

Hızlandırılmış HTML Kursu Notları:

Giriş

HTML nedir?

Herşeyden önce bazen yanlış anlaşıldığı gibi HTML bir programlama dili değildir. HTML bir mark-up (~işaretleme) dilidir, yani *tag* denen özel işaretlerle bir yazının tümünü ya da sadece bir kısmının görünümünü ve/veya işlevini değiştirmeye yarar. Örneğin herhangi bir tag olmadan default yazı tipinde görünen bir yazı, HTML içinde önüne ve arkasına `<I>` ve `</I>` işaretleri gelince *bu şekilde italik* görülür

Ayrıca, gelişen browser teknolojileri sayesinde, HTML içine yazılar dışında resim, ses, video, Java, JavaScript, VBScript gibi dillerle yazılmış programlar, ya da diğer nesneler de gömülebilir. Bu sayede bir HTML dökümanı düzenlenmiş, mizampajı yapılmış bir metin dosyası olmaktan çıkıp tam anlamıyla bir multimedia ortamı olmaya da hizmet edebilir. Örneğin, browser'a özel olarak yazılan gerekli plug-in'ler (~eklentiler) sistemde varsa, HTML'le online bir gazete, bir slide-show, şirket-içi yazılımların çalışabildiği bir platform yaratabilirsiniz.

HTML tag'leri

HTML tagleri bir çok farklı şekilde sınıflandırılabilir. Bir sınıflandırma şu şekildedir:

1. Bir açma bir de kapama tag'inden oluşan *container* tag'leri,
2. Tek başına bulunan tag'ler.

Örnek olarak, ilk gruptan

```
<B>Kalın yazı</B>  
<FONT COLOR="Maroon" SIZE="+1">Büyükçe, gri metin parçası</FONT>
```

ikinci gruptansa

```
<HR>  
<META HTTP-EQUIV="Refresh" CONTENT="5" URL="http://www.ulakbim.gov.tr/">
```

verilebilir.

HTML Dökümanlarının Yapısı:

Bir HTML dökümanı genel olarak (istisnaları vardır) aşağıdaki şekildedir:

```
<HTML>  
<HEAD>  
<!-- Head elemanları -->  
Burada genelde döküman içeriği dışında kalan karakter set tanımlamaları, ba  
</HEAD>  
<BODY>  
<!-- Body elemanları -->  
Burada ise dökümanın asıl içeriği vardır, içinde metin, ses, video vb içere  
<BODY>  
</HTML>
```

Kuraldışı durum olarak da *frameset*'ler verilebilir, framesetler <BODY></BODY> tag'lerini içermeyebilirler. Frameset'lere ileride gelinecektir.

İlk HTML Dökümanı:

Örnek olarak aşağıdaki HTML kodunu ele alalım:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<!--last modified on Tuesday, June 02, 1998 12:26 AM -->
<HTML>

<HEAD>

  <META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=iso-8859-9">
  <META NAME="GENERATOR" Content="Visual Page 1.1 for Windows">
  <META NAME="Author" Content="Cüneyt YILMAZ">
  <META NAME="Keywords" Content="deneme, learning HTML, dummy document">
  <TITLE>İlk HTML Dökümanı</TITLE>

</HEAD>

<BODY TEXT="#333333" BGCOLOR="White" LINK="Navy" VLINK="#006600" ALINK="#006600">

  <H1>Güzelinden bir balık atalım </H1>
  Bu Ulakbim'deki birlikte inceleyeceğimiz ilk HTML dökümanı.<BR>
  İçeriği imdilik önemli olmasa da ona da sıra gelir :)<BR>
  Bu satırın 2 altında bazı sayılar olmalı, acaba öyle mi?

  298 93 00
  <HR>
  Hemen üstümde irin bir çizgi var.

