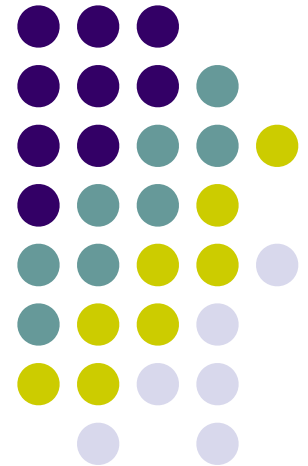


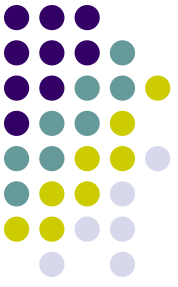
HTML & CSS

Öğr.Gör. M.Ersin AKAY



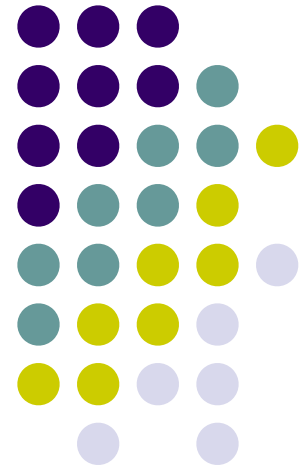
Neler Öğreneceğiz?

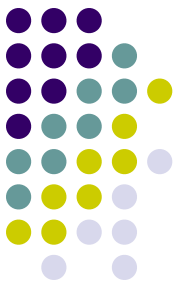
- HTML işaretleme dili
- CSS stil şablonları



HTML

Hyper Text Markup Language

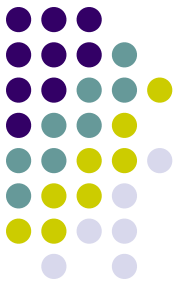




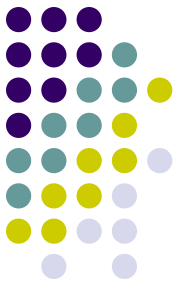
HTML'de Neler Öğreneceğiz?

- SEO nedir? SEO ve HTML
- Html işaretleme dili temel yapısı ve kavramları
- Html kod yapısı(head,body)
- Head içerisinde yer alan başlıca etiketler(title,meta,script,link)
- Başlık Etiketleri
- Metin düzenleme etiketleri(p,font,b,i,u,sub,sup,nbsp,br)
- Bağlantı(link) oluşturma
- Resim ekleme(img)

HTML'de Neler Öğreneceğiz?



- Listeler(`ul`,`ol`,`li`)
- Tablolar(`table`,`tr`,`td`)
- Formlar(`form`,`input`..)
- Yerleşim düzeni, anahat oluşturma(`table`,`div`)



SEO Nedir? SEO & HTML

- SEO (Search Engine Optimization)
- Arama motorları
 - Google
 - Yandex
 - Bing
 - ...
- İç SEO & Dış SEO
- Bir sitenin iç SEO yapısı HTML ile sağlanır

HTML

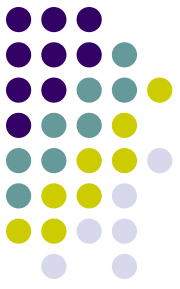
Temel Yapı ve Kavramlar



- Web Sayfalarının tüm görselliğini sağlayan HTML dilidir
- Yapısal bir dil değildir
- Kodları notepad ve/veya **Visual Studio Code** ile yazabiliriz
- Bilgisayarımızdaki web tarayıcılar(chrome, edge, firefox) tarafından derlenir
- Dosya uzantısı .html'dir
- Çift-tıkla çalıştır, çünkü web server'a ihtiyacı yoktur

HTML

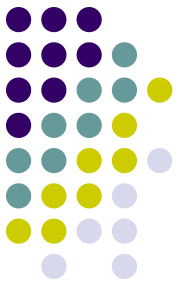
Kod Yapısı – Etiketler (Tag)



- Kodlar etiketlerden oluşur
- `<etiket adı>` ile başlar ve çoğunlukla `</etiket adı>` ile kapanır
- `<etiket adı özellik adı = “değer”>` şeklinde özellikler içerir
- Etiket ve özelliklerde kesinlikle Türkçe karakter kullanılmaz !

HTML

Kod Yapısı – `<html><head><body>`



```
<html>  
  <head>  
  .....  
  .....  
  </head>  
  <body>  
  .....  
  .....  
  </body>  
</html>
```

HTML

<head>



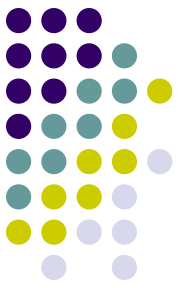
```
<html>
  <head>
    <title>EmlakBuluyoruz.Com - Emlak Bulmanın Yerel Yolu</title>
    <meta name="author" content="Finders Yazılım Teknolojileri">
    <meta name="copyright" content="Copyright (c) emlakbuluyoruz.com">
    <meta name="description" content="Türkiye'nin satılık veya kiralık daire,
      işyeri, arsa, arazi ilanlarının yer aldığı, ilk yerel ve tek
      interaktif emlak ilan sitesi">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <meta content="text/html; charset=iso-8859-9" http-equiv="Content-Type" />

    <script src="//www.google-analytics.com/analytics.js"></script>
    <link rel="stylesheet" type="text/css" href="/css/bootstrap.min.css">
  </head>
  .
  .
  .
</html>
```

HTML

Başlık Etiketleri



```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso8859-9" />
<title>Untitled Document</title>
</head>

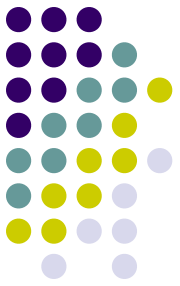
<body>
<h1>Başlık 1</h1>
<h2>Başlık 2</h2>
<h3>Başlık 3</h3>
<h4>Başlık 4</h4>
<h5>Başlık 5</h5>
<h6>Başlık 6</h6>
</body>
</html>
```



Başlık etiketleri yalnızca yazıyı boyutlandırmak için değildir. SEO(Arama Motoru Optimizasyonu) için önemlidir. İçerikte vurgu amaçlı kullanılır

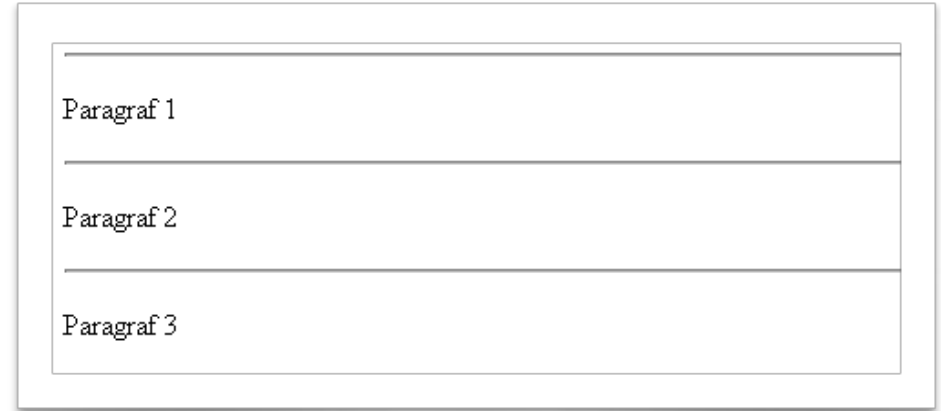
HTML

Çizgi(Ayıraç) – Açıklama Satırı



```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso8859-9" />
<title>Untitled Document</title>
</head>

<body>
<!-- Burası açıklama satırı -->
<hr><p>Paragraf 1</p>
<hr><p>Paragraf 2</p>
<hr><p>Paragraf 3</p>
</body>
</html>
```



`<hr color="" size="" width="" />` etiketi yatayda içeriği birbirinden ayırmak için kullanılır

`<!-- -->` açıklama etiketleri arasına yazılan içerik görüntülenmez

HTML

Paragraf, Yeni Satır, Boşluk



```
<html>
<body>
<p>HTML'de boşluklar ve satırlar dikkate alınmaz:</p>
<p>
  Korkma, sönmez bu şafaklarda yüzen al sancak;
  Sönmeden yurdumun üstünde tüten en son ocak.
  O benim milletimin yıldızıdır, parlayacak;
  O benimdir, o benim milletimindir ancak.
</p>
</body>
</html>
```

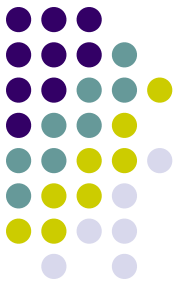
HTML'de boşluklar ve satırlar dikkate alınmaz:

Korkma, sönmez bu şafaklarda yüzen al sancak; Sönmeden yurdumun üstünde tüten en son ocak. O benim milletimin yıldızıdır, parlayacak; O benimdir, o benim milletimindir ancak.

HTML'de boşluklar ve satırlar dikkate alınmazlar. Bu boşlukları dizgide kullanabilmek için etiketlerden yararlanırız

HTML

Metin Düzenleme, Biçimlendirme



- Yeni nesil HTML kodlamada biçimlendirme CSS ile yapılmaktadır
- Etiket içinde CSS ile biçimlendirme;
 - style özelliği ile yapılır
 - `<p style=""></p>`
- `style="font-size:12pt"` metin büyüklüğü
- `style="color:red"` metin rengi
 - Renk değerleri `style="color:red"` renk ingilizce karşılığı veya `style="color:#FF0000"` şeklinde hexadecimal renk kodunu yazarak da verilebilir.
- `style="font-family:Arial"`

HTML

Metin Düzenleme, Biçimlendirme



```
<html>
<body>
<p>Normal Yazı</p>
<p><b>Kalın Yazı</b></p>
<p><strong>Kalın Yazı</strong></p>
<p><i>Eğik Yazı</i></p>
<p><em>Eğik Yazı</em></p>
<p><u>Altı çizili Yazı</u></p>
<h2>Yazı belirli bir oranda <small>küçük</small>
olarak görüntülenir</h2>
<h2>Yazıyı <mark>işaret</mark>lemek için kullanılır</
h2>
<p>Favori rengim <del>mavi</del> kırmızı.</p>
<p>H<sub>2</sub>O</p>
<p>X<sup>3</sup>Y</p>
```

Normal Yazı

Kalın Yazı

Kalın Yazı

Eğik Yazı

Eğik Yazı

Altı Çizili Yazı

Yazı belirli bir oranda küçük olarak görüntülenir

Yazıyı işaretlemek için kullanılır

Favori rengim ~~mavi~~ kırmızı.

H₂O

X³Y

- ‘ - ’ ve ‘<i> - ’ arasında tarayıcıda görsel olarak bir fark yoktur
- Arama motorları açısından ve yazıda vurguyu-önemi arttırmak anlamına gelmektedir

HTML

Bağlantılar(Link)



- ``
- Bağlantılar(linkler);
 - Dış bağlantılar
 - Site İçi bağlantılar
 - e-posta bağlantısı
 - dosya bağlantısı
 - Sayfa içi bağlantılar

HTML

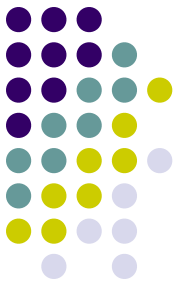
Dış Bağlantılar



- `Türkiye'nin Emlak İlan Sitesi`
- Yeni bir pencerede ve/veya sekmede
`Türkiye'nin Emlak İlan Sitesi`

HTML

Site İçi Bağlantılar



- Site içerisinde ziyaretçinin site içeriğine yönlendirilmesi için oluşturulan bağlantılardır.

- Anasayfa (index.html) => Ürünler(html/urunler.html)

```
<a href="html/urunler.html">Ürünler</a>
```

- Ürünler(html/urunler.html) => Anasayfa (index.html)

```
<a href="../index.html">Anasayfa</a>
```

HTML

E-Posta ve Dosya Bağlantıları



- E-Posta

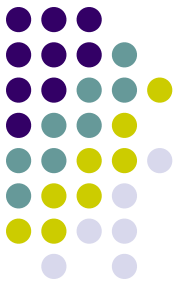
`E-posta gönder`

- Dosya (indirme-download)

`Bilgilendirme dosyası`

HTML

Sayfa İçi Bağlantılar



- Uzun içeriğe sahip sayfalarda kullanılan bir tekniktir.
- Örneğin;

```
<a href="#icerik1">İçerik 1</a>
```

...

