

HTML Cambazlıkları

Dr . Hakki ÖCAL

İçindekiler:

İÇİNDEKİLER:	2
WEBCİNİN TAKIM ÇANTASI	4
HTML'İ İYİ BİLİYOR MUSUNUZ?	6
DURAGAN HTML'İN İNCELİKLERİ	6
HTML DEYİP GEÇMEYİN!	10
DİNAMİK HTML'E HOŞGELDİNİZ!	14
BİR CSS ÖRNEĞİ	17
ZİYARETÇİNİN BROWSER'INI KOKLAYABİLİR MİSİNİZ?	22
ŞUNU NASIL YAPARIM?	33
Büyük Birinci Harf (Gömme):	34
Dikey Çizgi	35
Beyaz Boşluklar	36
Arkaplanda Yazı	36
İlk sayfa bir kaç saniye görünsün, kaybolsun!	37
Stil düzenlemesi yapmadan harf belirlemek için	38
Türkçenizi kaybetmemek için	38
Resimleri Bloklayın!	38
Noktasız Liste	39
HTML'e ENTER ve TAB girdilerini kabul ettirmek mümkün mü?	39
Form unsurlarını hizalamak istiyorsanız	39
E-Mektup bağlantısı	39
Geri Düğmesi	41
Pencereyi kapatın	41
Çerçeveye esir düşünce	41
Bağlantılarınıza Tool Tip ekleyin	42
Sayfanızı Sık Kullanılanlar'a eklettirin:	43
Geriye doğru sayım	43
GeoCities'in yüzen grafiğini kiskanıyor musunuz?	44
CSS Eğlenceleri	47
Maviden gına geldi ise	48
Yazıcıya göre sayfa kesme	49
Sayfada tarih	50
Javascript ile açılan menü	52
Tek pencere yetmiyorsa	53
Kaynağı kolayca görüntülemek	54

Ne sihir.. Ne keramet.. Web'de iki, haydi bilemediniz 12 satırlık komutla, mucizeler yaratmanın ne sihirle, ne kerametle ilişkisi var. Herşey, marifete dayanıyor. Bu kez marifetin adı, HTML püfleri ve DHTML.

Televizyonda en çok izlenen belgesellerden biri, film sanayiinin teknik hileleridir. Sinemada teknik hile, olmayanı varmış, ya da başka bir şeymiş gibi göstermek olduğu halde, Web'de "hile" olamaz. Web'de sihir ve keramet, sadece ziyaretçinin ekranında ancak video montaj yoluyla elde edilebilecek bir etkiyi, HTML kodları ile oluşturmayı amaçlar. Bu konuyu böylesine derinlemesine ele almamızın sebeplerinden biri sinema hilelerine meraklı sinema seyircilerine Hollywood tekniklerinin içyüzünü açıklayan belgesellerin yapım sebebiyle aynı: "Bakın ekranınızdaki bu inanılmaz etkiler aslında nasıl kolay sağlanıyor, görün!" demek. İkinci ve asıl amacımız ise, son altı ayda ülkemizde tanık olduğumuz Web sitesi sayısındaki patlamaya paralel olarak, tekniği zayıf Web sayfası sayısındaki yürekler acısı duruma engel olmak. Oradan buradan derlenmiş ve bu arada en hayatî satırları atılmış Javascriptler; parametreleri eksik Java Applet'ler; yukarıdan aşağı, mesela "HelveticaTürk" harfine atıf yapan, fakat bu harf türünün sadece kendi bilgisayarlarında olabileceğini hesaba bile katmayan Webmaster'ların incileri! Kısaca, Webciliğimizin biraz sihire, biraz keramete ve çok ölçüde marifete ihtiyacı var.

WEBCİNİN TAKIM ÇANTASI

Her sanat erbabının bir takım çantası ve içinde sanatın gerektirdiği alet edevat bulunur. Başka bir deyişle alet işler, el övünür. Bir Webcinin bilgisayarında ise bazı programların olması gerekir. Hele Web'de cambazlık ölçüsünde ustalaşmak istiyorsanız, bu programlar olmadan yola çıkamazsınız. İşte size gerekli alet-edevat:

Macromedia Dreamweaver

(www.macromedia.com)

Cascading Stylesheet yazmak ve Layer (katman) teknolojisinden yararlanmak için büyük kolaylıklar sağlıyor. Özellikle katmanlara (DIV, SPAN, LAYER VE ILAYER) hareket kazandırmak için gerekli JavaScript kodlarının yazılmasında, JavaScript dilini çok iyi bilmeyen tasarımcılar için eşi bulunmaz bir hazine.

GoLive CyberStudio

(www.golive.com)

Dreamweaver'ın Windows ortamında yaptığını ve daha fazlasını Macintosh ortamı için yapabilen bu program, tasarımı Mac'de yapanlar için gerekli bir araç. Sadece WYSIWYG görünümlü bir HTML editörü değil, fakat aynı zamanda son derece becerikli bir JavaScript yazarı.

Photoshop

(www.adobe.com)

Web'e veya kağıda yönelik bütün grafik işleriniz ve özellikle bitmap tabanlı resim, fotoğraf ve diğer grafikler için, bu amaçla yazılmış programların Rolls Royce'u. O kadar çok

rakibi olmasına rağmen, yıllardır bilgisayarda fotoğraf ve resim işlem yazılımlarının adeta standardı olan Photoshop ile, bugün ve yarın her türlü grafik sorunuza halledebilirsiniz.

HTML Editörü

HTML kodu yazmak için, aslında bir düz yazı programı yeter. Fakat bunun için bütün HTML kodlarını bilmeniz ve çok dikkatli olmanız gerekir. Özellikle açılan ve kapanan HTML kodlarında kapatma etiketini unutmak mümkündür. Microsoft FrontPage, Netobjects, Corel Web Suite, Adobe PageMill, ve daha niceleri, kelime-işlem programı kolaylığıyla ve HTML kodlarını bilmeden HTML yazmanızı sağlayabilir. Fakat bu programları, Web tasarımında ana araç olarak kullanmak, tasarımcıyı HTML'in inceliklerini öğrenmekten alıkoydukları ve özellikle DHTML alanında kendi bildiklerini tarzdan başkasına izin vermedikleri için tavsiye edilmez. Yine bu programlar gibi WYSIWYG tarzı, fakat kendi bildiğinde ısrar etmeyen, açtığınız HTML kodunu kapatan etiketi kendiliğinden yazan, bütün HTML etiketlerini gruplar halinde ekranın bir kenarında sunarak, hatırlatan HTML editörlerinden birini seçebilirsiniz. Paul Lutus'un Arachnophilia adlı editörü başta olmak üzere, bu işi yapabilecek çok program bulabilirsiniz.

HTML Kılavuzu

Başarılı her Web tasarımcısının masasının üzerinde bir HTML 4 kılavuzu bulunur. Bu kılavuzu edinmek çok kolay: Web Browser'ınızla "<http://www.w3.org/TR/REC-html40/struct/global.html#h-7.4.4.2>" adresine gidin ve görüntülenen belgeyi sabit diskinize kaydedin. İşte sizi HTML'in yerini XHTML dili alıncaya kadar idare edecek kılavuza sahip oldunuz! Bu belgenin İngilizce olması, HTML'e aşina bir tasarımcı için sorun olmamalıdır; çünkü bu belgede sayısız örnek yer alıyor.

HTML'İ İYİ BİLİYOR MUSUNUZ?

Web'de usta işi siteler ve sayfalar yapmak için, herşeyden önce HTML'in bütün inceliklerine hakim olmanız gerekiyor. Sonra, buna Dinamik HTML unsarlarını eklemeniz şart. Web sayfalarında gerçekten usta işi denilebilecek uygulamalar, doğru ve yerinde kullanılan HTML komutlarına, Javascript veya VBScript dilleriyle yazılmış programlar ve HTML unsurlarına *Cascading Stylesheets* denen stil komutları eklemekle mümkündür. Bir sirkte, tel üzerinde tek tekerlekli bisiklet süren ve elindeki uzun değneyi bir sağa bir sola oynatarak dengesini bulan bir cambaz düşünün. Benzetmek gibi olmasın, ama Web'de cambazlık yapacaksanız, üzerinde yürüdüğünüz HTML telini, dengenizi sağlayacak Script çubuğunu ve üzerine çıkacağınız Stil bisikletini iyi tanımanız gerekir. İşte, HTML'in inceliklerinden başlayabiliriz.

DURAGAN HTML'İN İNCELİKLERİ

Web'in ortak dili HTML, dördüncü sürümüne ulaşmış bulunuyor. *World Wide Web Consortium* (www.w3c.org) tarafından düzenlenen bir ortak çabanın ürünü olan HTML'in ziyaretçilerin birbirinden farklı Browser programları tarafından aynı şekilde yorumlanması ve tasarımcının niyet ettiği gibi görüntülenmesi için, W3C'nin koyduğu ilkelere harfi harfine riayet etmeniz şart.

Uyulması gereken temel ilkelerin başında ise, <TITLE>, <HEAD> ve <BODY> etiketlerinin sırası ve içlerine nelerin konabileceği konusundaki kurallar geliyor. Bir HTML dosyasının HEAD bölümünde yer alması gereken etiketler, bu etiketin dışına konursa, istenmeyen iki durum meydana gelir: Ziyaretçinin Browser'ı sayfayı sizin istediğiniz gibi yorumlayamaz; sayfanız Internet'in endeksini çıkartan belli başlı ara-bul firmaları

tarafından doğru tasnif edilemez ve arandığında herhangi bir ihtiyaç sahibi tarafından arandığında bulunamaz.

Başarılı bir Web sayfası oluşturmak istiyorsanız, <HEAD> etiketinin içinde, şu unsurlara yer vermelisiniz:

BASE: İçinde yer aldığı sayfanın mutlak veya görel URL'ini belirtir ve çok sayfalı Web sitelerine Browser açısından istikrar kazandırır. Browser'ların Geri düğmesi, bir sayfadan önceki sayfaya gidebilmek için buradaki bilgiyi kullanır. İçinde üç özellik unsuru (*attribute*) yer alır:

HREF: Sayfanın URL'i belirtir.

TARGET: Frameset etiketi ile birlikte kullanıldığında, çerçevelere isim verilmiş sayfanın yeralacağı çerçeveyi belirtir.

FRAMESET (Sayfanın yükleneceği pencere Target'ta belirtilmemişse, ya isim verilmiş bir çerçeveye ya da Browser'ın varsaydığı , *_blank*, *_parent*, *_self* veya *_top* çerçeveye yükleneceğini belirtir. Bunlar, sırasıyla, yeni ve adlandırılmamış bir Browser penceresini, mevcut sayfayı açtıran sayfanın çerçevesini, mevcut sayfanın kendi çerçesini ve çerçevesi de olsa mevcut Browser penceresinin tümünü kasteder.

META: Bir Web sayfasındaki bilgiler hakkındaki bilgileri (meta/üst bilgileri) içerir.

İki unsuru sık kullanılır:

NAME: Sayfanın yazarı, sayfanın yayın tarihi, vb., bilgiler içerir. <META NAME="bilginin türü" CONTENT="bilgi"> şeklinde yazılır.

HTTP-EQUIV: İçinde yer aldığı sayfanın, Web Server tarafından ziyaretçiye gönderilmesinde oluşturulacak Response Header (Karşılık Başlığı) bölümünde yer alacak bilgiler içerir. <META HTTP-EQUIV="bilginin türü" CONTENT="bilgi"> şeklinde yazılır.

