color: rgb(0, 96, 255) }
</style>
</head>
<body>
    <a href="http://bimmag.sdu.edu.tr">BİM Magazin</a>
</body>
</html>

```

Burada “link” ziyaret edilmemiş, “visited” ziyaret edilen, “active” tıklanınca, “hover” mouse imleci üzerine getirilince linkin özelliklerini tanımlar. “text-decoration” ile alt çizginin durumu tanımlanır, “rgb” ile de renk tanımlanır (kırmızı, yeşil ve mavi renklerin karışımı).

## 2.14. Uygulama 1

```
<html>
<head>
  <title>Benim Sevgili Köpeğim</title>
</head>

<body bgcolor="pink">
  <table>
    <tr>
      <td align="center">
        <p>
          <img src='kopek.jpg'>
          <br>
          <h2>Köpeğim Karabaş</h2>
        </p>
      </td>

      <td>
        <p>
          <h3>Karabaş'ın Özellikleri</h3>
          <ul>
            <li>Ağırlık: </li>
            <li>Renk: </li>
            <li>Boy: </li>
          </ul>
        <p>
      </td>
    </tr>
  </table>

  <p>
    Bu benim köpeğim Karabaş. O benim köpeğim ve en yakın arkadaşım. Karabaş ve ben
    kısa bir süredir arkadaşız. Karabaş son günlerde kendini oldukça yalnız hissediyor. Onun
    kendini iyi hissetmesi için ona <a href=mailto:karabaş@hotmail.com> buradan</a> e-
    mail göndermek ister misiniz?
  </p>

</body>
</html>
```

## 2.15. Uygulama 2

[illegible]

```

        <li>elma</li>
        <ol type="A">
            <li>elma</li>
            <li>armut</li>
            <li>hidayet</li>
        </ol>
        <li>armut</li>
        <li>hidayet</li>
    </ol>

    <li>armut</li>

    <li>hidayet</li>
</ul>

<ol type="circle">
    <li>elma</li>
    <li>armut</li>
    <li>hidayet</li>
</ol>

<ol type="disc">
    <li>elma</li>
    <li>armut</li>
    <li>hidayet</li>
</ol>

<ol type="square">
    <li>elma</li>
    <li>armut</li>
    <li>hidayet</li>
</ol>

<ol type="1">
    <li>elma</li>
    <li>armut</li>
    <li>hidayet</li>
</ol>

<ol type="a">
    <li>elma</li>
    <li>armut</li>
    <li>hidayet</li>
</ol>

<br>
<br>
<br>
</body>
</html>

```

## 2.16. Uygulama 3

```
<html>

<body>

<ul>
<li>1.madde</li>
<li>2.madde</li>
<li>3.madde</li>
</ul>
<hr>

<ol type="1" >
  <li>1.madde
    <ol type="square" >
      <li>1.madde</li>
      <li>2.madde</li>
      <li>3.madde</li>
      <li>4.madde</li>
    </ol>
  </li>

  <li>2.madde</li>
  <ol type="disc" >
    <li>1.madde</li>
    <li>2.madde</li>
    <li>3.madde</li>
    <li>4.madde</li>
  </ol>

  <li>3.madde</li>
  <ol type="circle" >
```

```
    <li>1.madde</li>
    <li>2.madde</li>
    <li>3.madde</li>
    <li>4.madde</li>
  </ol>

  <li>4.madde</li>
  <ol type="I" >
    <li>1.madde</li>
    <li>2.madde</li>
    <li>3.madde</li>
    <li>4.madde</li>
  </ol>

  <li>5.madde</li>
  <ol type="i" >
    <li>1.madde</li>
    <li>2.madde</li>
    <li>3.madde</li>
    <li>4.madde</li>
  </ol>

  <li>6.madde</li>
  <ol type="a" >
    <li>1.madde</li>
    <li>2.madde</li>
    <li>3.madde</li>
    <li>4.madde</li>
  </ol>
</ol>

</body>
</html>
```

## 2.17. ÇERÇEVELER

Çerçeveler (frame), HTML'e sonradan eklenmiş bir özelliktir. Bu yüzden eski sürüm browser'lar çerçeve kullanılmış bir sayfayı görüntüleyemezler. (Microsoft Internet Explorer 3.0 ile Netscape Navigator 2.0 ve üstü sürümleri frameleri destekliyor). Bunun yanında çerçeve kullanımına karşı düşmanca (!) tavır sergileyen tasarımcılar da hatırı sayılır miktarda ve hala birçok site, ziyaretçisine çerçeveli ve çerçevesiz sunum seçeneği sunuyor.

Çerçeveler sayfanın bir tarafı sabit kalırken, diğer tarafını, kullanıcının isteğine göre değiştirme imkanı verir (bu sayfalarda olduğu gibi). Bunun anlamı ise kullanılacak çerçeve miktarınca ek HTML sayfası oluşturmak demektir. Browser çerçevenin bir bölümüne yerleştirilecek olan kaynak dosyayı bulamazsa hata mesajı verecektir.

Aşağıdaki örneği inceleyelim:

frame.htm dosyası:

```
<html>
<head>
<title>Çerçeveler</title>
</head>

<frameset cols="*,*">
<frame name="sol" src="htm1.htm">
<frame name="sag" src="htm2.htm">
</frameset>

<noframes>
<body>
</body>
</noframes>

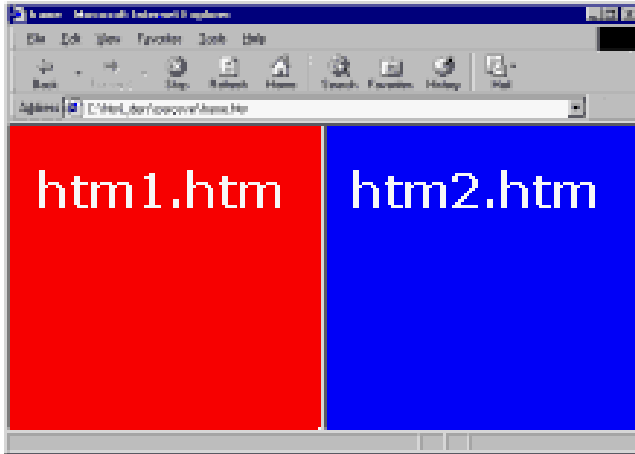
</html>
```

**htm1.htm dosyası:**

```
<html>
<head> </head>
<body bgcolor="red">
<font size="7" color="#ffffff">htm1.htm</font>
</body>
</html>
```

### htm2.htm dosyası:

```
<html>
<head> </head>
<body bgcolor="blue">
<font size="7" color="#ffffff">htm2.htm</font>
</body>
</html>
```



<frameset>...</frameset>

Çerçeve oluşturmada kullandığımız etiket frameset, cols ifadesi açılacak çerçevelerin yanyana sütunlar şeklinde olacağını belirtiyor, buraya row yazarsak çerçeveler alt alta satırlar görünümünde açılacaktır.

cols="\*", "\*" ifadesi ile açılacak çerçevelerin pixel cinsinden ebatlarını belirliyoruz, buraya örneğin cols="140,500" gibi sayı yazılabilir, cols="25%,75%" ile browser penceresinin o anki ebadına göre verilen % oranlarına göre şekil alması sağlanabilir yada \* sembolü ile açılacak çerçevenin ebadı browser'a bırakılır, cols="140,\*" ifadesi ilk pencerenin 140 pixel olacağını ikincinin ise ebadının browser'a bırakıldığını gösteriyor.

Bu şekilde aynı zamanda açılacak çerçeve sayısını da belirtmiş oluyoruz. cols=".." ifadesine iki değer (ya da yıldız) verirse bu iki pencere aç anlamındadır, 3 değer 3 pencere açar.

frame name="..." src="..." etiketi çerçevelere, bağlantıların TARGET kısmındaki kullanabilmek amacıyla isim (name) vermeyi sağlıyor (bu konuya aşağıda değineceğiz). Src pencerede görüntülenecek html dosyasının yerini gösteriyor. resimler bahsinde belirttiğimiz yer (kaynak) gösterme kuralları burada da geçerli. Açılacak çerçeve sayısı kadar frame name="..." src="..." etiketi kullanıyoruz.

Eski sürüm browserların çerçeve etiketini tanımadıklarından bahsetmiştik, işte noframes kısmı bu browserlara hitap ediyor. body kısmını bu eski browserlar görüntüleyebiliyor. Buraya örneğin sayfanın çerçeve kullanılarak yapıldığını gösteren bir mesaj yazılabilir, böylece ziyaretçinin artık yeni sürüm bir browser kullanması gerektiğini hatırlatabilirsiniz.



### 3. CSS (STİL ŞABLON)

Css'in açılımı ile söze başlayalım. Cascading Style Sheets. Biz kısaca konularımız dahilinde Stil Şablon olarak bahsedeceğiz. Gelelim Stil Şablonun varoluş amacına. Sizde bildiğiniz üzere Html yazım şekli olarak etiket türünde bir yazım dili. Bu yüzden pek fazla özelliklere sahip değil. Bu sahip olamadığı özellikler nedeniyle sayfanın dizaynında bize tam esneklik veremiyor. Css bu amaçla üretilmiş bir dil. Kullanım kolaylığı ve kullanışlılığı ile Html'e eklenmesinden itibaren çoğu web tasarımcısının gözdesi oldu. Çünkü her türlü sayfa dizaynını bize bırakarak müthiş bir esneklik sağlıyor. Ayrıca ilerideki konularımızda bahsedeceğimiz üzere bağlantılı stil şablonlar aracılığı ile de birden çok sayfaya etkiyebiliyor. Bu da bize sitenin görünümün değiştirmek istediğimizde elimizdeki onlarca belki de yüzlerce sayfanın kodlarını değiştirmeden sadece css dosyasının değiştirerek bu imkanı sağlıyor.

Stil Şablon'un tarayıcılara eklenmesinden sonra iki versiyonu çıktı. Bunlar Css 1 ve Css 2. Ayrıca bazı konularda MSIE (Internet Explorer) ve NN (Netscape Navigator) tarayıcıları aynı kodları kabul etmiyorlar. Biz derslerimizde her iki tarayıcıda da etkin olan veya etkin olmayıp dizaynı bozmayan (A:hover gibi) Stil Şablon özelliklerini göreceğiz. Şimdi derslerin içeriğinde neler var onları görelim :

#### 1. STİL ŞABLON ÇEŞİTLERİ :

Css'in en çok beğenilen yönü istendiğinde sadece bir öğeye etkimesi, istendiğinde tüm sayfaya etkimesi, istendiğinde site içindeki tüm html dosyalarına etkimesidir. Bunlar kısaca Stil Şablonun kullanım çeşitleridir.

#### 2. HTML ETİKETLERİ İLE CSS :

Bu dersimizde Html'deki font,background gibi çeşitli özelliklerin Stil Şablon tarafından nasıl belirlenebileceğini göreceğiz.

#### 3. SEÇİCİLER (SELECTORS) :

Kimi zaman Html etiketlerinden fontu hepimiz kullanırız. Örneğin bir sayfa içerisinde font etiketine birden çok görünüm eklemek isteriz. Bu durumda seçiciler imdadımıza yetişir.

#### 4. GENEL KULLANIM ŞEKİLLERİ :

Bu dersimizde ise A (link) etiketinin çeşitli kullanım biçimleri ile birlikte bir Stil Şablonun nasıl kullanırsak işimize daha fazla yarayacağını göreceğiz. Siz buradaki kullanım tarzına göre Css'i kullanırken kendinize nasıl bir yön izleyeceğinize karar vereceksiniz.

### 3.1. STİL ŞABLON ÇEŞİTLERİ

Css'in (Stil Şablon) 3 farklı kullanım alanı vardır. Bunlar ;

- Yerel, yani sayfada sadece bir kez, Yerel stil şablonlar bir html etiketi için özel olarak kullanılırlar.
- Global, yani tüm sayfa için, Global stil şablonlar sayfadaki tüm html etiketlerinin belirlenen özellikte olması istendiğinde kullanılırlar.
- Bağlantılı, yani birden çok sayfa için, Bağlantılı stil şablonlar birçok sayfada aynı biçimde olması istendiğinde kullanılırlar.

#### 3.1.1. Yerel Stil Şablonu

Başlangıçta belirttiğimiz gibi Yerel Stil Şablonlar, uygulanacak etiketi sadece bir kez bulunduğu yerde(yerel) etkiler. Şimdi bir örnek verelim.

```
<html>
<head>
<title>Css</title>
</head>
<body>
<h2>Web Teknikleri</h2><br>
<h2 style=font-size:20pt; color:blue>Web Teknikleri</h2>
</body>
</html>
```

Bu örneğimizi css.htm adıyla kaydedip tarayıcı yardımıyla açtığımızda iki tane Web Teknikleri yazısıyla karşılaşacağız. Fakat bunların yazım tarzı farklı olacak. Çünkü biz ikinci h2 adlı etiketimize etkimek üzere bir stil şablon ekledik.

#### 3.1.2. Global Stil Şablonu

Global Stil Şablonları bir önceki örnekte yaptığımız h2 etiketinin tüm sayfada aynı özellikte olması istendiğinde kullanılır. Bunu için Stil Şablon özellikleri sayfanın başlangıcında (<head></head> etiketleri arasında) tanımlanmalıdır.

Örnek ile biraz daha ayrıntılı inceleyelim.