...

...

```
<h2 id="icerik1">İçerik 1</h2>
```

...

```
<h2 id="icerik2">İçerik 2</h2>
```

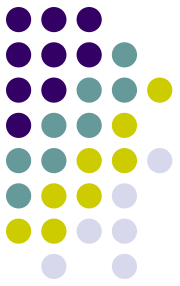
...

```
<h2 id="icerik3">İçerik 3</h2>
```

...

HTML

Bağlantılar & Renklendirme



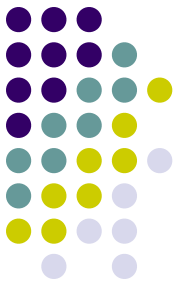
```
<html>
<head>
<style>
a:link {color: green;text-decoration: none;}
a:visited {color: pink;text-decoration: none;}
a:hover {color: red;text-decoration: underline;}
a:active {color: yellow;text-decoration: underline;}
</style>
</head>
<body>
<p>Linklerin varsayılan renkleri değiştirilebilir</p>
<a href="html_images.asp">HTML Linkler</a>
</body></html>
```

Linklerin varsayılan renkleri değiştirilebilir

[HTML Linkler](#)

- Linklerin renklerini değiştirmek için CSS kodları kullanılır
- “color” özelliği ile yazı rengi
- “text-decoration” ile altı çizili olup olmaması durumu belirlenir
- a:link => ilk durumu a:visited => ziyaret edilmiş
a:hover => üzerinde a:active => tıklanma anı

HTML Resimler



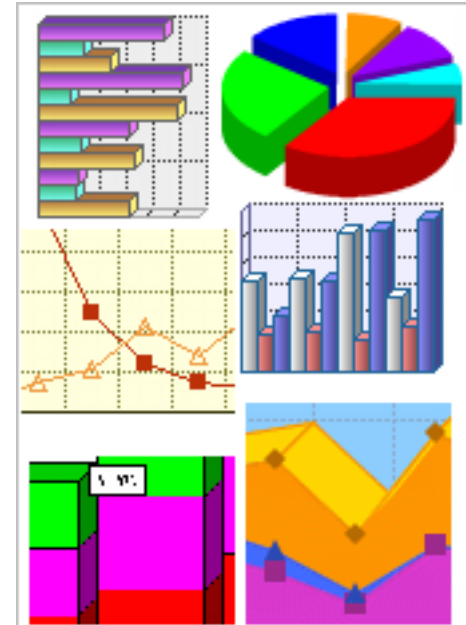
JPG



GIF



PNG



HTML

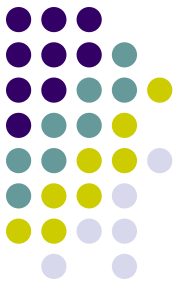
Resimler



- ``
- “src” : bu alana resmin uzak veya yerel adresi yazılır
- “alt” : alternatif yazı anlamında olup, temelde resim görüntülenmediğinde görünecek yazıdır. Ayrıca arama motorları açısından resimle ilgili açıklayıcı bilgilerin tanımlanması tavsiye edilen önemli bir özelliktir.
- “width” : resim genişliği
- “height” : resim yüksekliği

HTML

Resimler



Örnek;

```

```

```

```

```

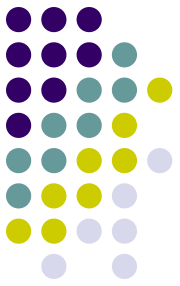
```

```

```

HTML

Resimler



Resime Link Vermek

```
<a href="http://www.cicekci.com">  
  
</a>
```

HTML5 versiyonunda "border" desteklenmeyecek olup CSS özellikleri kullanılacaktır

```
<a href="http://www.cicekci.com">  
  
</a>
```

HTML

Listeler



- Sırasız Listeler
- Sıralı Listeler
- Açıklama Listeleri
- İç içe Listeler
- Yatay Listeler

HTML

Listeler



- Sırasız Listeler(**u**nordered **l**ist)

```
<ul>  
<li>Kahve</li>  
<li>Çay</li>  
<li>Süt</li>  
</ul>
```

Sırasız Liste

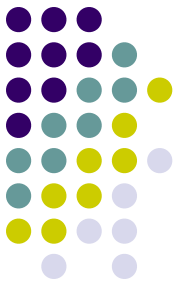
- Kahve
- Çay
- Süt

- Liste tipi

```
<ul type="disc|circle|square">
```

HTML

Listeler



- Sıralı Listeler(ordered list)

```
<ol>  
<li>Apple Iphone</li>  
<li>Samsung Note</li>  
<li>Nokia Lumia</li>  
</ol>
```

- Liste tipi

```
<ol type="1|A|a|I|i">
```

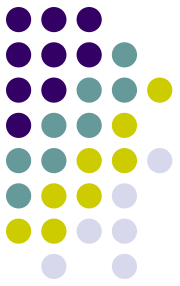
- | | |
|---|------------------------|
| 1 | : Sayı (varsayılan) |
| A | : Alfabetik Büyük harf |
| a | : Alfabetik Küçük harf |
| I | : Büyük Roma rakamı |
| i | : Küçük Roma rakamı |

Sıralı Liste

1. Apple Iphone
2. Samsung Note
3. Nokia Lumia

HTML

Listeler



- Açıklama Listeleri(**d**efinition **l**ist)

<d1>

```
<dt><strong>ul</strong> ekileti</dt>
```

```
<dd>Sırasız Listeler için kullanılır</dd>
```

```
<dt><strong>ol</strong> etileti</dt>
```

```
<dd>Sıralı Listeler için kullanılır</dd>
```

```
<dt><strong>dl</strong> etileti</dt>
```

```
<dd>Açıklama Listeleri için kullanılır</dd>
```

</d1>

Açıklama Listesi

ul ekileti

Sırasız Listeler için kullanılır

ol etileti

Sıralı Listeler için kullanılır

dl etileti

Açıklama Listeleri için kullanılır

HTML

Listeler



- İç içe Listeler

```
<ul>
```

```
<li>Tıp Fakültesi</li>
```

```
<li>Mühendislik Fakültesi
```

```
<ol>
```

```
<li>Bilgisayar</li>
```

```
<li>Elektronik</li>
```

```
<li>Gıda</li>
```

```
</ol>
```

```
</li>
```

```
<li>Fen-Edebiyat Fakültesi</li>
```

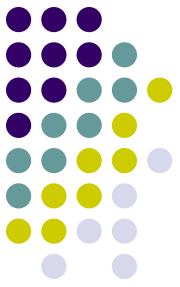
```
</ul>
```

İç İçe Listeler

- Tıp Fakültesi
- Mühendislik Fakültesi
 1. Bilgisayar
 2. Elektronik
 3. Gıda
- Fen-Edebiyat Fakültesi

HTML

Listeler



- Yatay Listeler (menu)

```
<ul id="menu">  
  <li><a href="/htmlders.html">HTML</a></li>  
  <li><a href="/cssders.html">CSS</a></li>  
  <li><a href="/jsders.html">JavaScript</a></li>  
  <li><a href="/phpders.html">PHP</a></li>  
</ul>
```

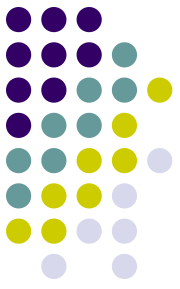
Normal liste kodunu html olarak hazırlıyoruz.

Liste elemanları blok elemanlardır, yanyana gelmezler

Ama...

HTML

Listeler



```
<style>
ul#menu { padding: 0;}
ul#menu li { display: inline;}
ul#menu li a {
    background-color: black;
    color: white;
    padding: 10px 20px;
    text-decoration: none;
}
ul#menu li a:hover { background-color: orange;}
</style>
```

Yatay Liste(Menu)

HTML

CSS

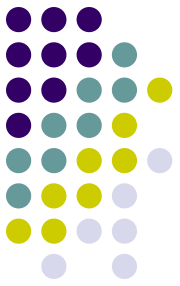
JavaScript

PHP

Html kodumuzdan önce yazacağımız CSS kodları ile listenin yapısına müdahale edebilir ve yatay görünümlü bir liste elde edebiliriz.

HTML

Tablolar



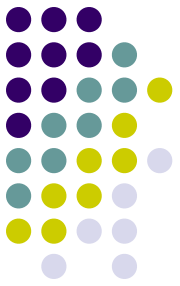
```
<table border="1">
  <tr>
    <th>Ay</th>
    <th>Gelir</th>
    <th>Gider</th>
  </tr>
  <tr>
    <td>Ocak</td>
    <td>200</td>
    <td>120</td>
  </tr>
  <tr>
    <td>Şubat</td>
    <td>180</td>
    <td>100</td>
  </tr>
</table>
```

- table
- tr(**t**able **r**ow) : satır
- th(**t**able **h**header) : başlık
- td(**t**able **d**ata) : hücre

Ay	Gelir	Gider
Ocak	200	120
Şubat	180	100

HTML

Tablolar - Table Özellikleri



- Tablo hizalama

```
<table align="left|right|center">
```

- Tablo zemin rengi

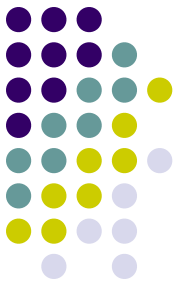
```
<table bgcolor="#00FF00">
```

- Tablo kenar çizgisi

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

- Tablo genişliği

```
<table width="350">
```



HTML

Tablolar - Table Özellikleri

- Tablo Hücre içi boşluk

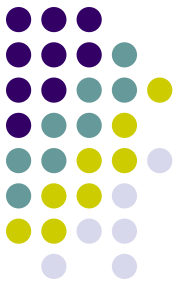
```
<table cellpadding="10">
```

Tablo cellpadding: yok

Ay	Gelir
Ocak	100 TL

Tablo cellpadding: var

Ay	Gelir
Ocak	100 TL



HTML

Tablolar - Table Özellikleri

- Tablo Hücrelerarası boşluk

```
<table cellpadding="10">
```

Tablo cellpadding: yok

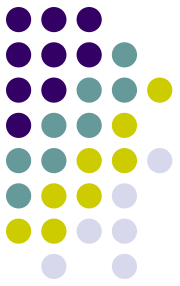
Ay	Gelir
Ocak	100 TL

Tablo cellpadding: var

Ay	Gelir
Ocak	100 TL

HTML

Tablolar - tr Özellikleri



- Satırda yatay hizalama

```
<tr align="left|right|center|justify">
```

- Satırda dikey hizalama

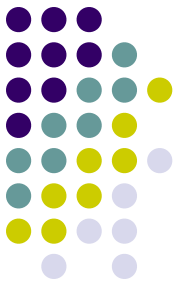
```
<tr valign="top|middle|bottom">
```

- Satırda zemin rengi

```
<tr bgcolor="#00FF00">
```

HTML

Tablolar - td,th Özellikleri



- Hücrede yatay hizalama

```
<td align="left|right|center|justify">
```

- Hücrede dikey hizalama

```
<td valign="top|middle|bottom">
```

- Hücrede zemin rengi

```
<td bgcolor="#00FF00">
```

- Hücrede genişlik-yükseklik

```
<td width="400" height="250">
```

HTML

Tablolar - td,th Özellikleri



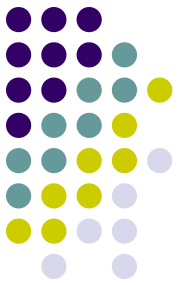
- Hücre Sütun birleştirme

```
<table border="1" cellpadding="10">
  <tr>
    <th>Ay</th>
    <th>Gelir</th>
  </tr>
  <tr>
    <td>Ocak</td>
    <td>200</td>
  </tr>
  <tr>
    <td>Şubat</td>
    <td>180</td>
  </tr>
  <tr>
    <td colspan="2">Toplam: 380</td>
  </tr>
</table>
```