Sık Kullanılan Meta Bilgilerinden örnekler:

```
<META NAME="keywords" CONTENT="kelime1 kelime2 kelime3">
```

"keywords" Internet'teki belli başlı ara-bul kurumlarının siteleri araştırır ve içeriklerine göre gruplandırırken, sınıflandırmada kullanmaları istenen kilit kelimeyi belirtir. CONTENT hanesine sitenizin hangi kelimelerle arandığında bulunmasını istiyorsanız, o kelimeleri yazın. Söz gelimi, Yahoo'da "Türkçe MP3" kelimeleriyle arama yapan birisinin sizin sitenizi bulmasını istiyorsanız, bu etiketi şöyle yazmalısınız:

```
<META NAME="keywords" CONTENT="Türkçe Turkçe türkçe turkçe, Türk  
Turk türk turk mp3 MP3 MPEG mpeg müzik muzik music">
```

```
<META NAME="description" CONTENT="metin">
```

Bazı ara-bul firmaları "description" etiketini tanır ve karşılığı olan metni, sitenin tanımlaması olarak arama yapan kişiye bildirir. Bu etiketi tanımayan ara-bul firmaları ise sayfanın ilk bir kaç kelimesini tanımlama aracı olarak kullanır. Örnek:

```
<META NAME="description" CONTENT="Telif hakkı yasası yakamıza  
yapışincaya kadar Türkçe MP3 müzik dosyalarını bu sitede  
bulabilirsiniz.">
```

```
<META NAME="author" CONTENT="Adı Soyadı">
```


Bazı ara-bul firmaları arama yapanlara siteleri sunarken sitenin yazarını "author" etiketini okumak suretiyle tanır ve karşılığı olan metni, arama yapan kişiye bildirir. Örnek:

```
<META NAME="author" CONTENT="Hakkı Öcal">
```

```
<META HTTP-EQUIV="expires" CONTENT="Wed, 04 Apr 1999 23:59:59 GMT">
```

Bu sayfanın 4 Nisan günü Griniç saatiyle gece yarısından sonra her türlü ara-bul firmasının endeksinden (ve ziyaretçilerin bilgisayarlarında geçici Internet dosyaları deposundan) silinmesini sağlar.

```
<META HTTP-EQUIV="refresh" CONTENT="5; URL=http://sitem.com/iki.htm">
```

Bu sayfanın 5 saniye sonra Browser penceresinden silinmesini ve yerine sitem.com adresindeki "iki.htm" belgesinin görüntülenmesini sağlar.

SCRIPT: Web sayfalarına, Javascript ile VBScript dilleri, HTML'in kendi orijinal komutları arasında bulunmayan işlevler kazandırır. Javascript ile Java programlama dili, adlarındaki benzerliğe rağmen uzaktan-yakından ilgili değildir; Javascript gerçekte C dilinden türetilmiştir. VBScript ise Microsoft'un Uygulamalar İçin Visual Basic, VBA, dilinden türetilmiştir. Bir Web sayfasında kullanılacak Script dilinin tipi ve kaynağı, HEAD bölümünde deklare edilmek suretiyle, bütün HTML'de bir daha tip ve kaynak belirtilmeden kullanılabilir. Ancak tasarımcı isterse, HEAD bölümü dışında bir yerde mahallî Script deklarasyonunda da bulunabilir. HEAD bölümünde Script dili kullanılarak oluşturulacak fonksiyonlara, sayfanın ileri bölümlerinde isimleriyle atıfta bulunmak mümkündür.

STYLE: Web Browser programlarının HTML'in ilk günlerinden kalma 18 metin biçimlerdirme, 12 metin sunuş etiketi vardır. Buna yapısal belge etiketlerinden altı başlık

etiketini de eklerseniz, bir sayfada herhangi bir metni biçimlendirmek için 26 adet önceden-tanımlanmış şekil kullanabilirsiniz.

HTML DEYİP GEÇMEYİN!

Web cambazlığına özenenlerin çoğu, HTML'in yorumu Browser'lara bırakılmış varsayılan biçimlendirme etiketleri ile çok şey yapılamayacağı şeklinde yanlış bir kanısı vardır. İşte bu etiketlerden bazıları ile oluşturabileceğiniz etkilerden bir kaç örnek:

<ACRONYM>..**<ACRONYM>**: İçindeki metnin kısaltma olduğunu gösterir ve ziyaretçi Mouse işaretçisini bu kelimenin üzerinde tutarsa, kelimenin açık şeklini gösterir. Bu etiket, Windows veya Macintosh'un yerel yardım ve ipucu kutuları gibi, bilgi kutuları oluşturmakta kullanılabilir:

```
<p>Bu konuda, <ACRONYM TITLE="PC WORLD  
TURKIYE dergisi, Mart 1999, sayfa  
98">PCW</ACRONYM> çok güzel şeyler  
yazmıştı.</p>
```

Bu konuda, PCW çok güzel şeyler yazmıştı.
PC WORLD TURKIYE dergisi, Mart 1999, sayfa 98

<CITE>..**</CITE>**

İçindeki metnin bir yerden alıntı olduğunu gösterir:

```
<p>Bu konuda,<CITE>HTML Tasarım Rehberi</CITE>  
kitabına bakabilirsiniz.</p>
```

Bu konuda, *HTML Tasarım Rehberi* kitabına bakabilirsiniz.

`<CODE>..</CODE>` İçindeki metnin bilgisayar programı veya benzeri bir Script parçası olduğunu gösterir:

```
<p>Bu paragrafın stili şöyle  
olmalıdır:</p><p><CODE>.shadow {<br>font-size:  
16pt;<br>color: #000000;<br>font-family:  
"Arial";<br>}<br></CODE></p><p>Daha sonra  
normal paragrafa dönebilirsiniz.</p>
```

Bu paragrafın stili şöyle olmalıdır:

```
.shadow {  
font-size: 16pt;  
color: #000000;  
font-family: "Arial";  
}
```

Daha sonra normal paragrafa dönebilirsiniz..

`_{..}` İçindeki metni, daha küçük harfle ve normal satırın altına indirerek, *subscript* şeklinde yazdırır. Özellikle bilimsel formül yazarken kullanılabilir:

```
<BODY>Dolayısıyla, suyun formülü,  
H<SUB>2</SUB>O şeklinde yazılır.</BODY>
```

<SUP>..
</SUP>

İçindeki metni, daha küçük harfle ve normal satırın üzerine çıkartarak, *superscript* şeklinde yazdırır. Özellikle bazı isimlerde kullanılabilir:

```
<BODY>Romanın kahramanının adı Mike  
M<SUP>c</SUB>Cormick idi.</BODY>
```

<DL>, <DT> ve <DD>

Özellikle tanımlama ve benzeri liste türü açıklamalarda, tanımlanan kelime ile tanımlama paragrafını, kelime-işlem veya masaüstü yayıncılık programı hassasiyetiyle, içerden başlatmakta bu iki etiket birlikte kullanılabilir:

```
<body><DL COMPACT><DT>DHTML<DD>Kelime anlamı  
Dinamik Hyper Text İşaretleme Dili olup, Web  
sayfalarına dinamik nitelik kazandıran  
teknolojiler bütünüdür.<DT>ASP<DD>Kelime anlamı  
Aktif Server Sayfaları olup, Microsoft'un  
Dinamik Web sitesi yapımına getirdiği yeni  
teknolojileri içerir.</DL></body>:
```

DHTML

Kelime anlamı Dinamik Hyper Text İşaretleme Dili olup, Web sayfalarına dinamik nitelik kazandıran teknolojiler bütünüdür.

ASP Kelime anlamı Aktif Server Sayfaları olup, Microsoft'un Dinamik Web sitesi yapımına getirdiği yeni teknolojileri içerir,

Ne var ki, bu görüntülenme tarzı önceden belirlenmiş biçim etiketleri, bu tarzın her zaman tasarımcının elde etmek istediği etkiye uygun olmaması ve HTML etiketlerinin bazen bir Browser'dan diğerine farklı şekil göstermesi çoğu zaman tasarımcıya "*Ah, şu metne şu şu özellikleri verecek bir etiket olsa!*" dedirtir. İşte bu gibi durumlarda tasarımcının imdadına koşmak amacıyla icad edilmiş birinci araç Script ise ikinci araç Style etiketidir.

Bir HTML belgesi, üç ayrı tarzda stil komutu kabul eder ve bunları tabir yerinde ise üstüste yığarak, en son talimatı yerine getirir. Bu sebeple, HTML'in bir uzantısı olan stil belirleme yoluyla belge biçimlendirme tekniğini, *Cascading Style Sheets*, denir. *Cascading*, "birbirinin üzerine yığılma," *Style Sheets* ise stil komutlarının içinde yer aldığı talimat belgesi anlamına gelir.

Bir Web sayfasına ayrı bir belge halindeki stil kağıdını ekleyebilirsiniz; veya Web sayfasının HEAD bölümünde belgenin tümü için geçerli stil komutları koyarsınız; ya da en zor yola giderek, her bir paragrafı bizzat kendi <P> etiketinin içindeki stil komutları ile biçimlendirirsiniz. Dışarıdan bir stil kağıdı ekleme yöntemi, sadece bu kağıttaki komutları

değiştirerek, bir sitenin (bu kağıt iştirilmiş) bütün sayfalarındaki biçimlendirmeleri değiştirmenize imkan sağlar. HEAD bölümünde yapacağınız stil değişikliği ise o belgedeki bütün biçim etiketlerini etkiler. Bir paragrafın kendi biçim komutları ise o paragraf bittiği anda hükmünü kaybeder. İyi bir Web tasarım tekniği, bütün siteyi bir Style Sheet'e bağlamak ve stil değişikliklerini bu belge üzerinde yapmaktır. Bu sitenin bütün sayfalarına tutarlılık kazandırır. Böyle bir sitede öyle bir sayfa olur ki, sitenin tümüne biçim veren Stil Kağıdı yerine, yeni HEAD bölümündeki stil komutları ile biçimlendirilmesi gerekir. Bu durumda, sayfa, dışarıdaki stil kağıdına bağlı bile olsa, kendi HEAD bölümündeki stil kuralları geçerli olur. Fakat bir sayfada öyle bir paragraf olur ki, sitenin tümünü biçimlendiren veya o sayfayı şekillendiren stiller yerine kendi özel stiline ihtiyaç duyabilir. Bu durumda o paragrafın stilini, kendi <P> etiketini biçimlendirerek belirlemek en kolay yol olur. Bu durumda ne sitenin stil kağıdı, ne de sayfanın stil bölümü geçerli olur; o paragraf stilini kendi yerel stil komutundan alır.

DİNAMİK HTML'E HOŞGELDİNİZ!

Gerçekte dinamik HTML diye bir şey olmadığını söylersek, Web tasarımcısı olarak hayalleriniz yıkılmaz, herhalde! Dinamik HTML başlığı altında ele alabileceğimiz herşeyi, yukarıda "Duragan HTML" unsurları arasında gördük bile: Script ve CSS.

Script, Browserların 100'e yakın kelime içeren HTML diline, yeni kelimeler kazandıran bir ek sözlük gibidir. Javascip veya VBScript, programlama dili olarak, Browser ve onun vasıtasıyla Windows, Unix veya MacOS tarafından yorumlanarak anlam kazanırlar. "Yorumlanan" dil, işletim sistemine bir yorumlayıcı eklenmesini gerektirir. Mesela C dili ile yazdığınız bir programın yorumlanması gerekmez; çünkü bu program, işletim sisteminin

istediđi ve anladığı bütün komutları bilir. Oysa Javascript veya VBScript komutunun, Browser'a bir iş yaptırtabilmesi için önce tercüme edilmesi, yorumlanması gerekir. Bu zorunluk gerçeri Script dili ile yazılmış "program" veya "programcıđın" yavaşlamasına yol açar, ama Script dilleri genellikle türetildikleri programlama dilinden daha basittir ve daha kolay öğrenilir.