```
<html>
<head>
<title>Css</title>
<style type="text/css">
<!-- h2 {font-size:20pt; color:blue} -->
</style>
</head>
<body>
<h2>Web Teknikleri</h2>
</body>
</html>
```

Burada ne yapmış olduk? Sayfa içerisinde kullanacağımız tüm h2 etiketlerinin özelliklerini sabitlemiş olduk. Yani sayfa içerisinde nerede kullanırsanız kullanın h2 etiketinin stil özellikleri hep aynı olacaktır. Yazım kurallarına biraz değinirsek, Stil Şablon tanımlamaları **<head> </head>** etiketleri arasında **<style type="text/css">** ile başlayıp **</style>** ile bitmelidir. **<!--** Etiketini Css'den anlamayan tarayıcıların bu kısmı geçmesini sağlıyoruz. Bu saklama işlemi **-->** etiketi ile son bulur.

### 3.1.3. Bağlantılı Stil Şablon

Global stil şablonu ise sitemiz içerisindeki tüm sayfalarda aynı stil özelliklerini kullanmak istediğimizde kullanırız. Şöyleki; Her zaman olduğu gibi stillerimizi yukarıda örneklerini verdiğimiz şekilde hazırlarız. Fakat bunu html dosyamızın içerisinde değil de boş bir sayfaya yazarız. Sonra onu kaydederken css uzantılı bir şekilde kaydederiz. Ardından da html dosyamızın içerisine yine **<head> </head>** etiketleri arasına **<link rel="stylesheet" type="text/css" href="dosya\_ismi.css">** şeklinde ekleriz. Şimdi hemen bir örnek verelim.

```
h1 {font-size:13pt; color:green}
h2 {font:20pt; color:blue}
h3 {font-size:15pt; color:red}
```

Bu dosyamızı **stil.css** olarak kaydedelim. Şimdi de html dosyamıza gelelim. Html dosyamızın kodları da şu şekilde olmalıdır.

```
<html>
<head>
<title>Css</title>
<link rel="stylesheet" type="text/css" href="stil.css"> </head>
<body>
<h1>Web Teknikleri</h2>
<h2>Web Teknikleri</h2>
<h3>Web Teknikleri</h2>
</body>
</html>
```

Html dosyasının kodları arasında geçen **<link rel="stylesheet" type="text/css" href="stil.css">** kodu stil.css dosyasındaki stil özelliklerini kullanmamızı sağlar. Bu kodu istediğimiz diğer html dosyalarına da eklediğimizde orada da kullanabiliriz. Böylelikle her sayfada stil özellikleri tanımlamamış, başlangıçta tanımladığımız stil özelliklerini kullanarak koddan tasarruf etmiş oluruz.

### 3.2. HTML ETİKETLERİ İLE CSS

Biz bundan sonraki tüm Stil Şablon örneklerimizde Global Stil Şablonu kullanacağız. Bu yüzden Css ile Html dosyalarımız beraber olacak böylece de konuyu kavrama ve anında uygulamanız daha kolay olacak. Şimdi Css'in etkidiği Html etiketlerini hangi özelliklerini değiştirdiğini görelim.

#### 3.2.1. Font Özellikleri

Adı üzerinde Font özelliklerini değiştirmeye yarayan bir stil şablon özelliğidir. Nasıl kullanıldığına hemen bir bakalım.

```
<html>
<body>
<head>
<title>Css</title>
<style type="text/css">
```

```
<!--  
p {font-size : 12pt;font-family : Arial;font-weight : bold;font-style : italic;color :  
#00FFFF;}  
-->  
</style>  
<body>  
<p>Web Teknikleri</p>  
</body>  
</html>
```

Alt özellikleri tanıyalım.

**font-size** : Font büyüklüğünü belirtir.

İsterseniz aşağıdaki gibi standart değerleri seçersiniz;

xx-large (en büyük )

x-large (biraz büyük)

large (büyük)

medium (orta)

small (küçük)

x-small (biraz küçük)

xx-small (en küçük)

isterseniz direkt olarak punto(pt) değerini verebilirsiniz.

**font-family** : Font tipini belirler.

Arial, Courier, Verdana gibi font isimlerini alabilir.

**font-weight** : Fontun kalınlı incelik durumunu belirler.

bold : Fontu kalın yapar.

normal : Fontun normal halde olmasını sağlar.

Bu özellik yazılmadığında normal özellik alınır.

**font-style** : Fontun yatık olup olmamasını sağlar.

italic : Yazının sağa doğru yatık olmasını sağlar.

normal : Fontu normal halde olmasını sağlar.

Bu özellik yazılmadığında normal özellik alınır.

**color** : Fontun rengini belirler.

Blue, red, green gibi renklerin ingilizce karşılıklarını alabilir.

### 3.2.2. Text Özellikleri

Text özelliği ile de font özelliğinin sahip olmadığı bazı özellikleri etiketimize ekleriz.

Örnek ile açıklayalım.