Ay	Gelir
Ocak	200
Şubat	180
Toplam: 380	

HTML

Tablolar - td,th Özellikleri



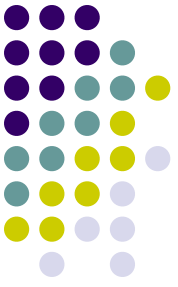
- Hücre Satır birleştirme

```
<table border="1" cellpadding="10">
  <tr>
    <th>Ay</th>
    <th>Gelir</th>
    <th>Toplam</th>
  </tr>
  <tr>
    <td>Ocak</td>
    <td>200</td>
    <td rowspan="2">380</td>
  </tr>
  <tr>
    <td>Şubat</td>
    <td>180</td>
  </tr>
</table>
```

Ay	Gelir	Toplam
Ocak	200	380
Şubat	180	

HTML

Tablolar - Örnekler



Adı	Soyadı	Notu
Ahmet	Beyaz	50
Ayşe	Mavi	94
Mehmet	Kara	80

HTML

Tablolar - Örnekler



A	
B	C
D	

HTML

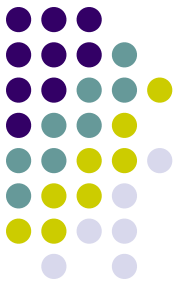
Tablolar - Örnekler



A	B	
	C	D

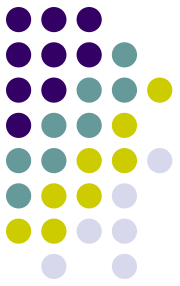
HTML

Tablolar - Örnekler



A	B	
E	C	D

HTML Formlar



`<form>`

..... •

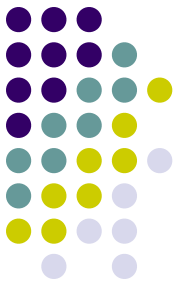
..... •

`</form>`

- Formlar web sayfalarında veri göndermek amacıyla kullanılır
- Örnek : Web sitelerindeki üye kayıt, üye giriş ve arama alanları

HTML

Formlar - Form Özellikleri



- Veri Gönderme Metodu

```
<form method="post|get">
```

- post metodu veriyi gizli gönderir ve güvenlidir
- get metodu veriyi adres satırında açıktan gönderir ve güvenli değildir

- Verilerinin Gönderileceği Adres

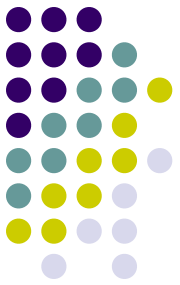
```
<form action="kayit.php">
```

- Form adı

```
<form name="form1">
```

HTML

Formlar - Form Özellikleri



- Örnek

```
<form method="post" action="kayit.php" name="form1">
```

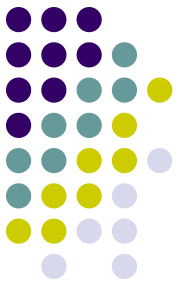
```
..... •
```

```
..... •
```

```
</form>
```


HTML

Formlar - Form Elemanları



- Form elemanları form etiket bloğunun içerisine yazılır
- Form elemanlarının çoğu

```
<input type="" />
```

etiketi ile oluşturulur.

Ancak;

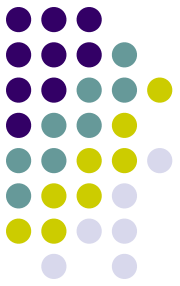
```
<textarea></textarea>
```

```
<select><option></option></select>
```

ile oluşturulan form elemanları da mevcuttur.

HTML

Formlar - Form Elemanları



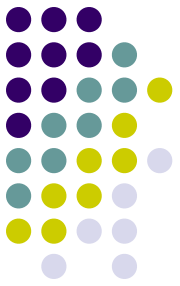
- **Text Kutusu**

```
<input  
    type="text"  
    name="ad"  
    size="20"  
    maxlength="10"  
>
```

- type : form elemanının türünü belirler. 'text' olmalıdır
- name : tüm form elemanlarında nesnenin adını tanımlamak için kullanılır
- size : Text kutusunun genişliğini karakter uzunluğu cinsinden belirler
- maxlength : En fazla kaç karakter veri girileceğini belirler

HTML

Formlar - Form Elemanları



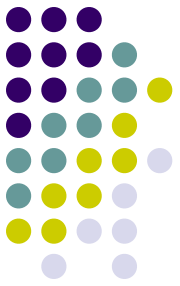
- Maskelenmiş Text Kutusu (Şifre Kutusu)

```
<input  
    type="password"  
    name="sifre"  
    size="20"  
    maxlength="10"  
>
```

- type : 'password' olmalıdır

HTML

Formlar - Form Elemanları



- Seçim Kutuları

```
<input  
type="checkbox" name="s1" value="f" checked  
>  
Futbol
```

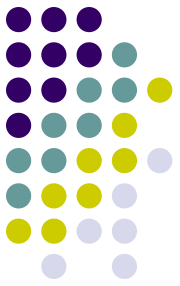
```
<input  
type="checkbox" name="s2" value="b"  
>  
Basketbol
```

- type : 'checkbox' olmalıdır
- value : değer belirlenebilir
- checked : Seçili olmasını sağlar

Futbol Basketbol

HTML

Formlar - Form Elemanları



- Çoktan Tek Seçim Kutuları

```
<input  
type="radio" name="s1" value="e" checked  
>  
Erkek
```

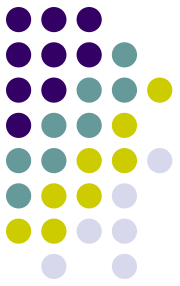
```
<input  
type="radio" name="s1" value="k"  
>  
Kadın
```

- type : 'radio' olmalıdır
- name : hepsinde aynı olmalıdır
- value : değer belirlenebilir
- checked : Seçili olmasını sağlar

Erkek Kadın

HTML

Formlar - Form Elemanları



- **Gizli Eleman**

```
<input type="hidden" name="id" value="5" />
```

- Ziyaretçiye görünmez
- type : 'hidden' olmalıdır
- value : değer belirlenebilir

- **Dosya Yükleme Elemanı**

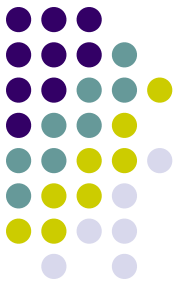
```
<input type="file" name="dosya"/>
```

- type : 'file' olmalıdır

Dosya seçilmedi

HTML

Formlar - Form Elemanları



- Butonlar

```
<input type="submit" name="onay" value="Gönder" />
```

```
<input type="reset" name="iptal" value="İptal" />
```

```
<input type="button" name="btn" value="Özel" />
```



- type : 'submit' => formdaki verilerin 'action' alanında tanımlanan adrese gönderir
- type : 'reset' => formdaki verilerin ilk durumuna gelmesini sağlar
- type : 'button' => özel işlevi olan buton oluşturulmasını sağlar
- value : Buton üzerinde görünen yazıyı belirler

HTML

Formlar - Form Elemanları



- **Text Alanı**

```
<textarea rows="5" cols="25" name="adres">
```

Bu alana çok satırlı bilgi yazılabilir..

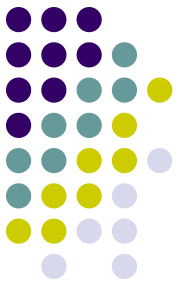
```
</textarea>
```

A rectangular text area form with a thin border. It contains the text "Bu alana çok satırlı bilgi yazılabilir.." in a black, sans-serif font. The text is positioned at the top of the form, leaving significant white space below it. A small double-slash icon is visible in the bottom right corner of the form's border.

- textarea : text alanının etiketleridir
- rows : text alanının kaç satırlık yükseklikte görüneceğini belirler
- cols : text alanının kaç karakterlik genişlikte görüneceğini belirler

HTML

Formlar - Form Elemanları



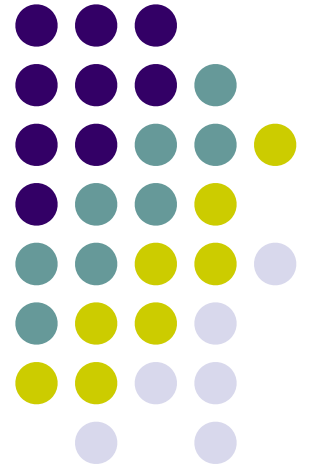
- Liste Kutusu

```
<select size="1" name="sehirler">  
  <option value="01">Adana</option>  
  <option value="06">Ankara</option>  
  <option value="27" selected>Gaziantep</option>  
  <option value="33">Mersin</option>  
  <option value="34">İstanbul</option>  
</select>
```

- select : liste kutusunun etiketleridir
- select => size : listenin görünür eleman sayısını belirler. Varsayılan değer '1' dir.
- option : liste elemanını belirler
- option => value : liste elemanının değeri belirlenebilir
- option => selected : liste elemanının seçili olmasını sağlar

CSS

Cascading Style Sheet



CSS

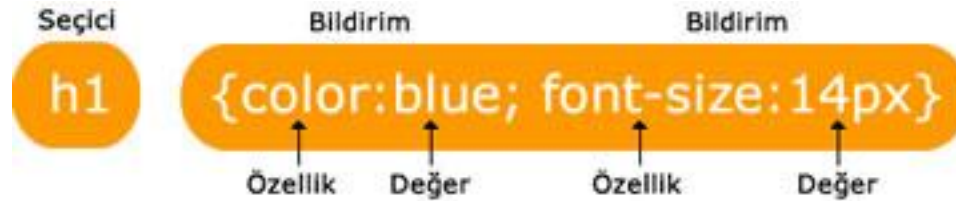


CSS, açılımı “Cascading Style Sheet” olan ve dilimize Stil şablonları olarak yerleşen basit ve kullanışlı bir işaretleme dilidir. Stil şablonu HTML öğelerine (yazı, paragraf, kenar çizgisi, resim, bağlantı...) stil vermek amacıyla kullanılır. Diğer bir deyişle sayfamızın içeriğinin biçimlendirilme işleminin yapıldığı kısımdır.

CSS'nin Yapısı



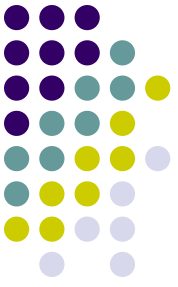
CSS'nin söz dizimi aşağıda görüldüğü gibi “Seçici” ve “Bildirim” olarak iki ana bölümden oluşur. Bildirim ise kendi içinde özellik ve değer olarak iki kısma ayrılmaktadır.



- •Seçici olarak tüm HTML etiketlerini kullanabileceğimiz gibi ileride göreceğimiz ID (kimlik) ve Class (sınıf) seçicilerini de kullanabiliriz.
- •Bildirim bloğu süslü parantez ile açılır ve kapanır. Bildirimler arasında “;” (noktalı virgül) kullanılır. Her Bildirim, "Özellik" ve "Değer" içerir. Özellik ve değer arası “:” (iki nokta üst üste) ile ayrılır.
- •Özellik, değiştirilmek istenen stil özneliğidir. Örnek; color, font-size vb.
- •Değer ise stil özneliklerine verilen değerlerdir. Örnek; blue, 14px vb.

CSS Seçiciler

- Sınıf(Class) Seçicisi
- Kimlik(ID) Seçicisi
- Etiket(Tag) Seçicisi



CSS Seçiciler

Sınıf(Class)Seçicisi



- Sınıf seçicileri birden fazla öğeye uygulanabilir ve birden fazla kullanılabilir
- Sınıf seçici olarak belirlediğimiz ismin önüne “.” (nokta) koymamız gerekmektedir

```
<style>
.onemli{
  color: #F00;
  font-weight:bold;
}
</style>
```

```
<h1 class="onemli">Bu bir Başlık</h1>
<p class="onemli">Bu bir deneme yazısıdır</p>
```

CSS Seçiciler

Kimlik(ID)Seçicisi



- ID seçiciler ise yalnız bir ögeye atanır ve bir kez kullanılırlar
- ID seçici olarak belirlediğimiz ismin önüne “#” koymamız gerekmektedir
- Gerek sınıf (class) seçici olsun, gerekse ID seçici olsun isim verirken Türkçe karakter kullanılmamalı ve rakamla başlamamalıdır.