</BODY>
</HTML>
```

Yukarıdaki kodun ilk satırı SGML (HTML'in atası) sözdizimine göre aşağıdaki satırların bir HTML 3.2 dökümanı olduğunu belirtmektedir, genelde ihmal edilebilir, ancak Internet üzerinde bazı HTML validator (sözdizimini kontrol eden programlar) tarafından kullanılır ve eğer kullanılmazsa dökümanın HTML 2.0'a uygun olduğu varsayılır.

2. satırda bir HTML comment'i (açıklaması) vardır. HTML commentleri <!-- ile başlar ve --> ile biter. Bu iki tag arasında kalan bütün yazılar browser tarafından ihmal edilir (JavaScript hariç, ileride bu konuya geri dönecektir).

3. satırda dökümanımız resmi olarak açılmış bulunmakta.

4. satırda browser'ın dökümanı gösterirken işine yarayabilecek bazı bilgiler vb elemanların bulunduğu HEAD kısmı açılmaktadır.

5. satırda dökümanın render edilirken (mizanpajının yapılıp ekrana dökülürken) kullanılacak karakter seti verilmektedir. Sık kullanacağımız setler ISO-8859-1 (Latin1) ve ISO 8859-9 (Türkçe)'dur. Bazı Windows programlarının (isimleri lazım değil) kullandığı bir set ise standart olmayan Windows-1254 (Unix'te görüntülenemeyen bir Türkçe)'dür.

6. satırda döküman hazırlanırken hangi HTML editörünün kullanıldığı yazmaktadır.

7. satırda dökümanı hazırlayan kişinin ismi vardır.

8. satırda daha çok AltaVista, HotBot gibi bazı arama makinelerinin kullandığı bir bilgi bulunmaktadır. Sözelimi AltaVista'da arama yapan birinin sizin dökümanınıza ulaşmasını istiyorsanız, verebileceği anahtar kelimeleri burada tanımlarsınız.

9. satırda döküman gösterilirken browser'da pencerenin başlığı olarak ne görüleceği bilgisi vardır. Bu bilgi ayrıca sayfanızı bookmark'ına (Navigator) ya da favorites'ine (Explorer) ekleyen kişilerde görülecek olan isimdir.

10. satırda HEAD kısmı kapanmaktadır.

11. satırdan itibaren dökümanın içeriği hemen hemen başlamaktadır. Burada sayfanın görüntülenirken ana görünümün nasıl olacağı belirlenir. BODY'den sonra gelen 5 eleman sırasıyla şunları ifade eder:

- Düz metin rengi
- Arka fon rengi
- Ziyaret edilmemiş linklerin rengi
- Ziyaret edilmiş linklerin rengi
- Fareyle link'e tıklandığı andaki (düğme halen basılıken) link'in rengi
- Arka fonda kullanılacak olan resim

Burada renkleri verirken 2 tip gösterim kullandık, #RRGGBB ve Renk_İsmi. #RRGGBB, hexadecimal olarak 2şer bayttan Kırmızı, Yeşil ve Mavi bileşenleri verir. Örneğin saf kırmızı #FF0000, camgöbeği #00FFFF'dir. Diğer gösterim daha kolaydır ve HTML standardında bulunan 16 (Navigator için yüze yakın) renkten birini verir. White, Blue, Gold gibi bir çok renk bulunmaktadır.

Arka fonda resmin kullanıldığı ve tabii ki kullanıcıda gösterilebilen bir resmin olduğu dökümanda arka fon renginin işi ne? Internet'e bağlı kullanıcıların bir kısmı hatları yavaş olduğu için resimleri otomatik yüklemeler. Böyle bir durumda dökümanınızın istediğinize en yakın görülebilmesi ve, bazen olduğu gibi siyah üstüne siyah yazan, yani boş gibi görünen bir sayfayla karşılaşmamaları için verilmesi tavsiye edilir.

Renkleri verdikten sonra artık sıra geldi içeriğe. Önce <H1>ve </H1> arasına kocaman bir başlık koyduk. Burada H1 ve H6 arasında istediğiniz birini başlık veya altbaşlık atmak için kullanabilirsiniz. H1 en önemli başlıklar içindir ve en büyüğüdür. H6 ise en önemsiz başlıklar içindir ve en küçüktür. Bu başlıkların kullanıcı tarafında nasıl görüleceğini bilemezsiniz, ancak elinizle yapacağınız bir düzenlemeden (puntoyu 2 arttır vs gibi) daha kolay ve garantilidir, çünkü browser uygun bir büyüklük seçecektir.

Başlığı attıktan sonra biraz metin yazılmış ve satır sonunda browser'ın bir alt satırdan devam etmesini söyleyen
 tag'i var. Devam edelim, alt satıra geçilmiş ve tekrar
 verilmiş. Onun altındaki satırda yazdığına göre iki alt satırda bir telefon numarası olması gerekiyormuş. Ancak browser'da bakıldığı zaman böyle olmadığı görülecektir, çünkü HTML
, <P>, <H3> gibi tagler dışında döküman içindeki **bütün** satır sonu karakterlerini, tab'lerini ve birden fazla boşluğu, üçüne birden whitespace de denir, ihmal eder ve tek bir boşluk karakteriyle değiştirir.

Sonraki satırda, <HR> yatay bir çizgi çektik (maalesef <VR> diye bir tag'imiz yok).

Sonra biraz daha metin ve sırasıyla BODY'yi ve HTML dökümanımızı kapattık.

HTML 4.0 ve StyleSheet'lerin kullanıma girmesiyle değişen yukarıdaki BODY tanımı artık desteklenmemektedir.

```
<BODY TEXT="#333333" BGCOLOR="White" LINK="Navy" VLINK="#006600" ALINK="#006600">
```

yerine

