

BLM3580

Sistem Programlama

2020-2021 GÜZ YARIYILI (UZAKTAN EĞİTİM)

DR.ÖĞR.ÜYESİ GÖKSEL BIRICIK

Web Programlama

HTML, CSS, SCRIPTING

SGML - Standard Generalized Markup Language

Tanımlayıcı işaretleme dilinde uyulması gereken standartlar bütünüdür. Elektronik verinin gösteriminde kullanılan cihaz ve sistemden bağımsız bir yöntemi tanımlar.

- Verinin belge içinde nasıl sunulacağından (presentation) ziyade nerede olacağını yani yapısal düzeni (structural organization) belirler.
- Verinin nasıl görüntüleneceğinden web gezginleri, kelime işlemcileri (word processor) veya metin düzenleyicileri (text editor) sorumludur.

Etiketleme dili; tanımlayıcı (declarative) olmalı, belgenin yapısını ve içinde yer alacak olan öznitelikleri (attributes) belirlemek için kullanılmalı, belge üstünde gerçekleştirilecek olan işlemlere yönelik bir tarif barındırmamalı, gelecekte oluşabilecek, önceden öngörülememiş işlemler ve teknikler ile çatışma (karışma) olasılığının düşük olmasına imkân sağlamalıdır.

DTD (Document Type Definition), etiketleme dilinde ifade edilen verinin doğru bir şekilde hazırlanıp hazırlanmadığını kontrol etmek üzere geliştirilmiş bir kontrol şablonudur.

Web'in Gelişimi

Web 1.0: Salt Oku (Statik)

- Kullanıcılar sadece kendilerine sunulanlar içinden arama yaparak bulabildikleri içeriği okuyabilirler. Etkileşim yoktur, kullanıcılar içerik üretmez.

Web 2.0: Oku-Yaz-Yayınla

- Kullanıcı ile web arasındaki etkileşim Facebook, Twitter, Flickr, YouTube ve Instagram gibi sosyal içerik paylaşım siteleri ile gündeme gelmiştir. Böylece web içeriği kullanıcılar tarafından da üretilebilir.

Web 3.0: Oku-Yaz-Çalıştır

- Web 2.0 ile oluşan bilginin semantic markup ile anlamlandırılması, bilgisayarlarca işlenerek (web service) uygulamaların doğrudan kendi aralarında konuşabilmesi hedefleniyor.

HTML HyperText Markup Language

Bilgilerin web sayfaları şeklinde düzenlenmesini sağlamak üzere geliştirilmiş özel amaçlı bir etiketleme (tag) dilidir. ASCII düz metin olarak yazılır.

- HTML5 W3C tarafından Ekim 2014'de geçerli bir standart olarak onaylanmıştır. Yeni özellikler eski gezginlerde zarifçe azaltılır (bilmiyorsa anonim eleman olarak algılar, istersek stil tanımlayabiliriz.)

HTML dili, etiket (tag) olarak adlandırılan < ve > işaretleri arasında yazılan görevi tanımlanmış komutlardan oluşur.

- Etiketler kapanmalıdır, ör: <p>...</p> ya da

- Etiketlerin varsa öznitelikleri, öznitelik_ismi="değer" biçiminde verilmelidir.
- Etiketler ağaç yapısında (nested) açılıp kapanmalıdır.
- Etiket isimlerini küçük vermek doğru olacaktır (DOM model)

HTML Belge Yapısı

`<html>` ve `</html>` etiketleri arasında kalan bölüm bir HTML belgesi olarak tanımlanır.

`<head>` `</head>` etiketleri arasında belge içinde bulunabilecek bazı özel tanımlamalar yer alabilir.

- `<meta>` `</meta>`, `<link>` `</link>`, `<title>` `</title>`, ...
- Sayfa ile birlikte çalışması düşünülen betik dilleri kullanılarak kodlanmış işlevler (`<script>` `</script>`)
- Stil ile ilgili tanımlamalar (`<style>` `</style>`)

`<body>` `</body>` etiketleri arasında ise sayfanın içinde yer alacak bilgi ve bunlar ile ilgili düzenlemeleri sağlayacak olan diğer etiketler ile görselliğe yönelik işlevler yer alır.

HTML 5 önceki veriyonlardan çok farklı değildir, bazı tanımlamaları sadeleştirmek ve ilerideki anlambilimsel (semantic) web çalışmalarına uygun şekilde verinin ifade edilebilmesini kolaylaştıracak olan iyileştirmeleri içerir.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset=UTF-8" />
```

```
<title> HTML Belgesinin iskelet yapısı </title>
```

```
</head>
```

```
<body>
```

```
Gövde içinde yer alacak olan bilgiler
```

```
</body>
```

```
</html>
```

HTML 4.x vs HTML5

```
<!DOCTYPE html PUBLIC "-//W3C//DTD  
XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
strict.dtd">
```