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

```
<p id="haber_baslik">Bu bir Başlık</h1>  
<p>Bu bir deneme yazısıdır</p>
```

CSS Seçiciler

Etiket(Tag)Seçicisi



- Etiket seçicileri aynı türden tüm öğelere (etiketlere), istediğimiz stili uygulamak için kullanılır

```
<style>
```

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

```
  p{color: #F00;font-weight:bold;}
```

```
</style>
```


CSS Kullanım Alanları



- Yerel Kullanım Alanı
- Global Kullanım Alanı
- Bağlantılı Kullanım Alanı

CSS Kullanım Alanları

Yerel Kullanım Alanı

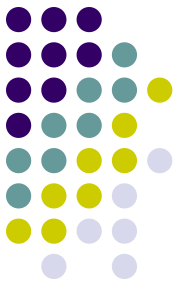


- Etiket içinde stil kullanımı (In-line Style) olarak da adlandırılır.
- Bu kullanım şeklinde CSS direkt etiketin “style” öz niteliği kullanılarak uygulanır
- Yerel CSS kullanımı, tek tek her etiketin içine yazılması gerektiği ve yönetiminin zor olması sebebiyle çok gerekmedikçe önerilmemektedir.

```
<p style="color:red;text-align:center">  
  Bu bir deneme yazısıdır  
</p>
```

CSS Kullanım Alanları

Global Kullanım Alanı



- Dahili stil kullanımını (Internal Style Sheet) olarak da adlandırılır.
- CSS kodumuz sayfamızın <head> veya <body> kısmında, <style>...</style> etiketi içinde yer alır
- Global (dahili) olarak yerleştirdiğimiz CSS kodumuz sadece yazıldığı sayfayı etkileyecektir

```
<html>
<head><title>Css Dersi</title></head>
<body>
  <style>
  p{
    text-align: center;
    color:blue;
  }
</style>
  <p id="haber_baslik">Bu bir Başlık</h1>
  <p style="color:red;text-align:center">Bu bir deneme yazısıdır</p>
</body>
</html>
```

CSS Kullanım Alanları

Bağlantılı Kullanım Alanı



- Harici stil kullanımını (External Style Sheet) olarak da adlandırılır.
- Stilimizi “.css” uzantılı stil şablonu dosyasına kaydederek, kullanmak istediğimiz sayfalarda <link> etiketi yardımıyla çağırırız.
- HTML kısmını ile CSS kısmını birbirinden tamamen ayırmış oluruz

```
<html>
<head>
<title>Css Dersi</title>
<link href="stil.css" rel="stylesheet" type="text/css" />
</head>
<body>
.....
</body>
</html>
```

Stil Şablonu Özellikleri



- Zemin Özellikleri
- Font Özellikleri
- Metin(Text) Özellikleri
- Kutu Modeli Özellikleri
- Tablo Özellikleri
- Pozisyon Özellikleri

Stil Şablonu Özellikleri

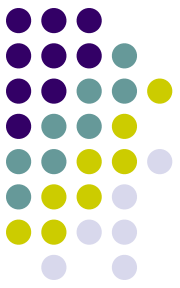
Zemin Özellikleri



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

Zemin Özellikleri

Zemin Rengi



```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<style type="text/css">
h1 {background-color:#3070E7;}
p {background-color:#9FFFFFF;}
div {background-color:#FFFF6F;}
</style>
</head>
<body>
<h1>CSS zemin rengi örneği</h1>
<div>
  Bu yazı div ögesi içindedir.
  <p>Bu paragraf kendi zemin rengine sahiptir.</p>
  Bu kısım hala div nesnesi sınırları içindedir.
</div>
</body>
</html>
```

CSS zemin rengi örneği

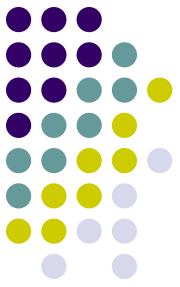
Bu yazı div ögesi içindedir.

Bu paragraf kendi zemin rengine sahiptir.

Bu kısım hala div nesnesi sınırları içindedir.

Zemin Özellikleri

Zemin Resmi



- “background-image” özelliği öğelere zemin resmi eklemek için kullanılır.

```
body { background-image:url(“zemin.jpg”); }
```


Zemin Özellikleri

Zemin Resmi Tekrarı

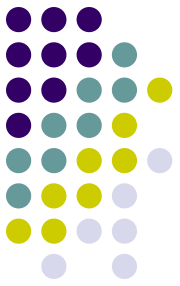


- “background-repeat”
- background-repeat:repeat; (Görselin hem yatay olarak hem de dikey olarak tekrar etmesini istiyorsak bu özelliği kullanırız. Bu özellik yazılmasa bile CSS varsayılan olarak bunu yapacaktır.)
- background-repeat:no-repeat; (Eklediğimiz görselin hiçbir şekilde tekrar etmesini istemiyorsak bu özelliği kullanırız.)
- background-repeat:repeat-x; (Eklediğimiz görsel yatay olarak yani x eksenini boyunca tekrar etsin istiyorsak bu özelliği kullanırız.)
- background-repeat:repeat-y; (Eklediğimiz görsel dikey olarak yani y eksenini boyunca tekrar etsin istiyorsak bu özelliği kullanırız.)

```
body {  
  background-image:url( “logo.gif”);  
  background-repeat : repeat;  
}
```

Zemin Özellikleri

Zemin Resmi Pozisyonu



- "background-position"
- Görselin sayfadaki pozisyonunu belirler

top left
center left
bottom left

top center
center center
bottom center

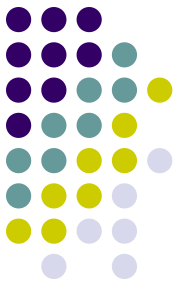
top right
center right
bottom right

```
body {  
background-image:url( "logo.gif");  
background-position : top left;  
// veya  
background-position : 60% 70%;  
// Vera  
background-position : 100px 200px;  
}
```

```
body{  
background:url("logo.gif") no-repeat center center;  
}  
  
body{  
background: url("logo.gif") no-repeat 100px 200px;  
}
```

Zemin Özellikleri

Zemin Resmi İliştirme



- "background-attachment"
- Zemine yerleştirdiğimiz görselin kaydırma çubuğu ile birlikte hareket edip etmeyeceğinin kontrolünü sağlar
- Görselin kaydırma çubukları ile birlikte hareket etmesini istiyorsak değer olarak **“scroll”** kullanırız
- Görselin kaydırma çubukları ile birlikte hareket etmemesini, sabit kalmasını istiyorsak değer olarak **“fixed”** kullanırız

```
body {  
background-image:url( "logo.gif");  
background-attachment : scroll;  
}
```

Stil Şablonu Özellikleri

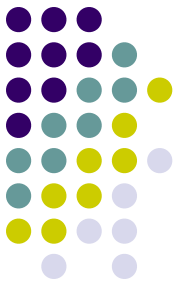
Font Özellikleri

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



Font Özellikleri

Font Ailesi



- "font-family"
- Bir metne ait font ailesini belirlemeye yarar.
- Birden fazla font ailesi ismi içerebilir, her font ailesi ismi virgülle ayrılır

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

```
h1{ font-family: Georgia, "Times New Roman", serif; }
```

Font Özellikleri

Font Boyutu



- “font-size”
- Yazı karakterlerinin, dolayısıyla da metnin boyutunu ayarlamaya yarar.
- Boyutu piksel olarak " **px** " , yüzde olarak " **%** " ya da " **em** " ile belirtebiliriz.

`body {font-size:100%;}` // varolan font boyutunun tamamını(%100) kullan.

`h1 {font-size:2.5em;}` // h1 için font boyutu(2.5em=2.5*16px =40px)belirler.

`h2 {font-size:25px;}` // h2 için font boyutu(25px) belirler.

Font Özellikleri

Font Kalınlığı



- “font-weight”
- Fontun kalınlık, incelik değerini belirlemeye yarar
- 100 (ince)- 900 (kalın) arasında bir değer alabildiği gibi bold, bolder, normal ve lighter değerlerini de alabilir.

```
p{  
font-weight: bold;  
}
```

Font Özellikleri

Font Stili



- “font-style”
- Metni eğik (italik), az eğik veya normal yapmak için kullanılır.

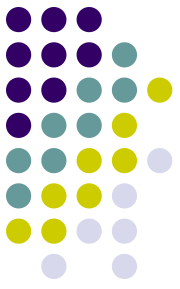
```
p { font-style:normal; } // Metin normal görünür.
```

```
p { font-style:italic; } // Metin italik yani eğik yazı olarak görünür.
```

```
p { font-style:oblique; } // Metin az eğik görünür(Çok fazla desteklenmez.).
```


Font Özellikleri

Satırlar Arası Mesafe Ayarı



- “line-height”
- Metin içindeki satırların arasındaki mesafeyi ayarlamak için kullanılır.

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

Stil Şablonu Özellikleri

Metin(Text) Özellikleri



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

Metin(Text) Özellikleri

Metin Rengi



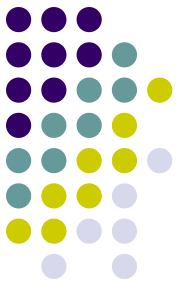
- “color”
- Metne renk vermek için kullanılır

```
h1 { color: #449BDB; }
```

```
p { color: black; }
```

Metin(Text) Özellikleri

Metin Hizalama



- “text-align”
- Metinleri sağa, sola,iki yana yaslayabilir veya ortalayabiliriz

text-align:left; (Metni sola hizalamak için kullanılır.)

text-align:center; (Metni ortaya hizalamak için kullanılır.)

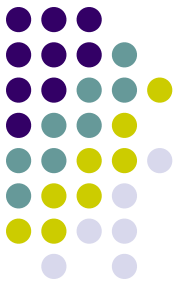
text-align:right; (Metni sağa hizalamak için kullanılır.)

text-align:justify; (Metin içeriğinin sağ taraftan kalan boşluklarını tamamlar, okumayı kolaylaştırır.
Gazete ve magazin dergilerinde olduğu gibi.)

```
h1 { text-align: left; }
```

Metin(Text) Özellikleri

Metin Dekorasyonu



- “text-decoration“
- Genellikle bağlantıların (link) altındaki çizgileri kaldırmakla birlikte metinlerin üst tarafını çizmeyi, altını çizmeyi, üzerini çizmeyi ve onlara yanıp sönme (blink) efekti vermeyi sağlar

text-decoration:none; (Genellikle a etiketine uygulanır, çizgiyi kaldırır.)

text-decoration:underline; (Metni altı çizili yapar.)

text-decoration:overline; (Metnin üst tarafını (tepesini) çizili yapar.)

text-decoration:line-through; (Metnin üzerinden (ortasından) çizgi geçirir.)

text-decoration:blink;(Metnin yanıp sönmesini (bir görünüp bir görünmemesi) sağlar.)

```
a {text-decoration:none;}
```

```
h1 {text-decoration:overline;}
```

```
h2 {text-decoration:line-through;}
```

```
h3 {text-decoration:underline;}
```

```
h4 {text-decoration:blink;}
```

Metin(Text) Özellikleri

Metin Dönüşümü



- “text-transform“
- Bir metindeki karakterleri büyük veya küçük harfe çevirebilir ya da metindeki kelimelerin ilk harflerini büyütebilir

text-transform:uppercase; (Metindeki tüm harfleri büyük harfe çevirir.)

text-transform:lowercase; (Metindeki tüm harfleri küçük harfe çevirir.)

text-transform:capitalize; (Metindeki kelimelerin ilk harflerini büyük hale çevirir.)

`.buyuk{text-transform:uppercase;}`

`.kucuk{text-transform:lowercase;}`

`.ilkharfbuyuk{text-transform:capitalize;}`

Metin(Text) Özellikleri

Metin Girintileme



- “text-indent“
- Paragraf başı yaparken cümlenin ne kadar içten başlayacağını ayarlamak için kullanılır

`p {text-indent:50px; }` // Her paragraf 50px içerden başlar.

