

T.C.
MİLLİ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

ENDÜSTRİYEL OTOMASYON TEKNOLOJİLERİ

SUNUCU SERVİSLERİ-3

ANKARA 2008

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	iii
GİRİŞ	1
ÖĞRENME FAALİYETİ – 1	3
1. WEB SAYFASI TEMELLERİ	3
1.1. Web Data Hazırlama	3
1.1.1. Metin Editörü	4
1.1.2. Yayınlama	4
1.1.3. Sistemin Çalışması	4
1.1.4. Web Tarayıcıları	4
1.1.5. Web Sayfasının Temel Yapısı	5
1.1.6. Koda Açıklama Ekleme	7
1.2. Resimlerle Çalışma	7
1.3. Metinlere Bağlantı Kazandırmak	9
1.3.1. Gerçek Dünyada Tasarım	10
1.4. Listeler	11
1.4.1. Yan Yana Birden Çok Boşluk Kullanımı ve Özel Karakter Kodları	12
1.5. Çerçeveler	13
1.6. Tablolar	14
1.7. Formlar	17
1.7.1. Form Türleri	17
1.7.2. Metin Kutusu	17
1.7.4. İşaret Kutusu	17
1.7.5. Şifre Giriş Formu	17
1.7.6. Gönder Tuşu	17
1.7.7. Sıfırla Tuşu	18
UYGULAMA FAALİYETİ	19
ÖLÇME VE DEĞERLENDİRME	21
ÖĞRENME FAALİYETİ – 2	23
2. CGI PROGRAMLAMA	23
2.1. Perl Dili	24
2.1.1. Linux İçin İse Aşağıdaki Aşamaların Yerine Getirilmesi Gereklidir	24
2.2. CGI Programının Temelleri	25
2.3. C Dili Kullanımı	26
2.4. Perl Dili Kullanımı	26
2.4.1. Perl’de Değişkenlerin Kullanımı	28
2.4.2. HTML Etiketlerinin CGI Programında Kullanımı	28
2.4.3. Özel Karakterlerin Yazdırılması	29
2.4.2. For Döngüsünün Kullanımı	29
2.4.3. Dizi Yapısının Oluşturulması	29
2.4.4. Foreeach Döngüsünün Kullanımı	30
2.4.5. Form Etiketinin Kullanımı	30
2.4.6. HTML’deki A Etiketinin kullanımı	31
2.4.7. HTML’deki FRAME Etiketinin Kullanımı	32
2.4.8. Sprintf Fonksiyonu	33
2.4.9. Time Fonksiyonu	34

2.4.10. Localtime Fonksiyonu	34
2.4.11. Ana Programdan Alt program Çağırma.....	36
2.2.12. Dosyalarla Çalışma.....	37
2.4.13. Dosya Açılmadığı Zaman Hata Mesajının Görüntülenmesi	39
2.4.14. Dosyada Yer Alan Tüm Kayıtların Görüntülenmesi	40
2.4.15. Okuma Dosyası İçin Program ve Kayıt	42
2.4.16. Erişim Sayıcı Program Yapma	44
UYGULAMA FAALİYETİ	48
ÖLÇME VE DEĞERLENDİRME	49
MODÜL DEĞERLENDİRME	50
CEVAP ANAHTARLARI.....	51
KAYNAKÇA	52

AÇIKLAMALAR

KOD	481BB0082
ALAN	Endüstriyel Otomasyon Teknolojileri
DAL/MESLEK	Endüstriyel Kontrol Teknisyenliği
MODÜLÜN ADI	Sunucu Servisleri - 3
MODÜLÜN TANIMI	Web sayfası hazırlamak amacıyla gerekli HTML etiketleri ve sunucu tabanlı programlama yapmak için PERL dili komutlarının uygulama becerilerinin verildiği öğretim materyalidir.
SÜRE	40/32
ÖN KOŞUL	Sunucu Servisleri - 2 modülünü almış olmak.
YETERLİK	Sunucu servisleri kurmak.
MODÜLÜN AMACI	Genel Amaç Sunucu servislerini doğru olarak kurabileceksiniz. Amaçlar 1. HTML ile web sayfa tasarımını doğru olarak yapabileceksiniz. 2. CGI programlamayı doğru olarak yapabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam: Bilgisayar laboratuvarı Donanım: Bilgisayar, bilgisayar çevre birimleri.
ÖLÇME VE DEĞERLENDİRME	Her faaliyetin sonunda ölçme soruları ile öğrenme düzeyinizi ölçeceksiniz. Araştırmalarla, grup çalışmaları ve bireysel çalışmalarla öğretmen rehberliğinde ölçme ve değerlendirmeyi gerçekleştirebileceksiniz.

GİRİŞ

Sevgili Öğrenci,

Bu modülden itibaren temel HTML etiketlerini kullanmaya başlayacaksınız. HTML etiketleri değişik amaçlar için hazırlanan Web sayfalarının tasarımında kullanılmaktadır. Sadece HTML etiketlerini kullanarak hazırlanan sayfalar statik bir yapıda olurken, kullanıcı ile etkileşimli dinamik web sayfaları oluşturmak için tarayıcı tabanlı ya da sunucu tabanlı programlama yöntemlerinin kullanılması gerekmektedir.

Bu şekilde tarayıcı üzerinde ya da sunucu üzerinde çalışan programlara betik (script) adı verilmektedir. Sunucu Servisleri 3 modülü, 2 öğrenme faaliyetinden oluşmaktadır.

- HTML etiketleri ile web sayfası hazırlama,
- PERL dili kullanarak CGI (Common Gateway Interface) programları yazma.

Öğrenme faaliyetlerinde konu genel olarak değil örnekler üzerinde anlatılmıştır. Bu yöntem, yapacağınız uygulamalara rehberlik edecektir. Örneklerde, yazılacak kodların ekran görüntüsü ve tarayıcıda elde edilen görüntüsü verilecektir. Bu yol ile uygulamalarda yararlanacağınız işlem basamaklarını daha iyi anlayacaksınız.

ÖĞRENME FAALİYETİ-1

AMAÇ

HTML ile web sayfa tasarımını doğru olarak yapabileceksiniz.

ARAŞTIRMA

Bu öğrenme faaliyetinden önce aşağıdaki hazırlıkları yapmalısınız.

- Web sayfası tasarımında kullanılan paket programlar hakkında araştırma yapınız.
- Web sayfalarında kullanılmak üzere gerekli resim ve animasyonları hazırlamak için kullanılan programlar hakkında araştırma yapınız.