```
<html>
<body>
<head>
<title>Css</title>
<style type="text/css">
<!--
p {
text-transform : lowercase;
text-decoration : underline;
text-align : left;
line-height : 20px;
text-indent : 15px;
}
-->
</style>
<body>
<p>Web Teknikleri</p>
</body>
</html>
```

Alt özellikleri tanıyalım.

#### **text-transform :**

lowercase : Yazının tümünün küçük harf olmasını sağlar.

uppercase : Yazının tümünün büyük harf olmasını sağlar.

capitalize : Yazının istenilen şekilde kalmasını sağlar.

#### **text-decoration :**

underline : Yazının altının çizili olmasını sağlar.

overline : Yazının üstünün çizili olmasının sağlar.

line-through : Yazının üstünün çizili olmasını sağlar.

none : Yazının herhangi bir yerine çizgi çekilmemesini sağlar.

### **text-align:**

left: Yazının sola bitişik olmasını sağlar.

center : Yazının ortada olmasının sağlar.

right : Yazının sağa bitişik olmasını sağlar.

line-height :Yazının normal satırdan çizgi yüksekliğini belirler. 3px, 5px gibi değerler alır.

text-indent : Yazının soldan ne kadar boşlukla içeriden başlayacağını belirler. 5px, 10px gibi değerler alır.

### **3.2.3. Background Özellikleri**

Background ile html sayfamızın arkafonlarının özelliklerini değiştirmemizi sağlar.

```
<html>
<body>
<head>
<title>Css</title>
<style type="text/css">
<!--
p {
background-color :#00ff00;
background-image : url ("resim_adi.gif");
background-position : center;
background-repeat : repeat-y;
}
-->
</style>
<body>
<p>Web Teknikleri</p>
</body>
</html>
```

**background-color :**

Arka fonun rengini belirler. Css'te renkleri blue, red gibi tanımlayabileceğimiz gibi Html kodunu vererekte tanımlayabiliriz.

**background-image :**

Arka fonu bir resim dosyası yapmak için kullanılır. url etiketinin içine resim dosyasının yolu ve ismi tam olarak yazılmalıdır.

**background-position :**

left : Arka fondaki resmin sadece sol tarafta olmasını sağlar.

center : Arka fondaki resmin sadece sol tarafta olmasını sağlar.

right : Arka fondaki resmin sadece sol tarafta olmasını sağlar.

**background-repeat :**

Arkafondaki resmin tekrarlanması istendiğinde kullanılır.

repeat : Tüm yönlerde tekrar edilmesini sağlar.

repeat-x : X (yatay) yönünde tekrar edilmesini sağlar.

repeat-y : Y (dikey) yönünde tekrar edilmesini sağlar.

no-repeat : Resmin tekrar edilmeyerek bir kere gösterilmesini sağlar.

### 3.2.4. List Özellikleri

Bu Css özelliği <UL> ve <LI> html etiketleri ile oluşturduğumuz listelerin özelliklerini belirlemek için kullanılır.

```
<html>
<body>
<head>
<title>Css</title>
<style type="text/css">
<!--
li {
list-style-type : circle;
list-style-position : inside;
list-style : decimal;
list-style-image : url ("resim.gif");
}
-->
```



```
</style>
<body>
<ul>
<li>Web Teknikleri
<li>Html
<li>Javascript
<li>Css
<li>Web Grafik
</ul>
</body>
</html>
```

#### **list-style-type :**

disk : Liste biçiminin disk (içi dolu yuvarlak) şeklinde olmasını sağlar.

circle : Liste biçiminin çember şeklinde olmasını sağlar.

square : Liste biçiminin kare olmasını sağlar.

decimal : Liste biçiminin rakamlardan oluşmasını sağlar.

#### **lower-roman :**

Liste biçiminin i,ii,iii, gibi roma rakamlarının küçük harfi olmasını sağlar.

upper-roman: Liste biçiminin I,II,III gibi roma rakamlarının büyük harfi olmasını sağlar.

lower-alpha : Liste biçiminin a,b,c şeklinde olmasını sağlar.

upper-alpha: Liste biçiminin A,B,C şeklinde olmasını sağlar.

none : Listenin imgesiz olmasını sağlar.

#### **list-style-position :**

inside : Listenin ikinci satırının en soldan başlamasını sağlar.

Outside : Listenin ikinci satırının ilk satır ile aynı yerden başlamasını sağlar.

#### **list-style-image :**

Liste biçiminin resim olmasını sağlar.

### **3.2.5. Position Özelliği**

Html'de kullandığımız Layer (katman) etiketlerinin html üzerindeki yerleştirme işlemi için kullanılır. Hemen bir örnek ile görelim.