Metin(Text) Özellikleri

Harfler Arası Mesafe



- “letter-spacing“
- Metin içinde yer alan karakterlerin arasındaki boşluğu ayarlamak için kullanılır

```
<html>
  <head>
<meta http-equiv="Content-Type"content="text/html; charset=iso-8859-9">
<title>CSS Öğreniyorum</title>
  </head>
  <body>
    <p style="letter-spacing:normal">Bilişim Teknolojileri Alanı </p>
    <p style="letter-spacing:5px;">Bilişim Teknolojileri Alanı</p>
    <p style="letter-spacing:-2px ;">Bilişim Teknolojileri Alanı</p>
  </body>
</html>
```

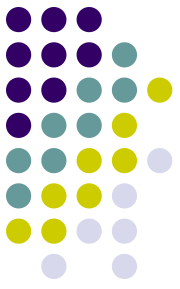
Bilişim Teknolojileri Alanı

Bilişim Teknolojileri Alanı

Bilişim Teknolojileri Alanı

Metin(Text) Özellikleri

Kelimeler Arası Mesafe



- “word-spacing“
- Metin içinde yer alan kelimelerin arasındaki mesafeyi (boşluğu) ayarlamak için kullanılır

```
p {  
  word-spacing : 12px; // Kelimeler arası mesafeyi 12 px yapar.  
}
```

Metin(Text) Özellikleri

Dikey Konum



- “vertical-align“
- Bir ögenin dikey olarak hizanlaması için kullanılır.

baseline : Orta (metin içeriğine göre)

sub : Altsimge

super : Üstsimge

top : Yukarı (elemente göre)

text-top: Yukarı (metin içeriğine göre)

middle : Orta (elemente göre)

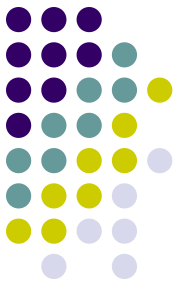
bottom : Alt (elemente göre)

text-bottom : Alt (metin içeriğine göre)

px ve yüzde (%) değerleri de verilebilir.

Metin(Text) Özellikleri

Dikey Konum



```
img { vertical-align: text-top;}
```

CSS Öğreniyorum

Varsayılan Değer

CSS Öğreniyorum

text-bottom değeri verilmiş hali

CSS Öğreniyorum

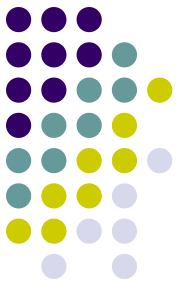
text-top değeri verilmiş hali

CSS Öğreniyorum

middle değeri verilmiş hali

Metin(Text) Özellikleri

Dikey Konum



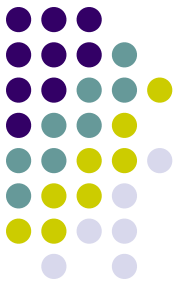
```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Başlıksız Belge</title>
  <style type="text/css">
    #mySpan
    {
      background-color:red;
      vertical-align:-50px;
    }
  </style>
</head>
<body>
  <p>
    Örnek <span id="mySpan" style="vertical-align:-50px;">Dikey Hizala
  </span>
  </p>
</body>
</html>
```

Örnek

Dikey Hizala

Stil Şablonu Özellikleri

Kutu Modeli Özellikleri



- Tablosuz tasarımların gereklilik kazandığı günümüzde, bu iş için esas etiketimiz olan “<div>” etiketi, kutu modeli (box model) oluştururken de esas etiket konumundadır.
- Kutu modeli üç ana ögeden oluşur;
 1. Kenar Dışı Boşluk (margin)
 2. Kenar Çizgisi (border)
 3. Kenar İçi Boşluğu (padding)
 4. Float, Clear ve Display

Stil Şablonu Özellikleri

Kutu Modeli Özellikleri



Kutu Modeli Özellikleri

Kenar Dışı Boşluk(margin)



- “margin“
- Kutumuzun diğer öğelerle arasındaki mesafeyi belirler
- **margin-top** (üst kenar dışı boşluğu)
margin-right (sağ kenar dışı boşluğu)
margin-bottom (alt kenar dışı boşluğu)
margin-left (sol kenar dışı boşluğu)
margin (kenar dışı boşlukları) (Kısaltma amaçlı kullanılır.)

`body { margin-top: 80px; }` // sayfamızın üst kenar dışı boşluğunu 80 px yapar.

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

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

Kutu Modeli Özellikleri

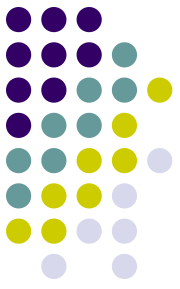
Kenar Çizgisi(border)



- “border“
- Kenar çizgisi özellikleri ile kenar çizgisinin biçimini, rengini ve kalınlığını ayarlanabilir
- **border-style** (kenar çizgisi biçemi)
- **border-width** (kenar çizgisi kalınlığı)
- **border-color** (kenar çizgisi rengi)
- **border** (kısa kullanım)

Kutu Modeli Özellikleri

Kenar Çizgisi Biçemi



- “border-style“
- Kenar çizgisinin nasıl olacağını (noktalı, kesik kesik, düz, çift, vb.) belirler. Herhangi bir değer yazılmadığı zaman “none” varsayılan olarak geçerlidir. Aşağıdaki değerleri alabilir.

dotted - noktalı
dashed - kesik
solid - düz
double - çift
groove - içe kabarık
ridge - dışa kabarık
inset - içe gömülü
outset - dışa gömülü

```
.border_1 { border-style:dotted;}
```

```
.border_2 {  
border-top-style: solid;  
border-bottom-style: solid;  
border-left-style: double;  
border-right-style: double;  
}
```

Kutu Modeli Özellikleri

Kenar Çizgisi Kalınlığı



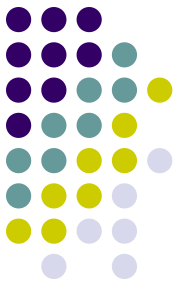
- “border-width“
- Kalınlığı ayarlamak için piksel (px) olarak değer verebilir veya thin,medium,thick değerlerini kullanabiliriz.

```
.border_1 {border-style: solid; border-width:15px; }
```

```
.border_1 {  
  border-style: solid;  
  border-top-width: 1px;  
  border-bottom-width: 1px;  
  border-left-width: 2px;  
  border-right-width: 2px;  
}
```

Kutu Modeli Özellikleri

Kenar Çizgisi Rengi



- “border-color“
- Kenar çizgisine renk vermek için kullanılır. Kenarların hepsine birden renk verebileceğimiz gibi tek tek de renk verebiliriz.

```
.border_1 {  
border-style:solid;  
border-color:blue;  
}
```

```
.border_1 {  
border-style:solid;  
  
border-width: 5px;  
border-top-color: #F60;  
border-left-color: #060;  
border-bottom-color: #F60;  
border-right-color: #060;  
text-align: center;  
}
```

Kutu Modeli Özellikleri

Kenar Çizgisi Kısaltma

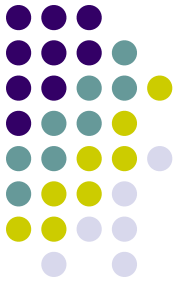


- “border“

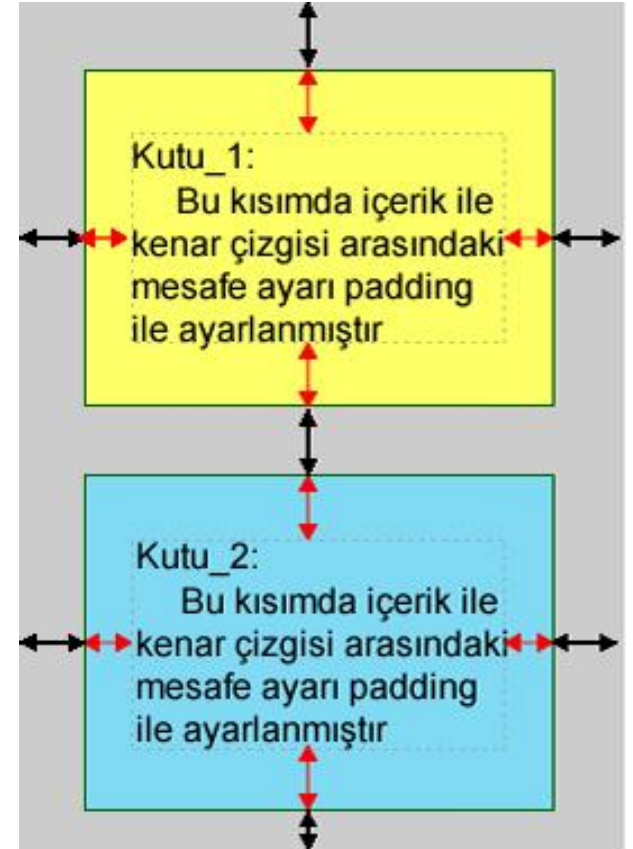
```
.border_1 { border 1px solid #F60 ;}
```

Kutu Modeli Özellikleri

Kenar İçi Boşluğu(padding)

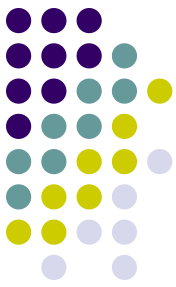


- “padding“
- Kenar çizgileri ile içerik arasındaki mesafeyi ayarlar
- **padding-top** (üst kenar içi boşluğu)
- **padding-right** (sağ kenar içi boşluğu)
- **padding-bottom** (alt kenar içi boşluğu)
- **padding-left** (sol kenar içi boşluğu)
- **padding** (kısaltma)



Kutu Modeli Özellikleri

Kenar İçi Boşluğu(padding)



```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Başlıksız Belge</title>
  <style type="text/css">
    .kutu_1 {
      text-align: justify;
      border:1px solid #060;
      padding: 10px 10px 10px 10px;
      margin:20px 10px 10px 20px;
      background-color:#FF6;
    }
    .kutu_2 {
      text-align: justify;
      border:1px solid #060;
      padding: 10px 10px 10px 10px;
      margin:20px 10px 10px 20px;
      background-color:#81DAF3;
    }
  </style>
  body {
    width: 45%; font-family:Verdana;
    font-size:12px; background-color:#CCC;  }
</head>
<body>
  <div class="kutu_1">
    İçerik ile kenar çizgisi arasındaki mesafe, üstten 10px,
    sağdan 5px alttan, 10px ve soldan 5px olarak ayarlanmıştır.
  </div>
  <div class="kutu_2">
    İçerik ile kenar çizgisi arasındaki mesafe, üstten 10px,
    sağdan 5px alttan, 10px ve soldan 5px olarak ayarlanmıştır.
  </div>
</body>
</html>
```

İçerik ile kenar çizgisi arasındaki mesafe, üstten 10px, sağdan 5px alttan, 10px ve soldan 5px olarak ayarlanmıştır.

İçerik ile kenar çizgisi arasındaki mesafe, üstten 10px, sağdan 5px alttan, 10px ve soldan 5px olarak ayarlanmıştır.