1. WEB SAYFASI TEMELLERİ

1.1. Web Data Hazırlama

HTML, (Hyper Text Markup language) özel etiketler kullanılarak belgelerin biçimlendirildiği bir dildir. Bu belgeler tarayıcı (browser) denilen programlar aracılığıyla görüntülenir. HTML yalnızca yayınlanacak bilgileri içeren bir metin dosyasıdır. Bir programlama dili değildir. HTML'nin farklı sürümleri, farklı özelliklere sahiptir (Html 1.0, Html 3.0 gb). Her yeni sürüm tam anlamıyla geriye doğru uyumludur (kendinden önceki sürümlerle uyumludur) ve HTML standartlarını birkaç adım daha ileri götürmektedir. Farklı bilgisayarlar ve kelime-işlem programları arasında, yazı dosyalarının biçimlendirilmesinde ortak bir yöntem bulma çabası, 1986'da, Dünya Standartlar Enstitüsü tarafından Standart Genelleştirilmiş İşaretleme Dili (Standart Generalized Markup Language, SGML) adıyla birleştirildi.

SGML ile ondan türetilen HTML ve XML dilleri kullanılarak oluşturulan belgeler, programlama dillerinden farklı olarak doğrudan bilgisayarın işletim sistemine değilse bile bir yazılıma, örneğin kelime-işlemcisine ya da tarayıcıya "aşağıdaki veriyi ekranda şöyle göster, yazıcıdan da şöyle çıkart" anlamına gelen komutları içerir.

Web sayfalarının, programlama dillerinde olduğu gibi bir kaynak kodu bulunur. Web sayfaları bu doğrultuda iki türlü hazırlanmaktadır. Bunlardan bazıları frontpage, dreamweaver gibi profesyonel programları kullanarak web sayfalarının hazırlanmasıdır. Diğer yolu ise, notepad gibi bir metin editörü ya da bazı editör programları kullanmaktır. Bir metin editörü ile oluşturulacak web sayfaları, bu konuda bir temel olması açısından önemlidir.

Web sayfasının hazırlanmasında kullanılan gerekli programlar şunlardır:

1.1.1. Metin Editörü

Windows'ta NotePad ya da WordPad, Linux'ta Vi ya da Gedit gibi basit metin editörlerini kullanabiliriz. Metin editörlerinde yazılan HTML kodu her zaman için mutlaka text file (metin dosyası - Salt Metin) olarak dosya uzantısı da .html olarak kaydedilmelidir. ".txt" olarak kaydedilen metin dosyasının uzantısı .html olarak da değiştirilebilir. Eğer dosya uzantısı olarak en fazla üç karakteri destekleyen işletim sistemleri kullanılıyorsa, dosya uzantısı .htm olarak verilebilir. Bir tarayıcıda görüntülenen web sayfasının kaynak kodunu elde etmek için ise sayfanın üzerinde, farenin sağ tuşu tıklandıktan sonra, kaynağı görüntüle seçeneği tıklanır. Bundan başka, internet explorer için görünüm menüsünden "kaynak" komut satırı tıklanarak aynı şekilde sayfanın kaynak kodu görüntülenebilir. Ayrıca, kod yazarken büyük küçük harf ayrımı yoktur.

1.1.2. Yayınlama

Web sunucu kurulu bilgisayarlarda işletim sistemine bağlı olarak, belirlenen yerde yeni bir dizin oluşturulur. Bu dizine girilerek boş bir metin dosyası oluşturulup dosya adı ve uzantısı index.html olarak değiştirilir. Index.html her zaman için ilk sayfa olacaktır. Web üzerine yüklendiğinde web sunucu index.html'yi ilk sayfa olarak görecektir. Index.html metin editöründe açılarak web sayfası kodu yazılabilir duruma getirilebilir. Web sayfası için hazırlanan index sayfasının, varsayılan olarak ayarlanmış dizine yerleştirilmesi gerekmektedir.

1.1.3. Sistemin Çalışması

Bir tarayıcının internet üzerinden web iletişimi şu şekilde gerçekleşir. Tarayıcı iletişim kuracağı web sunucudan istekte bulunur. Web sunucu üzerinde yüklü olan web sitesinin içeriğini istemci bilgisayara geri döndürür. Günümüzde kullanılan birçok farklı tarayıcı çeşidi vardır ve her geçen gün yeni versiyonları çıkmaktadır. Bu şekilde destekledikleri özellik sayısı artmaktadır. Bu yüzden bazen web sayfaları farklı tarayıcılarda, farklı görünümlere sahip olabilir.

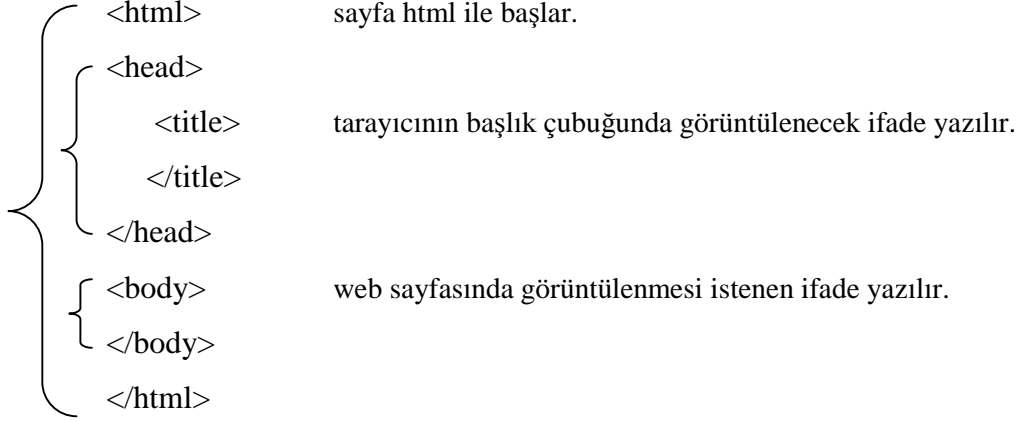
1.1.4. Web Tarayıcıları

Web tasarımcısının, HTML komutları kadar, hatta onlardan da çok iyi bilmesi gereken, tarayıcıların HTML'i nasıl yorumladığıdır. Bu nedenle bir Web tasarımcısının bilgisayarında, web sunucu yazılımı bulunmayabilir (sayfalarına başka bir internet web sunucu hizmeti veren kişi veya firma ev sahipliği yapıyor olabilir), ama mutlaka piyasaya mevcut web tarayıcılarının hemen bütün geçerli sürümleri bulunmalıdır. Netscape firmasının Navigator ve Communicator adıyla piyasaya sürdüğü farklı sürümleri ayrı ayrı dizinlerde durmak şartıyla aynı bilgisayarda çalışabilir. Ancak Microsoft firmasının Internet Explorer adlı programının farklı sürümleri aynı Windows ortamında bir arada bulunamazlar.