```
<STYLE type="text/css">
BODY { background: white; color: #333333;
background-image:URL("resim.jpg");}
A:link { color: navy }
A:visited { color: #006600 }
A:active { color: #009999 }
</STYLE>
```

kullanılması önerilmektedir.

Temel HTML Tag'leri:

<HTML></HTML>, <HEAD></HEAD> ve <BODY></BODY>: HTML'in en temel blok tag'leridir. Bir HTML dökümanı genelde <HTML> ile başlar ve </HTML> ile biter, ancak verilmesi bile, browser dökümanı sorunsuz olarak görüntüleyebilir. Aynı şey <BODY> için de gereklidir. Ancak, <HEAD>'in verilmesi ve döküman içinde HEAD bloğu varsa, <BODY>'nin de verilmesi zorunludur. Parametre olarak *bir çok blok elemanı gibi* LANG=Dil alır. Örneğin <BODY LANG=tr>.

<TITLE></TITLE>: İlk öğrenilmesi gereken tag'lerden biridir. Kullanımı basittir ve HEAD bloğunda kullanılır.

<P></P>: Paragraf açma kapama tag'leri. Yeni bir paragraf açarken/kapatırken kullanılır. Browser <P> tag'iyle karşılaştığı yerde yukarıdan ve aşağıdan biraz boşluk bırakır. Çoğu browser kapama tag'inin olmasına ihtiyaç duymaz.

Yararlı bir parametre olarak ALIGN=LEFT|CENTER|RIGHT olabilir. <P ALIGN="CENTER">Metin...</P> dediğiniz zaman aradaki bütün metin sol tarafa yaslı değil ortalanmış görülür. RIGHT ise yazıyı sağa dayar.

: Satır sonu tag'i. Kapama tag'i yoktur. <P> gibi altta, üstte boşluk bırakmadan yeni bir satır açar.

<HR>: Yatay çizgi tag'i. Parametre olarak ALIGN=LEFT|CENTER|RIGHT, WIDTH=xx%, SIZE=x ve NOSHADE olabilir. WIDTH ile yüzde olarak genişliğini verebilirsiniz. SIZE'nin varsayılan değeri 2'dir. NOSHADE ileyse, 3 boyut etkisini istemediğinizi belirtebilirsiniz.

: Kalın yazı tag'i. İçeride kalan bütün metin kalın harflerle görüntülenir. Ancak yeni HTML standartlarını hazırlayan organizasyon yerine *ileriki günlerde* kullanılmasını önermektedir.

<I></I>: İtalik yazı tag'i. İçeride kalan bütün metin italik harflerle görüntülenir. Yine gibi, <I></I> yerine (emphasis) kullanılması önerilmektedir.

<H1></H1>...<H6></H6>: Başlık tag'leri. Dökümanı çeşitli bölümlere ayırmak ve bunları belirtmek için kullanabilirsiniz. ALIGN=LEFT|CENTER|RIGHT parametresi olabilir.

 ve : Numaralandırılmış liste (ordered list) tag'leri. ve listeyi açma ve kapama için, ise yeni bir eleman tanımlamak için kullanılır. Numaralar otomatik olarak verilir.

 ve : Numaralandırılmamış liste (unordered list) tag'leri. ve listeyi açma ve kapama için, ise yeni bir eleman tanımlamak için kullanılır. Numara yerine bullet konur.

 : Karakter tipini, puntosunu, rengini vermek için kullanılır. Çok kullanılan tag'lerden biridir. Parametre olarak FACE=Font_İsmi (Arial, Helvetica vs gibi), SIZE=x|+/- x (SIZE="3", SIZE="-1" vs gibi), COLOR=#RRGGBB|Renk_İsmi (COLOR="#808080" vs gibi) alabilir. HTML 4.0'da iptal edilmiştir.

<BASEFONT> : Dökümanın genel yazı tipini belirtmek için kullanılır. Parametreleri tag'ininkilerle aynıdır. HTML 4.0'da iptal edilmiştir.

<A> : Anchor tag'i. HTML'in HTML olmasını sağlayan tag'dir. Metin içinde bir başka yere, Internet üzerinde herhangi bir dökümana, resme ya da herhangi bir servise ulaşılabilmesini sağlar. En gerekli parametre olarak HREF=Kaynak_İsmi alır. Kaynak kısmında kullanılabilecek isimlere örnek olarak:

- HREF=table_of_contents.html
- HREF=".."
- HREF=../MySubDir2/TarDosyası.tar
- HREF="http://www.ulakbim.gov.tr" /
- HREF="ftp://cu:cork@efe.ulakbim.gov.tr/" (ftp client'ını açar, efe'ye bağlanır ve kullanıcı ismi olarak cu, password olarak cork verir (şifrem tabii ki cork değil :)
- HREF=telnet://efe.ulakbim.gov.tr/"
- HREF=mailto:cu@ulakbim.gov.tr?subject=HTML%20hakkında+cc=cu%40reborn.com (e-mail client'ını açar, cu@ulakbim.gov.tr adresine, subject'i HTML hakkında olan ve bir kopyası da cu@reborn.com'a gönderilecek bir e-mail taslağı çıkartır. Dikkat edilirse, boşluk yerine %20, @ işareti yerine %40 kullanılmıştır. %Karakter_Hex_Kodu olarak istediğiniz karakteri kullanabilirsiniz. İlk parametreden önce ?, sonrakilerden önce de + kullanılır.)
- HREF="javascript:alert('Gidebilir miyim')"

Ayrıca bir diğer parametresi NAME=Anchor_İsmi'dir. Bu şekilde bir anchor tanımladığınız zaman döküman içinde istediğiniz bir yerden tam bu noktaya zıplamanız mümkündür. Örneğin sayfa en tepesine <A> dediğiniz zaman sayfanın en altına koyduğunuz bir Click here to go to top. link'iyle dökümanın en üstüne geri dönülebilmesini sağlarsınız. # işareti önemlidir ve link'in hedefinin bir anchor olduğunu gösterir. Ayrıca bla bla.. gibi bir link'le başka bir döküman içindeki bir anchor'a da link koyabilirsiniz.

Tabii ki Anchor tanımlarken hem NAME hem de HREF parametrelerini verebilirsiniz. Böylece aynı anda hem dışarıya link vermiş, hem de dışarıdan link verilebilecek bir blok tanımlamış olursunuz. Örneğin, .

<BASE> : Döküman içinde bütün URL'leri uzun uzadıya vermemek için bir ön-ek URL tanımlamak için kullanılır. Parametre olarak HREF=Ön-ek_URL'i alır. Örneğin <BASE

HREF=http://www.ulakbim.gov.tr/> verildiği zaman içindekiler bulunan directory'deki değil http://www.ulakbim.gov.tr'e yönelir. Yine de bu şekilde bir kullanım pek tavsiye edilmemektedir. HEAD bloğunda kullanılmaktadır.

 : Döküman içine bir resim yerleştirmek için kullanılır. Parametre olarak SRC=Resim_URL'i, WIDTH=Genişlik, HEIGHT=Yükseklik, ALT=Alternatif_Text, LOWSRC=Düşük_Çözünürlükteki_Resim_URL'i alabilir BORDER=Çerçeve_Kalınlığı, HSPACE=x, VSPACE=x (Yatay ve dikey marjinler) alabilir. Çok sık olarak bir bloğu içine yerleştirilir ve görsel bir link yaratılmış olur.

 ve : Sırasıyla superscript ve subscript yazı için kullanılır. Superscript yaptığınız yazı yukarıda, subscript yaptığınız yazı aşağıda görüntülenir. Örneğin x (superscripted) ve $M_{1,2}$ (subscripted).

<BLOCKQUOTE></BLOCKQUOTE> : Alıntı yapmak için kullanılır. Genelde ise bir bloğu toptan içeri indent etmek için kullanılır. Bu dökümandaki kod örnekleri bu tag'le içeri doğru alınmıştır. Indent etmek için kullanılması yüzünden, alıntı yaptığınızda tırnak (") işareti koymanız tavsiye edilmez.

<TT></TT> : Teletype (monospaced, sabit genişlikte) karakter kullanmak için kullanılır. Courier tipik bir teletype karakter tipidir. Bu sayfalardaki bütün kod örnekleri de Teletype stilindedir.

<PRE></PRE> : Önceden düzenlenmiş (preformatted) metin blokları için kullanılır. Normalde, HTML birden çok whitespace'i tek bir boşluk karakteriyle değiştirdiği için, görüntülenmesi başka türlü mümkün olmayan metin blokları bu tag yardımıyla görüntülenebilir. Örneğin:

```
Welcome to my little corner of tutorial
Here are some formatting that is otherwise impossible
```