Kutu Modeli Özellikleri

Float, Clear Özelliđi

- Float özelliđini řu örnekle açıklayabiliriz: Bir su birikintisi üzerindeki kâğıt parçasının rüzgarla sađa sola doğru kaydığını (floating) düşünün, işte bu kâğıt parçasının sađa sola kayması gibi biz de nesnelerimizi float özelliđini kullanarak; sađa (float:right) ve sola (float:left) kaydırma işlemine tabi tutabiliriz

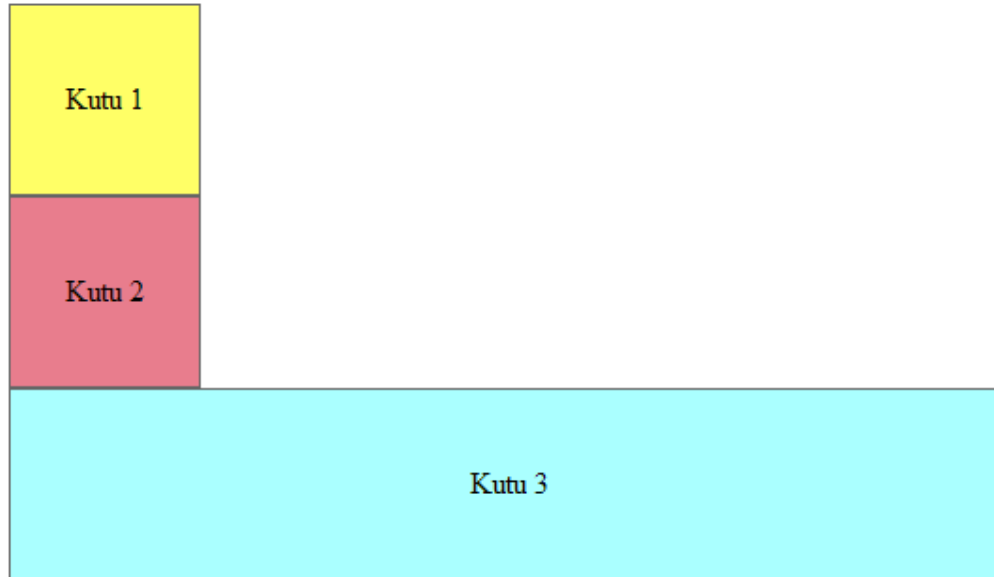


Kutu Modeli Özellikleri

Float, Clear Özelliği



- Aşağıdaki örnek float uygulanmamış durumdur. Div ile yapılmış üç adet kutu bulunmaktadır;
- div blok-level bir elemandır ve normal akışı aşağıda olduğu gibidir.

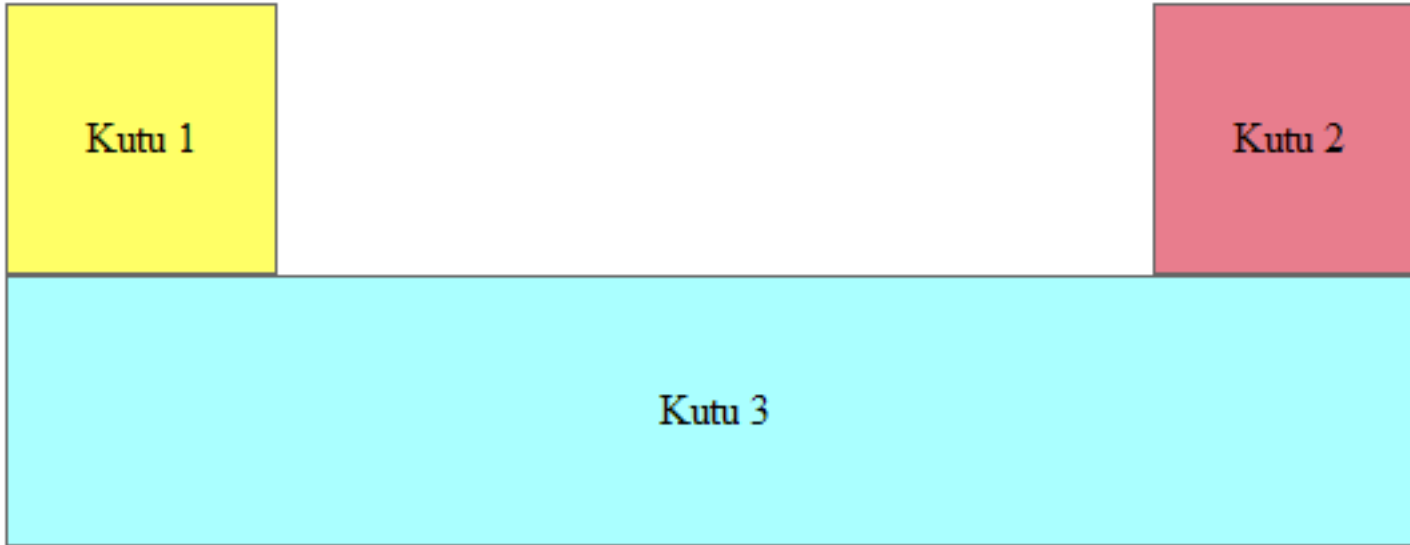


Kutu Modeli Özellikleri

Float, Clear Özelliği



- Şimdi bu kutulara float özelliğini uygulayarak kaydırma işlemine tabi turalım



Burada Kutu 1'e **float:left**; kutu 2'ye **float:right** uygulanmıştır. Kutu 3'ün konumunun sayfanın normal akışında devam etmesi, yani kaydırma işinin sona ermesi, için aşağıdaki özellik uygulanmıştır;

`clear:both;`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Başlıksız Belge</title>
<style type="text/css">
#kapsayici{
width:50%;}
#kutul{
float:left; /*sola kaydır */
background-color:#FF6;
border:1px solid #666;
width:100px;
height:100px;
text-align:center;
vertical-align:middle;
line-height:100px; }
#kutu2{
float:right; /* sağa kaydır */
background-color:#E87D8D;
border:1px solid #666;
width:100px;
height:100px;
text-align:center;
vertical-align:middle;
line-height:100px; }
#kutu3{
clear:both; /*her iki yönde de kaydırma işini bitir*/
background-color:#AFF;
border:1px solid #666;
width:100%;
height:100px;
text-align:center;
vertical-align:middle;
line-height:100px;
}
</style>
</head>
<body>
<div id="kapsayici">
<div id="kutul">
Kutu 1
</div>
<div id="kutu2">
Kutu 2
</div>
<div id="kutu3">
Kutu 3
</div>
</div>
</body>
</html>
```



Kutu Modeli Özellikleri

Blok - Inline Elemanlar (Display)



Normal olarak bir web sayfasındaki nesnelerin akışı, o elemanın seviye (level) türüne göre değişir. Seviye (level) türüne göre iki tip eleman vardır;

- **Blok-level elemanlar** : Blok şeklinde işlenirler. Sayfadaki normal akışları yukarıdan aşağı şekilde alt alta sıralanırlar. Blok-level elemanlar inline elemanları ve diğer blok-level elemanları içerebilirler. Tipik özellikleri, yeni bir satır başlangıcı yapmalarıdır. Örnek: div, h3, p, li...
- **Inline-level elemanlar**: Inline-level elemanlar aynı satırda işlenirler. Sayfadaki akışları soldan sağa doğrudur, başka bir inline elemanı içerebilirler. Yeni satır veya boşluk oluşturmazlar. Örnek: a, strong, span...

Kutu Modeli Özellikleri

Blok - Inline Elemanlar (Display)



Bu elemanların başlangıç değerlerini **display** özelliği ile değiştirebiliriz. Yani inline- level eleman olan bir linki (`<a>....`), blok-level bir elemana dönüştürebiliriz. Aynı şekilde blok-level elemanı, inline elemente dönüştürebiliriz.

Aşağıdaki örnek, liste elemanlarını "**inline**" olarak göstermektedir:

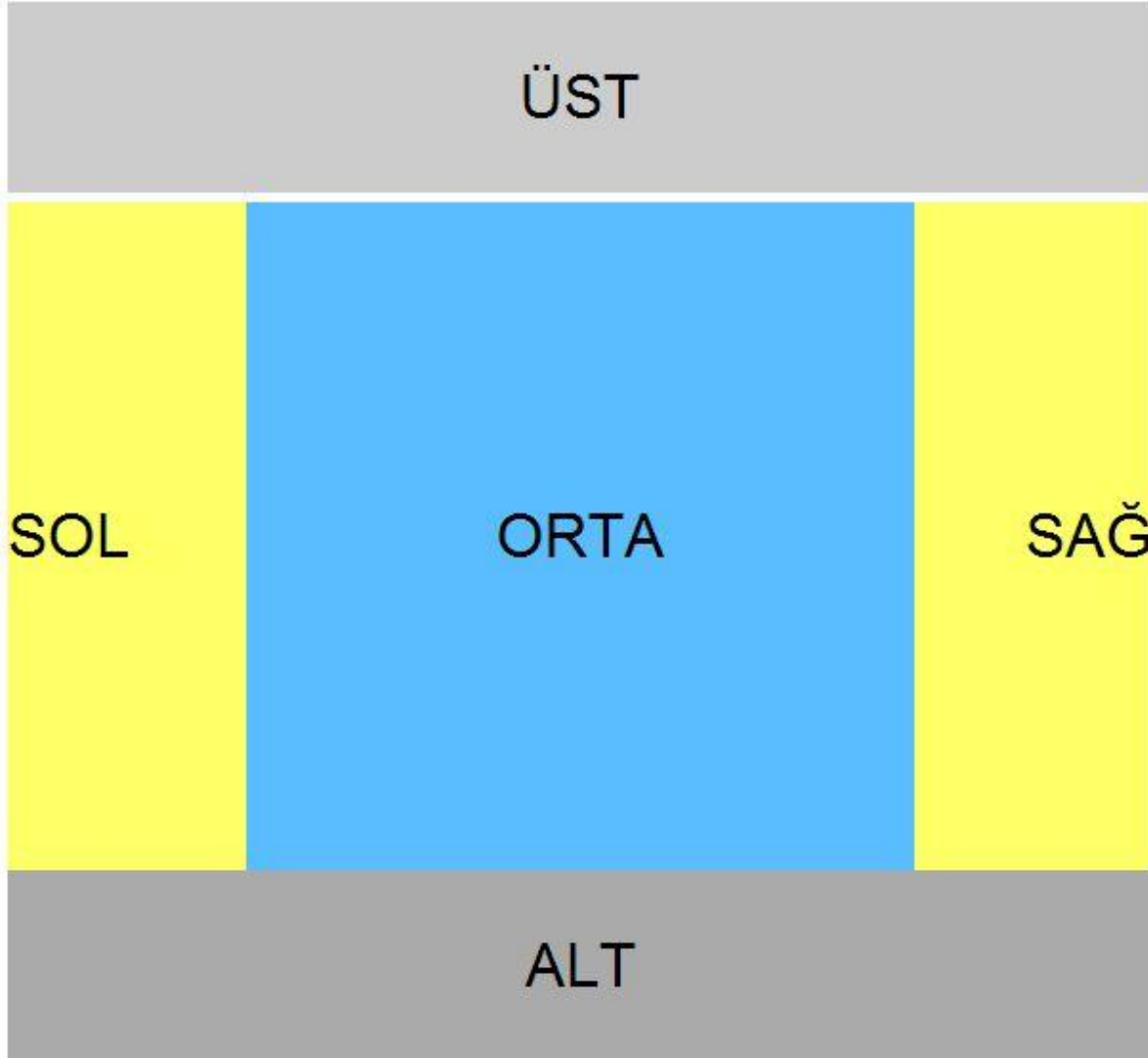
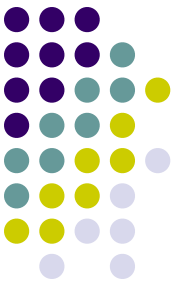
```
li {  
  display:inline;  
}
```

Aşağıdaki örnek, span elemanlarını "**block**" olarak göstermektedir:

```
span {  
  display:block;  
}
```

Kutu Modeli Özellikleri

Tablosuz Tasarım

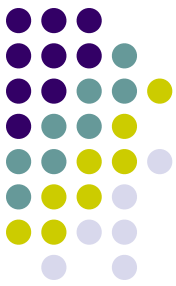


Kutu Modeli Özellikleri

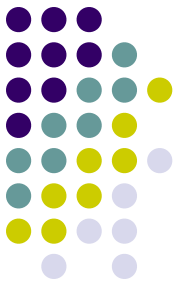
Tablosuz Tasarım

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<style type="text/css">

#ana {
    width:600px; margin:auto; font-size:32px; color:#000;
    font-family: Arial, Helvetica, sans-serif; }
#ust {
    height: 100px; background-color:#CCC; margin-bottom: 5px;
    text-align:center;vertical-align:middle; line-height:100px;}
#sol {
    float:left; width:125px; height:350px; background-color:#FF6;
    line-height:350px;}
#orta {
    float: left; width:350px; height:350px; background-color:#59BDFE;
    text-align:center; line-height:350px; }
#sag {
    float: right; width:125px; height:350px; background-color: #FF6;
    text-align:right; line-height:350px;}
#alt {clear: both; height: 100px; background-color: #aaa;
    margin-top: 5px; text-align: center; line-height:100px; }
</style>
</head>
<body>
<div id="ana">
    <div id="ust">ÜST</div>
    <div id="sol">SOL</div>
    <div id="orta">ORTA</div>
    <div id="sag">SAĞ</div>
    <div id="alt">ALT</div>
</div>
</body>
```