Web'de gerçekten usta bir tasarımcı olmaya niyet eden kişinin, HTML'den hemen sonra, belli başlı iki Browser'ın da ortak Script dili olan Javascript'i derhal öğrenmesi gerekir. VBScript öğrenmek, aktif ve ziyaretçi ile etkileşmeli yeni bir Server teknolojisi olan ASP (*Active Server Pages*) öğrenmeye ve uygulamaya kararlı kişiler için daha çok kazançlı olacaktır; ancak Netscape bugünkü haliyle VBScript komutlarını tanımamaktadır. Sitesini her iki Browser'ı da kullanan ziyaretçileri dikkate alarak tasarlayan bir Web ustasının, Browser için Javascript, Server için VBScript öğrenmekten başka çaresi yoktur.

"Öğrenmek" dediğimiz zaman, bundan, bilgisayar okullarında öğretildiđi tarzda aylar süren bir formel eğitimle bir programlama dilini öğrenmeyi kastetmiyoruz. Çođu zaman bu iki dilin de nerede ve nasıl kullanıldığını ve bu dilin komutlarının sentaksını (cümle kuruluşunu) öğrenmek yeterlidir. İnternet'te istemediğiniz kadar çok bulacađınız Script'leri, kendi sitenize uyarlarken nelere dikkat edeceđinizi bilmeniz gerekir. Bir yerde unutulmuş bir tırnak, eşittir yerine iki nokta üstüste, ve benzeri mini-mini hatalar yüzünden, bir Script beklediğiniz sonucu vermeyebilir. Bir Web tasarımcısının Script dili öğrenirken amacı, o dili kullanarak, etkileşmeli oyun programı yazmaktan çok, sitesine dinamik unsurlar kazandırmak için yeterli düzeyde, en temel komutları ve bunların yazım tekniđini öğrenmekten ibarettir.

Dinamik HTML tekniğine dayanan Web sayfası ile yapılabilecek işlerin zirve noktalarından birini *www.htmlguru.com* adresinde görebilirsiniz. Sayfalardaki hemen her unsurun görsel olduğunu, buna karşılık metinle yapılabilecek hiç bir işlemin grafiğe yüklenmediği ve dolayısıyla sayfaların siteden ziyaretçinin bilgisayarına aktarılması için gerekli zamanı asgaride tutan, yine de ziyaretçinin ekranına yansıyacak herşeyi ziyaretçinin talebine bırakan bir dinamizme sahip bir sayfa on adımda elde edilebilir. Buna isterseniz, DHTML'in anayasası da diyebilirsiniz:

1. Kağıt üzerinde hazırlık: Fikrinizi önce kağıda dökün; reklam ajanslarının reklam filmi yaptığı gibi bir senaryo şeması çıkartın. Bu, birinci sayfadan sonuncuya kadar, sitenizdeki bütün sayfaların ekran görüntüsünü kağıt üzerinde resmetmek demektir.

2. Teknikleri belirleme: Hayalinizdeki sitenin oluşumu için kullanılacak teknikleri belirleyin; zayıf olduğunuz tekniklerde kimden yardım isteyeceğinizi düşünün. Günümüzün Web teknolojisi, grafikçinin programcı, metin yazarının grafikçi olmasını gerektiriyor ve kolaylaştırıyor. Hayatında bir suluboya resim yapmamış kişi bile bir grafik programı ve telif hakkından vazgeçilmiş bir kaç grafikte yola çıkabilir ve ortaya en görkemli Web sitelerinde kullanılacak kalitede grafikler çıkartabilir. Bu tür bir el yatkınlığını elde etmek için, bol bol alıştırmak yapmak gerekir. Alıştırmanın zamanı ise Web sitesinin çatısını çözmeye başlamadan öncedir.

3. Donanım ve yazılım denetimi: Hayalinizdeki sitenin oluşturulması için gerekli grafik programları, HTML editörü, site yönetim programı, kelime-işlem programı bilgisayarınızda yüklü mü? Bilgisayarınız bu programların hepsini aynı anda çalıştıracak (ve çökmeyecek) RAM, sanal bellek, sabit disk ve grafik kartı ve ekrana sahip mi?

4. Grafik hazırlama: Kullanacağınız grafik unsurları hazırlayın. Zemin grafiği, ziyaretçilerinizin sayfalarınız arasında gidip gelmesini sağlayacak Navigation (seyir) düğmeleri olarak kullanılacak grafikler, içeriğin gerektirdiği grafikler, başlıklar, süsleme unsuru olarak kullanılacak resimler, sabit diskte, belirli bir yerde toplanmış durumda mı? Bu iş söylendiği kadar kolay değildir. Birinci maddede çizdiğiniz sayfa görüntülerinin gerektirdiği bütün unsurları biraraya getirip, telif hakları ile ilgili sorunları halletmek zaman alır. Ayrıca bütün grafiklerinizde aynı tür isimlendirme tekniğini uygulayın: ya hepsinin adı büyük harfle, ya da hepsinin adı küçük harfle yazılsın. Günün birinde dinamik sayfanızda zemin grafiği yerine kırmızı çarpı işareti görmek istemiyorsanız, sitenizin duracağı Server'ın dosya adları konusunda büyük harf-küçük harf ayırt edip etmediğini şimdiden bilmeniz gerekir.

5. Stil belirleme: Sitenizde uygulanacak metin, başlık ve diğer biçimlendirme stillerini belirleyin: Başlıklarınız arasında kavram sıralaması yapın ve yukarıdan aşağı doğru, bütün unsurlarınıza nasıl şekil vereceğini tayin edin. Bunların gerektirdiği stilleri, bir CSS dosyası olarak yazın. CSS dosyası, düz yazı dosyasıdır ve herhangi bir düz yazı programı ile oluşturulabilir. CSS yöntemiyle, HTML'in temel etiketlerinin stillerini değiştirebileceğiniz gibi, kendi vereceğiniz adla kendi stillerinizi oluşturabilirsiniz.

BİR CSS ÖRNEĞİ

Düz yazı dosyası ile oluşturabileceğiniz bir Cascading Style Sheets örneği:

/*

CSS Dosyası

*/

```
#turkuaz{
    BackGround : AquaMarine ! important;
    Font : 100%;
}

.Koyumavi{
    BackGround : Beige ! important;
    Color : DarkBlue ! important;
    Font : bold xx-large/1 Sans-Serif, Arial ! important;
}

.Mavi{
    Color : #0000FF ! important;
    Font : inherit inherit/inherit Sans-Serif ! important;
}

body, p{
    Color : #008080 ! important;
    Font : normal normal 100%/1 Sans-Serif, Arial ! important;
}

H1{
    Color : #800080 ! important;
    Font : oblique xx-large/1 Serif, 'Times New Roman' ! important;
    Text-Align : center ! important;
    Text-Decoration : underline ! important;
}
```

Web dilinde stili belirlenebilen etiketlere *Selector* denir. Bu CSS dosyasında, bir ID (kimlikli) türü, iki Class (sınıf) türü Selector tanımlanıyor; iki HTML Selector'ü, varsayılan değerleri yerine yeni biçimlere kavuşturuluyor.

ID türü Selector'ler, adlarının başındaki # işaretinden tanınırlar ve HTML belgesinde kendilerine kimlikleriyle atıfta bulunularak, herhangi bir etikete biçim vermekte kullanılırlar. Örneğin, buradaki "#turkuaz" adlı ID Selector'ünü bir katmanın içindeki yazıyı biçimlemek üzere kullanmak isterseniz, HTML kodunuz şöyle olacaktır:

```
<BODY>
```

```
<P>HTML programcılığının temel ilkesi:</P>
```

```
<SPAN ID=turkuaz">Hiç yılmadan çalışmak ve öğrenmek; sonra herşeye yeniden başlamak</SPAN>
```

```
<P>Bu ilkeyi benimseyen, sadece başarılı HTML programcısı değil, her işte başarılı bir bir kişi olur</P>
```

Buradaki katman etiketi SPAN'ın içindeki metni biçimlendiren, "turkuaz" adlı Selector başka hiç bir şey yapmayıp sadece biçimlendirdiği unsura aquamarin arkaplan rengi verecektir.

"Class" sayılan Selector'leri, mevcut herhangi bir HTML biçim etiketi ile birlikte kullanılabilir: p.mavi, H2.koyumavi gibi. "Mavi" sınıfı Selector ile birleştirilen HTML Selector'ü, "mavi" sınıfının özelliklerini alır. Buradaki örnekte ".mavi" Selectörünün harf rengini mavi yapmaktan başka bir özelliği yok: diğer bütün stil özelliklerini "inherit" ediyor, yani bağlandığı ve değiştirdiği HTML Selector'ünden miras alıyor. Burada gördüğümüz "Important" ifadesi ise Browser'a, ziyaretçi kendi bilgisayarında Browser ayarlarını değiştirmiş bile olsa, bilgisayar sahibinin değil, bizim verdiğimiz değerlerin geçerli olması gerektiğini bildiriyor.

CSS yoluyla, HTML'in kendi stil belirleyebilen etiketlerini, veya teknik adıyla HTML Selector'lerini de yeniden biçimleyebiliriz. Normal olarak HTML'in H1 başlığı sağa yatık bir harfle görüntülenmez; ancak "stillerim.css" dosyasını bağladığınız her Web sayfasında H1 başlığı, italik, çok büyük ve Times Roman olacaktır.

Stil kağıdı, herhangi bir isimle kaydedilebilir; fakat dosya adının uzatması, tercihan ".css" olmalıdır. Bir CSS dosyası, HTML sayfasına HEAD etiketi içinde LINK etiketiyle bağlanır. Diyelim ki bu dosyayı "stillerim.css" adıyla kaydettiniz. Bu dosyayı HTML sayfasında şöyle bağlamanız gerekir:

```
<HTML>
<HEAD>
<TITLE>Stil kağıdı bağlama örneği</TITLE>
<SCRIPT LANGUAGE="Javascript">.....</SCRIPT>
<LINK REL=stylesheet TYPE="text/css" REF="stillerim.css">
</HEAD>
.....
```

Bazen bir sitede birden fazla aslî stil belgesi olabilir. Bazı sayfalara "stillerim1.css" dosyasını, bazı sayfalara "stillerim2.css" dosyasını bağlayabilirsiniz.

6. Script dosyası oluşturmaya başlayın. Tıpkı CSS gibi, HTML sayfasının Script bölümü de dışardan dosya olarak HTML'e bağlanabilir. Başkalarının sitelerinde gördüğünüz Javascript programlarını çok hoşlandınız diye alıp, kullanamazsınız. "Kim görecektir!" diye düşünmek en azından ahlakî bir tutum olmaz. Ayrıca buna ihtiyaç da yok. Bir kere başkalarının sayfalarındaki Javascript bölümleri, büyük bir ihtimalle açıklama ve yorum satırları içermediği için içinden çıkılması imkansız olabilir. Oysa Web tasarımcılarına, kullanılmaya hazır, hemen her fonksiyonu yorum satırları ile açıklanmış,

tefif hakkı sorunu olmayan Javascriptler sunan binlerce Internet sitesi bulabilirsiniz. Bu tür yerlerden edineceğiniz güvenli ve anlaşılır Javascript'leri, sabit diskinizde biriktirebilirsiniz. Ayrıca büyük bir ihtimalle Javascript öğrenmek için alacağınız kitabın da içi binlerce örnekle dolu bir CD-ROM'u olacaktır!