Neden değişik tarayıcıların değişik sürümlerine ihtiyacınız var? Bu sorunun cevabı, HTML'in internet'in ortak dili olduğu gerçeğine bir ölçüde gölge düşürecektir. Çünkü ortak

bir HTML dili bulunmasına rağmen, tarayıcıların ve aynı tarayıcının farklı sürümlerinin HTML'i yorumlayışı farklıdır.

1.1.5. Web Sayfasının Temel Yapısı



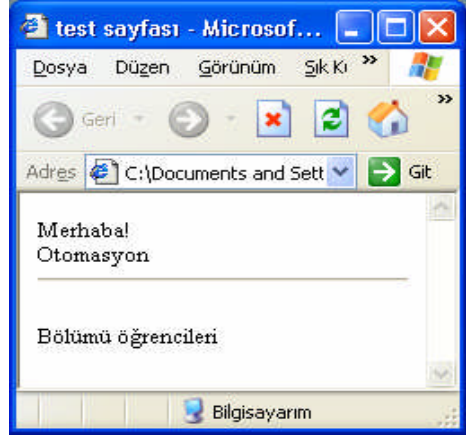
HTML etiketleri < > şeklinde ifade edilen işaretlerin arasına yazılan HTML deyimleridir. Hemen hemen her kodun başlangıcı < > ve bitişi </ > şeklindedir. Bu ibareler arasında kalan işlemler ilgili etiket tarafından icra edilir. Bazı deyimlerin bir veya daha fazla parametresi vardır. Bu parametrelerin hepsi seçimlidir. Yani kullanma zorunluluğu yoktur. Fakat kullanıldığı takdirde deyim zenginlik katar. Ayrıca bu parametrelerin kullanılması için belli bir sıralama yoktur. Bundan başka yukarıdaki etiketleri kullanmadan da, tarayıcımıza bir şeyler yazdırabiliriz. Fakat amacımız sadece yazı yazdırmak değil de, bazı programlar(script) kullanarak sayfamızı daha etkileşimli hale getirmek ise, bu etiketlerin yazılmaması durumunda, sayfa görüntülenemeyecektir.

HTML belgesi, iki bölüme ayrılır. Baş taraf (başlangıç) (<HEAD>) ve gövde (<BODY>) bölümleri. Web tarayıcılar, bir belgeyi istediğiniz tarzda yorumlayabilmek için, HTML etiketini gördükten sonra, HEAD ve BODY etiketlerini arar ve ekrandaki sayfayı buna göre biçimlendirirler. Sayfanın baş tarafı (head), sayfanın en üstünde, örneğin bir gazetenin başlığı gibi gösterilen bir metin olmayıp, belge hakkında genel bilgileri kapsayan bölümdür. Burada yer alabilecek genel etiketler (meta tag), örneğin;

<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">, HTML sayfalarının İngilizce metinlerden oluştuğunu varsayan tarayıcıya, sayfamızın Türkçe olduğunu bildirmek için kullanılır.

Örnek 1.1:

```
<html>
<head>
  <title>test sayfası</title>
</head>
<body>
  Merhaba!<br>
  Otomasyon
  <hr>
  <p>Bölümü öğrencileri</p>
</body>
</html>
```



Şekil 1.1: hr ve br etiketi

Yukarıdaki örnekte “merhaba” ifadesi yazıldıktan sonra
 etiketi ile bir alt satıra geçilir, daha sonra Otomasyon ifadesi yazdırılır. <hr> etiketi, yatay bir çizgi çizmek için kullanılır. <p> etiketi ise metne paragraf özelliği kazandırır.

Örnek 1.2:

```
<html>
<head>  <title>test sayfası</title>
</head>
<body>
  <h1>otomasyon bölümü</h1>
  <h2> otomasyon bölümü </h2>
  <h3> otomasyon bölümü </h3>
  <h4> otomasyon bölümü </h4>
  <h5> otomasyon bölümü </h5>
  <h6> otomasyon bölümü </h6>
</body>
</html>
```



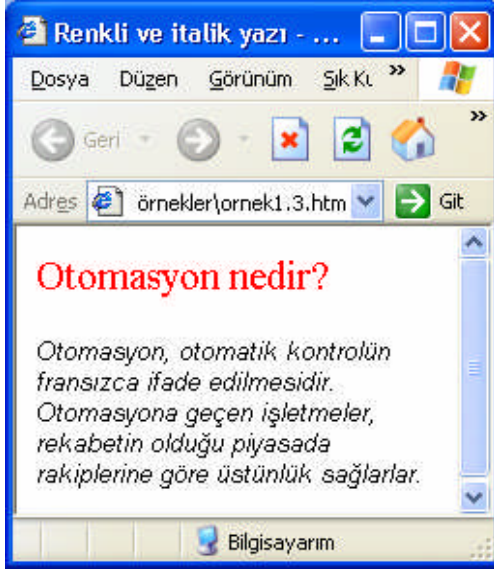
Şekil 1.2: Başlık etiketi

Yukarıdaki örnekte h1, h2, h3, h4, h5, h6 etiketleri ile farklı başlık seviyeleri oluşturulmuştur.

Örnek 1.3:

```
<html>
<head>  <title>Renkli ve italik yazı</title>
</head>
```

```
<body><p>
  <font color="red" size="+2">Otomasyon nedir?</font></p><i>
  <font face="arial">Otomasyon, otomatik kontrolün<br>Fransızca ifade
edilmesidir. <br>Otomasyona geçen işletmeler,<br>Rekabetin olduğu piyasada
<br>rakiplerine göre üstünlük sağlarlar. </font></i>
</body>
</html>
```



Şekil 1.3: Font Etiketi

1.1.6. Koda Açıklama Ekleme

<!-- --> Bu etiket küçüktür işaretiyle başlar ve büyüktür işaretiyle biter. Ünlem işareti, tarayıcının ondan sonra gelen bilgileri sanki yokmuş gibi dikkate almasını sağlar. Bu etiketin içine bir yorum eklenerek, HTML dokümanı bölümlere ayrılabilir. Yorum etiketleri, açıklama eklemek istenilen herhangi bir yerde kullanılabilir.