```
<html>
```

```
...
```

```
<meta http-equiv="Content-Type"  
content="text/html; charset=UTF-8" />
```

```
<!DOCTYPE html>
```

```
<html lang="tr">
```

```
...
```

```
<meta charset="UTF-8" />
```

```
...
```

HTML Etiketleri

HTML 5 ile birlikte kullanılmaya başlayan bazı yeni etiketler olduğu gibi daha önceki HTML versiyonlarında kullanılmakta olan etiketlerden bir kısmı,

- geçerliklerini ya tamamen yitirmiş (<frame>, <frameset> ve <noframe> gibi)
- yerlerini başka etiketlere bırakmış (<acronym> yerini <abbr>, <applet> yerini <object> ve <dir> yerini gibi)
- veri ifade etmekten ziyade görselleştirme ile ilgili oldukları için görevlerini CSS'e (<basefont>, <big>, <center> ve <strike> gibi) bırakmışlardır.

HTML 5 ile kullanılmaya başlayan etiketleri,

- anlambilimsel (semantic) elemanlar (<header>, <footer>, <article> ve <section> gibi)
- yeni form kontrol öznitelikleri (number, date, time, calendar ve range gibi)
- yeni grafik tasarım etiketleri (<svg> ve <canvas> gibi)
- çoklu ortam (multimedia) etiketleri (<audio> ve <video> gibi) olarak özetlemek mümkündür.

HTML5 Yapısal Elemanlar

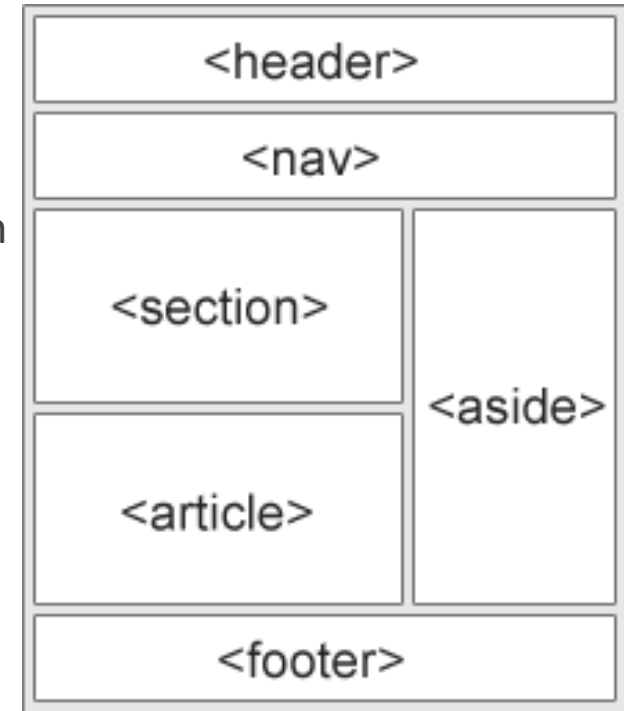
daha önceleri sayfada class ve id öznitelikleri ile belirlenen, amaca yönelik anlambilimsel parçaların farklı platform, cihaz ve ortamda çalışacak şekilde (media-independent) yerini almışlardır.

Bir belgenin esas içeriği **<main>** **</main>** etiketi ile belirlenir.

- Bu içerik bir şekilde belge içinde başka yerlerde tekrarlanmamalıdır.
- Bu etiket **<article>**, **<aside>**, **<footer>**, **<header>** veya **<nav>** etiketlerinin alt elemanı olarak da kullanılamaz.

Başlıklar **<header>** **</header>** etiketi ile belirlenir.

- Bu etiket kendi içinde **<h1>**–**<h6>** ve **<hgroup>** elemanlarını da barındırabildiği gibi yönlendirme (**<nav>**), içindekiler tablosu, logo, arama için form, alt bilgi (**<footer>**) hatta başka bir başlık (**<header>**) da içerebilir.



HTML5 Yapısal Elemanlar

Bir bölüm (section) içindeki, gerek aynı sayfa içindeki (baş, sona, içindekiler kısmına vb.) bir noktaya gerek ise farklı sayfalarla bağlantıları sağlamak için **<nav> </nav>** kullanılır.

- Ancak her türlü yönlendirmenin de bu etiket içinde yapılmaya çalışılması doğru bir kullanım şekli değildir.

İçeriğin bir temaya bağlı olarak mantıksal olarak düzenlenmesi amacıyla **<section> </section>** kullanılır.

- İçinde **<h# > </h#>** ve **<article> </article>** etiketlerini de barındırabilir.

Sayfanın tamamından bağımsız olarak okunabilecek olan içeriği belirlemek için **<article> </article>** kullanılır.

Bir başlığı onun altında yer alacak olan alt-başlıklar ile ilişkilendirmek için **<hgroup> </hgroup>** kullanılır.

- Kendi içinde **<h1>—<h6>** elemanlarını da barındırır.

Kullandığımız her türlü resim, tablo vb. görsel materyale bir isim ve numara vermek için **<figure> </figure>**, içinde ** ** ile resim/tablo belirtilirken ile **<figcaption> </figcaption>** de o görsel materyale verilecek etiket ifade edilir.

HTML5 Yapısal Elemanlar

Çevrelediği içerik ile ilgili ancak ondan ayrı olarak (ek olarak) gösterilmesi gereken bir içeriği belirlemek için **<aside> </aside>** kullanılır. İlgili kısmın altında yer alacaktır.

Oluşturulmuş olan belgenin kim tarafından hazırlandığı, ne zaman güncellendiği, telif hakları ve sayfa başına atlamayı sağlayacak olan bağlantıları içere kısımlar **<footer> </footer>** ile tanımlanır.

Kullanıcının isteğine bağlı olarak, gösterilecek veya saklanacak olan ayrıntıları belirlemek için **<details> </details>** etiketi kullanılır.

- Sayfa yüklenirken görünsün isteniyorsa open özniteliği kullanılmalıdır.

<summary> </summary> etiketi **<details> </details>** etiketi içinde kullanıldığında içeriği ekranda görünecek ve üzerine tıklandığında açılacak/kapanacaktır.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title> HTML 5 Anlambilimsel Etiketler</title>
</head>
<body>
  <main>
    <nav>
      <a href="#KA">Kara Araçları</a> | <a href="#HA">Hava Araçları</a> | <a href="#DA">Deniz
araçları</a>
    </nav>
    <h1>Araçlar</h1>
    <p>Kara, hava ve deniz araçlari olarak sınıflandırılırlar.</p>
    <article>
      <h1 id="KA">Kara Araçları</h1>
      <p>Karayollarını kullanan araçlar, yola temas eden tekerlerin yapısına göre; Lastik tekerlekli
veya paletli olabilirler</p>
      <figure>
        <p><figcaption>Resim 1 - Otomobil</figcaption></p>
      </figure>
      <details>
        <summary>Lastik Tekerlekli Kara Araçları</summary>
        <p>Bisiklet, motorsiklet, otomobil, minibüs, otobüs gibi insan taşıma, kamyonet, kamyon,
çekici gibi yük taşımada kullanıldıkları gibi traktör, kepçe vb. iş makineleri de olabilir.</p>
      </details>
      <details>
        <summary>Paletli Kara Araçları</summary>
        <p>Karayolları gibi düzgün satıhlardadışındaki arazide hizmet vermek için tasarlanmış kepçe,
grayder, ekskavatör gibi iş makineleri, askeri amaçlı tank ve personel taşıyıcılar ile kar üzerinde
hareket eden motorsikletlerdir.</p>
      </details>
    </article>
```

```
<article>
  <h1 id="HA">Hava Araçları</h1>
  <p>Planör gibi motorsuz, iş jeti, savaş, yolcu, kargo uçağı vb. motorlu, kalkış/iniş için uzun pist
gerektiren hava araçları yanında helikopter gibi dikine kalkış/iniş yapabilen araçlar.</p>
  <figure>
    <p><figcaption>Resim 2 - İş Jeti</figcaption></p>
  </figure>
</article>
<article>
  <h1 id="DA">Deniz Araçlar</h1>
  <p>Suyun kaldırma gücünden yararlanarak yolcu ve yük taşımada kullanılan sal, sandal, yat, yelkenli,
şilep, yolcu gemisi vb. su üzerinde giden araçlardır.</p>
  <figure>
    <p><figcaption>Resim 3 - Yolcu Gemisi</figcaption></p>
  </figure>
</article>
<footer>(c) 2019 - BLM3580 Sistem Programlama</footer>
</main>
</body>
</html>
```

HTML5 Yapısal Elemanlar

Bir belge içindeki bir kısmın sanki fosforlu kalem ile çizilmiş gibi gösterilmesi için **<mark>** **</mark>** etiketinden yararlanılır. CSS ile renkleri değişebilir.

<time> **</time>** etiketi ile gün/saat bilgisini hem kullanıcı hem de makine tarafından kullanılan gün/saat formatına göre ifade edilebilir.

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklama	HTML3.x	HTML4.x	HTML 5.x
<code> </code>	Bold: Yazıyı koyu yazar. HTML5 de başka çare olmadığı sürece kullanılmamalıdır. Vurgulama için <code></code> , önemli metinler için <code></code> ve fosforlu kalem ile çizilmiş gibi göstermek için <code><mark></code> etiketi kullanılmalıdır.	✓	✓	✗
<code><basefont color="*" size="?" face="*"></code>	Base Font: <code><head></code> elemanın içinde kullanılan bu eleman ile belge içinde kullanılacak olan font büyüklüğü, rengi ve font ailesi belirlenir Bu işlem artık CSS ile yapıldığı için yeni nesil web gezginlerinde desteklenmez.	✓	✓	✗
<code><bdi> </bdi></code>	Bi-directional Isolation: Sağdan sola doğru yazılan diller ile soldan sağa doğru yazılan dillerin bir arada kullanılması durumundaki düzeni sağlamak için kullanılır	✗	✗	✓
<code><bdo dir="{rtl ltr}"> </bdo></code>	Bi-Directional Override: Yazının yönünü (dir) sağdan-sola (rtl) veya soldan-sağa (ltr) olacak şekilde belirler	✗	✓	✓
<code><big> </big></code>	Big: Yazının büyük yazılmasını sağlar	✓	✗	✗

Gösterim/Düzenleme Etiketleri

<code><blockquote> </blockquote></code>	Block Quote: Yapılan alıntılarını belirlemek için kullanılır. Genel olarak web gezginleri bu eleman içindeki metni bir miktar sağa kayık olarak gösterir.	✓	✓	✓
<code>
</code>	Break: Satır başı	✓	✓	✓
<code><caption> </caption></code>	Caption: <code><table></code> elemanı içinde kullanıldığında tanımlı olduğu tablonun numarası ve açıklamasını belirler. Eleman içindeki yazı tablonun konumuna göre ortalananır .	✓	✓	✓
<code><caption align="{left right top bottom}"> </caption></code>	Caption: Tablo, şekil numara ve isimlerini vermek için kullanılır. align özneliği hizalamayı belirler	✓	✓	✗
<code><center> </center></code>	Center: Yazıyı ortalayarak yazar	✓	✓	✗
<code><cite> </cite></code>	Citation: Kitap, resim vb. bir eklentinin alt yazısıdır.	✓	✓	✓
<code><code> </code></code>	Code: Bilgisayar kodunu belirler.	✓	✓	✓

Gösterim/Düzenleme Etiketleri

<code><dfn title="** id="**"> </dfn></code>	Define: HTML içinde bir terimin açıklanması için kullanılır . Genel olarak eğik yazı olarak görülür. title özniteliği kullanılırsa imlecin üzerine geldiğinde değer olarak verilen açıklama görünür. id özniteliği kullanıldığında <code> </code> kullanımı ile başka bir noktadan erişilmesi de mümkündür.	✓	✓	✓
<code><dir compact> .. </dir></code>	Directory List: Listelenmiş elemanları belirler. compact özniteliği kullanıldığında daha küçük görünmesi beklenir, ancak pek web gezgini bunu desteklemez.	✓	✓	✗
<code><dd> </dd></code>	Describe each a term: Her terime ait bir açıklama/tanım yapmayı sağlar.		✓	✓
<code> </code>	Delete: İçeriğin üzerinde bir çizgi çizerek silinmiş olduğunu belirler.	✓	✓	✓
<code><dl> </dl></code>	Description list: Tanım/açıklama listesi oluşturur.		✓	✓
<code><dt> </dt></code>	Define a term: Tanım listesi içinde bir terim oluşturur		✓	✓
<code> </code>	Emphasize: Yazıyı eğik yazar. <code><i> </i></code> yerine kullanılır.	✓	✓	✓