7. Microsoft Internet Explorer ve Netscape Navigator başta olmak üzere yaygın Web Browser programlarını özelliklerini, benzerliklerini ve farklılıklarını öğrenmeye başlayın. Dinamik HTML adı altında yer alan tekniklerin çoğu, IE veya Netscape'in Belge Nesneleri Modeli adı verilen ve bir açtıkları bir pencerede görüntüledikleri bir Web sayfasının özelliklerini nasıl tanıdıkları ve hangi komutlarla değiştirdikleri bilgisine dayanır.

Bir örnekle bu karmaşık kavramı açalım: HTML'de tablo oluşturmak için `<TABLE></TABLE>` etiketlerinin içini doldurmanız gerektiğini biliyorsunuz. Fakat IE için bir tablonun 22 adet belirlenebilir özelliği varken, Netscape bir tablonun sadece 9 özelliğinin belirlenebilmesine izin verir. Bu yüzden TABLE etiketinin içinde, mesela `BORDERCOLORDARK="koyu-renk"` ve `BORDERCOLORLIGHT="açık-renk"` unsurlarını kullanacak olursanız, IE'de tablonuzun dış çerçevesinin üst ve sol kenarı koyu-renk yerine yazdığınız renkle, alt ve sağ kenarı ise açık-renk yerine yazdığınız renkle görüntülenir; sitenizi Netscape'i kullanarak ziyaret edenler ise tablonun tüm çerçevesini tek renk olarak görürler. İki Browser arasındaki farklar, bu kadar "zararsız" olsa iyi! Örneğin, ziyaretçilerden bilgi derlemekte kullandığınız formların vazgeçilmez ögesi INPUT etiketi, IE'de 21 adet belirlenebilir özelliğe sahip ve bunlardan birini kullanarak, formunuzdaki INPUT kutusunu bir veritabanından alınacak bilgi ile doldurtmak isteyebilirsiniz. Bu, özellikle dinamik bir sayfada, ziyaretçinizin size daha önce bildirdiği bazı bilgilere, bazı

tercihlere dayanıyor olabilir. Oysa ziyaretçiniz, Netscape kullanıyor ise, sayfanız bu dinazmden yoksun kalacağı gibi, formdaki eksiklik, ziyaretçiye tasarımcı olarak sizin hatanız gibi görünebilir.

İyi bir Web tasarımcısı, bu sebeple her iki programı çok iyi tanımak ve farklarını bilmek zorundadır. Güvenli yol, iki Browser'ın asgarî ortak noktalarına hitap etmektir. Bu, bazen Web sayfanızın daha az dinamik, daha az hareketli, daha az marifetli olmasına yol açacaktır. Ama iyi bir Web tasarımcısı, ziyaretçilerine hangi Browser'ın daha iyi olduğuna dair ders vermez; sadece ve sadece sitesinin içeriğini her iki Browser'ın kullanıcılarına da ulaştırmaya çalışır. Sitenizi IE kullanarak görmeye gelen bir kişinin "Bu site Netscape için optimize edilmiştir," Netscape kullanarak gelen bir ziyaretçinin de "Bu site en iyi IE ile görülür!" yaftasını görünce, ne hissettiğini sanıyorsunuz? Size düşen, ziyaretçinizi sinirlendirmek değil, tersine sunmak istediğiniz içeriği, ziyaretçinin Browser'ına uyumlu hale getirmek. Bunun için ziyaretçinin Browser'ının türünü ve sürümünü belirlemeniz gerekir. Eğer bunu yapmayacaksanız, asla Netscape'de yanlış görüntülenecek, sadece IE'e özgü özelliklere hitap etmeyin. IE'nin bazı özelliklerinden yararlanmak istiyorsanız, bu sağlayacak kodun Netscape'de "hiç bir şey yaptırmayan" kodlar olmasına dikkat edin.

ZİYARETÇİNİN BROWSER'INI KOKLAYABİLİR MİSİNİZ?

Internet Browser programlarının bir kokusu olduğunu düşünmüyorsunuz, herhalde! Ama muzip bir programcı, ziyaretçilerin sitenize bağlantı sağladıkları Browser'ın markası ve sürümünü belirleme işine "Browser'ı koklamak" adını takmakta sakınca görmemiş olmalı! O günden beri bu amaçla kullanılan Javascript ve VBScript programlarına "Browser Koklayıcı" deniyor.

Ziyaretçinin Browser'ının türü ve sürümü iki yerde belirlenebilir: Ziyaretçinin bilgisayarında (Client-side) veya sitenize ev sahipliği yapan Web Server'da (Server-side). ASP gibi ileri bir Server teknolojisi kullanmıyorsanız, belirleme işini ziyaretçinin bilgisayarında yapmak uygun olur. HTML kodlarınızı iki Browser'ın ortak özelliklerine indirgemek işinize gelmiyorsa, sayfanın tümünü değilse bile bir bölümünü IE için ve diğerleri için farklı yapabilirsiniz. Bunu sağlamak için size gerekli bilgi, ziyaretçinin Browser'ı IE 4 ve üstü mü, 3 ve altı mı, yoksa diğerleri mi sorusunun cevabıdır. Bu soruyu, ziyaretçinin Browser'ına şu Javascript fonksiyonu ile sorabilirsiniz:

```
function msieversion()  
{  
    var ua = window.navigator.userAgent  
    var msie = ua.indexOf ( "MSIE " )  
    if ( msie > 0 )  
        return parseInt (ua.substring (msie+5, ua.indexOf (".", msie )))  
    else  
        return 0  
}
```

Bu fonksiyon, hemen hemen bütün belli başlı Browser'larda işler ve size Microsoft IE'nin sürüm numarasını bildirir. Diyelim ki Web sayfanız gerçekten IE 4 ve daha sonrası sürümlerin bir özelliğine mutlaka hitabetmek zorunda. O zaman, HTML'in daha sonraki bir bölümünde, IE 4 ve sonrası için ayrı, IE 3 ve öncesi için ayrı, ve diğerleri için ayrı HTML veya Script kullanacağınız zaman, yukarıdaki fonksiyonun elde etmiş olduğu bilgiyi şöyle kullanabilirsiniz:

```
if ( msieversion() >= 4 ) {
```

```

    [buraya IE 4 ve sonrası için yazacağınız kodlar girecek]
    }
else if ( msieversion() <= 3 ) {
    [buraya IE 3 ve öncesi için yazacağınız kodlar girecek]
    }
else {
    [buraya diğer Browser'lar için yazacağınız kodlar girecek]
}

```

Fakat bazen, birinci sayfanızı bile ziyaretçinin Browser'ına uyumlu olarak vermek isteyebilirsiniz. Bu durumda şu sayfa, ziyaretçinin Browser'ını koklayacak ve ona göre ana sayfa gönderecektir. (Netscape için sayfanızın adı "netscape.htm," IE için "ie.htm" ve diğerleri için "diger.htm" olması, sayfa adları başka ise bu kodda değişiklik yapmanız gerekir):

```

<html>
<head>
<title>Browser'a göre ana sayfa!</title>
<script>
<!-- Eski Browser'lardan gizle
var browser=navigator.appName + " " + navigator.appVersion;
var getkey=browser.substring(0, 12);
function hangiSayfa()
{
if (browser.substring(0, 8)== "Netscape")
    window.location="netscape.htm";
if (browser.substring(0, 9)== "Microsoft")
    window.location="ie.htm";
}

```



```

if ( (browser.substring(0, 8)!="Netscape") && (browser.substring(0, 9)!="Microsoft")
)

        window.location="diger.htm";

}

// -->

</script>

</head>

<body onload=hangiSayfa()>

</body>

</html>

```

Genellikle, ziyaretçinin Browser'ını soruşturduğunuz zaman aldığınız sonucu bir değişkene (variable) yazarak, bir kenarda tutmak isteyebilirsiniz. Bu değişken, ilerde başka Java fonksiyonlarında veya HTML içinde özellikle BODY etiketinde kullanılır. Bu amaçla biraz daha gelişmiş bir Browser belirleme fonksiyonu şöyle olabilir:

```

function Brw() {

    var agent = navigator.userAgent.toLowerCase();

    this.major = parseInt(navigator.appVersion);

    this.minor = parseFloat(navigator.appVersion);

    this.ns = ((agent.indexOf('mozilla')!=-1) && ((agent.indexOf('spoofer')== -1) &&

(agent.indexOf('compatible') == -1)));

    this.ns2 = (this.ns && (this.major == 2));

    this.ns3 = (this.ns && (this.major == 3));

    this.ns4 = (this.ns && (this.major >= 4));

    this.ie = (agent.indexOf("msie") != -1);

    this.ie3 = (this.ie && (this.major == 2));

    this.ie4 = (this.ie && (this.major >= 4));

```

```
this.op3 = (agent.indexOf("opera") != -1);  
}
```

```
var brw = new Brw()
```

Bu Javascript fonksiyonu, size biri doğru (true), altısı yanlış (false) yedi nesne (object) kazandırır. Diyelim ki, ziyaretçinin Browser'ı Netscape 4.0; o zaman brw.ns4 doğru olacaktır, mesela brw.ie4 yanlış! Sayfanıza ilerde, "Eğer brw.ns4 doğru ise şunu yap... eğer brw.ie4 doğru ise bunu yap" şeklinde rutinler ve fonksiyonlar koyabilirsiniz.

8. Javascript'lerinizi hazırlayın: Dinamik HTML'in iki ayağından biri CSS ise diğeri Javascript yoluyla, bu stilleri katmanlara uygulamaktır. Ziyaretçinin Browser'ına, "Şimdi, pencerede x koordinatı 120, y koordinatı 15 olan noktadan itibaren eni 100, yüksekliği 110 olan bir katman aç ve içine "Burayı tıklayınız!" yaz; bu yazının rengi sarı, zemini mavi olsun; Mouse işaretçisi bu katmanın üzerine geldiği zaman yazının rengi kırmızıya, zeminin rengi turuncuya dönsün!" demek sadece HTML kullanarak hemen hemen imkansız. Oysa, Javascript ile kullanılmaya hazır fonksiyonlar oluşturup (veya bulup), bunları HTML sayfasında kullanmak, işi olağanüstü kolaylaştırır. Javascript ile oluşturulacak fonksiyonlar da, tıpkı CSS komutları gibi bir HTML dosyasına ya dışarıdan eklenir, ya da HTML'in içinde (ya baş tarafında, ya da kullanılacağı yerde) kullanılır.

Javascript ile oluşturacağınız fonksiyonları bir sitedeki bütün sayfalarda, hatta başka-başka sitelerin sayfalarında kullanmak haricî Javascript dosyasının HTML'e bağlanması ile mümkün olur. Javascript fonksiyonlarını içeren bir düzyazı dosyası, adının uzatması ".js" olan bir dosyada toplanır ve HTML'e LINK etiketi ile bağlanır. Tıpkı stil komutları gibi, sırası geldiğinde HTML'in içinde bu dosyadaki bir fonksiyonu işbaşına çağırırsınız. İyi bir tasarımcının çok zengin bir haricî Script belgesi koleksiyonu olur.