1.2. Resimlerle Çalışma

 Bu etiket, resmin kaynağını (yolunu) kullanarak, tarayıcıda göstermek için kullanılır. Bu ifadede resim.jpg dosyasının, kullanılan html sayfasıyla aynı klasörde olduğu varsayılmıştır.

Aşağıdaki örnek için Fedora işletim sisteminde bulunan html klasörünün altına (/var/www/html) winter.jpg isimli bir resim kopyalayınız. Ayrıca html klasörünün altına resim isimli bir klasör oluşturunuz. Resim klasörünün içine sunset.jpg isimli bir resim kopyalayınız. Son olarak www (/var/www) klasörünün altına bluehills.jpg isimli bir resim kopyalayınız.

 etiketi metnin rengini, boyutunu ve tipini değiştirmek için kullanılır. "Size" parametresindeki "+2" ifadesi metnin normalden iki kat daha büyük yazılacağını ifade eder. "Color" parametresindeki "red" değeri yerine renklerin hexadecimal kodları da yazılabilir. Örneğin ifadesi de metne renk özelliği kazandırır. <i> etiketi ise, yazının italik yazılmasını sağlar.

Örnek 1.4:

<!--sayfa en son Ocak 2007'de güncellendi-->

<html>

<head>

<title>resim ekleme </title>

</head>

<body>

<p><h2>resimleri web sayfalarına ekleyebiliriz</h2></p>

<p> bahar mevsiminin güzelliklerini sergileyen bir

resim <p>yaz mevsiminin güzelliklerini sergileyen bir

resim <p> sonbahar mevsiminin

güzelliklerini sergileyen bir resim

</body>

</html>



Şekil 1.4: img etiketi

Bu örnekte resimlerin bulunduğu yer dikkate alınmalıdır. Birinci bölümde winter.jpg resim dosyası web sitesinin bulunduğu dizinin içindedir. Dolayısıyla yol tanımlamaya gerek yoktur. İkinci bölümde sunset.jpg resim dosyası Resim dizini içindedir. Bu dizin web sitesinin içinde bulunduğu dizin altında oluşturulmuştur. Dolayısıyla yol tanımlaması Resim/sunset.jpg şeklinde olmalıdır. Üçüncü bölümde ise bluehills.jpg resim dosyası web sitesinin içinde bulunduğu dizinin üstünde açılmıştır. Dolayısıyla yol tanımlama ../bluehills.jpg şeklinde olacaktır.

Etiketler	Parametreler	Fonksiyon
	<ul style="list-style-type: none"> ➤ Src ➤ Alt ➤ Width, Height ➤ Align ➤ Border 	<ul style="list-style-type: none"> ➤ Resmi web sayfasına yerleştirir. ➤ Resim üzerine geldiğimizde görüntülenecek mesajı belirler. ➤ Resmin boyutlarını belirler ➤ Yerleştirildiği alan için sola, sağa veya ortaya hizalama yapar. ➤ Resmin çerçeve kalınlığını belirler.

Şekil 1.5: img etiketi parametreleri

1.3. Metinlere Bağlantı Kazandırmak

Anchor, etiketi (<A>..<), metinlere bağlantı özelliği kazandıran bir etikettir. Aşağıdakiler bu etiketin parametreleridir.

HREF="url": URL, (Uniform Resource Locator) internette adres demektir. Bu adres, kendi sabit diskinizde bir klasör (ve alt-klasörler) içindeki bir dosyanın adı olabileceği gibi, HTTP, FTP veya elektronik posta yoluyla ulaşılabilecek bir web alanı ve o alanın içindeki bir dosya olabilir.

NAME="metin": Anchor'a isim vererek, daha sonra bu noktaya atıfta bulunma imkânı kazandırır. Bunu, bir sayfanın kendi içinde, belirli yerleri, örneğin baş tarafı, belirlemek için kullanabilirsiniz. Çok uzun bir sayfanın baş tarafına

 şeklinde bir "isimlendirilmiş Anchor" noktası koyarsanız ve aşağıda baş tarafa dönüşü kolaylaştırmak için şöyle bir bağlantı yapılabilir.

Baş tarafa dönmek için burayı tıklayınız

TARGET="pencere": Bu bağlantı sağlandığı zaman alınacak HTML sayfasının nerede kullanılacağını gösterir. FRAMESET etiketi bulunan bir sayfada kullanılması halinde, alınacak sayfanın hangi çerçeveye konulacağını gösterir. Burada "pencere" kelimesinin yerine şu değerler yazılabilir:

"çerçeve adı": Oluşturulan çerçevelere önceden isim verilmiş ise, o isimler buraya yazılmak suretiyle, alınacak HTML sayfasının hangi çerçeveye yerleştirileceği belirtilebilir. Çerçeve isimleri mutlaka rakam veya harfle başlamalıdır.

blank: Alınacak sayfa veya unsur için yeni bir tarayıcı penceresi açılır.

parent : Alınacak unsur, o anda açık sayfayı oluşturmuş bir ana sayfa varsa onun yerine konulur.

self : Alınacak sayfa mevcut sayfanın bulunduğu tarayıcı penceresine konulur.

top : Alınacak sayfa mevcut pencereye en üstten itibaren konulur.

Örnek 1.5:

```
<html>
<head>
<title>Link verme </title>
</head>
  <body>
    <p><a name="top"></a>
    <h1>Sayfanın en üstü.</h1>
    <p><a href="#bottom">sayfanın en altına gitmek için tıklayınız.
    </a><br><br><br><br><br><br><br><br><br><br><br><br><br>
    <br><br><br><br><br><br><br><br><br><br><br><br><br><br>
    <br><br><br><br><br><br><br><br><br><br><br><br><br><br>
    <p><a href="#top">sayfanın en üstüne gitmek için tıklayınız</a>
    <p><a name="bottom"></a>
    <h1>Sayfanın en altı.</h1>
  </body>
</html>
```

1.3.1. Gerçek Dünyada Tasarım

İnternette web sayfalarına erişilirken bir sayfanın yüklenmesini beklemek bazen sıkıcı olabilir. Kodunuzu daha verimli hale getirmeyi düşündüğünüzde, resim dosyalarından birkaç kilobyte fedakârlık edilebilir. Ancak bu durumda koda da bakmaya gerek olabilir. Fazladan konulan boşluk, gereksiz bir etiket ve bir cümlenin sonunda bir nokta olabilir. Dosyadaki her karakter dosya boyunu, yani web üzerinden aktarım süresini arttırır. En sık görülen zaman harcayıcılar, boş . ve <I>.</I> etiketleridir.

Koda bakmanın bir diğer faydası, küçük hataların sorun haline gelmeden önce tespit edilmesidir. Bu şekilde ilerde bulması zorlaşacak hataları önceden tespit etmek yerinde olacaktır.