Gösterim/Düzenleme Etiketleri

<code></code> <code></code>	Font: Yazıyı "?" ile belirtilen büyüklükte yazar. Yerine CSS kullanılması önerilir.	✓	✗	✗
<code><h?> </h?></code>	Heading: ? İşareti yerine 1-6 arasında bir rakam yazılabilir. Başlıklar arasındaki alt-üst ilişkisini belirler. align özneliği başlığın hizalamasını belirler.	✓	✓	✓
<code><h? align="{left center right}"> </h?></code>		✓	✓	✗
<code><hr /></code>	Horizontal Rule : Yatay çizgi çizer. HTML 5.x de temanın bitimini belirler.	✓	✓	✓
<code><hr align="{left center right}" size="?" width="{% ??}"></code> <code></hr></code>	align özneliği hizalamayı , size özneliği kalınlığı ve width özneliği çizginin yatayda kaplayacağı alanı belirler.	✓	✓	✗
<code><i> </i></code>	Italic: Eğik yazar. HTML5 de başka çare olmadığı sürece kullanılmamalıdır. Vurgulama için , önemli metinler için , fosforlu kalem ile çizilmiş gibi göstermek için <mark>, atıflar için <cite> ve tanımlamalar için <dfn> kullanılmalıdır.	✓	✓	✗

Gösterim/Düzenleme Etiketleri

<code><ins> </ins></code>	Insert: Eklenen yazının altına bir çizgi çizer.	✓	✓	✓
<code><nobr> </nobr></code>	No Break: Yazıyı bölmeden ,tek satıra yazar.	✓	✗	✗
<code><p></code>	Paragraph: Bir satır boş bırakıp satır başına geçirir. align özneliği hizalamayı belirlemek ile birlikte bunun için CSS kullanılması önerilir.	✓	✓	✓
<code><p align="{left center right justify}"> </p></code>		✓	✗	✗
<code><pre> </pre></code>	Pre-formatted: Metnin kullanıcının yazdığı şekilde görünmesini sağlar		✓	✓
<code><s> </s></code>	Strikethrough text: Hatalı olan, geçerliliğini yitirmiş içeriği belirtir. HTML 4.x'de kullanımdan kalkmış ancak HTML 5.x de hatalı içeriği belirtmek için yeniden kullanılmaya başlanmıştır.	✓	✗	✓

Gösterim/Düzenleme Etiketleri

<code> </code>	Strong: Koyu yazar. yerine kullanılır.	✓	✓	✓
<code><sub> </sub></code>	Subscript : Alt indis belirtir.	✓	✓	✓
<code><sup> </sup></code>	Superscript: Üst indis belirtir.	✓	✓	✓
<code><tt> </tt></code>	Typewriter: Sabit aralıklı yazdırır.	✓	✗	✗
<code><u> </u></code>	Underline: Altı çizgili yazar. HTML 4.x de kullanımı kalkmış lmasına rağmen HTML 5.x de yine normalden farklı olarak hatalı yazımları vurgulamak için kullanılır.	✓	✗	✓
<code><wbr> </wbr></code>	Word Break Opportunity: Kelimenin çok uzun olduğu durumlarda web gezgininin bunu uygunsuz bir şekilde bölmesini engeller.	✗	✗	✓

HTML'de Kullanılan Özel Karakterler

Karakter	Açıklama	ISO 8859-9	
		Kullanım -1	Kullanım -2
"	Çift tırnak	"e;	"
&	Ampersand	&	&
'	Tek tırnak (apostrof)		'
<	Küçük	<	<
>	Büyük	>	>
	Boşluk	 	
£	Pound	£	£
©	Copyright	©	©
®	Registred Trademark	®	®
€	Euro	€	

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklaması	HTML3.x	HTML4.x	HTML 5.x
<code><body</code> <code>alink="{color_name hex_number rgb_number}"</code> <code>background="URL"</code> <code>bgcolor="{color_name hex_number rgb_number}"</code> <code>link="{color_name hex_number rgb_number}"</code> <code>text="{color_name hex_number rgb_number}"</code> <code>vlink="{color_name hex_number rgb_number}"</code> <code></body></code>	Body: alink: aktif bağlantı rengi background: arkaplanda yer alması istenen resmin URL'si bgcolor: arkaplan rengini belirler. color_name : rengin ismi "red", "green", "blue" hex_number : rengin kodu "#ff0000", "#00ff00", "#0000ff" rgb_number: RGB olarak "rgb(255,0,0)", "rgb(0,255,0)", "rgb(0,0,255)" olarak verilebilir. link: mevcut bağlantıların rengi vlinks: ziyaret edilmiş bağlantıların rengi text: yazı rengi	✓	✓	✗
<code><body></code> <code></body></code>	HTML5 de öznitelikler desteklenmemektedir.	✓	✓	✓

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklaması	HTML3.x	HTML4.x	HTML 5.x
<pre> </pre>	<p>Anchor:</p> <p>charset: Yönlendirilecek olan sayfanın karakter kodlamasının ne olduğunu belirler.</p> <p>coords: Özellikle sensitive map yapısında alan belirlemek için kullanılır.</p> <p>href bağlantıya ait URL'yi gösterir</p> <p>hreflang: Yönlendirilen sayfanın dilidir.</p> <p>name: belge içindeki bir bölüme (section) bağlantı gerçekleştirebilmek için kullanılır.</p> <p>rel : Bu öznitelik web gezginleri tarafından kullanılmaz. Arama motorları ise ilgili bağlantının mevcut belge ile olan bağıntı anlamlandırmak için kullanılmaktadır.</p> <p>rev: rel'in tam tersidir. Yani mevcut sayfanın yönlendirilen ile bağıntı anlamlandırır.</p> <p>shape: sensitive map yapısında yönlendirme yapmak için tıklanacak olan bölgeyi belirlemek için kullanılır.</p> <p>target: yönlendirilen belgenin nerede açılacağını belirlemede kullanılır.</p> <p>type: yönlendirilecek belgenin medya tipini belirler. audio/mp3, application/pdf, application/zip, image/jpeg, image/png, message/http, multipart/form-data, multipart/mixed, text/html, text/javascript, video/avi vb.</p>	✓	✓	✗

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklaması	HTML3.x	HTML4.x	HTML 5.x
 	Anchor: download: Bağlantının kayıt edilmesini sağlar her gezgin tarafından desteklenmez. href: http://, ftp://, gopher:// gibi bir servis bağlantısı için kullanılabileceği gibi mailto:gbiricik@yildiz.edu.tr?Subject=test şeklinde bir e-posta bağlantısı da olabilir. media: Bağlı olan belgenin ne tür medya/cihaz için optimize edildiğine bağlı olarak bazı özelliklerin tanımlanmasına izin verir.	x	x	✓

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklama	HTML 3.x	HTML 4.x	HTML 5.x
<code></code> <code></code>	<p>Image: Belge içinde yer alacak olan resim türündeki bilgileri belirlemek için kullanılır.</p> <p>alt: Resimin gösterilmesinin mümkün olmadığı durumlarda onun yerine ve/veya resim üzerine imleç geldiğinde gösterilecek olan alternatif ifadedir.</p> <p>crossorigin : canvas içinde, üçüncü taraf kabul edilen resimlerin kullanımına yönelik bir özelliktir.</p> <p>height: resimin yüksekliğini pixel cinsinden belirler</p> <p>ismap: sunucu taraflı bir sensitive map olması durumunda kullanılmaktadır. Buna göre sunucu tarafına tıklana yerin koordinatları yollanır.</p> <p>src: resimin URL'si</p> <p>usemap: istemci taraflı bir sensitive map olması durumunda kullanılır.</p> <p>width: resimin pixel cinsinden genişliğini belirler.</p>			✓
<code></code> <code></code>	<p>align: çevresindeki diğer elemanlara bağlı olarak konumunu belirler</p> <p>border: resim etrafında bir çerçeve oluşturur</p> <p>hspace: resimin sağ ve sol yanındaki boşlukları pixel cinsinden belirler</p> <p>longdesc: çoğu web gezgini tarafından desteklenmez. Ancak resim ile ilgili uzun açıklananın yer aldığı belgeye ait URL'dir.</p> <p>vspace: resimin üst ve altındaki boşlukların pixel cinsinden değeridir.</p>		✓	

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklama	HTML 3.x	HTML 4.x	HTML 5.x
<code><table sortable> </table></code>	Tablo: sortable: tablonun sıralanabileceğini belirler.	✓	✓	✓
<code><table align="{left center right}" bgcolor="{ rgb(x,x,x) #xxxxxx colorname}" border="{ 1 0}" cellpadding="??" cellspacing="??" frame="{ void above below hsides lhs rhs vsides box border }" rules="{ none groups rows cols all}" summary="**" width="{?? % }"> </table></code>	align: hizalamayı belirler bgcolor: arka plan rengini belirler. border: tablonun çerçevesini belirler. cellpadding: hücre içeriğinin tablo kenarından olan uzaklığını belirler. cellspacing: tablo hücreleri arasındaki boşluğu belirler. frame: tablo hücreleri arasında hangi çizgilerin görüneceğini belirler. rules: tablo içindeki bölme çizgilerinin ne şekilde kullanılacağını belirler. summary: tablonun içeriğine yönelik özetler. width: tablo genişliğini belirler.	✓	✓	✗