Böyle bir hariç JS dosyasında şu bölümler bulunur:

Browser Belirleme Fonksiyonu: DHTML uyumlu bir Browser, CSS ve katman teknolojilerini destekler. Bununla birlikte bütün Browser'lar bütün Javascript fonksiyonlarını desteklemez. Burada kritik nokta, Netscape 4.0 ve daha üstü ile IE 4 ve daha üstünün Browser penceresindeki bir belgeye nasıl referansta bulunduğudır. Netscape, belge unsurlarına *document.nesne* şeklinde hitabedilmesini isterken, IE, *document.all.nesne* şeklindeki referans gerektirir. Ayrıca Netscape'te stil belgelerine *document.stil_adi.nesne* şeklinde atıfta bulunursunuz, IE'de ise bu *document.all.stil_adi.style.nesne* şeklinde yazılır. Javascript'inizde bu unsurlara, diyelim ki, sürekli belge ve stil diye atıfta bulunabilirsiniz, fakat baş tarafa, Browser belirlediğiniz yere "Eğer Netscape ise belge = *document.stil_adi.nesne*, eğer IE ise belge = *document.all.nesne*" anlamında bir rutin yazarsınız.

Grafikleri önceden yükleme: DHTML önemli ölçüde grafik unsurlara yer verme sanatıdır; ve grafikler bir sayfanın ziyaretçiye aktarılmasında en önemli yavaşlık sebebidir. Burada iki büyük düşünce ekolü var: "Sitemizdeki bütün grafikleri henüz birinci sayfa görüntülenmeden ziyaretçiye aktaralım ve sayfalarımız daha sonra çabuk görüntülensin" diyenler; ve "Birinci sayfa ne kadar çabuk görüntülenirse o kadar iyi, ziyaretçinin bir kere ilgisini çekmek şart!" diyenler. İster birinci grupta olun, ister ikinci grupta, hiç değilse birinci sayfanın grafiklerinin tümü, sayfa görüntülenmeye başlamadan önce ziyaretçiye aktarılmalıdır. Bunu, BODY etiketine gömeceğiniz `onLoad` komutuna bir Javascript önyükleme fonksiyonu bağlayarak yapabilirsiniz. Şu Javascript fonksiyonu, önyükleme işini yapacaktır:

```
Function onYukleme() {
```

```
resim1 = new Image();  
resim1.src = "dizin/resim1.gif";  
}
```

Kapanan kıvrık parantezle fonksiyonu bitirmeden önce, resim2, resim3, vd., sıralayabilirsiniz. Böylece önceden yüklediğiniz grafikler, ziyaretçinin bilgisayarının belleğinde kalacak ve ihtiyaç anında sür'atle görüntülenecektir.

Javascript konusunda bir son tavsiyeye yer verelim: Gerçekten Javascript öğrenmeye zamanınız yoksa, o zaman sizin için Javascript yazacak bir programı, en ince ayrıntılarına kadar öğrenin. DHTML'in bütün unsurlarını HTML sayfalarınıza yazabilecek ve bu arada sizin için yüzlerce satır Javascript programı oluşturabilecek WYSIWYG tarzı HTML editörlerinin sayısı hızla artıyor. Macintosh dünyasında **GoLive CyberStudio**, Windows dünyasında **Macromedia Dreamweaver** bu tür programlar arasında sayılabilir.

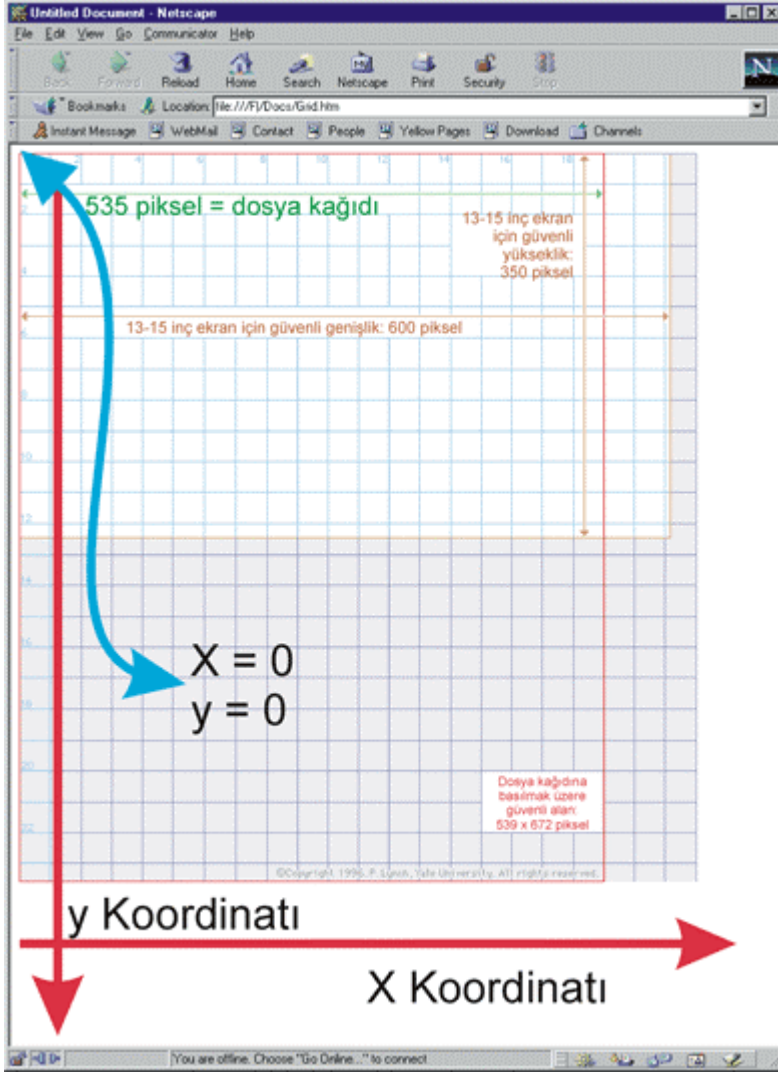
9. Katmanlar: Denir ki, Internet, televizyon ile dergi arasında bir ortamdır.

Doğru; Web sitelerinde bir gazetede olabileceğinden çok, ancak uzmanlık alanında yayın yapan bir dergide ya da kitapta bulunacak kadar bilgi, bir televizyon atraksiyonu ile ziyaretçiye sunulabilir. Web sayfalarınızda doğrudan video dosyaları da verebilirsiniz; fakat çok pahalı *streaming video* (akan video) teknolojisi kullanmıyorsanız, bir AVI dosyasının ziyaretçiye aktarılması için gerekli süre, ziyaretçiye bilgisayarını kapattırıp, televizyonunu açtırtabilir!

Fakat Web sayfalarınızda televizyon ve sinema teknolojisinin temeli olan grafiklerin (veya metinlerin) belirli bir sırayla görüntülenmesi, sırası gelenin "açılması," sırası geçenin "kapanması" sağlanabilir. Bunu katman teknolojisi sağlar. Bir HTML sayfasında, ziyaretçinin Browser'ına ActiveX, Flash, Shock ve benzeri animasyonları gerçekleştirme

için gerekli *plug-in* dosyalarının yüklenmesini zorunluk olmaktan çıkartan da katman teknolojisidir. DIV, LAYER, ILAYER ve SPAN etiketleri ile oluşturacağınız katmanlar, piksel düzeyinde bir hassasiyetle, ziyaretçinin ekranında, Browser penceresinde belirli bir yere oturtulabilir ve belirli bir sırayla görünür veya görünmez hale getirilebilir. DHTML'in *dinamik* tek unsuru da budur. (Bunlardan sık kullanacağınız DIV ile SPAN'dir ve aralarındaki tek fark, DIV, blok elemanıdır, yani kendisinden sonra yeni paragraf başlatır; buna karşılık SPAN satır elemanıdır, kendisinden sonra hiç bir şey yapmaz.)

"Piksel düzeyinde hassasiyet" ne demek? Bir Browser, klasik HTML belgesini aldığı zaman içindeki komutları yorumlayıp, sonucunu kendi penceresinde görüntülemeye, sıfır-sıfır noktasından başlar. Bir Browser penceresini, soldan sağa x, yukarıdan aşağı y koordinatları olarak düşünün.



Başarılı bir Web sitesi, 14 inçlik bir ekranı, 640x480 çözünürlükte bir grafik kartı bulunan tipik bir ziyaretçi dikkate alınarak kurulduğuna göre, Windows ve MacOS'in aldığı yerler, Browser'ın kendi penceresi, başlık ve menü çubukları, çerçeveler çıktıktan sonra, tasarımcı olarak size "temiz alan" olarak 600'e 350 piksel kalıyor. (Microsoft, ve Britannica ansiklopedisi bu alana kendi sitelerini sığdırdığına göre, herhalde siz bir günlük gazete sitesi sığdırabilirsiniz!) Düz bir yazı içeren Web sayfası, nerede başlayıp, nerede biterse bitebilir; önemli değil. Ama, diyelim ki, ziyaretçi Mouse işaretçisini bir grafiğin belirli bir yerine getirdiği zaman, aynı grafiğin 155:220 koordinatında başlayan ve eni 35, yüksekliği

48 piksel olan bir bölüme farklı grafik görüntülenecek! Bu ölçüde hassas *positioning* (yerleştirme işlemi), ancak katmanlarınızı doğru tanımlamakla mümkündür. "Doğru tanımlamak" ise önce ziyaretçinin o andaki Browser boyutlarını bilmekle mümkündür. Kim olduğunu bile bilmediğiniz bir kişinin, ne olduğunu bile bilmediğiniz Browser'ının o anda enini boyunu nasıl bilebilirsiniz? Ve üstelik bunu, her ziyaretçi için yapacaksınız. Oysa işin kolayı, sayfanın bir kenarına, ziyaretçiye hakaret edercesine, "Ekran çözünürlüğünüz 800'den aşağı ise, yandınız!" diye bir uyarı koymak iken!

Bu işleme ziyaretçinin Browser penceresinin çözünürlüğünü belirleme denir, ve Netscape ile Internet Explorer, farklı sorular sormak şartıyla, sorarsanız, size o andaki boyutlarını verirler. Diyelim ki bir hariç JSavascrypt dosyasını HTML belgenizin baş tarafında:

```
<SCRIPT LANGUAGE="JavaScript" SRC="javascript.js"></SCRIPT>
```

şeklinde bir SCRIPT etiketi ile sayfanıza bağlamış bulunuyorsunuz ve bu JS dosyasının içinde daha önce ele aldığımız türden size "brw.ie4" veya "brw.ns4" nesnelere veren bir fonksiyon var. Şimdi bu nesnelere BODY etiketi içinde onLoad olayını kullanarak test edebilir ve ziyaretçinin Browser'ı Netscape ise Netscape için, IE ise IE için geçerli soruyu sorarak, ziyaretçimizin o anda Browser ekranının enini boyunu öğrenebiliriz. İşte size kullanabileceğiniz bir HTML-Javascript parçası:

```
<BODY onLoad="
    if(is.ns4) {
        ekranEn =innerWidth;
        ekranBoy=innerHeight;
        preLoad();
    } else if(is.ie4) {
```

```
ekranEn=document.body.clientWidth;

ekranBoy=document.body.clientHeight;

preLoad();

}"

onResize="history.go(0)">
```

Bir parantez açalım: buradaki "onResize="history.go(0)" ifadesi, ziyaretçinin sitemize bağlandıktan sonra Browser'ının boyutlarını değiştirmesi ihtimaline karşı, sayfamızın ziyaretçinin sabit diskindeki Cache bellekten yeniden yüklenmesini sağlayan bir güvenlik önlemidir. Bir başka güvenlik önlemi Browser'ın sayfa görüntülediği anda penceresinin sağına ve altına otomatik kaydırma çubuğu koymasını istemiyorsanız, BODY etiketine şu özellikleri de verebilirsiniz:

```
BACKGROUND="images/horizon.jpg" MARGINWIDTH="0" MARGINHEIGHT="0" SCROLL="NO"
```