Örnek 1.6:

```
<html>

<head><title>test sayfası</title></head>
  <body>
    <p><sup>super</sup>script, <sub>sub</sub>script
    <p><strike>strike</strike>, <u>altı çizgili</u>
    <blockquote>
    Tablolar sayfa biçimlendirmede en uygun yöntemdir.
    </blockquote>
    <pre>
      1  2  3  4  5  6  7
```


8 9 10 11 12 13 14

-- sıralı sayılar --

```
</pre>
</body>
</html>
```

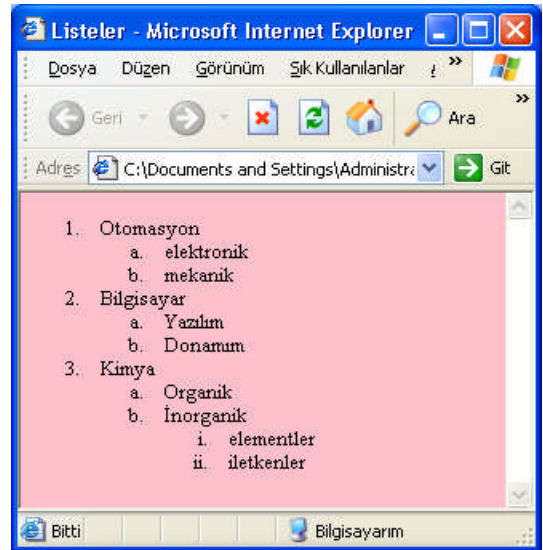
Burada kullanılan **<sup>** etiketi üst simge yapmak için, **<sub>** etiketi ise alt simge yapmaya yarar. **<strike>** etiketi ise metnin üzerini çizmek için kullanılır. **<blockquote>** etiketi yazı bloklarını cazip bir şekilde sunmak için etkili bir yoldur. Sayfa boşluklarının ve diğer boşlukların ayarlanması için kullanılan bir yöntemdir. **<pre>** etiketinden sonra kullanılan tüm ifadeler olduğu gibi tarayıcıda gösterilecektir.

1.4. Listeler

Listeler sıralı ve sırasız olmak üzere ikiye ayrılır. Aşağıdaki örnek sıralı liste oluşturmayı göstermektedir.

Örnek 1.7

```
<html>
<head><title>Listeler</title>
</head>
<body bgcolor=pink>
<ol type="1">
<li>Otomasyon
<ol type="a">
<li>elektronik
<li>mekanik
</ol>
<li>Bilgisayar
<ol type="a">
<li>Yazılım
<li>Donamım
</ol>
<li>Kimya
<ol type="a">
<li>Organik
<li>İnorganik
<ol type="i">
<li>elementler
<li>iletkenler
</ol>
</ol>
</ol>
</body>
</html>
```

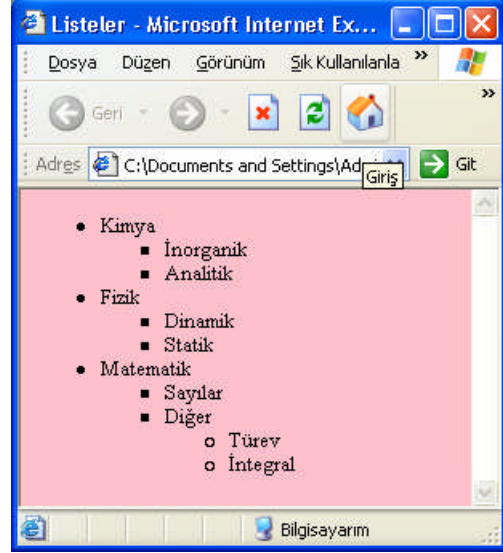


Şekil 1.6: Sıralı listeler

Aşağıdaki örnek sırasız listelemeyi göstermektedir.

Örnek 1.8:

```
<html>
<head>
<title>Listeler</title>
</head>
<body bgcolor=pink>
  <ul type="disc">
    <li>Kimya
      <ul type="square">
        <li>İnorganik
        <li>Analitik
      </ul>
    <li>Fizik
      <ul type="square">
        <li>Dinamik
        <li>Statik
      </ul>
    <li>Matematik
      <ul type="square">
        <li>Sayılar
        <li>Diğer
          <ul type="circle">
            <li>Türev
            <li>İntegral
          </ul>
        </ul>
      </ul>
  </ul>
</body>
</html>
```



Şekil 1.7: Sıralı olmayan listeler

1.4.1. Yan Yana Birden Çok Boşluk Kullanımı ve Özel Karakter Kodları

Yan yana birden çok boşluk koymanın bir yolu özel karakterler kullanmaktır. Normalde klavyeden giremediğiniz özel karakterleri, karşılık gelen kodlarla yazabiliriz. Tüm özel karakter & ile ; arasında yer alan kodlarla gösterilir. En çok kullanılan özel karakterlerden bazıları şunlardır.

 	: Boşluk
©	: © işareti
<	: <
>	: >
&	: &

Yan yana birden çok yazarak birden çok boşluğun görüntülenmesini sağlanabilir. Böylece paragraflara içeriden başlanabilir veya iki kelime arasında birden çok boşluk bırakılabilir. #include <stdio.h> ifadesi #include <stdio.h> , ifadesi ise şeklinde tarayıcıda görüntülenecektir.

1.5. Çerçevesler

Örnek 1.9:

index.html

```
<html>
<head>
<title>Çerçeve yapılması</title>
</head>
<frameset cols="20%,80%">
  <frame src="a.html" name="ilk">
  <frame src="b.html" name="ana">
</frameset>
</html>
```

a.html

```
<html>
<head><title>Tablo yapılması</title>
</head>
<body bgcolor=pink>
<a href="link.htm" target="ana">merhaba
</a>
</body>
</html>
```

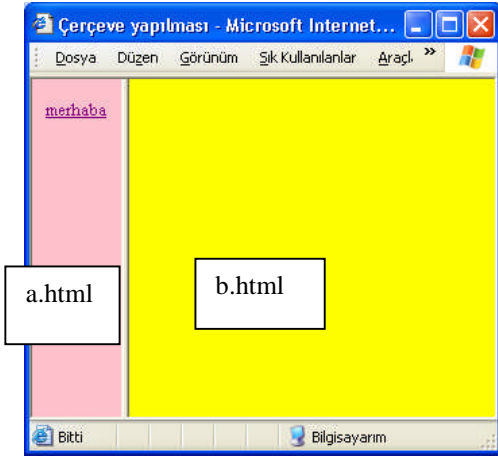
link.html

```
<html>
<head>
<title>Çerçevede Link</title>
</head>
<body bgcolor=pink>
MERHABA DÜNYA
</body>
</html>
```