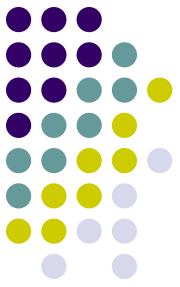
Liste Özellikleri



- `list-style-type`
- Sıralı veya sırasız listenin tipi belirlenir

```
ul.a {  
    list-style-type: circle;  
}  
ul.b {  
    list-style-type: square;  
}  
ol.c {  
    list-style-type: upper-roman;  
}  
ol.d {  
    list-style-type: lower-alpha;  
}  
ol, ul {  
    list-style-type: none;  
}
```

Liste Özellikleri



- **list-style-image**
- Sıralı veya sırasız liste işaretçisini resim olarak belirler

```
<style>
  ul{
    list-style-image: url('sq.gif');
  }
</style>
<ul>
  <li>İngilize</li>
  <li>Almanca</li>
  <li>Fransızca</li>
</ul>
```

- **İngilize**
- **Almanca**
- **Fransızca**

Liste Özellikleri



- **list-style-position**
- Sıralı veya sırasız liste işaretçisinin pozisyonunu belirler

```
ul{  
  list-style-position: inside;  
}
```

• İngilizce
• Almanca
• Fransızca

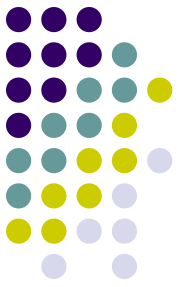
```
ul{  
  list-style-position: outside;  
}
```

• İngilizce
• Almanca
• Fransızca

- **Kısa kullanım**

```
ul {  
  list-style: square inside url("sq.gif");  
}
```

Liste Özellikleri



```
<html>
<head>
<style>
ul {
  border-left: 5px solid red;
  background-color: #f1f1f1;
  list-style-type: none;
  padding: 10px 20px;
}
</style>
</head>
<body>

<p>List with a red left border:</p>
<ul>
  <li>İngilizce</li>
  <li>Almanca</li>
  <li>Fransızca</li>
</ul>

</body>
</html>
```

İngilizce
Almanca
Fransızca

Liste Özellikleri



```
<html>
<head>
<style>
ul {
  list-style-type: none;
  padding: 0;
  border: 1px solid #ddd;
}
ul li {
  padding: 8px 16px;
  border-bottom: 1px solid #ddd;
}
ul li:last-child {
  border-bottom: none
}
</style>
</head>
<body>
<ul>
  <li>İngilizce</li>
  <li>Almanca</li>
  <li>Fransızca</li>
</ul>
</body>
</html>
```

İngilizce

Almanca

Fransızca

Tablo Özellikleri



```
table {  
  font-family: Arial, Helvetica, sans-serif;  
  border-collapse: collapse;  
  width: 100%;  
}
```

```
td,th {  
  border: 1px solid #ddd;  
  padding: 8px;  
}
```

```
tr:nth-child(even) {background-color: #f2f2f2;}
```

```
tr:hover {background-color: #ddd;}
```

```
th {  
  padding-top: 12px;  
  padding-bottom: 12px;  
  text-align: left;  
  background-color: #4CAF50;  
  color: white;  
}
```

Firma Adı	Yetkili	Şehir
A Firması	Ahmet Beyaz	Ankara
B Firması	Mehmet Siyah	İstanbul
C Firması	Zeynep Mavi	Gaziantep

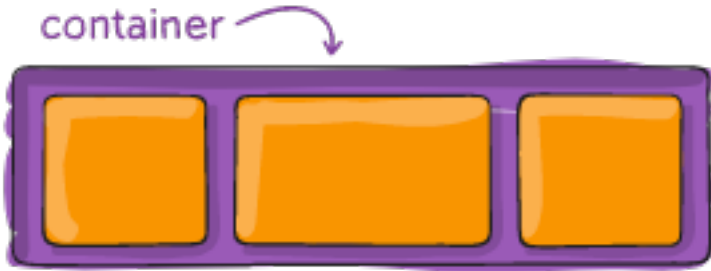
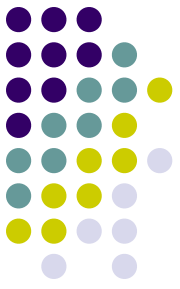
Flexbox Yapısı



- Flexbox yapısında ögeler ve bu ögeleri içinde bulunduran kapsayıcı kavramları söz konusudur.
- Flexbox yapısından önce, inline veya block eleman yönetiminde yetersiz ve uğraştırıcı bir kullanım söz konusuyken, esnek ve duyarlı bir yapıya sahip olan flexbox'ın çıkışından sonra bu işlemler çok daha kolay uygulanabilir hale geldi.
- Bilindik css yöntemi ile ögeleri sağa-sola konumlamak için float kullanır, dikey ortalamak için çeşitli css yöntemleri bir arada kullanılır. Ancak flexbox yapısı ile, artık kapsayıcıya ve içindeki ögelerine esneklik getirebiliyoruz.

Flexbox Yapısı

Kapsayıcı (Container) Özellikleri - display



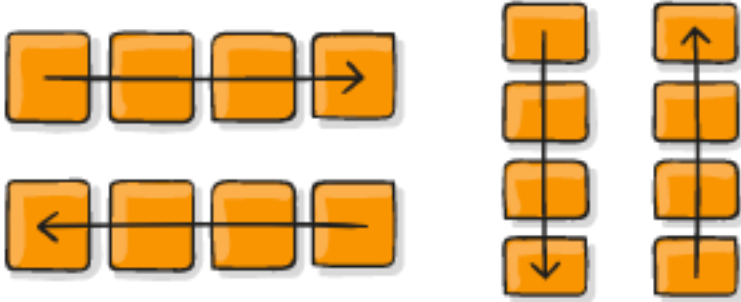
```
.container {  
  display: flex; /* ya da inline-flex */  
}
```

- `display: flex;` ya da `display: inline flex;` kullanılarak kapsayıcı içerisindeki öğelere artık esneklik vermeye başlayabiliriz. Bu tanımla artık bu kapsayıcının esnek bir görünümde olduğunu belirtiyoruz.

`flex` ile `inline-flex` farkı, `block` ile `inline` tanımları ile aynıdır.

Flexbox Yapısı

Kapsayıcı (Container) Özellikleri - flex-direction

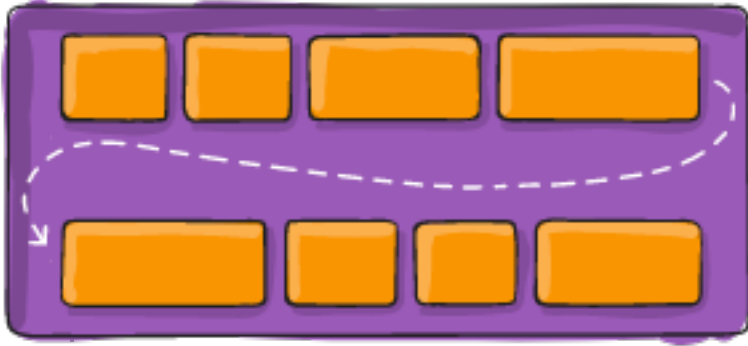


```
.container {  
  flex-direction: row | row-reverse | column | column-reverse;  
}
```

- Öğelerin yönünü belirlemek için kullanılır. Varsayılan olarak flex container yataya doğru uzar.
 - `row` (varsayılan) = soldan sağa doğru sıralar.
 - `row-reverse` = sağdan sola doğru sıralar.
 - `column` = yukarıdan aşağı doğru sıralar.
 - `column-reverse` = aşağıdan yukarı doğru sıralar.

Flexbox Yapısı

Kapsayıcı (Container) Özellikleri - flex-wrap



```
.container {  
  flex-wrap: nowrap | wrap | wrap-reverse;  
}
```

- Varsayılan olarak esnek öğeler tek bir satıra sığmaya çalışırlar. Gerektiğinde birden fazla satıra yaymak için bu özelliği kullanabilirsiniz.
 - `nowrap` (varsayılan) = Tüm öğeleri tek bir satıda yan yana listeler.
 - `wrap` = Öğeler gerektiğinde birden fazla satırda yukarıdan aşağı doğru listelenir.
 - `wrap-reverse` = Öğeler gerektiğinde birden fazla satırda aşağıdan yukarı doğru listelenir.

Flexbox Yapısı

Kapsayıcı (Container) Özellikleri - justify-content



flex-start



flex-end



center



space-between



space-around



space-evenly



```
.container {  
    justify-content: flex-start  
| flex-end | center | space-  
between | space-around | space-  
evenly;  
}
```

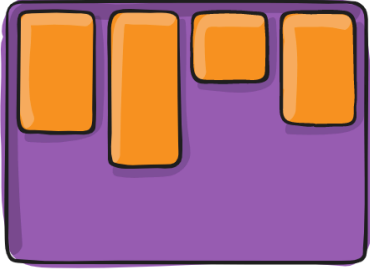
- Öğelerin yatay ekseninde hizasını ayarlamak için kullanılır.
 - `flex-start` (varsayılan) = Öğeleri başlangıç noktasında (sola dayalı) hizalar.
 - `flex-end` = Öğeleri bitiş noktasında (sağa dayalı) hizalar.
 - `center` = Öğeleri ortada hizalar.
 - `space-between` = İlk öge sola dayalı, son öge sağa dayalı ve diğerleri ortalı olacak şekilde hizalar.
 - `space-around` = İlk ve son öge hariç diğerlerinin arası eşit olacak şekilde hizalar.
 - `space-evenly` = Tüm öğelerin arası aynı eşitlikte olacak şekilde hizalar.

Flexbox Yapısı

Kapsayıcı (Container) Özellikleri - align-items



flex-start

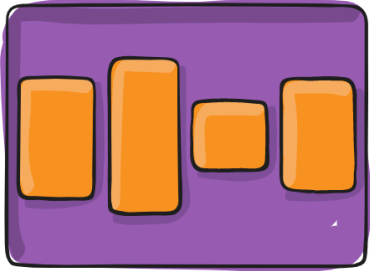


flex-end

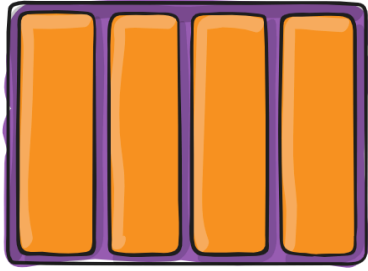


```
.container {  
  align-items: stretch | flex-start | flex-end | center | baseline;  
}
```

center



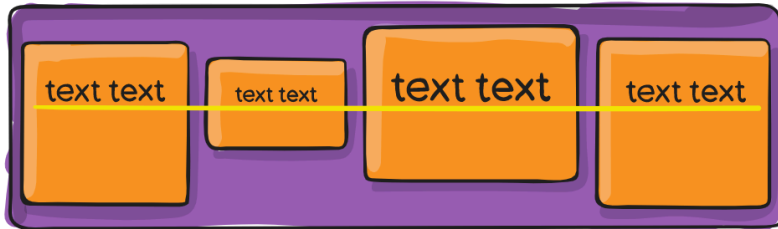
stretch



- Öğelerin dikey ekseninde hizasını ayarlamak için kullanılır.