Peki, şimdi elinizde iki ölçü oldu: ekranEn ve ekranBoy. Bu bilgileri nasıl kullanacaksınız? İşte bu bilgi, ziyaretçinize "Ekranını 800x600 piksel yap, öyle gel!" demekten sizi kurtaracak, söz gelimi bir grafiği 460:710 koordinatlarına denk gelmek yerine "ziyaretçinin ekranının sağ kenarından 200 piksel solda başlamasını sağlayacak olan bilgidir. Bunu yapabilmek için grafiğinizi bir katman nesnesinin (<LAYER>..</LAYER>, .. veya <DIV>..</DIV> etiketlerinin içine koyarsınız; katman nesnesi, "çıplak" konulacak grafikten () farklı olarak, STYLE özelliğine (attribute) sahiptir ve dolayısıyla, pozisyon, başka bir nesnenin altında veya üstünde olmak, görünür veya görünmez olmak gibi kontrollere sahiptir. Örneğin, içinde sayfalarınız arasında gidip gelmeyi sağlayan düğmelerden birincisini içeren katmanın şöyle oluşturulduğunu düşünün:

```
<DIV ID="katman_dugme1" STYLE="position: absolute left: ekranEn-200;
top ekranBoy-200; width: 120; height: 120; z-index: 1; visibility:
hidden;">
```


Hangi IMG etiketi böylesine denetlenebilir? İşte bu dinamizmdir; ziyaretçi ile etkileşmektir. Ziyaretçi ile etkileşmek, her ziyaretçinin bilgisayarına bir çerez bırakmaktan ibaret değildir. Katmanlar nesne oldukları için, isimleriyle hitabedilebilirler; herhangi bir Mouse olayına göre *attribute* değiştirebilir, örneğin görünür hale gelebilirler; görünmez olabilirler. Dinamik bir katman yerleştirme işlemi gibi yönetimi de Javascript ile mümkündür.

10. Sitenizi sınamın: Çok kolay gibi görünmesine rağmen, Internet, farklı Browserlar, farklı modem hızları, farklı bilgisayarlar, farklı ekran kartları ve ekranlarla sınanmamış sitelerle dolu. Bir çok tasarımcı için sına, en fazla bütün dosyaları sitenin üretildiği bilgisayardan başka bir bilgisayara aktararak, bağlantıları kontrol etme anlamına geliyor. Yani acaba herhangi bir LINK, mutlak bir dosyayı mı çağırıyor? IMAGE dizinine yerleştirilmemiş grafik kaldı mı? Hayır, hayır.. Sınama bu değil. Sınama, mesela siz İstanbul'da oturuyorsanız, Artvin'deki veya Van'taki hedef kitlenize mensup olduğuna inandığınız gerçek Internet kullanıcılarını bir 14.4'lük modem bağlantısı olan, 66 MHz'lük bir CPU'su ve 1 MB bellekli, 640x480 çözünürlükte ekran kartı bulunan ve 15 inç ekrana sahip bir bilgisayarın karşısına geçirip, yüzlerini incelemektir.

Ve bunu yaparsanız, sizden daha iyi Web Sihirbazlığı yapan olmayacağına inanabilirsiniz.

ŞUNU NASIL YAPARIM?

Ne kadar usta olursa olsun, bu soruyu, hiç olmazsa arada bir sormayan Web tasarımcı düşünmek imkansız gibidir. İşin kötü tarafı, insan kendisini Web tasarımcısı saymaya başladıktan sonra, bu tür soruları da açıkça sormaz! İşte, sormaya çekindiğiniz

veya çekinmediğiniz HTML soruları ve yanıtları. Ama hemen eklemek gerekir ki, yapılacak herhangi bir işin, çözülecek herhangi bir tasarım güçlüğünün, HTML 4 gibi, CSS, CSS-P, Javascript ve VBScript ile donanmış bir dilde, sadece bir doğru yanıtı olamaz. Buradaki çözümleri, bulanabilecek çözümlerden sadece biri olarak değerlendirmek gerekir.

Büyük Birinci Harf (Gömme):

Gazete ve dergi yazılarında bazı paragrafların ilk harfi, diğer harflerden çok büyük ve aşağı doğru iki, hatta üç satır yüksekliğinde olur. Gazetecilik dilinde buna **Gömme Harf** denir. Yazılı basının favori masaüstü yayıncılık programı QuarkExpress'te bile otomatik yapmak için programın beşinci sürümünü beklediğimize göre, HTML'de birinci harfin büyük olmasını otomatik olarak sağlamak için biraz daha bekleyeceğiz demektir. Gerçi CSS-1 kuralları arasında bunu sağlayan bir eleman var (first-letter), fakat kuralların zorunlu değil tavsiye bölümünde yer aldığı için ne Nestcape 4.5 ve öncesi, ne de IE 4.01 ve öncesinde uygulanmış değil. Bütün umudumuz, IE 5 ile CSS-1 ve CSS-P'nin bütün kurallarının uygulanabilir hale gelmesi. Eğer bu mümkün olursa, bu imkandan yararlanmak için, örneğin, şöyle bir stil tanımlamanız yeterli olacak:

```
P.first-letter {font-size: 200%;}
```

Tabii, ilk harfinizi yüzde 200 yerine isterseniz yüzde 400 de büyütebilirsiniz. O zamana kadar şu seçeneklerle idare edin:

1. Grafik olarak harf:

```
<IMG SRC="Buyuk_b.gif" HSPACE=1 ALIGN=LEFT>üyük ilk harf elde etmenin  
çaresi.
```

2. Tek kutulu tablo:

```
<TABLE ALIGN=LEFT CELLSPACING=0 CELLPADDING=0 BORDER=0><tr align="left" valign="top"><TD><FONT size=+4>B</FONT></TD></TR></TABLE> harf elde etmenin aresi..
```

3. Stil vererek:

HTML belgesinde HEAD blmnde Őu stili oluŐturun:

```
<style> .gomme { width: 1em; height: 1em; float: left; text-align: center; font-size: 20pt; color:red; font-style:italic } </style>
```

Sonra, birinci harfini byk yapmak istediĐiniz paragrafa Őyle baŐlayın:

```
<p><SPAN class="gomme">B</span>harf byk, kırmızı ve yksekliĐi  satır olacak. Bu paragrafı biraz daha uzatalım.. Biraz daha.. Biraz daha.. Biraz daha.. Biraz daha.. Biraz daha.. Biraz daha.. Biraz daha.. Biraz daha.. Biraz daha.. Eh bu kadar yeter.</p>
```

Dikey izgi

HTML'de yatay izgi (<HR>) var da, dikey izgi yok. Tasarımınız, sayfada dikey izgi gerektiriyorsa, tablodan yararlanabilirsiniz. Grafik programınızda eni ve boyu 1 piksel ve rengi siyah bir kutu resmi yapın. Bunu 1psiyah.gif adıyla kaydedin. Dikey izgiye ihtiyacınız olan yerde bir tablo oluŐturun; metnelerinizin bir kısmı ve tablonun iine girebilir; girmeyebilir. İŐte bir rnek:

```
<table width="75%" border="0" cellspacing="0" cellpadding="0"> <tr> <td align="center" valign="top">Burada metin var.</td> <td></td> <td align="center" valign="top"> Burada metin var </td></tr> </table>
```

Çizginin yüksekliği ne kadar olacaksa, çizginin bulunduğu kutudaki IMG etiketinin yüksekliğini o kadar arttırın.

Beyaz Boşluklar

Bazen, iki satır arasında tam istediğiniz ölçüde boşluk elde etmek zor olur. <P> çok,
 ise az boşluk bırakır. Kolayı var. Eni 1, boyu 2 piksel, saydam bir GIF dosyası yapın ve 1psaydam.gif adıyla kaydedin. Boşluğa ihtiyacınız olan iki satırın arasına şu etiketi koyun:

```
<IMG scr="1psaydam.gif" width=1 height=1 hspace=20 vspace=49>
```

Burada 20 ve 49 yerine, kendi ihtiyacınız olan boşluğun değerini yazacaksınız.

Bir satırı arzu ettiğiniz kadar içerden başlatmak amacıyla da aynı teknikten biraz farklı ifadeyle yararlanabilirsiniz:

```
<IMG hspace="25" scr="1psaydam.gif">Bu satır 25 piksel içerden başlayacak.
```

İki sütun yazı arasına arzu ettiğiniz kadar boşluk vermek için, sütunları bir tablonun hücrelerine alın, ortalarındaki hücrenin boyunu ve enini istediğiniz boşluk ölçüsüne getirin.

Sadece Netscape'de bulunan <SPACER> etiketi de beyaz boşluk sağlayabilir:

```
<SPACER TYPE="block" WIDTH="30" HEIGHT="45">
```

IE, bu kodu görünce, hiç bir şey yapmaz.

Arkaplanda Yazı

Bazen güzel bir düzenleme ile biş başlığın arkasında, aynı veya farklı kelimelerin tekrarlanarak zemin oluşturduğunu görürsünüz. Üstelik de sayfa yıldırım hızıyla gelmiştir;

yani arkaplandaki yazıların, grafik olmadığı anlaşılıyor. Bunu katman teknolojisi ile başarabilirsiniz:

```
<SPAN STYLE="position: relative; top: 60px; left: 40px; z-index: 2; font-family: Verdana, Arial, Helvetica; font-weight: bold; font-size: 24pt; color: #ff3300">ÖNDEKİ METİN<BR></SPAN><BR>
```

```
<SPAN STYLE="z-index: 1; font-family: Serif, Times, Georgia; font-size: 10pt; color: #ffcc99">Arkadaki metin Arkadaki metin Arkadaki metin Arkadaki metin<BR>Arkadaki metin Arkadaki metin Arkadaki metin Arkadaki metin <BR>Arkadaki metin Arkadaki metin Arkadaki metin Arkadaki metin <BR> </SPAN>
```

Arkadaki metin Arkadaki metin Arkadaki metin Arkadaki metin
Arkadaki metin Arkadaki metin Arkadaki metin Arkadaki metin
Arkadaki metin Arkadaki metin Arkadaki metin Arkadaki metin

ÖNDEKİ METİN

İlk sayfa bir kaç saniye görünsün, kaybolsun!..

Buna "Splash page" (suya dalma sayfası) da deniyor; bir kaç saniye görünüyor, sitenizin niteliği hakkında fikir veriyor, ilgi çekiyor, sonra yerini ana sayfaya bırakıyor.:

```
<META HTTP-EQUIV="Refresh" CONTENT="5; URL=ANASAYFA.HTML"><BODY><IMG width="300" height="200" alt="Giriş grafiği" SCR="giris.gif"><BODY>
```

Bu tekniği iki sayfasının arasında reklam amacıyla uygulayanlar bile var!

Stil düzenlemesi yapmadan harf belirlemek için

Sayfanızda, sadece bir yerde normal stil komutlarının dışına çıkarak, bir metnin belirli bir harfle görüntülenmesini istiyorsanız, hemen önünde şu etiketi kullanabilirsiniz:

```
<FONT FACE="Courier">Bu yazı Courier harfle görüntülenecek</FONT>
```

Türkçenizi kaybetmemek için

Bilgisayarla uğraşalı beri, kendi dilimiz sık sık kazaya uğruyor, ama sayfalarımızın dilini korumak elimizde. Her sayfanın başına şu META etiketi koyabilirsiniz:

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
```

```
<meta http-equiv="Content-Language" content="tr">
```

Resimleri Bloklayın!

Sayfanızdaki grafiklerin hepsinin sol tarafı aynı hizaya gelsin istiyorsanız, sadece yazı için kullanılır sandığımız <BLOCKQUOTE> etiketinden grafiklerde de yararlanabilirsiniz:

```
<BLOCKQUOTE><IMG src="resim.gif"></BLOCKQUOTE>
```

Bütün grafikleriniz soldan yaklaşık 40 piksel hizalanacaktır! Resimleriniz sayfanın sağına hizalansın istiyorsanız, buna bir de ALIGN ekleyebilirsiniz:

```
<BLOCKQUOTE><IMG src="resim.gif" ALIGN=right></BLOCKQUOTE>
```

<BLOCKQUOTE> etiketi kendi içinde kullanırsa, içindeki unsuru 80 piksel sağa iter.