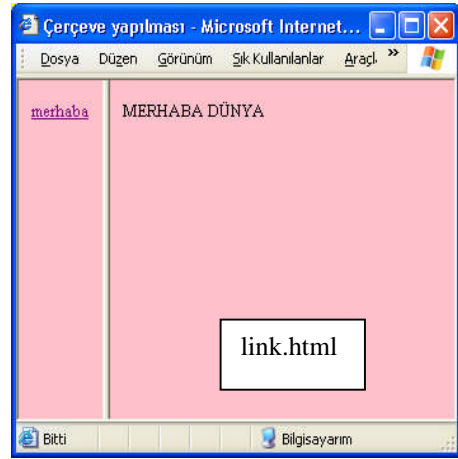
b.html

```
<html>
<head>
<title>Tablo yapılması</title>
</head>
<body bgcolor=yellow>
</body>
</html>
```

Linke Tıklandıktan Sonra



Şekil 1.8: Frame etiketi



Şekil 1.9: Frame etiketi ve link verme

Çerçevelerin kullanımında ihtiyacınız olan en az iki adet farklı web sayfasıdır. Bunların tek bir web sayfasında belirli oranlarda görüntülenmesi için frame yapısı kullanılır. Çerçeve kullanımı web sayfanızı daha görsel hale getireceği için tercih edilebilir.

Frame etiketinin name parametresi ile çerçevelere isim verilebilir. Bu isimler link verilirken kullanılarak, sayfanın istenilen çerçevede görüntülenmesini sağlamak için kullanılabilir.

1.6. Tablolar

Table etiketinin birçok parametresi vardır. Komutun en basit şekli:

```
<table> <tr> <td> </td> </tr> </table>
```

Bu komuta eklenebilecek parametreler aşağıda tanımlanmıştır.

“Width” tablonun genişliğinin piksel olarak belirtilmesidir. Tablo, belirlenecek bu değerden büyük olamayacaktır.

“Border” tabloda kullanılacak olan çerçevenin piksel olarak kalınlığıdır.

“Cellspacing” parametresi, tablonun içinde bulunan kutuların birbirleri arasındaki mesafeyi piksel olarak belirler. Bu değer boşluk mesafesinin değeridir.

“Cellpadding” parametresi de aynı şekilde tablonun içinde bulunan öğelerin etrafındaki yakınlık mesafesidir.

“bgcolor” parametresi tablonun zeminindeki rengi html kodlarıyla belirlenmesini gerçekleştirir.

“bordercolor” parametresiyle, çerçevelerde kullanılacak renkleri belirler.

“<tr> </tr>” satırları tanımlar

“<td> </td>” sütunları tanımlar.

1 satır ve 2 sütundan oluşan tablo

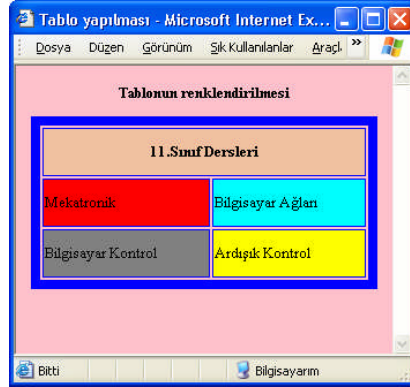
```
<tr>
<td>1</td><td>2</td>
</tr>
```

2 satır ve 1 sütundan oluşan tablo

```
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
```

Örnek 1 .10:

```
<html>
<head>
<title>Tablo yapılması</title>
</head>
<body bgcolor="pink">
<p>
<table border="8" bordercolor="blue" width="300" height="150" align="center">
<caption align="center"><b>Tablonun renklendirilmesi</b></caption>
<tr>
<td colspan="2" bgcolor="fucia" align="center"><b>11.Sınıf Dersleri</b>
</td></tr>
<tr>
<td bgcolor="red">Mekatronik</td><td bgcolor="aqua">Bilgisayar Ağları</td>
</tr>
<tr>
<td bgcolor="gray">Bilgisayar Kontrol</td>
<td bgcolor="yellow">Ardışık Kontrol </td>
</tr>
</table><br>
</body>
</html>
```



Şekil 1.10: Table etiketi

<Table> etiketinin içerisinde tablo için herhangi bir açıklama yapmak istiyorsak caption etiketi kullanılır. Yukarıdaki örnekte tablonun ilk satırında iki sütunun birleştirmek amacıyla <td> etiketinin colspan parametresi kullanılmıştır. Eğer satırlar birleştirilmek istenirse kullanılacak etiket rowspan olacaktır. Hücrelerdeki metinler biçimlendirilmek istenirse bu durumda <td> etiketinin içerisinde verilmek istenen biçimin parametresi kullanılmalıdır. Tablonun tarayıcıda kaplayacağı yer, width ve height parametreleri ile piksel cinsinden girilebileceği gibi, tarayıcının belirli bir bölümünü kaplaması için % cinsinden de girilebilir.

Örnek 1.11:

```

<html>
<body>
<table border="1" width="75%">
<tr>
<td rowspan="2" align="center">Kutu 1</td>
<td align="center">Kutu 2</td>
<td align="center">Kutu 3</td>
</tr>
<tr>
<td colspan="2" align="center">Kutu 4</td>
</tr>
</table>
</body>
</html>

```



Şekil 1.11: Rowspan ve Colspan parametresi

1.7. Formlar

Web sayfalarında kullanılan formlar sayfayı ziyaret eden kişinin siteye formlar aracılığıyla metin yollamasını, bir liste varsa bu listeden bir seçenek seçmesini sağlayan HTML komutlarından oluşmaktadır.

1.7.1. Form Türleri

Metin, radyo düğmesi, işaret kutusu, şifre, gönder tuşu, sıfırla tuşu, HTML dilinde kullanılan formlardır.