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklaması	HTML3 .x	HTML4 .x	HTML 5.x
<thead> </thead>	Table Head:	✓	✓	✓
<thead align="{ left right center justify char}" char = "***" charoff="?" valign="{top middle bottom baseline}" > </thead>	Table Head: align: dikey hizalama için kullanılır. char ve charoff çoğu web gezgini tarafından desteklenmez. Ancak amaç içeriğin hizalanmasıdır valign: içeriğinin yatay daki hizasını belirler.	✓	✓	x
<tbody> </tbody>	Table Body:	✓	✓	✓
<tbody align="{ left right center justify char}" char = "***" charoff="?" valign="{top middle bottom baseline}" > </tbody>	Table Body: align: dikey hizalama için kullanılır. char ve charoff çoğu web gezgini tarafından desteklenmez. Ancak amaç içeriğin hizalanmasıdır valign: içeriğinin yatay daki hizasını belirler.	✓	✓	x
<tfoot> </tfoot>	Table Foot:	✓	✓	✓
<tfoot align="{ left right center justify char}" char = "***" charoff="?" valign="{top middle bottom baseline}" > </tfoot>	Table Foot: align: dikey hizalama için kullanılır. char ve charoff çoğu web gezgini tarafından desteklenmez. Ancak amaç içeriğin hizalanmasıdır valign: içeriğinin yatay daki hizasını belirler.	✓	✓	x

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklaması	HTML3.x	HTML4.x	HTML 5.x
<tr> </tr>	Table Row:	✓	✓	✓
<tr align="{ right left center justify char}" bgcolor="{ rgb(x,x,x) #xxxxxx colorname}" char="*" charoff="?" valign="{ top middle bottom baseline}" </tr>	align: dikey hizamaları belirler. bgcolor: arka plan rengini belirler. char ve charoff çoğu web gezgini tarafından desteklenmez valign: yatay hizalamayı belirler.	✓	✓	✗

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklaması	HTML 3.x	HTML 4.x	HTML 5.x
<code><th abbr="**" colspan="?" headers="header_id" rowspan="?" scope="{col colgroup row rowgroup}" sorted="{reversed number reversed number number reversed}" > </th></code>	Table Header: Tablo sütunları üzerindeki başlık alanlarını belirlemek için kullanılır. abbr: Başlık için kullanılanın kısaltılmış halini tutumak için kullanılır. colspan: başlığın kaç sütüne yayılacağını belirler. headers: id özneliği kullanılmış olan <th> elamanları ile ilişkilendirmek için kullanılır. scope: başlık hücresinin ne için tanımlandığını belirler. sorted: sıralama şeklini belirler.	✓	✓	✓
<code><th align="{ left right center justify char}" axis="category_name" bgcolor="{ rgb(x,x,x) #xxxxxx colorname}" char = "***" charoff="?" headers="header_id" height="{pixels %}" nowrap rowspan="?" scope="{col colgroup row rowgroup}" sorted="{reversed number reversed number number reversed}" valign="{top middle bottom baseline}" width="{pixels %}" > </th></code>	Table Header: align: dikey hizalama için kullanılır. axis: tablo başlıklarını kategorilere ayırmak için kullanılır. Görsel bir etkisi yoktur. bgcolor: arka plan rengini belirler char ve charoff çoğu web gezgini tarafından desteklenmez. Ancak amaç içeriğin hizalanmasıdır. height: hücre yüksekliği belirler. nowrap: başlık satırının içeriğinin alt satıra kaymasını engeller. rowspan: başlık içeriğinin kaç satıra yayılacağını belirler. sorted: sırlamanın şeklini belirler. valign: başlık içeriğinin yatay daki hizasını belirler. width: başlık hücresinin genişliğini belirler.	✓	✓	✗

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklaması	HTM L3.x	HTM L4.x	HTM L5.x
<code><td colspan="??" headers="header_id" ></code> <code></td></code>	Table Data:	✓	✓	✓
<code><td</code> <code>abbr="**"</code> <code>align="{ left right center justify char}"</code> <code>bgcolor=" { rgb(x,x,x) #xxxxxx colorname}"</code> <code>char = "**"</code> <code>charoff="?"</code> <code>height="{pixels %}"</code> <code>nowrap</code> <code>rowspan="??"</code> <code>scope="{col colgroup row rowgroup}"</code> <code>valign="{top middle bottom baseline}"</code> <code>width="{pixels %}" ></code> <code></td></code>	abbr: hücre içeriğinin kısaltılmış halidir. align: dikey hizalama için kullanılır. bgcolor: arka plan rengini belirler char ve charoff çoğu web gezgini tarafından desteklenmez. Ancak amaç içeriğin hizalanmasıdır. height: hücre yüksekliği belirler. nowrap: hücre içeriğinin alt satıra kaymasını engeller. rowspan: içeriğin kaç satıra yayılacağını belirler. sorted: sırlamanın şeklini belirler. valign: içeriğinin yatay daki hizasını belirler. width: hücrenin genişliğini belirler.	✓	✓	✗

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklaması	HTM L3.x	HTM L4.x	HTM L5.x
<code><col span="?"></code> <code><col></code>	<code><col></code> etiketi bir tablodaki bir sütuna stil uygulamak istendiğinde kullanılır. Böylece her satırda yer alan hücelere (<code><td></code>) ayrı ayrı stil uygulamak gerekmediği için tanım işlemini kolaylaştırır. span: <code><col></code> elemanının tabloda kaç sütuna yayılmış şekilde göstereileceğini belirler.	✓	✓	✓
<code><col</code> align="{left right center justify}" char="*" charoff="?" valign="{ top middle bottom baseline}" width="{pixels % relative_length}" <code></col></code>	align, char, charoff, valign ve width öznitelikleri HTML5 tarafından desteklenmez. Bunlara görselliğe yönelik oldukları için CSS yardımı ile düzenlenmelidir.char ve charoff çoğu web gezini tarafından desteklenmez.	✓	✓	✗
<code><colgroup span="?"></code> <code></colgroup></code>	<code><colgroup></code> etiketi tablo (<code><table></code>) yapısı içinde bir veya birden fazla kolon (<code><col></code>) tanımını içerir. Genelde tabloda her satırdaki (<code><tr></code>) hücelere (<code><td></code>) aynı stilin verilmesinin gerektiği durumlarda tanımlamayı sadeleştirmek için kullanılır. Yapısı gereği <code><table></code> elemanı bir alt elemanı olarak <code><caption></code> elemanından sonra, <code><thead></code> , <code><tbody></code> , <code><tfoot></code> ve <code><tr></code> elemanlarından önce kullanılır.	✓	✓	✓
<code><colgroup</code> align="{left right center justify}" char="*" charoff="?" valign="{ top middle bottom baseline}" width="{pixels % relative_length}" <code></colgroup></code>	align, char, charoff, valign , width öznitelikleri HTML tarafından desteklenmez. Bunlara görselliğe yönelik oldukları için CSS yardımı ile düzenlenmelidir.	✓	✓	✗