```
<BLOCKQUOTE><BLOCKQUOTE><IMG src="resim.gif" ALIGN=right>
</BLOCKQUOTE></BLOCKQUOTE>
```

Tabii, en istikrarlı hizalama aracı olarak daima tablodan yararlanabilirsiniz.

Noktasız Liste

HTML'in noktasız listesinin satır araları çok açık; noktalarının araları iyi, fakat siz nokta istemiyorsunuz. O halde HTML'i biraz kandırabilirsiniz:

```
<UL>Madde 1<BR> Madde 2<BR> Madde 2<BR> Madde 4<BR></UL>
```

İşte size aralığı dar, ama noktasız liste!

HTML'e ENTER ve TAB girdilerini kabul ettirmek mümkün mü?

Hayır, değil! Düz yazı programınızda HTML belgede bir metne ne kadar ENTER ve TAB girseniz de, Browser bildiğini okur ve bunları yok eder. Bir şartla! İsteddiğiniz gibi biçimlemek istediğiniz bölümün başına <PRE> , sonuna </PRE> etiketlerini koyarsanız, bütün ENTER ve TAB'leriniz korunacaktır.

Form unsurlarını hizalamak istiyorsanız

Formlarda, isim yazılacak INPUT kutusuyla adres yazılacak INPUT kutusunu hizalayabiliyor musunuz? Elbette, hayır! Bir çaresi bütün unsurları bir tablonun hücrelerine serpiştirmektir. Fakat bir başka çaresi daha var. <PRE> etiketinden yararlanmak.

"İsim" kelimesi dört harfli, "Adres" kelimesi beş harfli.. Demek ki, İsim'in INPUT kutusunun Adres'in INPUT kutusu ile hizalanması için İsim'den sonra bir boşluk olması gerekiyor. <FORM> etiketinden önce <PRE> etiketini kullanırsanız, bütün form unsurlarınıza bu hesabı yaparak, INPUT kutularını kolayca hizalayabilirsiniz.

E-Mektup bağlantısı

Sitenizdeki bir bağlantıyı tıklayan ziyaretçinin elektronik posta programı açılınsın; boş bir mektup kağıdı görüntülensin; alıcı hanesine sizin (veya arzu ettiğiniz

bir başka kişinin ve kurumun) adresi yazılsın istiyorsunuz. Kolay! Bir mailto: bağlantısı bu işi halleder. Bu mektup birden çok kişiye gitsin istiyorsanız, o zaman kod şöyle olacak:

```
<A HREF="mailto:birincikişi@domain.com?cc=ikinci_kişi@domain.com">
```

Bu mektubu bir üçüncü kişi de alsın istiyorsanız, bir ekleme yeter:

```
<A HREF="mailto:birincikişi@domain.com?cc=ikincikişi@domain.com&bcc=üçüncükişi@domain.com">
```

Peki, sitenizde kime elektronik mektup göndereceği kararını neden ziyaretçiye bırakmıyorsunuz? Bir "açılan liste" kutusuyla bu işi halledebilirsiniz:

```
<form NAME="mektupformu">
```

```
<select SIZE="1" NAME="liste">
```

```
<option value="birincikişi@domain.com"> Birinci Kişi </option>
```

```
<option value="ikincikişi@domain.com"> İkinci Kişi</option>
```

```
<option value="üçüncükişi@domain.gov"> Üçüncü Kişi</option>
```

```
</select>
```

```
<input LANGUAGE="JavaScript" TYPE="button" VALUE="Mektup Gönder"
```

```
ONCLICK="location.href = &quot;mailto:&quot; +
```

```
document.mektupformu.liste.options[document.mektupformu.liste.selectedIndex].value" NAME="Mektup Gönder">
```

```
</form>
```

Bu kodu yazarken, Javascript bölümünde tırnak işaret tırnak içinde olmasın diye """ kodunun kullanılacağına dikkat edin!

Geri Düğmesi

Browser'ın Geri düğmesi özellikle Çerçeveli sayfalarda nereye geri döneceğini bilmeyebilir. Bu bakımdan sayfalarınızda arzu ettiniz yere kendi Geri düğmenizi koyabilirsiniz:

```
<FORM><INPUT TYPE="button" VALUE="Geri dönmek için burayı tıklayınız!"
onClick="history.go(-1)"></FORM>
```

Aynı kodu grafik yerine bir veya daha fazla kelimeye de kazandırabilirsiniz:

```
<a href="javascript:history.go(-1)">Geri dönmek için burayı tıklayın</a>
```

Aynı kodu, HTML-vari düğmelere de uygulayabilirsiniz:

```
<FORM><INPUT TYPE = "BUTTON" VALUE = "Back" onClick =
"window.history.go(-1);"><INPUT TYPE = "BUTTON" VALUE = "Forward"
onClick = "window.history.go(1);"></FORM>
```

Pencereyi kapatın

Bir ziyaretçinin Browser penceresini kapatmak için sayısız seçeneği var. Ama yine de kolay bir yere Kapat düğmesi koyabilirsiniz:

```
<A HREF="javascript>window.close()">Bu pencereyi kapatmak için
burayı tıklayınız</A>
```

İsterseniz, yazı yerine bir grafik de kullanabilirsiniz.

Çerçeveye esir düşünce

Bir başka sitede sizin sayfanıza bağlantı veriliyor; fakat uyanık site sahibi sizin sayfanızı kendi çerçevelerinden birinde görüntülettiriyor. Oysa siz bu tür bağlantılarda sayfanız Browser'ın tüm penceresini kaplasın istiyorsunuz.

Sayfalarınızın başına şu minik kodu yazın; hiç kimse sayfalarınızı kendi çerçevesine hapsedemez:

```
<head>
<script language = javascript>
if(top.location != document.location.href) {
    top.location = document.location.href;}
</script>
</head>
```

Veya şu kodu da yazabilirsiniz:

```
<script> <!--
if (window != top )
top.location.href = window.location.href;
// -->
</script>
```

Bağlantılarınıza Tool Tip ekleyin

Tool Tip, Windows'da Mouse işaretçisini araç menüsünde bir simgesinin üzerine götürdüğünüzde açılan ve içinde bu aracın ne işe yaradığını belirten küçük bir sarı kutu içindeki yazılara verilen ad. Ziyaretçi Mouse işaretçisini bağlantılarınızın üzerine götürdüğünde böyle bir Tool Tip görüntülensin isterseniz, bağlantı kodunu şöyle yazabilirsiniz:

```
<A HREF="sayfanın_adı.htm"><IMG scr="images/resim.gif" TITLE="Burayı
tıklayın, filanca sayfaya gidin!"></A>
```

Sayfanızı Sık Kullanılanlar'a eklettirin:

IE 4.0 ve üstünü kullanan ziyaretçilere, isterlerse, bir kelimeyi tıklattırarak, sayfanızı Windows Sık Kullanılanlar klasörüne eklettirebilirsiniz. Buradaki kod, kendi Browser belirleme fonksiyonunu da içeriyor; dolayısıyla başka tür ve sürüm Browser'ı olan ziyaretçiler açısından arzu etmedikleri bir durum olmayacaktır:

```
<SCRIPT>

<!--

if ((navigator.appVersion.indexOf("MSIE") > 0) &&
(parseInt(navigator.appVersion) >= 4)) {

    document.write("<U><SPAN STYLE='color:blue;cursor:hand;'
onclick='window.external.AddFavorite(location.href, document.title);'>
Add this page to your favorites</SPAN></U>");

}

//-->

</SCRIPT>
```

Geriye doğru sayım

Diyelim ki, özel sitenizde doğum gününüze kaç gün kaldığını belirterek, ziyaretçilerinize, size alacakları armağan için hazırlık fırsatı vermek istiyorsunuz. Ya da tuttuğunuz takımın büyük maçına! Ya da 21'nci yüzyıla! (Bu Script'i kullanmak isterseniz, başındaki kredi satırını atmamak şartıyla serbestçe kullanabilirsiniz):

```
<script>

<!--//
```

```
/*  
  
Script by Jari Aarniala (foo@mbnet.fi)  
  
*/  
  
today = new Date()  
  
hedef = new Date("December 31, 1999") // buraya geriye sayılacak  
  
// tarihi yazın  
  
hedef.setYear = today.getYear;  
  
kalangun = (hedef.getTime() - today.getTime()) / (1000*60*60*24);  
  
kalangun = Math.round(daysLeft);  
  
document.write("Dogum günüme sadece"+ kalangun +" gün kaldı");  
  
//-->  
  
</script>
```

GeoCities'in yüzen grafiğini kısıkanıyor musunuz?

GeoCities bütün sayfalarında yarı-saydam bir loga grafiği, sayfası yukarı da indirseniz, aşağı da indirseniz, sağ alt köşede kıpırdamadan durur. Bu etkiyi elde etmek isterseniz, sadece IE 4 ve sonrasında işlemek üzere, işte kodu:

```
<body style="background-image: url('resim.gif'); background-attachment:  
fixed; background-repeat: no-repeat; background-position: right bottom">
```

Fakat aynı etkiyi Netscape de de elde etmek için, bir Javascript'e ihtiyaç var.

Önce, DIV ile bir katman oluşturalım:

```
<DIV id="waterMark" style="position:absolute"><A  
href="/index.html"><IMG src="watermark.gif" width=90 height=90  
border=0></A></DIV>
```

Kullanacağınız grafiğin dosya adı ne ise, burada grafik kaynak (scr=) yerinde bu ismi kullanacaksınız. Bunu hallettikten sonra, DIV bölümünden sonra ve tercihan BODY etiketinin kapanmasından önce, şu kodu yazın. (Bu Script'i , telif hakkı satırını atmamak şartıyla istediğiniz gibi değiştirerek, kullanabilirsiniz.)