```
<html>
<head>
<title>Css</title>
<STYLE type="text/css">
<!--
div {
position:absolute;
top:20px;
left:10px;
width:200px;
height:200px;
clip:auto;
overflow:scroll;
z-index:auto;
visibility:visible;
}
-->
</style>
<body>
<div>
Web Teknikleri<br>
Html<br>
Javascript<br>
Css<br>
Grafik<br>
</div>
<p> Web Teknikleri </body>
</html>
```

### **position :**

absolute : Katmanın yerinin kesin olarak belirlenmek istendiğinde kullanılır.

relative : Katmanın yerinin görelî(diğer öğelere göre değışebilen) olarak belirlenmek istendiğinde kullanılır.

static : 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.

**clip** : Katmanın görünmesi istenen bölgeyi içeren kutucuk.

**overflow** : Katmanın belirtilen yükseklik ve genişliğe sığmayan kısmına ne olacağını belirler.

auto : Otomatik olarak belirlenir.

**scroll** : Kaydırma çubukları ekler.

**visibility** : Katmanın görünebilirlik ayarı yapar

visible : Görünür hale getirir.

hidden : Gizler.

**z-index** : Katmanın sayfa üzerindeki sıra sayısı.

### 3.3. SEÇİCİLER (SELECTORS)

Css'te seçiciler en çok kullanılan öğelerdendir. Örneğin H1 etiketine Css yardımıyla belli bir şablon yüklediniz. Ama sayfanızda kullanacağınız H1 etiketlerinin tümünün aynı şekilde olmasını istemiyorsunuz. Bu durumda bize seçiciler yardımcı olur.

Şimdilik üç çeşit seçici göreceğiz. Bunlar :

1. Class Selector (Sınıf Seçicisi)
2. Id Selector (Id seçicisi)

### 3.3.1. Class Selector (Sınıf Seçicisi)

Bu seçiciyi sayfanızdaki h1 gibi etiketlerin tümünün aynı olmasını istemediğiniz durumlarda kullanırsınız. Böylelikle genel bazı özellikleri koruyarak farklı özellikleri özelleştirebilirsiniz. Sınıf seçicisinin iki türü vardır. İlk önce birinci şeklini görelim.

Hemen bir örnekle bu seçiciyi tanıyalım.

```
<html>
<head>
<title>Css</title>
<style type="text/css">
<!--
h1.mavi {color:blue}
h1.kirmizi {color:red}
-->
</style>
</head>
<body>
<h1 class=mavi>Mavi sınıf seçicisi ile </h1><br>
<h1 class=kirmizi>Kırmızı sınıf seçicisi ile </h1>
</body>
</html>
```

Burada sınıf seçicisini sadece h1 için tanımladık. Sınıf seçicisinin ikinci türünde genel bir sınıf seçicisi tanımlamaktır. Bunu da bir örnekle görelim.

```
<html>
<head>
<title>Css</title>
<style type="text/css">
<!--
.mavi {color:blue}
.kirmizi {color:red}
-->
</style>
</head>
```

```
<body>
<h3 class=mavi>Mavi sınıf seçicisi ile </h1><br>
<h4 class=kirmizi>Kırmızı sınıf seçicisi ile </h1>
</body>
</html>
```

### 3.3.2. id selector (id seçicisi)

Id Selector'lerini tanımlayıcı adlarının önündeki # işaretinden tanırız. Html belgesinde kendi tanımlayıcı adlarına gönderme yaparak herhangi bir Html etiketine stil vermekte kullanılırlar. Bu etiketler span'dan tutunda paragraf(p)'a kadar olabilir. Bir örnekle açıklayalım.