1.7.2. Metin Kutusu

Form üzerinde bilgi alınmasını sağlayan boş kutucuklardır.

```
<input type="Text" name="isim" value="deger" align="TOP" size="10"
maxlength="10">
```

1.7.3. Radyo Düğmesi

Bu düğme seçmeli sınavlarda kullanılan bir ya da birden çok ögenin seçilebilmesi için kullanılan, bilginin seçimini gösteren bir düğmedir.

```
<input type="Radio" name="isim" value="deger" align="TOP">
```

1.7.4. İşaret Kutusu

İşaret kutusunun Radyo kutusundan belirgin bir farkı, radyo kutusunun bir defa seçildiğinde seçimin bir daha kaldırılamamasıdır. Ancak işaret kutusunda bu seçim tekrar üzerine tıklanıldığında kaldırılabilir.

```
<input type="Checkbox" name="isim" value="deger" align="TOP">
```

1.7.5. Şifre Giriş Formu

Şifrelerin yazılabilmesi için özel bir kutucuk bulunmaktadır. Bu kutucuk yazılan şifrenin o anda başkası tarafından okunmaması için yazılanları **** olarak göstermektedir.

```
<input type="Password" name="isim" align="TOP" size="10" maxlength="12">
```

1.7.6. Gönder Tuşu

Gönder tuşu(submit) bir bilginin gönderilmesi gerektiği zaman, bu tuşa basılarak gönderme işlemi gerçekleştirilir.

```
<input type="Submit" name="isim" value="basınız" align="TOP">
```

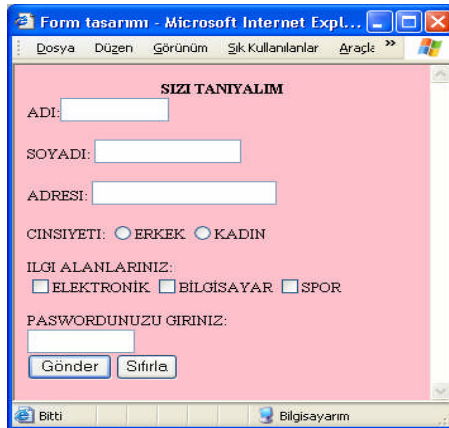
1.7.7. Sıfırla Tuşu

Web sayfasında formlara girilen bilgilerin sıfırlanması istendiğinde, yapılması gereken bir reset tuşu hazırlamaktır. Bu tuşa basılmasıyla girilen tüm bilgiler anında silinecek ve sayfa yeni yüklenmiş gibi olacaktır.

```
<input type="Reset" name="isim" value="Sıfırla" align="TOP">
```

Örnek 1.12:

```
<html><head><title>Form tasarımı</title></head>
<body bgcolor="pink">
<caption><center><b>SIZI TANIYALIM</b></center></caption>
<form>ADI:<input type="Text" name="isim" align="TOP" size="10"
maxlength="10"> <br><br>
SOYADI:<input type="Text" name="isim" align="TOP" size="15" maxlength="10">
<br><br>
ADRESI:<input type="Text" name="isim" align="TOP" size="20" maxlength="10">
<br><br>
CINSİYETİ: <input type="Radio" name="isim" value="erkek">ERKEK<input
type="Radio" name="isim" value="kadın">KADIN <br> <br>
ILGI ALANLARINIZ:<br><input type="Checkbox" name="isim" value="değer"
align="TOP">ELEKTRONİK<input type="Checkbox" name="isim" value="değer"
align="TOP">BİLGİSAYAR<input type="Checkbox" name="isim" value="değer"
align="TOP">SPOR<br><br>
PASSWORDUNUZU GIRINIZ:<br><input type="Password" name="isim"
align="TOP" size="10" maxlength="12"><br><input type="Submit" name="isim"
value="Gönder" align="TOP"> <input type="Reset" name="isim" value="Sıfırla"
align="TOP"></form>
</body>
</html>
```



Şekil 1.12: Form etiketi

UYGULAMA FAALİYETİ

Aşağıdaki sorulara ilişkin uygulama faaliyetini yapınız.

- Aşağıda verilen tabloyu html etiketleri kullanarak tarayıcınızda oluşturunuz.
(Tablo tarayıcınızın tamamını kapsasın ve kenarlık değeri “1” olsun.)

1			
2	3		6
4	5		

- Çalıştığınız klasörden bir üst seviyedeki klasörde bulunan “sayfa1.html” adlı dosyaya çalıştığınız klasörde bulunan bir resim üzerinden link verip yeni bir sayfada görüntülenmesini sağlayınız.
(resmin üzerine tıklandığında sayfa1.html yeni bir sayfada açılacak)

- Aşağıda olduğu gibi bir tablo oluşturarak tablonun ilgili hücrelerine resim yerleştiriniz.
1. Resim : Genişlik 40 Yükseklik= 80 Çerçeve= yok
 2. Resim : Genişlik ve Yükseklik= 80
- Tablo : Kenarlık = var



İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Temel html etiketlerini (html, head, body) etiketlerini kodunuza yerleştiriniz.➤ Amacınıza uygun kullanacağınız etiketlerin varsa parametrelerini yerleştiriniz.➤ Yazdığınız dosyanın uzantısı .html (/var/www/html klasörünün içine) olacak şekilde kaydediniz.➤ Tarayıcı kullanarak oluşturduğunuz sayfayı görüntüleyiniz.	<ul style="list-style-type: none">➤ İhtiyacınız olan etiketleri tespit ediniz. Gereksiz etiket kullanımından kaçınız.➤ İç içe geçen etiketler kullanırken, etiketlerin uygun sırada yerleştirildiğinden emin olun.➤ Yazdığınız kodu daha sonra incelerken kolaylık olması açısından, etiketlerin bazılarının diğerlerinden daha girintili yazılması gerektiğini unutmayın.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları cevaplayarak bu faaliyette kazandığınız bilgileri ölçünüz.

OBJEKTİF TEST (ÖLÇME SORULARI)

1. En büyük başlık oluşturmak için gerekli HTML etiketi aşağıdakilerden hangisidir?
A) <head>
B) <h1>
C) <h6>
D) <heading>
2. Bir satır boşluk vermek için gereken HTML etiketi hangisidir?
A) <hr>
B) <lb>
C) <break>
D)

3. Aşağıdakilerden hangisi sayfanın arka plan rengini değiştirmek için kullanılır?
A) <background>yellow</background>
B) <body color="yellow">
C) <body bgcolor="yellow">
D) <body background="yellow">
4. Bir metne link vermek için gereken HTML kodu aşağıdakilerden hangisinde doğru olarak verilmiştir.
A) Sınırsız Bilgi Paylaşımı
B) Sınırsız Bilgi Paylaşımı
C) <a>http:// www.mazharzorlu.k12.tr
D) Sınırsız Bilgi Paylaşımı
5. Bir linki yeni bir tarayıcı sayfasında açmak için hangi HTML kodu kullanılır?
A)
B)
C)
D)
6. Aşağıda verilenlerden hangilerininin tamamı tablo etiketleridir?
A) <table><tr><td>
B) <table><body><td>
C) <table><tr><td>
D) <table><head><tfoot>

7. Bir tablo hücresinin içeriğini sola hizalı yapmak için gereken etiket hangi şıkta doğru olarak verilmiştir.

- A) <td valign="left">
- B) <td left>
- C) <td leftalign>
- D) <td align="left">

8. Sayılardan oluşan sıralı bir liste oluşturmak için hangi etiket kullanılır?

- A)
- B) <list>
- C)
- D) <dl>

9. "checkbox" oluşturmak için gerekli HTML etiketi aşağıdakilerden hangisidir?