```
          May 1998
Mon  Tue  Wed  Thu  Fri  Sat  Sun
-----
  1    2    3    4    5    6    7
  8    9   10   11   12   13   14
 15   16   17   18   19   20   21
 22   23   24   25   26   27   28
 29   30   31
```

<STRIKE></STRIKE> : Strike-Through (üstü çizili yazı) için kullanılır. Vurgu vermek ya da bir şeyin iptal edildiğini göstermek için kullanılabilir ancak pek rastlanmamaktadır. HTML 4.0'da iptal edilmiştir.

 : Altçizgi çekmek için kullanılır. Kullanılması pek tavsiye edilmeyen bir tag'dir, çünkü link'lerin altında da genellikle altçizgi olduğu için altı çizili düz metinle link'lerin birbirine karışma ihtimali vardır. HTML 4.0'da iptal edilmiştir.

<BIG></BIG> ve <SMALL></SMALL> : Yazıyı yeni bir paragraf açmadan geçici olarak büyütmek ve küçültmek için kullanılır.

<BLINK></BLINK> : Netscape Navigator'ın meşhur yanıp sönen tag'i. Bir çok HTML yazarı için suyunu kaçırsa da, efektif bir kullanım hala mümkündür.

<CITE></CITE> : Atıf (citation) yapıldığı zaman kullanılır. Genelde italik olarak render

edilir.

`<CODE></CODE>` : Örnek kod (C, Pascal vs gibi) verildiği zaman kullanılır. Genelde Teletype olarak render edilir.

`<ADDRESS></ADDRESS>` : Adres bölümleri için kullanılır. Genelde italik olarak render edilir.

`<SAMP></SAMP>` : Örnek çıktı (program, script vs) için kullanılır. Genelde Teletype olarak render edilir.

`<ACRONYM></ACRONYM>` : Kısaltmalar için kullanılır. Parametre olarak `TITLE=Kısaltmanın_Açık_Hali` alır. Örneğin, `<ACRONYM title="World Wide Web">WWW</ACRONYM>`.

NOT: Yukarıdaki son 5 tag çifti genelde pek sık kullanılmazsa da kullanılması tavsiye edilmektedir. Dökümanlar içinde arama yapılırken ve bir arama makinesi tarafından endekslenirken yardımcı olabilmektedir. Ayrıca başka benzer tag'ler de mevcuttur.

`<APPLET></APPLET>` : Döküman içine Java applet'i gömmek için kullanılır. Parametre olarak `CODE=Java_Class_Dosyası_URL`'i, `CODEBASE=Class_Dosyasının_Bulunduğu_Dizin`, `ARCHIVE=JAR_Dosyası_URL`'i, `ALT=Alternatif_Metin`, `ALIGN=LEFT|CENTER|RIGHT`, `HEIGHT=Yükseklik`, `WIDTH=Genişlik`, `MAYSCRIPT` (JavaScript'lerle haberleşebileceği) ve `NAME=İsim_Bilgisi` alabilir. Ayrıca applet'in kullanabileceği parametreleri de, istenildiği kadar `PARAM NAME=Applet_Parametresi_İsmi VALUE=Applet_Parametresi_Değeri` kullanarak verebilirsiniz. HTML 4.0'da iptal edilmiştir. HTML 4.0'da şöyle bir kullanım mevcuttur:

```
<OBJECT codetype="application/octet-stream"
classid="java:Bubbles.class"
codebase="http://www.ulakbim.gov.tr/~cu/java/"
width="500" height="500">
Ekрана bir sürü baloncuk çizen irin bir applet
</OBJECT>
```

`<SCRIPT>`: Döküman içine JavaScript, VBScript gibi dillerle yazılmış programcıklar gömmek için kullanılır. Parametre olarak `LANGUAGE=Script_Dili_İsmi` alabilir. Ancak HTML 4.0'da `LANGUAGE` yerine `TYPE` parametresi getirilmiştir. Örneğin `<SCRIPT LANGUAGE="JavaScript 1.1">` yerine `<SCRIPT TYPE=text/javascript>` kullanılmalıdır.

`<META>`: HTML'in en kural tanımayan elemanıdır denilebilir. Parametre olarak bir çok şey alabilir, ancak en çok kullanılan iki tanesi `HTTP-EQUIV` ve `NAME`'dir. Genel olarak bir de `VALUE` ya da `CONTENT` parametresi takip eder. Örnek olarak:

```
<META HTTP-EQUIV="Expires" CONTENT="Tue, 10 Jun 1998 12:30:00 GMT">
<META HTTP-EQUIV="Content-Type"
CONTENT="text/html;CHARSET=iso-8859-9">
<META HTTP-EQUIV="Refresh" CONTENT="5"
URL="http://www.ulakbim.gov.tr/">
<META HTTP-EQUIV="Content-Script-Type" CONTENT="text/javascript">
<META NAME="Generator" CONTENT="Visual Page 1.1 for Windows">
<META NAME="Author" CONTENT="Cüneyt YILMAZ">
<META NAME="Keywords" LANG=tr CONTENT="deneme, learning HTML, dummy
document">
```

Yukarıdan da anlaşılabileceği gibi `HTTP-EQUIV`, HTTP protokolü tarafından desteklenen bazı

parametreleri vermek için kullanılır. Örnek olarak, 1. satırdaki tanımlama, dökümanın geçerlilik tarihinin 10 Haziran 1998’de dolacağını belirtir. Bu tarihten sonra kullanıcının bu dökümanı cache’den yüklemesi engellenir ve tekrar orijinal dökümana gitmesi zorlanır.

<EMBED></EMBED> : Döküman içine genelde sık kullanılan plug-in’lerin desteklediği ses, video, animasyon gibi nesnelerin gömülmesi için kullanılır. Parametreler nesnenin tipine göre değişir. Ancak ortak olan bir parametre SRC=Nesne_URL’idir. HTML 4.0’da iptal edilmiştir. Örneğin bir QuickTime 3.0 movie dosyasını şu şekilde gömebilirsiniz:

```
<EMBED SRC="myMovie.mov" CONTROLLER="FALSE" AUTOPLAY="TRUE"
CACHE="FALSE" WIDTH="240" HEIGHT="180">Plug-in’iniz yok mu? Ne
kötü!</EMBED>
```

HTML 4.0 kurallarına göre yeni gösterim şu şekildedir:

```
<OBJECT DATA="myMovie.mov" TYPE="application/mov">
<PARAM NAME="width" VALUE="240" VALUETYPE="data">
<PARAM NAME="height" VALUE="180" VALUETYPE="data">
<PARAM NAME="controller" VALUE="false" VALUETYPE="data">
<PARAM NAME="autoplay" VALUE="true" VALUETYPE="data">
<PARAM NAME="cache" VALUE="false" VALUETYPE="data">
Plug-in’iniz yok mu? Ne kötü!
</OBJECT>
```

Ayrıca HTML 4.0’da gelen yeniliklerden bir tanesi de DIR (direction, yön) parametresidir. LTR (left to right) veya RTL (right to left) değerini alabilir. Bir çok blok elemanı içinde kullanılabilir. Örneğin <HTML DIR="RTL"> veya <P DIR=LTR> gibi. Ancak hala bir çok browser tarafından desteklenmemektedir.

Tablolar:

Tablolar, HTML’in sayfa mizampajı açısından en çok kullanılan kısımlarındandır. Her ne kadar büyük bir tabloya elle müdahale etmek bir işkence olabilse de, yine de bilgi açısından ele alınacaktır.

Bir tablo genel olarak şuna benzer.