Tablo Kullanım Örneği

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>HTML Tablo Kullanımı-3</title>
</head>
<body>
  <table border="1" cellpadding="10">
    <tr> <th rowspan="2">No</th> <th colspan="2">Öğrencinin</th> </tr>
    <tr> <th>Adı</th> <th>Soyadı</th> </tr>
    <tr> <td>12011014</td> <td>Mehmet</td><td>Çalışkan</td> </tr>
    <tr> <td>11011021</td> <td>Ayşe</td> <td>Üstün</td> </tr>
  </table>
</body>
</html>
```

No	Öğrencinin	
	Adı	Soyadı
12011014	Mehmet	Çalışkan
11011021	Ayşe	Üstün

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklaması	HTML 3.x	HTML 4.x	HTML 5.x
<code><iframe</code> <code>height = "??"</code> <code>name = "***"</code> <code>sandbox = "{allow-forms </code> <code>allow-pointer-lock </code> <code>allow-popups </code> <code>allow-same-origin </code> <code>allow-scripts </code> <code>allow-top-navigation}"</code> <code>seamless</code> <code>src = "URL"</code> <code>srcdoc = "HTML_code"</code> <code>width = "??"></code> <code></iframe></code>	<p>Inline Frame</p> <p>height: çerçevenin yüksekliğini belirler</p> <p>name: <iframe>' e bir isim vermek için kullanılır. Bu isim elemana javascript, <a> veya <form> elemanları ile <input> veya <button> elemanları ile yapılan erişimlerde kullanılır.</p> <p>sandbox: <iframe> içinde ek sınırlamalar getirmek için kullanılır. İçeriğinin tek bir kaynaktan sağlanması, form yollarının engellenmesi, scriptlerin çalıştırılmasının engellenmesi, API'lerin kullanımının engellenmesi, içerikte plugin (<embed>, <object>, <applet> vb.) kullanımının engellenmesi, içeriğin daha üst düzeydeki bağlantıya yönlendirilmesinin engellenmesi, otomatik olarak tetiklenen özelliklerin(bir videonun otomatik olarak başlaması vb.) devre dışı bırakılması vb. Özniteliğe herhangi bir değer verilmez ise tüm sınırlamalar devrededir. Eger seçilmiş sınırlamalar kullanılacak ise boşluk ile ayrılmış olarak yazılır.</p> <p>seamless: her web gezgini tarafından desteklenmek ile birlikte<iframe> içeriğinin mevcut sayfanın bir parçası gibi görünmesi sağlanır.</p> <p>src: <iframe> içinde gösterilecek olan URL'i belirler.</p> <p>srcdoc: <iframe> içinde yer alanacak olan HTML kodudur.</p> <p>width: <iframe> genişliğini belirler.</p>	x	x	✓
<code><iframe</code> <code>align = "{left right top middle bottom}"</code> <code>border = "{1 0}"</code> <code>longdesc = "URL"</code> <code>marginheight = "??"</code> <code>marginwidth = "??"</code> <code>scrolling = "{yes no auto}"></code> <code></iframe></code>	<p>align: dikeydeki hizamayı sağlar</p> <p>border: çerçeve kullanılıp kullanılmayacağını belirler.</p> <p>longdesc: <iframe> için bir açıklamadır. Ancak kullanımı kullanışlı olmadığı için kullanımı önerilmez.</p> <p>marginheight: içeriğin <iframe> çerçesinden olan (dikey) uzaklığını belirler.</p> <p>marginwidth: içeriğin sağ ve soldan olan uzaklığını belirler.</p> <p>scrolling: kaydırma çubuklarının görünüp görünmemesini belirler.</p>	✓	✓	x

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklaması	HTML3. x	HTML4. x	HTML 5.x
<code><dir> .. </dir></code>	HTML 5.x de bunun yerine <code></code> kullanılması önerilir.	✓	✓	✗
<code><ol start="??" type="{ 1 A a I i }"></code> <code></code>	Ordered List: start: başlangıç değeri type: kullanılacak listeleme tipi	✓	✓	✓
<code><ol compact></code> <code></code>	compact: kullanılacak listeleme elemanlarının normalden daha ufak olmasını sağlar. Ancak tüm web gezginlerinde desteklenmez	✓	✓	✗
<code><ol reversed> </code>	reversed: liste sırasının tersten başlamasını sağlar. Azalan şekilde ilerler.	✗	✗	✓
<code> </code>	Unordered List Stilleri vermek için CSS tanımlarından yararlanılır. <code><ul style="list-style-type:square"></code>	✗	✗	✓
<code><ul compact type="{ disc square circle }"></code> <code></code>	compact: kullanılacak listeleme elemanlarının normalden daha ufak olmasını sağlar. Ancak tüm web gezginlerinde desteklenmez type: kullanılacak listeleme tipi	✓	✓	✗
<code><li value="??"></code> <code></code>	List Item: type: kullanılacak listeleme tipi value: değeri	✓	✓	✓
<code><li type="{ 1 A a I i disc square circle }"></code> <code></code>	List Item: type: kullanılacak listeleme tipi value: <code></code> ile kullanıldığında değer ilk <code><li value="5"></code> şeklinde verilmiş ise takip edenlerde value kullanılmadan 6, 7 şeklinde ilerler.	✓	✓	✗

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklama	HTML3.x	HTML4.x	HTML5.x
<code><form action="**" method="{post get}" id="**" autocomplete="on" {novalidate} > </form></code>	Genel form tanımı method yazılmaz ise get kabul edecek ve submit butonuna basılması ile form içinde yer alan input alanları name=value ikilileri olarak sunucu tarafında formu işlemek üzere yazılmış olan ve adı action ile belirlenen koda aktarılır. Form içinde birden fazla input alanı olması durumunda aktarılabacak veri arasında & işareti bulunacaktır.			
<code><form enctype="multipart/form-data"> </form></code>	Dosya yükleme tanımı			
<code><input type="**" tabindex="?"></code>	Genel giriş alanı tanımı			

Gösterim/Düzenleme Etiketleri

<code><input type="file"></code> <code><input type="file" name="*" {multiple}></code>	Dosya giriş alanı			
<code><input type="hidden" name="*" value = "*"></code>	Gizli giriş alanı tanımı			
<code><input type="text"</code> <code>name="*" value = "*"</code> <code>size="?"</code> <code>maxlength="?"</code> <code>{readonly disabled}</code> <code>{autofocus}</code> <code>{form="*" } {pattern=""}</code> <code>{placeholder="" } {required }></code>	Metin tipinde giriş alanında size verilmez ise 20 kabul eder. readonly seçildiğinde value ile belirlenen değer değiştirilemez. disabled verildiğinde giriş alanı üzerine tıklanamaz ve rengi soluk görünür. Form yüklendiğinde autofocus tanımı uyarınca imleç ilgili giriş alanında duracaktır. form özniteliği input elemanının herhangi bir form tanımının dışında olmak ile birlikte adı verilen formun bir parçası olarak davranacağını belirtmek için kullanılır. pattern ile regular expression tanımlanabilir. placeholder kullanılarak input alanına kullanıcıyı bilgilendiren, kullanıcı bu alana giriş yaparken silinen bir açıklama yazılır.			
<code><input type="radio" name="*" value="*" {checked}></code>	Yuvarlak buton şeklinde giriş alanı			

Gösterim/Düzenleme Etiketleri

<code><input type="checkbox" name="*" {checked}></code>	İşaretlenir kutu şeklinde giriş alanı			
<code><input type="password" name="*"></code>	Şifre tipinde giriş alanı, yazılan * veya ● simgeleri ile gizlenecektir.			
<code><input type="reset" name="*"></code>	Girilen alanlarını temizler			
<code><input type="submit" name="*"></code> <code><input type="submit" formaction="*" value="*" formenctype="multipart/form-data" formmethod="{ post get}" {formnovalidate} formtarget="_blank" }></code> <code><input type="image" src="*" alt="*" width="?" height="?"></code> <code><input list="*"></code>	Girilen bilgileri form elemanının action özniteliğinde adı verilen sunucu taraflı koda yollar. Name özniteliğine istenen şekilde yolla, ara, bul gibi formun kullanım amacına uygun olarak bir name verilmez ise SubmitQuery yazısı çıkar. formaction kullanılmış ise form elemanının action özniteliğinde hangi sunucu taraflı kodun verildiğine bakılmasızın formaction ile verilen sunucu taraflı koda aktarım yapılır. formenctype, sadece post methodu kullanılan formlar içinde yer alan type="submit" ve type="image" öznitelikli input elemanında kullanılır ve formda verilen formenctype özniteliğini yeniden tarifler. formmethod form elemanı içinde yer alan method değerini değiştirir. formtarget="_blank" ile sonucu yeni bir sayfaya yönlendirir. list="*" ile <datalist id="*"> elemanı ile eşleme sağlanır.			

Gösterim/Düzenleme Etiketleri

<code><select name="**" size="?" {mutiple}></code> <code></select></code>	Seçenekli giriş alanı			
<code><option value="**" {selected}></code>	Seçenekli giriş alanı seçeneklerini belirler			
<code><input name="**" list="**"></code> <code><datalist id="**"></code> <code><option value="**"></code> <code><option value="**"></code> <code><option value="**"></code> <code></datalist></code>	<p><code><datalist></code> etiketi ile bir <code><input></code> elemanı için seçeneklerin tanımlanmasına imkan verir. <code><input></code> etiketinin <code>list</code> özneliği <code><datalist></code> etiketinin <code>id</code> özneliği ile aynı olduğunda tanımlı yapılan <code><option></code> elemanlarının değerleri gösterilecektir.</p> <p>Kullanıcıların <code><input></code> alanına giriş yapması ile birlikte tanımlı seçenekler aşağı doğru açılan bir liste olarak görünecektir.</p>	x	x	✓

Gösterim/Düzenleme Etiketleri

<code><textarea name="**" rows="?" cols="?" wrap="{off hard soft}"> </textarea></code>	Yazının yazılmasını sağlayan metin alanı	✓	✓	✓
<code><textarea name="**" maxlength="??"> </textarea></code>		x	x	✓
<code><button name="**" type="{submit reset buton}" value="**"> </button> <input type="button" onclick="**" value="**"></code>	Buton tipinde giriş alanı			

Gösterim/Düzenleme Etiketleri

<code><input type="number" name="*" min="?" max="?" step="?" value="?"></code>	IE 9 ve öncesi web gezginleri tarafından desteklenmez.	x	x	✓
<code><input type="date" name="*"></code>	Kullanılan web gezginine bağlı olarak (Chrome) kullanıcının tarih tipinde veri girişi yapabilmezi için bir takvim çıkartır.	x	x	✓
<code><input type="color" name="*" value="#ff0000"></code>	Kullanılan web gezginine bağlı olarak renk seçiciyi çıkartır.	x	x	✓
<code><input type="range" name="*" min="?" max="?"></code>	Değeri seçmek için bir çubuk üzerinde kayan bar gösterir.	x	x	✓

Gösterim/Düzenleme Etiketleri

<input	type="month" name="**">	Kullanılan web gezginine bağlı olarak açılan takvim uygulamasından kullanıcının seçtiği ay ve yıl değerlerini almayı sağlar.	x	x	✓
<input	type="week" name="**">	Kullanılan web gezginine bağlı olarak açılan takvim uygulamasından kullanıcının seçtiği tarihin yılın kaçınıcı haftası olduğunu döndürür.	x	x	✓
<input	type="time" name="**">	Kullanılan web gezginine bağlı olarak açılan kutuda kullanıcının saat ve dakika belirlemesine imkan verir.	x	x	✓
<input	type="datetime" name="**">	Kullanılan web gezginine bağlı olarak açılan takvim uygulamasından kullanıcının tarih ve saat seçmesine imkan verir.	x	x	✓
<input	type="email" name="**">	Kullanılan web gezginine bağlı olarak verilen e-posta adresinin doğru olduğunu kontrol eder.	x	x	✓
<input	type="search" name="**">	Type="text" den farklı değildir.	x	x	✓
<input	type="url" name="**">	Kullanılan web gezginine bağlı olarak girilen URL'yi kontrol eder. Yazılan bilginin başında http:// veya ftp:// olması yeterlidir.	x	x	✓

Form Kullanım Örneği