```
<html>
<head>
<title>Css</title>
<style type="text/css">
<!-- Eski tür tarayıcılardan kodumuzu saklayalım -->
#mavi
{
background:blue;
color:white;
}
#yesil
{
background:green;
color:white;
}
-->
</style>
</head>
<body>
<span id=mavi>Bu yazının arka fon rengi mavi font rengi beyaz</span><br><br>
<span id=yesil>Bu yazının arka fon rengi yeşil font rengi beyaz</span>
</body>
</html>
```

### 3.4.CSS GENEL KULLANIM ŞEKİLLERİ

Css'i Html üzerinde kullanmak için 3 yöntem (yerel-global-bağlantılı) olduğunu daha önce belirtmiştik. Şimdi ise komple bir css dosyasını Html üzerinde nasıl kullanacağımız görelim. Fakat öncelikle Html'deki a etiketinin diğer etiketlerden farklı olarak bir kullanım tarzı bulunmakta. İlk önce ona değinelim.

A etiketinin Css ile kullanımı

Bildiğiniz üzere A etiketi Html'e çok büyük bir özellik katan link etiketidir. Bu etiket ile diğer bir web sayfasına veya bir mail adresine gönderme yapabiliriz. Bu etiketin belli durumlarda aldığı değişik değerler vardır. Yani link tıklandığında etiket artık visited (ziyaret edilmiş) pozisyonuna geçecektir. Biz Css yardımıyla A etiketinin aldığı pozisyonlara istediğimiz biçimi verebiliriz. Şimdi A etiketinin aldığı pozisyonları görelim :

İlk pozisyon linke herhangi bir tıklama olmadığındadır. Bu değer linkin sayfada görülecek ilk halidir.

Visited : Bu pozisyon link tıklandığından sonra etiketin aldığı değerdir.

Active : Bu pozisyon linkin aktif olduğu durumdur. Yani imleç linkin tıklandığı andaki durumdur.

Hover : Bu pozisyon Linkin üzerine gelindiği durumdur. Yani linkin üzerine gelindiğinde nasıl bir biçimde olması isteniyorsa stil o şekilde verilir. Şimdi A etiketi için bir stil dosyası yapalım.

```
<html>
<head>
<title>Css</title>
<style type="text/css">
<!--
A.normal
{
background-color:white;
color:blue;
}
A.ziyaret:visited
{
background-color:white;
color:maroon;
```

```

font-weight:normal;
}
A.aktif:active
{
background-color:white;
color:red;
font-weight:normal;
}
A.degisken:hover
{
background-color:blue;
color:white;
font-weight:bold;
}
-->
</style>
</head>
<body>
<a href="#" class="normal">Linkin normal durumu</a><br>
<a href="#" class="ziyaret">Linki tıklayın ve değiştiğini görün</a><br>
<a href="#" class="aktif">Linkin aktif durumu</a><br>
<a href="#" class="degisken">Linkin üzerine geldiğinde stil değişecek</a><br>
</body>
</html>

```

Şimdi A etiketinin özel durumunu da gördükten sonra esaslı bir css kullanma tekniğini görelim. Bu örneğimizde div , table , span , h1-2-.. , p , a gibi Html etiketlerini kullanırken nasıl bir yöntem izlememiz gerektiğini göreceğiz.

İlk öncelikle stillerimiz hem bağlantılı hem global hem de yerel kullanacağız. Bunu belirteyim. Böylelikle sizde nasıl bir yol izlemenize kara verin

Şimdi bağlantılı css dosyamızı hazırlayalım. Hatırlayacağınız üzere bu dosyanın uzantısı css olmalı. Bu css dosyasını Html dosyamızın içerisinde çağıracağız.

Aşağıdaki kodları stil.css adıyla kaydedelim.

```
A {font-style : normal;
color : navy;
font-family : Times New Roman ! important;
text-decoration : none; <!-- bu satır linkin altında satır olmamasını sağlar -->}
A:Visited {font-family : Times New Roman ! important;
font-style : italic;
color : olive; }
A:Active { font-family : Times New Roman;
color : red;}
A:Hover {text-decoration : underline;
font-family : Times New Roman ! important;
font-weight : bold;
font-style : normal;
color : maroon;}
BODY { background: white url("fon.gif");
background-repeat: repeat-y;
background-position: left; }
p#sol {position : relative;
visibility : visible;
left : 30pt;
width : 450pt;
font-family:"Verdana,Arial,Helvetica" ! important;
font:15pt;}
```

Aşağıdaki kodları da css.html adıyla kaydedelim. (Dikkat ! html uzantılı kaydedin )

```
<html>
<head>
<title>Css</title>
<style type="text/css">
<!--
.onemli {font-weight:bold;}
h4 {color:blue;
position : relative;
```