- flex-start = Öğeleri yukarıya doğru hizalar.
- flex-end = Öğeleri aşağıya doğru hizalar.
- center = Öğeleri ortaya doğru hizalar.
- baseline = Öğelerin içindeki ilk satırı baz alarak ortalama yapar.
- stretch (varsayılan) = Hizalama yapmaz, öğelerin yüksekliklerini kapsayıcıya göre uzatır.

baseline

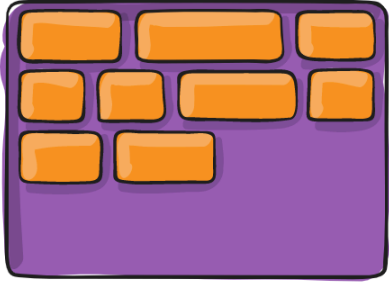


Flexbox Yapısı

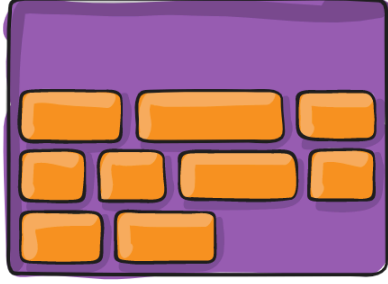
Kapsayıcı (Container) Özellikleri - align-content



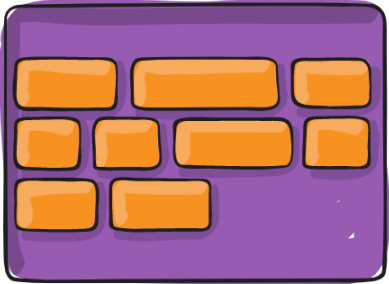
flex-start



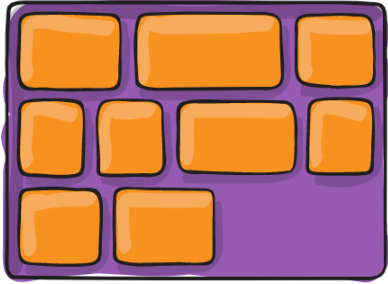
flex-end



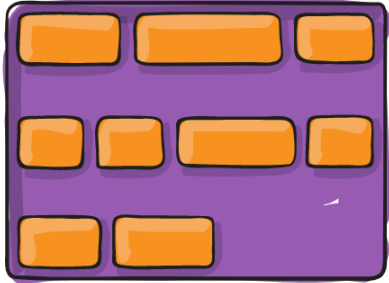
center



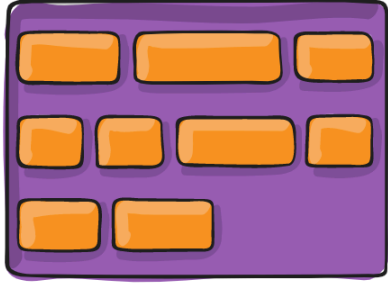
stretch



space-between



space-around

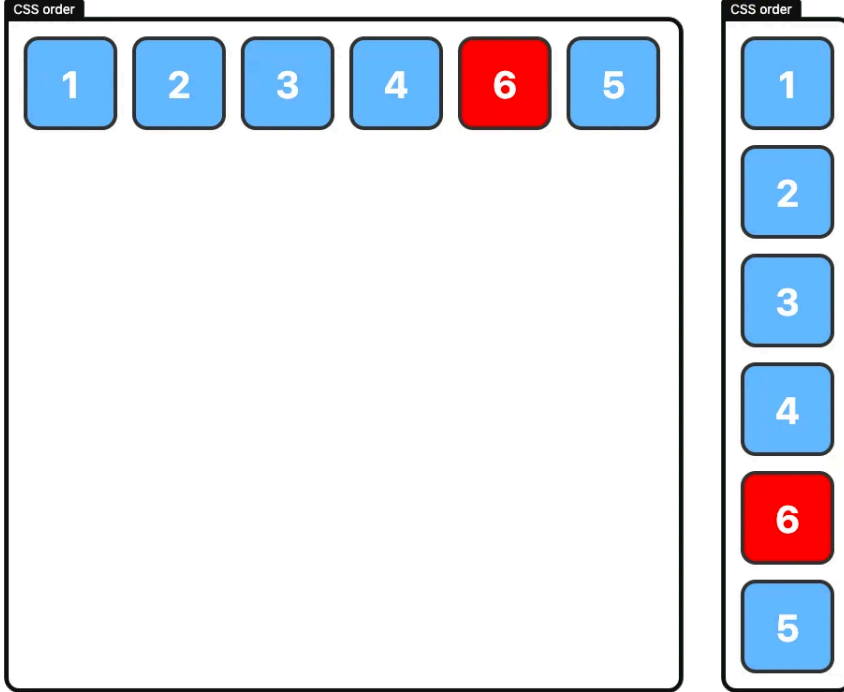
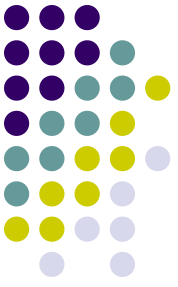


```
.container {  
  align-content: stretch |  
  flex-start | flex-end | center |  
  space-between | space-evenly;  
}
```

- Çok satırlı yapılarda öğelerin dikey ekseninde hizasını ayarlamak için kullanılır.
- `flex-start` = Öğeleri dikey ekseninde başta hizalar.
- `flex-end` = Öğeleri dikey ekseninde sonra hizalar.
- `center` = Öğeleri dikey ekseninde ortalı hizalar.
- `stretch` (varsayılan) = Öğeleri hizalamaz, yüksekliklerini uzatır.
- `space-between` = İlk satırdaki öğeler başta, son satırdaki öğeler sonra olmak üzere diğer satırdaki öğeleri dikey ekseninde ortalar.
- `space-evenly` = Öğeleri satırlar arası eşit oranda olacak şekilde dikey ekseninde ortalar.

Flexbox Yapısı

Öğeler (items) Özellikleri - order



```
.item5 {  
  order: 6;  
}  
.item6 {  
  order: 5;  
}
```

- Öğelerin sıralanışını deęiřtirmek için kullanılır.

Flexbox Yapısı

Öğeler (items) Özellikleri - flex-grow

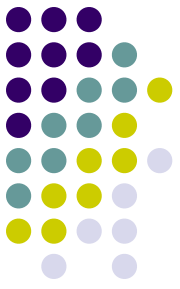


```
.item {  
  flex-grow: 1;  
}  
.item4 {  
  flex-grow: 2;  
}  
.item5 {  
  flex-grow: 4;  
}
```

- Bir öğeyi diğerlerine göre daha büyük göstermek için bu özelliği kullanıyoruz. Varsayılan değeri 0'dır.
- Eğer öğelerden birisinin değerini 2 yaparsanız diğerlerine göre daha büyük olacaktır ancak kalanı daralacaktır. İlk olarak tüm öğelerin değerini 1 yapıp, daha sonra bir öğenin değerini 2 yaparsanız, sonuç olarak diğer öğelere göre 2 kat daha büyük bir öğeye sahip olacaksınız.

Flexbox Yapısı

Öğeler (items) Özellikleri - flex-shrink

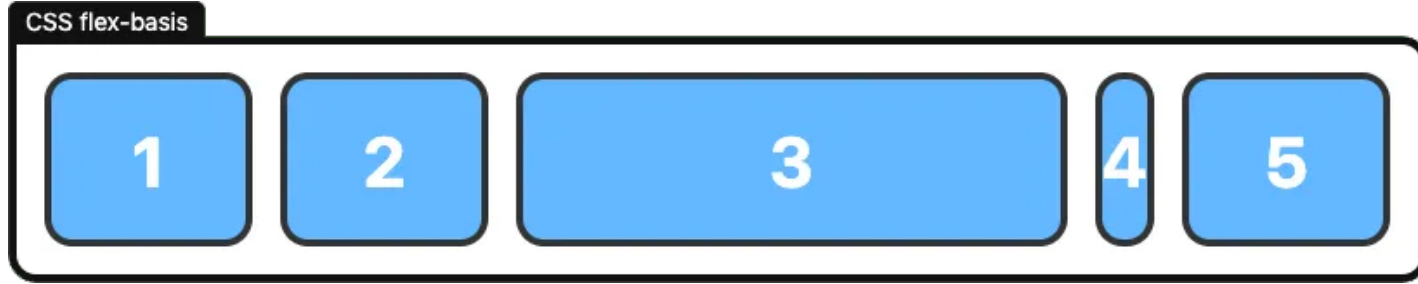


```
.item4 {  
  flex-shrink: 2;  
}  
.item5 {  
  flex-shrink: 3;  
}
```

- `flex-grow` özelliğinin tam aksine, küçültme işlemi için kullanılır. Varsayılan olarak değeri 1'dir.
- Eğer bir öğeye değer olarak 2 verirsiniz, diğerlerine göre küçülmesi gerektiğinde daha önce küçülecektir. Değeri artırırsanız örneğin 4. öğeye 2, 5. öğeye 3 değeri verirsiniz, küçülme esnasında her ikisinde küçülecektir ancak 5. öğe 4. öğeye göre daha hızlı küçülecektir.

Flexbox Yapısı

Öğeler (items) Özellikleri - flex-basis



```
.item3 {  
  flex-basis: 40%;  
}  
.item4 {  
  flex-basis: 0;  
}
```

- Öğelerin eğer alanı yetiyorsa olması gereken genişliği belirlemek için kullanılır. Değer olarak tüm uzunluk birimleri (px, pt, em vb.) kullanılabilir. Varsayılan olarak değeri auto'dur. Eğer değer olarak 0 belirlenirse, doğrudan sadece içindeki değer kadar genişliği olur

Flexbox Yapısı

Öğeler (items) Özellikleri - flex

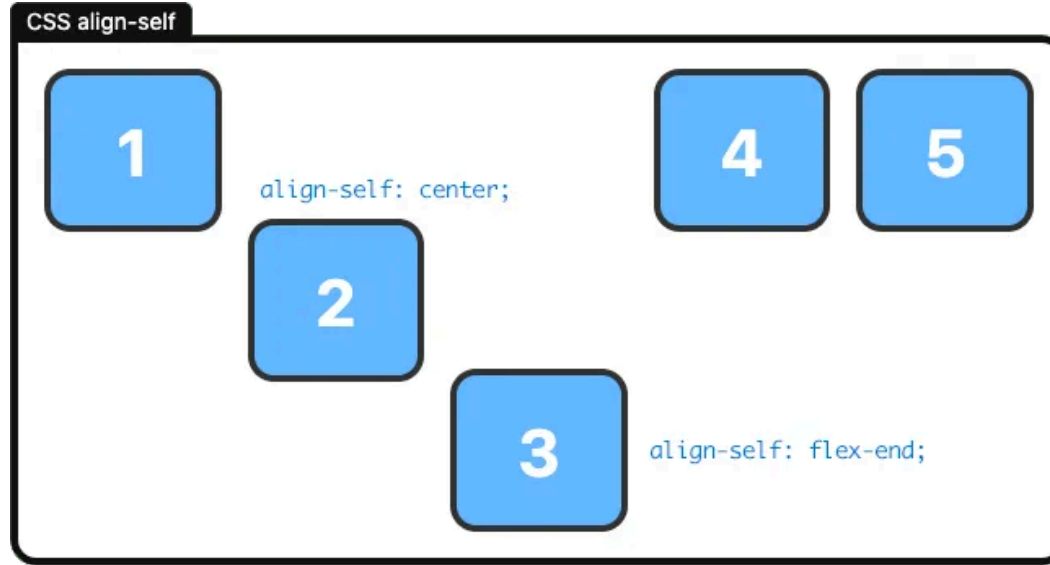
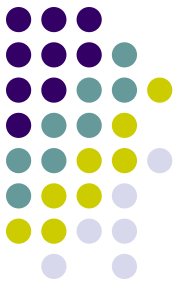


- Sırasıyla `flex-grow`, `flex-shrink` ve `flex-basis` özelliklerinin kısa kullanımımıdır. İlk değeri zorunlu olup (`shrink` ve `basis`) değerlerini belirtmek zorunlu değildir. Varsayılan olarak `0 1 auto` değerine sahiptir. Ve en sık kullanımını da şöyledir;

```
.item1 {  
  flex: 1 1 auto;  
}
```


Flexbox Yapısı

Öğeler (items) Özellikleri - align-self



- `align-content` ile aynı işlemleri yapar. Dikey ekseninde öğelerin nasıl hizalanacağı belirlenir. Tek farkı kapsayıcı yerine bunu öğelere uygulamasıdır.

```
.item {  
  align-self: auto | flex-start | flex-end | center | baseline | stretch;  
}
```