```
<html><body>
<form name="form1">
İsim      <input type="text" name="isim" size="10" maxlength="20" tabindex="1"><br>
Cinsiyet  <input type="radio" name="cinsiyet" value="E">Erkek
          <input type="radio" name="cinsiyet" value="K">Kadın<br>
          <input type="checkbox" name="gsm" tabindex="2">GSM
          <input type="checkbox" name="laptop" tabindex="3">Laptop
          <input type="checkbox" name="tv" tabindex="4">TV
          <input type="hidden" name="gizli" value="gizli olarak yollanan deger"> <br>
Şifreniz  <input type="password" name="şifre" value="" tabindex="5"> <br>
Arabanız  <select name="araba">
          <option value="audi" selected>Audi
          <option value="bmw">BMW
          <option value="Mercedes">Mercedes
          </select><br>
Gazeteler <select multiple name="gazete">
          <option value="milliyet" selected>Milliyet
          <option value="hürriyet">Hürriyet
          <option value="sabah">Sabah
          </select><br>
Yorum     <textarea name="yorum" rows="5" cols="45"> </textarea><br>
          <input type="submit" value="formu yolla">      <input type="reset" value="formu temizle">
        </form>

</body>
</html>
```

The screenshot shows a web form with the following elements:

- İsim :** A text input field.
- Cinsiyet :** Two radio buttons labeled "Erkek" and "Kadın".
- GSM :** A checkbox.
- Laptop :** A checkbox.
- TV :** A checkbox.
- Şifreniz :** A password input field with two dots.
- Arabanız :** A dropdown menu with "Audi" selected.
- Okuduğunuz Gazeteler :** A list box with "Milliyet", "Hürriyet", and "Sabah" options.
- Yorumunuz :** A text area.
- Formu yolla :** A submit button.
- Formu temizle :** A reset button.

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklama	HTML3.x	HTML4.x	HTML 5.x
<code><fieldset</code> <code>disable</code> <code>form="**"</code> <code>name="**"></code> <code></fieldset></code>	<p>disable, <frameset> elemanı içinde yer alan giriş alanlarını (<input>) girişe kapatır.</p> <p><fieldset> eğer farklı bir formun dışında ise ve bir form ile ilişkilendirilmek isteniyor ise form özniteliğinde ilgili olduğu <form> elemanının id özniteliğine verilen değerin verilmesi gereklidir.</p> <p>name özniteliği kullanıldığı durumlarda yapılan işe göre tetiklenen özellikleri ile ilişkilendirilerek bazı stil değişiklikleri yapılması mümkündür.</p>	x	x	✓
<code><legend ></code> <code></legend></code>			✓	✓
<code><legend</code> <code>align="{top bottom left right}"</code> <code></legend></code>			✓	x

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklama	HTML 3.x	HTML 4.x	HTML 5.x
<code><map name="**"></code> <code></map></code>	Map: name: Hassas haritanın ismi	✓	✓	✓
<code><area</code> alt="**" coords="coordinates" href="URL" shape="{ default rect circle poly}" target="{_blank _parent _self _top Framename}" > <code></area></code>	Area: alt: resim gösterilemediği durumlarda onun yerine alacak olan açıklama satırıdır. coords: dikdörtgen için x1,y1,x2,y2 (sol üst köşe- sağ alt köşe) daire için x,y,radius , poligon (çok kenar) için x1,y1,x2,y2,...,xn,yn href: seçilen koordinat ile ilişkilendirilen URL shape: belirlenen alanın ne tür bir şekil olduğunu belirler target: gidilecek URL'nin nerede gösterileceğini belirler.	✓	✓	✓
<code><area</code> download="filename" hreflang="language_code" media="media query" rel="{ alternate author bookmark help license next nofollow noreferrer prefetch prev search tag}" type="media_type"> <code></area></code>	download: hreflang: media: rel: type:	✗	✗	✓
<code><area nohref="value"> </area></code>	nohref: belirlenen alanın (coords) karşılığında herhangi bir link'in olmadığını belirtir.	✓	✓	✗

Gösterim/Düzenleme Etiketleri

Etiket Yapısı	Açıklama	HTML3.x	HTML4.x	HTML 5.x
<code><div style="**"></code> <code></div></code>	Division: blok düzeyinde bir ayırım oluşturmak ve buna yeni bir stil vermek için kullanılır.	✓	✓	✓
<code><div align="{left right center justify}"</code> <code>style="**"></code> <code></div></code>	align: hizalama için kullanılır. style: stil belirlemek için gerekli CSS verilebilir.	✓	✓	✗

Çoklu Ortam Etiketleri

Tipi	MIME tipleri
Resim	image/gif image/jpeg image/tiff image/pict image/x-bitmap
Ses	audio/x-wave audio/basic audio/aiff
Video	video/mpeg video/quicktime video/x-msvideo

Kullanım	Açıklama
Dinlemek için buraya basın 	Kullanıcının basması ile devreye giren bir multimedia uygulaması
<embed src="ses.wav" width = "15" height = "15" autostart = "false">	Uygulamanın içine gömülmüş bir ses dosyası
<bgsound src="ses.wav" loop = "{? infinite}">	Arka planda çalacak bir ses dosyası

Meta Etiketleri

Kullanım	Açıklama
<code><meta name="{ keywords description author generator formatter copyright}" content="****" /></code>	<code><meta></code> tanımı içinde ister anahtar kelime ister tanım verilebilir.
<code><meta name="robots" content="{noindex follow noimageindex noimageclick}" /></code>	Arama motorlarının nasıl çalışması gerektiğini belirler. Noindex sayfada indislenmez, Nofollow sayfadaki linkler takip edilmez, Noimageindex sayfadaki resimlerin indislenmez, Noimageclick doğrudan resim içeren linkleri indislemeyi engeller.
<code><meta http-equiv="expires" content="Wed, 26 Feb 1997 08:21:57 GMT" /></code>	Sayfanın hangi tarihe kadar geçerli olduğunu belirler.
<code><meta http-equiv="pragma" content="no-cache" /></code>	Sayfanın ön belleğe konulmasını engeller
<code><meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /></code>	Yollanan bilginin MIME tipinin html olduğunu belirler
<code><meta http-equiv="Content-Script-Type" content="text/javascript" /></code>	Yollanan bilginin MIME tipinin javascript olduğunu belirler
<code><meta http-equiv="Content-Style-Type" content="text/css" /></code>	Yollanan bilginin MIME tipinin css olduğunu belirler
<code><meta http-equiv="Content-Language" content="en-GB" /></code>	Sayfanın dil kodlamasını belirler
<code><meta http-equiv="Refresh" content="3; URL=http://www.some.org/some.html" /></code>	Verilen süre sonunda sayfa belirlenen URL'den yüklenir.
<code><meta http-equiv="Window-target" content="_top" /></code>	Sayfanın frame tabanlı başka bir sayfada görünmesini engeller.
<code><meta http-equiv="Set-Cookie" content="cookievalue=***; expires=Friday, 31-Dec-99 23:59:59 GMT; path=/" /></code>	Cookie yaratmak için kullanılır..

Diğer HTML Etiketleri

Etiket Yapısı	Açıklama	HTML 3.x	HTML 4.x	HTML 5.x
<code><abbr title="Türkiye Cumhuriyeti Büyük Millet Meclisi"> TCBMM </abbr></code> 23 Nisan 1920 de kuruldu. <code></p></code>	Özellikle kısaltamalar için kullanılır. üzerine gelince açık halinin görünmesini sağlar. Web gezginleri, tercüme sistemleri ve arama motorları için faydalı bilgilerin çıkartılmasına yardımcı olacaktır.		✓	✓
<code><acronym title="Türkiye Cumhuriyeti Büyük Millet Meclisi">TCBMM</acronym></code> 23 Nisan 1920 de kuruldu. <code></p></code>	Özellikle kısaltamalar için kullanılır. üzerine gelince açık halinin görünmesini sağlar. Web gezginleri, tercüme sistemleri ve arama motorları için faydalı bilgilerin çıkartılmasına yardımcı olacaktır.		✓	✗
<code><address></code> Yıldız Teknik Üniversitesi <code>
</code> Bilgisayar Mühendisliği Bölümü: <code>
</code> Davutpaşa-ESENLER <code>
</code> 34220 – İSTANBUL <code>
</code> <code></address></code>	<code><address></code> etiketi belgenin yazarını/sahibinin iletişim bilgilerini göstermek için kullanılır. HTML 4.x de <code><body></code> elemanı içinde olduğunda söz konusu belge ile ilgili irtibat kişisini, HTML .x de <code><article></code> elemanı içinde ise o makale ile ilgili kişiyi gösterecektir. Genel olarak bu etiketin içeriği eğitim yazı sitlinde gösterilir..		✓	✓