```

visibility : visible;
left : 25pt;
font-size:large;
.solic { color:brown;
font-family:"Verdana,Arial,Helvetica";
position : relative;
visibility : visible;
left : 20pt;
font-weight:bold; }
li { list-style-type : circle;
list-style-position : inside;
list-style : decimal;}
; -->
</style>
<link rel=stylesheet href="stil.css" type="text/css">
</head>
<body>
<table width="500" align="center">
<tr><td> <!-- Global -->
<h4>Bilgisayar;<a name="bsl">&nbsp;&nbsp;&nbsp;</a></h4>
<!-- Eğer koordinatları tam olarak ayarlamak istiyorsanız (MSIE ve NN için) Global
Stil Şablonu Kullanmalısınız. -->
    <!-- Bağlantılı -->
    <p id="sol">
Aldığı komutlar uyarınca, veri işleyerek problem çözen otomatik elektronik aygıtların
ortak adı. Bu tür aygıtlar, çalışma ilkeleri,donanım tasarımları ve uygulama alanları
bakımından örneksel, sayısal ve karma bilgisayarlar olarak <font
class="onemli">üç</font> ayrılır.</p>
<p id="sol">
<ul>
<li><a href="css.html#orneksel">Örneksel (analog) bilgisayarlar</a>
<li><a href="css.html#sayısal">Sayısal bilgisayarlar</a>
<li><a href="css.html#karma">Karma bilgisayarlar</a>
</ul>
</p> <p class="solic">

```

Örneksel (analog) bilgisayarlar<a name="örneksel">&nbsp;&nbsp; </a></p>  
<p id="sol">Açısal konum ya da gerilim gibi değişken nicelikleri temsil eden veriler  
üzerinde işlem yapar ve çözülmesi istenen matematiksel problemin fiziksel bir  
örneğini oluştururlar. Sıradan diferensiyel denklemleri çözebilen örneksel  
bilgisayarlar, sistem mühendisliğinde, özellikle bazı süre ve donatımların gerçek  
zamanlı benzetim modellerinin oluşturulmasına çok elverişlidirler. Bu bilgisayarların  
bir başka yaygın kullanım alanı da elektrik dağıtım sistemi gibi şebekelerin  
analizidir.<br>  
<a href="css.html#bsl">Başa Dön</a>  
</p>  
<p class="solic">Sayısal bilgisayarlar,<a name="sayısal">&nbsp;&nbsp; </a></p>  
<p id="sol">Çeşitli üretim süreçlerine, takım tezgahlarına , karmaşık laboratuvar ve  
hastane aygıtlarına kumanda etmekte kullanılırlar. Aynı özellikten, uçakların ve uzay  
araçlarının karmaşık iletişim sistemlerinin otomatizasyonunda da yararlanılır. Sayısal  
bilgisayarlar ayrıca, eğitimde yardımcı olarak (örn. temel dil ve matematik  
becerilerinin kazandırılmasında) , bilimsel araştırmalarda ise verilerin analizi ve  
matematiksel modellerin geliştirilmesi amacıyla kullanılır.  
<br> <a href="css.html#bsl">Başa Dön</a> </p>  
<p class="solic">Karma bilgisayarlar,<a name="karma">&nbsp;&nbsp; </a></p>  
<p id="sol">Örneksel ve sayısal bilgisayarların özelliklerine ve yararlarını birleştirirler;  
örneksel bilgisayarlara oranla daha fazla kesinlik, sayısal bilgisayarlara oranla daha  
fazla denetleme sağlarlar.  
<br><a href="css.html#bsl">Başa Dön</a>  
</p> </td> </tr> </table>  
</body>  
</html>

Burada birkaç konuya açıklık getirelim.

Bazı stil özelliklerinin sonunda gördüğümüz limportant ifadesi ile ziyaretçi kendi bilgisayarındaki tarayıcı özelliklerini değiştirmiş olsa dahi bu değerleri kullanmamasının bizim belirttiğimiz değerleri kullanmasını söylemiş oluyoruz. Font özelliklerinde çoğu zaman birden çok font ismi kullandık. Bunun nedeni eğer ziyaretçinin makinasında ilk font yoksa ikincisi o da yoksa üçüncü font kullanılır.Şayet o fontta yoksa tarayıcının kendi banko fontu kullanılır. Böylelikle bizde değişik ziyaretçi makinalarında sayfamızın nasıl görünebileceğini öncelikle kontrol altına almış oluruz.