```
<script language="JavaScript1.2">
<!--
// Watermark script by Paul Anderson, CNET Builder.com. All rights
reserved.

markW = 90; //resmin genişliği
markH = 90; //resmin yüksekliği
markX = 100; //sol kenardan uzaklık
markY = 100; //üst kenardan uzaklık
markRefresh = 20; //yenilenme süresi milisaniye olarak

if (!document.all) document.all = document;

if (!document.all.waterMark.style) document.all.waterMark.style =
document.all.waterMark;

wMark = document.all.waterMark.style;

wMark.width = markW;

wMark.height = markH;

navDOM = window.innerHeight; // Nav DOM flag

function setVals() {

barW = 0; // scrollbar compensation for PC Nav

barH = 0;

if (navDOM) {

if (document.height > innerHeight) barW = 20;

if (document.width > innerWidth) barH = 20;
```

```

} else {
innerWidth = document.body.clientWidth;
innerHeight = document.body.clientHeight;
}
posX = ((innerWidth - markW)-barW) * (markX/100);
posY = ((innerHeight - markH)-barH) * (markY/100);
}
function wRefresh() {
wMark.left = posX + (navDOM?pageXOffset:document.body.scrollLeft);
wMark.top = posY + (navDOM?pageYOffset:document.body.scrollTop);
}

function markMe() {
setVals();
window.onresize=setVals;
markID = setInterval ("wRefresh()",markRefresh);
}
window.onload=markMe;
//-->
</script>

```

Yüzen logo olarak kullanacağınız resmin eni-boyu kaç piksel ise, markW ve markX değişkenlerinin karşısına bu değerleri yazmalısınız. Ayrıca logonuzun ekranın neresinde sabit durmasını istiyorsanız, o noktanın koordinatlarını (yüzde olarak) markX ve markY değişkenlerinin karşısına yazabilirsiniz. Bu örnekte logo, ekranın sağ alt köşesinde duracak şekilde ayarlanmış bulunuyor. Bu değerler 0,0 olursa, logo ekranın sol üst köşesinde 50,50

olursa ortasında, 100,0 olursa sağ üst köşesinde, 0,100 olursa sol alt köşesinde durur. Yenilenme oranı olan milisaniye ise, Browser'ın logonun yerini kaç saniyede bir güncelleştirmesini istediğinizi gösterir. Bu süre ne kadar düşük olursa, logo o kadar yerinde sabit kalır.

CSS Eğlenceleri

CSS, kullanışlığı olduğu kadar eğlenceli de olabilir. DHTML-uyumlu veya CSS kodlarının tümünü anlayabilen bir Browser ile Mouse işaretçisinin görünümünü değiştirerek, ziyaretçilerinizi şaşırtabilirsiniz. Önce HTML belgesinin HEAD bölümüne şu stil kodlarını yazın:

```
<style>
<!--
.yesil{
color :green
}
.kahverengi{
color :brown
}
A.yardim {cursor:help}
A.mesgul {cursor:wait}
A.tasi {cursor:move}
A.oto {cursor:auto}
A:link {text-decoration:none;}
```

```
{ font-family: Verdana, Arial }
A:visited {text-decoration:none;}
A:hover {color: white; background:"#008080"; decoration=none
}
-->
</style>
```

Sonra, BODY bölümünde bir sınama yapalım:

```
<p><a class="yesil" href="#">Yeşil Bağlantı</a></p>
<p><a class="kahverengi" href="#">Kahverengi Bağlantı </a></p>
<p class="yesil">Bir Yeşil Daha Ama bu Bağlantı Değil</p>
<p><a class="mesgul" href="#">Meşgul.. Bekleyiniz</a></p>
<p><a class="yardim" href="#">Yardım</a></p>
<p><a class="tasi" href="#">Taşı</a></p>
<p><a class="oto" href="faq.html">Otomatik</a></p>
```

Bu kodun işleyebilmesi için ziyaretçinizin IE 4.0 sonrası bir Browser kullanıyor olması gerekir. NS kullanan ziyaretçiler etkiyi göremezler, o kadar.

Maviden gına geldi ise

Browserlar'a bağlantıları mavi harfle ve mavi çizgili çizgili yorumlattıran varsayılan stilden bıktı iseniz, aramıza hoşgeldiniz! Ama üzülmeyin. Mavilikten kurtulmanın bir yolu HEAD etiketi içinde şu stili oluşturmaktır:

```
<style>
<!--
a:link { text-decoration: none; color: #FF0000 }
A:visited {text-decoration:none; color:aqua}
```



```
a:hover { text-decoration: underline; color: #00FF00 }
-->
</style>
```

Tabii, renkleri zevkinize göre verebilirsiniz. Dikkat gerektiren tek nokta, ziyaret edilmiş bağlantı ile edilmemiş bağlantının renklerinin aynı olması, ziyaretçiyi şaşırtır.

Yazıcıya göre sayfa kesme

Özene bezene hazırladığınız sayfaların ziyaretçi tarafından yazıcıda bastırılması halinde ne şekil alacağını da hesaba katıyor musunuz? Bazı sayfaların, özellikle grafik bulunan sayfaların yazıcıdan nasıl çıkacağını önceden planlamak tasarımcının görevidir. Fakat yazıcılar arasındaki farklar ve ziyaretçinin varsayılan harfinin büyüklüğü veya küçüklüğü gibi faktörler yüzünden evdeki hesap bazen çarşığı tutmayabilir; en güzel grafiğiniz kağıda ortasından ikiye ayrılarak dökülebilir. Hiç değilse buna engel olabilir; sayfanızın dağıt üzerinde bir bütün oluşturmasını istediğiniz bölümünü belirleyebilirsiniz. HTML'in HEAD bölümüne şu stili girin:

```
<style type="text/css">
.yenisayfa (pagebreak-after: always)
</style>
```

Sonra, sayfanızı bölmek ve kağıt üzerinde yeni bir sayfa başlasın istediğiniz yerde, şu kodu yazın:

```
<br class="yenisayfa">
```

Sayfada tarih

Sayfalarınızda, günün adını, ay ve yılı, isterseniz, geri planında bir de güne belirten grafikle gösterebilirsiniz. Biraz uzunca görünmekle birlikte, herşeyi bir tablo içinde yapabilen bu kodu, istediğiniz sayfaya ithal ederek, bir kere yazdıktan sonra yeniden kullanmanız mümkün. (Bu kodu, yazarının adını belirten satıra yer vermek şartıyla, istediğiniz gibi kullanabilir ve değiştirebilirsiniz)

```
<TABLE WIDTH="75%" BORDER="0" CELLSPACING="5" CELLPADDING="5">
<tr><td WIDTH="100%">
<script LANGUAGE="JavaScript">
<!-- eski browseryardan gizleyelim
// DateCal   JavaScript v1.00 Copyright (c) 1998 Carolyn B. Allard
var bugun = new Date()
var gun = bugun.getDay()
var tablo
var ay = bugun.getMonth()
var tarih = bugun.getDate()
var yil = bugun.getYear()
var ay_uzun = "January"
var yil_uzun = 1900
var uzuntarih = "bugun"
if (gun == 0) {resim = "pazar.jpg"}
if (gun == 1) {resim = "pazartesi.jpg"}
if (gun == 2) {resim = "sali.jpg"}
if (gun == 3) {resim = "carsamba.jpg"}
```

```

if (gun == 4) {resim = "persembe.jpg"}
if (gun == 5) {resim = "cuma.jpg"}
if (gun == 6) {resim = "cumartesi.jpg"}
if (ay == 0) {ay_uzun = "Ocak"}
if (ay == 1) {ay_uzun = "Şubat"}
if (ay == 2) {ay_uzun = "Mart"}
if (ay == 3) {ay_uzun = "Nisan"}
if (ay == 4) {ay_uzun = "Mayıs"}
if (ay == 5) {ay_uzun = "Haziran"}
if (ay == 6) {ay_uzun = "Temmuz"}
if (ay == 7) {ay_uzun = "Ağustos"}
if (ay == 8) {ay_uzun = "Eylül"}
if (ay == 9) {ay_uzun = "Ekim"}
if (ay == 10) {ay_uzun = "Kasım"}
if (ay == 11) {ay_uzun = "Aralık"}
yil_uzun = 1900 + yil
uzuntarih = ay_uzun + " " + tarih + ", " + yil_uzun
tablo = '<TABLE WIDTH="120" BORDER="0" CELLSPACING="2" ';
tablo += 'CELLPADDING="0" HEIGHT="34">';
tablo += ' <TR>';
tablo += ' <TD WIDTH="120" HEIGHT="34"ALIGN="center"
VALIGN="bottom" BACKGROUND="';
tablo += resim;
tablo += '"><CENTER><FONT COLOR="#003399" FACE="Arial" SIZE="-1">';
tablo += uzuntarih;
tablo += ' </FONT></CENTER></TD>';

```

```
tablo += ' </TR>';  
tablo += '</TABLE>'  
document.write(tablo)  
// gizlemeye son-->  
</script></font><</td></tr></table>
```

Bu kodu yazarken, en çok dikkat edilmesi gereken bölümü, Javascript'e HTML yazdırtan son 16-17 satırıdır.; tek ve çift tırnakların yerine özellikle dikkat etmeniz gerekir.

Javascript ile açılan menü

Sayfanızın bir yerinde, Navigation çubuklarında gösterilemeyecek kadar çok belgeye veya sayfaya ulaşım imkanı vermek isteyebilirsiniz. Sözgelimi bir kitabın farklı bölümlerine ulaşma kolaylığı sağlamak isteyebilirsiniz. Ya da kendi sitenizin çeşitli sayfalarına bir menü ile ulaşılması daha az yer kaplaması bakımından uygun olabilir. Bunu FORM tekniği ile sağlamak mümkün. Önce HEAD bölümünde kısa bir Javascript yazalım:

```
<script language="JavaScript">  
<!--  
function goster(form) {  
var index=form.hedef.selectedIndex  
window.location=(form.hedef.options[index].value);  
}  
// -->  
</script>
```

Sonra, bu Script'in yardımıyla kullanacağımız açılan menü bölümü için bir

FORM oluşturalım:

```
<form name="menü">
<select name="hedef" size="1">
<option selected value="...">Bir sayfa seçin</option>
<option value="birinci.htm">Birinci Sayfa</option>
<option value="ikinci.htm">İkinci Sayfa</option>
<option value="ucuncu.htm">Üçüncü Sayfa</option>
<option value="dorduncu.htm">Dördüncü Sayfa</option>
</select>
<input type="BUTTON" value="Göster" onclick="goster(this.form)">
</form>
```

Tabii menüde gösterilecek kelimeler ve bunların karşılığı olan HTML sayfalarının adını kendi sitenize uygun şekilde düzeltmeniz gerekiyor.

Tek pencere yetmiyorsa

Diyelim ki sayfalarınızın içeriği öylesine arttı ki, artık size bir Browser penceresi yetmiyor ve ziyaretçi bir bağlantıyı tıkladığında bilgisayar ekranında ikinci bir pencere açılsın ve talep edilen içerik yeni pencerede gösterilsin istiyorsunuz. Yani size bir pop-up pencere gerekiyor. İkinci ve 640x480 piksellik pencere için kodumuz şöyle olabilir.

```
<script language="JavaScript">
<!--
function yeniPencere() {
```

```
popupWin = window.open('http://www.benimsitem.com/hedefsayfa.htm',
'remote', 'width=640,height=480')
}
// -->
</script>
```

Bu penceresi ziyaretçinin ekranında belirli bir konumda da açtırabiliriz.

Bunun için popupWin satırı şöyle olacak:

```
popupWin=window.open('http://www.psacake.com', 'remote', 'width=640,height=480,top=100,left=100')
```

Sonra, hangi bağlantı yeni bir pencere açtırsın istiyorsanız, o bağlantıyı şöyle yazabilirsiniz:

```
<form><input type=button value="Burayı tıklayınız!"
onClick="yeniPencere();"></form>
```

İsterseniz, bu kolaylıktan düz bağlantı metninde de yararlanabilirsiniz:

```
<a href="javascript:yeniPencere();">Burayı tıklayınız</a>
```

Açtıracığınız pencerenin menü ve araç çubuğu olup olmaması, kaydırma çubukları bulunup bulunmaması da bu arada belirtilebilir.

Kaynağı kolayca görüntülemek

Özellikle Internet, Web teknikleri, HTML ile ilgili bir siteniz varsa ve ziyaretçileriniz o anda izledikleri sayfanın kaynağını görmek için zahmet edip, Mouse'un sağ düğmesini tıklayıp, menü açarak kaynağı görüntülemekle uğraşmasınlar istiyorsanız, sayfanızın altına mesela bir "Kaynağı Görüntüle" satırı koyun, bu satırı şöyle bağlantılayın:

```
<script>document.writeln("<A ID='btm' HREF='view-source:"+document.location.href+"'>View Source</A>");</script>
```