<code><head></code> <code><base href=" resimler/" target="_blank" /></code> <code></head></code>	<code><head></code> elemanı içinde kullanılan <code><base></code> elemanının href özneliğinde verilen URL belge içinde kullanılan her görelili (relative) adresin başına eklenerek kullanılacaktır.		✓	✓

Web sayfalarının daha yüksek başarımla çalışması için ipuçları

HTTP isteklerin sayısını azaltmak: Web sunucusuna gelecek olan her istek onun başarımla azaltacaktır. Bu nedenle istek sayılarını azaltacak bazı düzenlemeler yapılarak istemcilerin cevap alma süreleri iyileştirilebilir. Bu amaçlar sayfa içinde birden fazla script ve CSS dosyası kullanmak yerine bu dosyaları birleştirilerek tek seferde yüklenmelerini sağlayarak istek sayısını azaltmak mümkündür.

İmkan dahilinde ise CDN (Content Delivery Network)'lerden yararlanın: Sayfa içeriklerinin birden fazla kaynaga dağıtarak sunulması (trafiğin bir sunucuya doğru yoğunlaşmasına mani olup) sayfaların daha hızlı yüklenmesine imkan sağlar. Kullanıcı açısından cevap süresinin %80-90'lık kısmının indirme (download) için kullanıldığı düşünüldüğünde içeriğin birden fazla içerik barındırıcısından paralel olarak temin edilmesi çok faydalı olacaktır.

Önbellekte tutulma süresinin iyi belirlenmesi: Sabit içerik için önbellekte tutulma süresi sürekli (never expires) iken dinamik içerik için uygun bir değer olarak belirlenmelidir. Bunun için bir geçerlilik son tarihi girilebildiği gibi web sunucusunun (Apache gibi) özellikleri arasında yüklemenin yapıldığı tarihe bağli olarak da süre belirlenebilmektedir.

Sıkıştırma ile aktarılan verinin küçültülmesi: HTTP 1.1 ile başlamak üzere web istemcileri sıkıştırma desteklemektedir. Yollanan istek başlığı içerisinde Accept-Encoding: gzip, deflate tanımını gören web sunucusu yolladığı cevabı belirlenen sıkıştırma metodlarından biri (genel olarak gzip) ile sıkıştırarak yollar. Yollanan cevap başlığı içinde Content-Encoded: gzip ile bu durumu istemciye de bildirir. Gzip kullanarak yollanan cevap verisi %70 civarında küçülebilmektedir. Sadece resim (JPEG, PNG, GIF vb) ile PDF dosyalar zaten sıkıştırılmış içerik barındırdıkları için sıkıştırılmazlar.

Web sayfalarının daha yüksek başarımla çalışması için ipuçları

Stilleri sayfanın başında tutmak: Stil tanımlarının sayfanın başında bulunması nedeni ile sayfanın görselleştirilmesi (render) aşamalı olarak sağlanarak sayfanın daha hızlı yüklenmesine imkan sağlar.

Scriptler sayfanın sonunda tutmak: HTTP 1.1 ile web gezginleri aynı anda ikiden fazla parçayı paralel olarak yüklemeyebilir. Eğer resim gibi içerikler farklı iki sunucu üzerinden alınıyor ise işlem paralel yapılırken is scriptlere geldiğinde kaynağı farklı bile olsa aynı anda tek yükleme gerçekleşir. Her durumda uygulanabilir olmamak ile birlikte <script> etiketi içinde kullanılacak olan defer özneliği ile bunların yüklenmesi sona bırakılabilir.

CSS ve JavaScript dosyalarının harici olarak kullanılması: CSS ve javascript dosyalarını harici olması bunların önbellekte tutulmasına imkan verdiği ve HTML dosyasının boyutunu küçültmeye imkan verdiği ve HTTP isteklerini azalttığı için performans artışına imkan verir.

DNS sorgularını azaltın: Kelimeler ile belirlenen adreslerin numara (IP) ile belirlenen adreslere dönüştürme işi DNS (Domain Name System) tarafından yapılır ve elde edilen dönüşüm sonuçları gerek işletim sistemi gerek ise web gezginleri tarafından bir önbellek alanında tutulur. Böylelikle yapılan DNS isteklerinin azaltılması hedeflenir.

Web sayfalarının daha yüksek başarımla çalışması için ipuçları

CSS ve JavaScriptlerin küçültülmesi (minification): Gerek kolay okunabilir olmaları, gerek ise daha rahat takip edilebilmeleri ve anlaşılabilmesi için yazılan kodun içinde bazı açıklama satırlarının eklenmesi, boşluklar ve sekmeler kullanılarak bazı düzenlemelerin yapılması bir alışkanlıktır. Ancak bunların karşı tarafa aktarılması da bir zaman alır. Oysa bu kodu çalıştıracak sistemin için kodun içine yerleştirilmiş açıklama satırları ile düzenlemeye yönelik olarak konan boşluk ve sekmelerin hiçbir anlamı yoktur. Bunların kod içinde kaldırılması ile yollanacak bilgi miktarı azalacaktır.

Sayfa yönlendirmelerinden (redirect) kaçınılmalıdır: HTTP başlığında yer alan 301 ve 302 kodları ile istek başka bir URL'ye yönlendirilmektedir. Bu durum sıklıkla yapılan bilinçsiz kodlama ile de oluşur. Örneğin URL olarak <http://www.ce.yildiz.edu.tr> yazılacak olursa bu 301 kodu ile <http://www.ce.yildiz.edu.tr/> şeklinde yeniden yönlendirilecek, böylece ikinci bir HTTP isteği yapılmış olacaktır. Aradaki temel fark adresini sonuna / sembolünün konmamış olmasıdır.

Çift kullanılan scriptlerin kaldırılması: Dikkatsizlik sonucu scriptler çift olarak sayfalarda yer alabilmekte, bu da sayfa yüklenme zamanının uzamasına neden olmaktadır.

Ajax kodunun ön belleğe alınması

Web sayfalarının daha yüksek başarımla çalışması için ipuçları

Önbelleğin zamanında önce boşaltılması; Sunucu tarafında kısmen oluşmuş HTML içeriğin istemciye yollanarak sunucunun arta kalan kısmı oluşturmak için uğraşırken istemcinin aldığı kadarını işleyerek zamandan kazanmak hedeflenmektedir.

Neyin önce neyin sonra yükleneceğine doğru karar vermek: Sayfalarda yer alan scriptler bu manada şekillendirilebilirler. Örneğin bir sayfa üzerinde sürükle-bırak tarzı işlemler yapacak olan scriptler, sayfanın temel kısmı oluşturulmadan yüklenmesi gerekmeyen özelliklerdir. Bu nedenle onları daha sonradan yüklemekte bir mani yoktur.

DOM elemanı kullanımının azaltılması: DOM elemanlarının sayısının artması performansı düşürür.

<iframe> kullanımını dengelemek: <iframe> farklı adreslerdeki, üçüncü taraflı yavaş içeriklerin sayfaya dahil edilebilmesi, bu işlemlerin kapalı bir kutu (sandbox) içinde yapılması ve scriptlerin paralel olarak yüklenmesine rağmen boş olsalar bile maliyetli, sayfa yüklenme aşamasında süreci bloke eden ve hiç bir anlambilimsel (semantic) özelliği olmayan bir elemandır.

Web sayfalarının daha yüksek başarımla çalışması için ipuçları

404 den kaçınin: Birşey bulunamadığında döndürülen 404 kodu kullanıcı deneyimini yavaşlatmanın yanı sıra hiçbir bilgi içermemesi açısından da faydasızdır.

Cookie büyüklüklerinin indirgenmesi: Cookie bilgileri sunucu ve istemciler arasında HTTP başlıkları üzerinden taşınırlar. Bunların sayısının artırılması başlık alanında daha çok şeyin taşınması dolayısı ile kullanıcıya yansıyan cevap süresinin uzamasına neden olacaktır.

Resim dosyaları optimize edilmeli: Zaman zaman GIF tipi resimlerin PNG'ye çevrilmesi boyutlarının daha ufalmasına imkan sağlayabilir.

Resimleri HTML ile boyutlandırmamak: Resmi büyük olarak yollayıp kullanılarak yeniden boyutlandırılması taşınan verinin büyüklüğünü değıştirmez. Bu cevap süresinin uzamasına neden olur.

Kullanılan parçaların 25KB dan küçük olmasını sağlamak

CSS

CSS (Cascading Style Sheets)

HTML elemanlarının nasıl gösterileceğini belirleyen stilleri tanımlamaya yarar

Stiller, Style Sheet olarak adlandırılan harici dosyalarda saklandığı gibi HTML dosyasının içinde de bulunabilir.

Birden fazla stil tanımı birbiri ardına basamaklanabilir.

HTML belgesi içinde `<style>` `</style>` etiketleri arasında yer almalıdır (head bloğu içinde).

- Gövde (`<body>` `</body>`) kısmı işlenmeden CSS tanımlarının devreye girmiş olması gereklidir.

Tanımlamalar `{` ve `}` işaretleri arasında yer alır.

Stil özellik:değer ikilileri kullanılarak belirlenir.

- Kullanılacak öznitelik için verilecek değer bir kelimedenden daha uzun ise tırnak içinde yazılmalıdır.
- Bir seferde birden fazla özellik birden ayarlanabilir (; ile ayrılarak)

Açıklamalar `/*` ve `*/` arasına yazılır.

Eleman (element) tipinde stil tanımı