```
<TABLE BORDER="2" CELLPADDING="3" CELLSPACING="1" HEIGHT="200"
WIDTH="90%" BGCOLOR="#A0A0A0">
<TR>
<TD>1. sütun 1. satır</TD>
<TD>2. sütun 1. satır</TD>
</TR>
<TR>
<TD>1. sütun 2. satır</TD>
<TD>2. sütun 2. satır</TD>
</TR>
<TR>
<TD>1. sütun 3. satır</TD>
<TD>2. sütun 3. satır</TD>
</TR>
</TABLE>
```

<TABLE> tag’ini açarken BORDER="2" ile çerçevenin kalınlığının 2 pixel, CELLPADDING="3" ile hücre içindeki yazı ile çerçevesi arasında 3 pixel, CELLSPACING="1" ile hücreler arasında 1’er pixel olacağını, HEIGHT="200" ile yüksekliğinin 200 pixel olacağını, WIDTH="90%" ile

genişliğinin browser penceresinin %90'ı olacağını ve en son olarak da BGCOLOR="A0A0A0" ile arka fon renginin gri olacağını vermiş olduk. Bunlardan hiçbirini vermemiş olsaydık, browser genişlik, yükseklik ayarlarını hücre içlerindeki dataya göre otomatik yapardı ve kalınlık ve arka fon rengini de varsayılanlardan atardı. Burada dikkat edilmesi gereken bir şey, verdiğiniz yükseklik veya genişlik değerleri, hücre içindeki elemanlardan küçükse, verdiğiniz değerler ihmal edilir ve hücre içindeki elemanları *tamı tamına* alabilecek en küçük değerler kullanılır.

Tablo tanımlamaları her zaman için satır satır yapılır. Yukarıda görüldüğü gibi `<TR>` (table row) ile yeni bir satır açıyoruz, ve altına da `<TD>` (table data) ile elemanlarımızı sıralıyoruz. Sütunla işimiz bittiğinde `</TD>` ile, satırla işimiz bittiğinde `</TR>` ile kapatıyoruz. Tabii ki tablomuzla işimiz bittiğinde de `</TABLE>` ile tablomuzu kapatıyoruz.

Peki burada birinci satırda 2 yerine 1 tane geniş sütun vermek isteseydik ne yapacaktık? İlk satır bilgisini şu şekilde değiştirecektik:

```
<TR>
<TD COLSPAN="2">1. satır, ba lık satırı olabilir</TD>
</TR>
```

Aynı şekilde 1. sütunda 3 satır yerine tek bir satır kullanmak için şunu yapacaktık:

```
<TR>
<TD ROWSPAN="3">1. sütun, ba lık sütunu olabilir</TD>
</TR>
```

Aşağıdaki gibi bir tablonun kaynak kodu şu şekildedir.

EN UST				
EN SOL	SOL IC UST		SAG IC UST	EN SAG EN UST
	SOL IC SOL ORTA	SOL IC SAG ORTA		EN SAG ORTA UST
	SOL IC ORTA		SAG IC ALT	EN SAG ORTA ALT
	SOL IC SOL ALT	SOL IC SAG ALT		EN SAG EN ALT

```
<TABLE BORDER="2" CELLPADDING="2" WIDTH="500" HEIGHT="300">
<TR>
<TD HEIGHT="60" VALIGN="MIDDLE" COLSPAN="5"><P ALIGN="CENTER">EN
UST</TD>
</TR>
<TR>
<TD WIDTH="100" VALIGN="MIDDLE" ROWSPAN="4"><P ALIGN="CENTER">EN
SOL</TD>
<TD HEIGHT="60" VALIGN="MIDDLE" COLSPAN="2"><P ALIGN="CENTER">SOL IC
```

```

UST</TD>
<TD WIDTH="100" VALIGN="MIDDLE" ROWSPAN="2"><P ALIGN="CENTER">SAG IC
UST</TD>
<TD WIDTH="100" HEIGHT="60" VALIGN="MIDDLE"><P ALIGN="CENTER">EN SAG
EN UST</TD>
</TR>
<TR>
<TD WIDTH="100" HEIGHT="60" VALIGN="MIDDLE"><P ALIGN="CENTER">SOL IC
SOL ORTA</TD>
<TD WIDTH="100" HEIGHT="60" VALIGN="MIDDLE"><P ALIGN="CENTER">SOL IC
SAG ORTA</TD>
<TD WIDTH="100" HEIGHT="60" VALIGN="MIDDLE"><P ALIGN="CENTER">EN SAG
ORTA UST</TD>
</TR>
<TR>
<TD HEIGHT="60" VALIGN="MIDDLE" COLSPAN="2"><P ALIGN="CENTER">SOL IC
ORTA</TD>
<TD WIDTH="100" VALIGN="MIDDLE" ROWSPAN="2"><P ALIGN="CENTER">SAG IC
ALT</TD>
<TD WIDTH="100" HEIGHT="60" VALIGN="MIDDLE"><P ALIGN="CENTER">EN SAG
ORTA ALT</TD>
</TR>
<TR>
<TD WIDTH="100" HEIGHT="60" VALIGN="MIDDLE"><P ALIGN="CENTER">SOL IC
SOL ALT</TD>
<TD WIDTH="100" HEIGHT="60" VALIGN="MIDDLE"><P ALIGN="CENTER">SOL IC
SAG ALT</TD>
<TD WIDTH="100" HEIGHT="60" VALIGN="MIDDLE"><P ALIGN="CENTER">EN SAG
EN ALT</TD>
</TR>
</TABLE>

```

Yukarıda yeni bir parametre olarak VALIGN (vertical alignment) kullanıldı. VALIGN ile hücre elemanlarının hücrenin yukarısı mı ortası mı yoksa altına mı yaslanacağını belirtebilirsiniz. Sola, sağa yaslamak veya ortalamak için de <P ALIGN=LEFT|CENTER|RIGHT>'ı kullanabilirsiniz.

Tablolarla ilgili iki önemli not:

1. Hücre içinde hiç bir data yoksa, browser o hücrenin çerçevesini göstermez. Örneğin