```
<style type="text/css"> p {font-family: "sans serif"; text-align:center; color:red } </style>
```

HTML belgesi içinde kullanılacak olan eleman, aksini gösterir bir tanım bulunmadığı sürece bu şekilde formatlanır.

Bileşik seçiciler, yalnızca verilen bir elemanın içinde ikinci tanımlanan eleman yer aldığı anda stil tanımının etkili olmasını sağlar.

```
ul ul { list-style-type : square }
```

- Noktalı listenin içinde yer alan bir noktalı (alt) liste, işaretçi olarak kare kullanır.

```
h1 em em { color : red }
```

- Bir başlık içinde yer alan vurgulu yazı içindeki vurgulu yazı kırmızı olarak yazılır.

Bileşik seçiciler, basit seçicilerden daha özeldir. Örneğin, aşağıdaki stil tanımlarının ikisi de yapıldığında,

```
p { color : red }
```

```
div p { color : blue }
```

<paragraph> bir <div> içinde yer aldığı anda ikinci kural uygulanır.

Öznitelik sınıfı (attribute class) tipinde stil tanımı

```
<style type="text/css">
```

```
    p.right {text-align: right}
```

```
    p.center {text-align: center}
```

```
</style>
```

```
...
```

```
<p class="right"> Sağa yaslanmış yazı</p>
```

```
<p class="center"> Ortalanmış yazı</p>
```

Her HTML elementi için sadece bir tek sınıf tanımı kullanılabilir.

Öznitelik sınıfı (attribute class) tipinde stil tanımı

Esnek kullanımını arttırmak üzere öznitelik sınıf tanımlarını tek bir HTML etiketine bağlamak yerine genel bir tanım da yapmak mümkündür:

```
<style type="text/css">
```

```
    .center {text-align: center}
```

```
</style>
```

```
...
```

```
<h1 class="center"> Bu başlık ortalanmıştır. </h1>
```

```
<p class="center"> Bu paragraf da ortalanmıştır.</p>
```

Sözde Sınıflar (Pseudo-classes)

Stil sınıflarına benzer, ancak burada etkilenen eleman kalıtım dışında bir ilişki ya da bir kullanıcı aksiyonu gerektirir

Stil sayfasında, sözde sınıf adının önüne “:” eklenir.

HTML sayfasında, sözde sınıf adı etkilenen etiketle birlikte kullanılmaz, çünkü dış etki ile harekete geçer.

Eğer eşik özellik seviyesine sahip iki kural aynı elemana etki ederse, **son gelen** stil tanımı kullanılır.

Aşağıdaki örnekte, a:hover sözde sınıfının etki etmesi için mutlaka a:link ve a:visited sözde sınıflarını takip etmesi gereklidir.

- İlk ikisinin sırası, birbirini dışlayan (her ikisi de aynı anda doğru olmayan) olduğu için önemli değildir.

a:link { color : red } Henüz tıklanmamış link kırmızı.

a:visited { color : green } Önceden tıklanmış link yeşil.

a:hover { color: yellow } İşaretçi linkin üzerine geldiğinde sarı.

Sözde Elemanlar

Sözde sınıflara benzerler ve açık tanımlarla değil, ilişki ile tanımlanırlar.

Sözde eleman, gerçek bir elemanın özel bir durumundan çok, o gerçek elemana refere eden sanal bir elemandır.

:first-line sözde elemanı, blok seviyeli bir elemanın içindeki yazının ilk satırına refere eder.

:first-letter sözde elemanı, bir eleman içinde yer alan metnin ilk harfine refere eder.

```
p { text-indent: 1em }
```

```
p.initial { text-indent: 0 }
```

```
p.initial:first-line { text-transform: uppercase }
```

```
p.initial:first-letter { font-size: 24pt }
```

Tüm normal paragraflar içerden başlasın. “Initial” sınıfından olan paragraflar içerlikli olmasın, ilk satırındaki tüm harfler büyük olsun, ilk harfi de çok büyük olsun.

Öznitelik kimlik (attribute ID) tipi stil tanımı

Farklı HTML etiketlerine stil vermek üzere kullanılmaktadır.

```
#red {color: red}
```

```
p#tip1 { text-align: center; color: red}
```

...

```
<h1 id="red">Başlık kırmızı olacak</h1>
```

```
<p id="red">p elementi için yapılmış id tanımı kullanılacak.</p>
```

CSS Kullanımı

Stiller, düşükten yüksek önceliğe göre:

- Web gezgininin varsayılan stil tanımlamaları (Browser style)
- Harici stil sayfasında yapılan tanımlamalar (External style)
- HTML kodunun içinde `<head> </head>` etiketi içinde kullanılan `<style> </style>` etiketleri ile yapılan tanımlamalar (Internal style)
- HTML etiketinin içinde style özniteliği kullanılarak yapılan stil tanımlamalar (Inline style)

Harici stil sayfasında yapılan tanım ile dahili stil tanımı çelişecek olursa dahili stilin önceliği yüksek olduğu için onun belirlediği stil geçerli olur.

Farklı olan tanımlamalar ise birleşim kümesi olarak kullanılır.

Harici CSS Kullanımı

Etiket Yapısı	Açıklama	HTML 3.x	HTML 4.x	HTML 5.x
<code><link href="URL" hreflang="language_code" media="media_query" rel="{stylesheet}" type="media_type" > </link></code>	<p>Link an external stylesheet : Harici bir CSS dosyasının sayfa ile ilişkilendirilmesi için kullanılır. Bu etiketin <head> bloğu içinde kullanılması gereklidir.</p> <p>href: harici CSS dosyasının URL'sidir</p> <p>hreflang: CSS dosyasının dilini gösteriren iki harfli kısaltmadır.</p> <p>media: kaynağın ne tür bir ortam/cihaz için optimize edildiğini belirler.</p> <p>rel: içinde bulunan belge ile link edilen belge arasındaki ilişkiyi belirlemek için kullanılır. Harici stil dosyalarında bunun değeri stylesheet olmalıdır.</p> <p>type: href özniteliği ile bir referans verildi ise bunun media tipini (MIME type) belirlemek için kullanılır.</p>		✓	✓
<code><link charset="*" rev="{stylesheet}" target="{ _blank _self _top _parent frame_name}"> </link></code>	<p>charset: bağlantılı dosyanın kodlama sistemini belirler</p> <p>rev: rel özniteliğinin tersi olarak kendini çağıran belge ile olan ilişkisini belirlemek için kullanılır.</p> <p>target: bağlantı kurulan belgenin hangi hedefde görüneceğini belirler.</p>		✓	✗
<code><link crossorigin="{ anonymous use-credentials}" sizes=" HeightxWidth"> </link></code>	<p>crossorigin: farklı kaynaklardan gelen isteklerin eleman tarafından ne şekilde değerlendirileceğini belirlemede kullanılır.</p> <p>sizes: rel="icon" olan durumda kullanılacak iconun boy ve en değerlerini belirlemek için kullanılır.</p>		✗	✓

Dahili CSS Kullanımı

```
<head>
```

```
  <style type="text/css">
```

```
    p {margin-left: 15px}
```

```
    hr {color: blue}
```

```
    h1 {text-align: center; color: red}
```

```
    body {background-image: url("images/arkaplan.gif")}
```

```
  </style>
```

```
</head>
```

Her dosyanın tek tek incelenerek gerekli düzeltmelerin yapılmasını gerektirir.

Bakım ve sürekliliğin sağlanması açısından dâhili stil kullanımı harici olana oranla daha zordur.

Inline CSS Kullanımı

Her HTML elemanı için bağımsız stil tanımlaması yapılabilir.

Bu tip tanımlamaların değiştirilmesi gerektiği zaman son derece yoğun uğraş gerekecektir !

- Çok özel durumlar dışında kullanılmaması daha uygundur.

`<p style="color: green; margin-left: 10px"> Tanımlama sadece bu paragraf boyunca geçerlidir.</p>`

Gelecek Ders

Client Side Scripting, Javascript