```

<TABLE BORDER="1" WIDTH="100" BGCOLOR="#C0C0C0">
<TR>
<TD WIDTH="50"></TD>
<TD WIDTH="50">&nbsp;</TD>
</TR>
<TR>
<TD WIDTH="50">&nbsp;</TD>
<TD WIDTH="50"></TD>
</TR>
</TABLE>

```

2. Eğer döküman içinde varsayılan bir font kullanıyorsanız bile tablonun her hücresi içinde bunu tekrar tanımlamanız gerekir, çünkü tablolar içinde dökümanın tanımlanan karakter tipinin üstüne varsayılan karakter tipi konur. Örneğin, döküman font'unuzu Helvetica tanımladıysanız ve browser'ın varsayılan font'u Times New Roman ise tablo içinde bütün yazılar Times New Roman olarak görünür.

Veri Giriş FORMları:

Veri giriş formları genelde bir sunucuya bilgi göndermek veya almak için kullanılır. Çok yaygın olarak sunucu tarafında CGI (Common Gateway Interface) kullanılarak gönderilen bilgi işlenir. CGI programları en çok Perl dilinde yazılmakta, ancak C, C++, Java, Unix shell'leri, Phyton, Dos batch file'leri, ya da Windows exe'leri de olabilmektedir. Ayrıca CGI'a alternatif olarak Servlet (Sun'ın tanımladığı Server-Side Java applet'leri), ASP (Microsoft'un tanımladığı Active Server Page'leri) ya da LiveWire (Netscape'in tanımladığı Server-Side JavaScript) de bulunmaktadır. Bu konular şimdilik bu dökümanın konusu dışındadır.

Internet'te çok sık olarak isminizi, yaşınızı, mesleğinizi vs istedikleri formlar görmüşsünüzdür. Örneğin aşağıdaki formu ele alalım:

Adınız:	asdas
E-mail adresiniz	asdas
Şifreniz:	*****
Kullanım tipi:	Ticari Kişisel
Mesleğiniz:	Öğrenci
Kullandığınız işletim sistemleri:	Linux is the best! Windows NOT Hiroshima 95 Tabii ki Solaris MacOS, almayayım
Düşünceleriniz:	güzel bi ii tabii
Yeni bir sürüm çıktığında e-mail adresime bilgi gönderiniz.	
Gönder Sil-baştan	

Yukarıda ilk iki eleman TEXT-BOX'tır. Sonraki sırada ikisinden yalnızca birini seçebildiğiniz RADIO-BUTTON'lar bulunmakta. Meslek seçilen kontrol ise bir DROP-DOWN COMBO-BOX (!)'tır. Altında birden çok elemanın arasından seçim yapılabilecek bir LIST-BOX var. Düşünceleri sorduğumuz büyük kontrolün adı TEXT-AREA. Altında evet/hayır gibi bir anlamı olan CHECK-BOX'ımız var. En altta ise formu doldurduktan sonra göndermek için kullandığımız SUBMIT ve form'u ilk haline getirmek (içeriği sıfırlamak zorunda değil) için kullandığımız RESET düğmesi var.Şimdi bu elemanlara yakından bakalım:

```
<FORM ACTION="http://www.ulakbim.gov.tr/cgi-bin/anket.cgi" METHOD="post">
  Adınız: <INPUT TYPE="text" NAME="isim" SIZE="25" VALUE="">
  E-mail adresiniz: <INPUT TYPE="text" NAME="email" SIZE="25" VALUE="">
  Şifreniz:<INPUT TYPE="PASSWORD" NAME="pass" SIZE="25" VALUE="">
  Kullanım tipi: <INPUT TYPE="radio" NAME="kultip" VALUE="tica" CHECKED>
                 <INPUT TYPE="radio" NAME="kultip" VALUE="kisi">Kişisel
  Mesleğiniz: <SELECT NAME="meslek">
                <OPTION SELECTED>Öğrenci</OPTION>
                <OPTION>Serbest</OPTION>
                <OPTION>Kamu</OPTION>
              </SELECT>
  Kullandığınız işletim sistemleri:
  <SELECT NAME="os" SIZE="5" MULTIPLE>
    <OPTION SELECTED VALUE=linux>Linux is the best!</OPTION>
    <OPTION SELECTED VALUE=winnt>Windows NOT</OPTION>
    <OPTION VALUE=win95>Hiroshima 95</OPTION>
    <OPTION SELECTED VALUE=solaris>Tabii ki Solaris</OPTION>
```

```
<OPTION VALUE=mac>MacOS, almayayım</OPTION>
</SELECT>
Dü ünceleriniz: <TEXTAREA NAME="TextArea" ROWS="7" COLS="35">güzel bi
<INPUT TYPE="CHECKBOX" NAME="sendmail" CHECKED>
Yeni bir sürüm çıktığında e-mail adresime bilgi gönderiniz.
<INPUT TYPE="SUBMIT" NAME="Submit" VALUE="Gönder">
<INPUT TYPE="RESET" NAME="Reset" VALUE="Sil-ba tan">
</FORM>
```

Şimdi bu elemanları daha da bir yakından inceleyelim :)

1. ve 2. sıradaki Input elemanlarımız Text-Box tipinde, NAME=İsim, TYPE=Eleman_Tipi, SIZE=Görünen_Büyüklik, VALUE=Varsayılan_Değer ve MAXLENGTH=Yazılabilecek_Maksimum_Karakter_Sayısı parametrelerini almakta. Name parametresine sonradan dönülecektir.

3. sıradaki yine bir Input elemanımız var, ancak bu sefer TYPE olarak *PASSWORD* verildiğine dikkat. Bu tip bir Text-Box'ın içine yazdığınız karakterler yerine standart bir karakter, genelde * karakteri (Mac'lerde – bullet) görülür. Parametre olarak Text-Box'lar

4. sırada 2 adet Radio-Button'ımız var. Radio-Button'lar da Input tipi elemanlardan. Parametre olarak NAME=İsim, VALUE=İçsel_İsim ve CHECKED almakta. NAME'leri aynı olan bütün Radio-Button'lar aynı grupta yer alır, ve kullanıcı bunların arasından *sadece bir* tanesini seçebilir. VALUE parametresi de CGI scriptine aktarılan bir parametredir, sonradan dönülecektir. CHECKED parametresi ise, başlangıçta grup içinden hangisinin ön-seçilmiş olduğunu belirtir. Birden fazla kullanılmaması gerekir.

5. ve 6. sırada SELECT elemanımız var. Evet, aslında ikisi de aynı eleman, yalnız görüntüleri değişik. Combo-Box şeklinde görülmesinin tek sebebi, görülen eleman sayısının 1 olarak verilmesi. Parametre olarak NAME=İsim, SIZE=Görünen_Eleman_Sayısı ve MULTIPLE olabilir. MULTIPLE parametresi verdiğiniz zaman kullanıcı birden çok elemanı seçebilir. Combo-Box'larda pek kullanılmaz. <SELECT> ve </SELECT> arasına istediğiniz kadar <OPTION></OPTION> yerleştirebilirsiniz. OPTION parametre olarak VALUE=İçsel_İsim ve SELECTED olabilir. VALUE, CGI script'ine aktarılan bir değerdir. SELECTED ile de önceden hangi liste elemanlarının seçili olacağını söylersiniz. <SELECT> tag'inde MULTIPLE vermemişseniz, birden fazla elemana SELECTED yazmanız tavsiye edilmez.

7. sırada büyükçe bir TEXTAREA'mız var. Parametre olarak NAME=İsim, ROWS=Satır_Sayısı ve COLS=Sütun_Sayısı olabilir. <TEXTAREA> ve </TEXTAREA> arasına yazdığınız herşey, sayfada önceden yazılmış olarak çıkacaktır. Ayrıca, sadece Netscape Navigator'ın desteklediği WRAP=OFF|SOFT|HARD parametresi vardır. Normalde, yazdığınız yazı, uzun bir satır halinde devam eder. WRAP=SOFT yaptığınızda browser içinde satır sonlarında kelimeler bir alt satıra geçer ancak sunucuya yine de tek bir satır halinde gönderilir. WRAP=HARD yaptığınızda hem browser'da hem de sunucuya giden metin satır sonlarından bölünmüş olacaktır. Varsayılan değer OFF'dur ancak WRAP=SOFT iyi bir seçim olacaktır.

8. sırada Input tipinden bir Check-Box'ımız görülüyor. NAME=İsim, VALUE=Değer ve CHECKED parametrelerini alabiliyor.

Son sırada ise birer SUBMIT ve RESET tipinden Input elemanlarımız var. RESET'le formunuzu FORM'un ACTION kısmında belirtilen adrese yollarsınız. Burada iki tip metod vardır: *POST* ve *GET*. GET metodu yeni yüklenecek bir sayfaya parametre olarak gider ve URL'in bir parçasıdır.

POST metodu ise tamamen ayrı bir veri akışı olarak gider ve sonucunda yüklenecek sayfa genelde bu parametrelerden bağımsızdır.

Bir formu doldurup SUBMIT düğmesine bastığınızda, sunucuya giden bilgi, POST veya GET metodundan hangisinin kullanıldığına göre değişir ancak genel görünüm şuna benzer:

```
ismi=Ivır+Zıvır&cinsi=erkek&medhali=müzmin&kafasız=evet
```

= işaretinin sol tarafında NAME, sağ tarafında VALUE bulunur ve birden fazla alanın gönderilmesi durumunda aralarında & vardır. Yukarıda `mailto:` linklerinde gördüğümüz gibi POST metodu kullanılırsa da aralarında + işareti bulunacaktır.

FRAMESET'ler:

Frameset'ler bir browser penceresi içinde birden çok sayfayı aynı anda görüntülemek için kullanılır. Frameset kullanarak, başka türlü yapılması mümkün olmayan bir çok şeyi yapabilirsiniz, çok sık kullanılan bir örnek olarak, yukarıda (veya solda) bulunan ve hareket etmeyen bir düğme grubundan istediğinize basarak, aşağıda (veya sağda) bulunan çerçeve içinde yeni bir sayfaya yönlenebilirsiniz.

Tipik bir frameset tanımlaması şöyledir:

```
<HTML>
<HEAD>
<TITLE>Frameset denemesi</TITLE>
</HEAD>

<FRAMESET COLS="20%,80%" BORDER="0" FRAMESPACING="0">
  <FRAME SRC="sol.html" NAME="sol" SCROLLING="NO" MARGINWIDTH="5" MARGINHEIGHT="10">
  <FRAME SRC="sag.html" NAME="sag" MARGINWIDTH="0" MARGINHEIGHT="0" RESIZABLE="NO">
</FRAMESET>

<NOFRAMES>
  <BODY TEXT="#333333" BGCOLOR="#FFFBF0" LINK="#000099" VLINK="#006600">
    <P>
      Browser'ınız frame'leri desteklemiyor.<BR>
      <A HREF="http://home.netscape.com/download/">Netscape Navigator'ı
    </BODY>
  </NOFRAMES>

</HTML>
```

NOT: Yukarıda BODY kısmı olmasına rağmen bir çok frameset tanımlamasında bu blok hiç yoktur.

<FRAMESET>'le yeni bir frameset'i tanımladık, ama parametresiz hiçbir işe yaramaz. Alabildiği parametreler:

- COLS=x|x%|*|x*, x|x%|*|x*....
- ROWS=x|x%|*|x*, x|x%|*|x*....
- BORDER=x
- FRAMESPACING=x

COLS ve ROWS ile frameset'inizi nasıl böleceğinizi belirlersiniz, ve *ikisinden birini vermeniz zorunludur*. Bölümlemeyi yaparken değerleri pixel cinsinden (x) veya yüzde cinsinden (x%)

verebilirsiniz. * işareti ise özel bir anlam taşır. * verdiğiniz zaman, browser önce x veya x% olarak verdiğiniz değerlere bakar ve bunlara gereken boşluğu ayarlar. Daha sonra kalan boşluk ise * işaretiyle belirtilen yerlere ayrılır. Örneğin, ekranımız 800x600 genişliğinde olsun ve biz de <FRAMESET ROWS=10%, 500, *> kullanmış olalım. Browser en üstteki frame'e 800x10%=80 pixel, bir altındakine 500 pixel ve en alttakine de kalan 800-(80+500)=220 pixel'lik alanı ayırır. * işaretini birden fazla kere kullanabilirsiniz, örneğin aşağıdakilerin hepsi geçerli tanımlamalardır:

```
<FRAMESET ROWS=*, 10%, 500, *>
<FRAMESET ROWS=*, *>
<FRAMESET ROWS=1*, 10%, 500, 10%, 3*>
```

Yukarıdaki 1* ve 3* kullandığımız zaman browser, 2. sıradaki 10%'i, 3. sıradaki 500 pixel'i ve 4. sıradaki 10%'i ayırdıktan sonra kalan boşluğu 4 eşit parçaya ayırıp en üstteki frame'e 1, en alttaki frame'e de 3 birimlik yer ayırır.

BORDER parametresiyle frame'ler arasındaki ayıraç çizgisinin kaç pixel olacağını ve FRAMESPACING'le de aralarında kaç pixel kalacağını belirlersiniz.

<NOFRAMES></NOFRAMES> ikilisiyle de browser'ı frameset'leri desteklemeyen kullanıcılarda görülecek olan metni girebilirsiniz.

NOT: <NOFRAMES></NOFRAMES> ikilisini vermeden BODY bloğu açarsanız, frameset'iniz yerine bu BODY bloğu görülecektir.

Ayrıca, Netscape Navigator'a özel olarak BORDERCOLOR=Ayıraç_Çizgisi_Rengi ve FRAMEBORDER=YES|NO parametrelerini de verebilirsiniz. FRAMEBORDER sanıldığından biraz farklı olarak, aradaki çizginin 3D etkisinin olup olmayacağını verir.

Frameset'ler içiçe gömülebilir, yani bir <FRAMESET></FRAMESET> ikilisi arasında bir tane daha <FRAMESET></FRAMESET> tanımlayabilirsiniz. Örneğin:

```
<FRAMESET ROWS=100, 500, *>
  <FRAME SRC=ENUST.HTML>
  <FRAMESET COLS=150, *>
    <FRAME SRC=TOC.HTML>
    <FRAME SRC=CONTENTS.HTML>
  </FRAMESET>
  <FRAMESET COLS=150, 200, *>
    <FRAME SRC=SOLALT.HTML>
    <FRAME SRC=ORTAALT.HTML>
    <FRAME SRC=SAGALT.HTML>
  </FRAMESET>
</FRAMESET>
```

ENUST.HTML, 100 pixel yükseklik,		
TOC.HTML, 150 pixel genişlik	CONTENTS.HTML, 500 pixel yükseklik ve kalan genişlik	
SOLALT.HTML, 150 pixel genişlik ve kalan yükseklik	ORTAALT.HTML, 200 pixel genişlik ve kalan yükseklik	SAGALT.HTML, kalan genişlik ve kalan yükseklik

Gelelim frame'imizi böldükten sonra içlerine konulacak sayfaları tanımlamaya. Bunu da <FRAME> elemanı ile yapıyoruz. Aldığı parametreler:

- NAME=Frame'in İsmi
- RESIZE|NORESIZE
- SCROLLING=YES|NO|AUTO
- MARGINHEIGHT, MARGINWIDTH=x
- SRC=Frame'in_URL'i

Bu parametrelerden RESIZE veya NORESIZE ile büyüklüğünün değiştirilip değiştirilemeyeceği bilgisini verirsiniz. SCROLLING ile de frame'in içeriğinin bir ekrana sığmaması halinde browser'ın kenara bir kayma çubuğu ekleyip eklemeyeceğini belirtirsiniz, varsayılan AUTO'dur. MARGINHEIGHT ve MARGINWIDTH'leyse frame'lerin kendilerine ayrılan alana yerleştirilirken sol üstten kaçır pixel'lik bir marjın bırakılacağını belirtirsiniz. SRC ile de içine yerleştirilecek dosyanın URL'ini belirtirsiniz, SRC'u belirtmediğiniz zaman browser herhangi bir hata çıkarmayacak, buraya varsayılan renklerde boş bir sayfa koyacaktır. NAME parametresi frame'lerin en önemli parametresi sayılabilir, çünkü frameset'lerde bir frame'in içindeki bir link bir başka frame'in içeriğini değiştirebilir, bunu da NAME parametresini kullanarak yapar. Örneğin tanımlamamız şu olsun:

```
<HTML>
<HEAD>
<TITLE>Frameset denemesi</TITLE>
</HEAD>
<FRAMESET COLS="200,*" BORDER="1" FRAMESPACING="0">
  <FRAME SRC="sol.html" NAME="sol" SCROLLING="NO" NORESIZE>
  <FRAME SRC="sag.html" NAME="sag" MARGINWIDTH="0" MARGINHEIGHT="0" RESIZE>
</FRAMESET>
</HTML>
```

ve *sol.html* içinde şöyle bir link'imiz bulunsun: İçindekiler. Bu link'e tıkladığımızda *sag.html*'in bulunduğu frame'e *table_of_contents.html* yüklenecektir. Hatırlarsanız, frameset'i olmayan sayfalarda bir link'e tıkladığınız zaman, yeni dosya aynı pencere içinde açılırdı.

NOT: Bu parametreyi kullanarak mükemmel şekilde dolaşılabilir siteler

hazırlayabileceğiniz gibi, gittikçe içiçe açılan frame'lerden oluşan bir siteniz de olabilir!

Ayrıca, TARGET olarak verebileceğiniz bazı önceden-tanımlı isimler vardır, bunlar:

- `_top`: tıklanan link'i bütün frame'lerden çıkarak, en üst seviyede açar.
- `_parent`: tıklanan link'i kendi frameset'indeki bütün frame'leri silerek, o seviyede açar.
- `_self`: tıklanan link'i, link'in bulunduğu frame'in içinde açar.
- `_blank`: tıklanan link'i yeni boş bir pencerede açar.

NOT: TARGET parametresini verdiğiniz bir link'in TARGET'ı gerçekte yoksa, etkisi `_blank` ile aynı olacaktır, yani link yeni bir pencerede açılacaktır.

Relative (göreceli) ve Absolute (mutlak) Adreslemelerin Farkları:

Şu ana kadar verilen link'lerde kimi zaman relative (ör. ../index.html), kimi zaman da absolute (ör. http://www.ulakbim.gov.tr/cgi-bin/anket.cgi) adresleme kullandık Her ikisinin de hem avantajı hem de dezavantajları vardır:

- Relative adreslenerek oluşturulan sayfaların bir makineden diğerine taşınması kolaydır, ancak bir üst seviyedeki sayfa taşındığı zaman dağılırlar.
- Absolute adreslemede yazılan linkler daha uzundur, ve bakımı daha zordur ancak bazı dosyaların mutlaka sunucudan yüklenmesini istiyorsanız (bir kullanıcının sayfanın bir kopyasını kendi makinesine indirmesi durumunda) kesin sonuç verir.

İkisi arasındaki farkları bilerek hazırlanan bir sitede normalde relative ve absolute adresleme beraber kullanılır, ve relative adreslerin sayısı genelde daha çok olur. İyi bir sitede yapılabilecek bir düzenleme, bütünlüğünün bozulmaması gereken "blok" sayfalarda, relative adresleme kullanmak, ve her yüklendiğinde değişen sayfalara da absolute link vermektir, örneğin saatbaşı değişen istatistikler gibi.

Bütün notlar Cüneyt YILMAZ tarafından hazırlanmıştır. e-mail: cu@reborn.com