- A) <checkbox>
- B) <input type="check">
- C) <check>
- D) <input type="checkbox">

10. Web sayfasına bir resim yerleştirmek için gereken HTML kodu hangisidir?

- A) image.gif
- B)
- C)
- D) <image src="image.gif">

DEĞERLENDİRME:

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

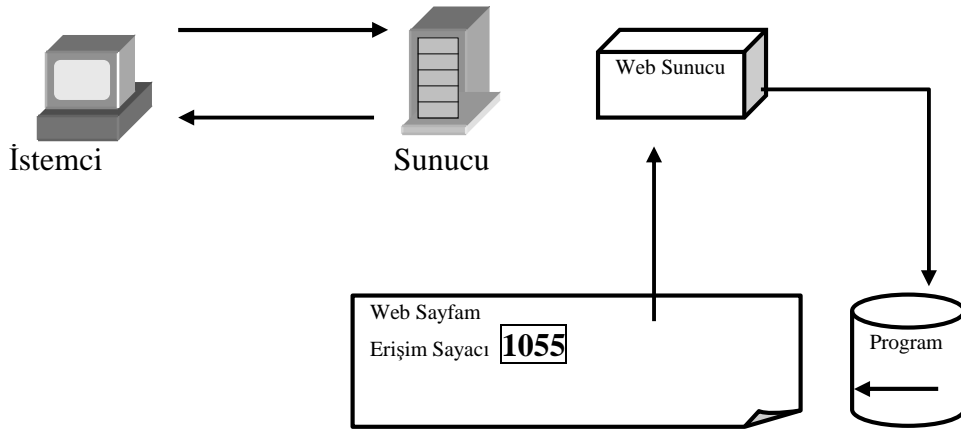
CGI (Common Gateway Interface) programlamayı doğru olarak yapabileceksiniz.

ARAŞTIRMA

- Tarayıcı tabanlı programlama ile sunucu tabanlı programlama arasındaki farklılıklar nelerdir?

2. CGI PROGRAMLAMA

CGI (Common Gateway Interface) web sitelerinde çeşitli servisler sağlar. CGI, bir programlama dili ismi değildir. CGI, program ve Web sunucu arasında iletişimi başarmak için bir standarttır. CGI ile ziyaretçi tarafından sunucuya bir iş yaptırılır. Bu iş ziyaretçinin veritabanı olarak kullanılan dosyalara belli konularda kayıt yapabilmesi şeklinde olabileceği gibi ziyaretçi sayısının sayılması, web üzerinden mail gönderilmesi, ziyaretçinin kişisel bilgilerinin tutulması, bir anket uygulaması olabilir. CGI programlarında “c”, “visual basic scripting” gibi diller kullanılabilir de genel olarak CGI programlarında perl dili kullanılır. Bunun sebebi geniş web sunucu desteği ve perl dilinin köklü ve geniş bir dil oluşudur.



Şekil 2.1: CGI programının çalışması

Yukarıda giriş sayısını tutan bir sayıcı örneği gösterilmiştir. İstemciden gelen istek, yukarıda gösterildiği gibi sunucu tarafından alınır ve web sunucu üzerinde çalışan CGI programı kullanılarak sunucu makinesi üzerinde işlenir. Üretilen sonuç istemcide çalışan tarayıcı üzerinde görüntülenir.

Perl'in sayısal gösterimi C diline benzer. Perl'i derlemeye gerek yoktur. Bu yüzden, programlama'dan sonra, işlemler test edilmeye hazırdır. Ayrıca, Perl varsayılan olarak UNIX ve Linux işletim sisteminde kurulu gelir.

2.1. Perl Dili

Perl, çok zor ve karışık görevleri yerine getirmek için yeterince güçlü bir dildir. Yaygın olarak kullanılır ve çoğu zaman Linux sunucu üzerinde kuruludur. Perl programlarını derlemeye gerek yoktur. Yazılan kodlar, kaydedilir ve çalıştırılır. Programın kendisi sadece bir metin dosyasıdır, Perl yorumcusu bütün işlemi gerçekleştirir. Bunun avantajı, programınızı küçük değişikliklerle herhangi bir makineye taşıyabilmedir. Derleme ihtiyacı olmadığı için, programı çalıştırmadan, herhangi bir hata tespit edilemez.

CGI programlarınızı (script-betik) kendi yerel makinenizde ya da Linux kabuğunda çalışan herhangi bir programla yazabilirsiniz (pico, gedit, vi vb.). Programınızı uzak bir bilgisayarda yazıyorsanız, bir FTP programı aracılığı ile bunu web sunucunuza yüklemeniz gerekecektir (upload).

Burada şunu söylemek gerekli olacaktır ki CGI programlarınızı ASCII (plain text files) olarak yüklemeniz gereklidir. Eğer binary olarak yüklenirse satır sonlarında birçok kontrol karakteri ile karşılaşılacaktır. Ve bu da programınızda hatalara sebep olacaktır.

Web sunucunun kurulu olduğu Linux makine için, programınız, sunucunun cgi dosyalarını çalıştıracakını belirttiği klasörünün içerisinde olmalıdır(cgi-bin). Ayrıca, bu dosyanın çalıştırılabilir olması için, izinleri değiştirilmelidir, (herkes için okunabilir – çalıştırılabilir, sadece sahibi için tüm haklar).

Bundan başka, Windows işletim sisteminde çalışan apache2triad gibi bir program kullanarak yazılan cgi programları çalıştırılabilir. Bu programı internette indirebilir (<http://sourceforge.net/projects/apache2triad/>) ve cgi programlarını deneyebilirsiniz. Bu program kurulduktan sonra dosyaların belirli bir klasörün içerisinde yer alması gereklidir. Örneğin, "C:\apache2triad\htdocs\apache2triadcp" gibi. Bu klasör programın sürümüne göre değişebilir.

2.1.1. Linux İçin İse Aşağıdaki Aşamaların Yerine Getirilmesi Gereklidir

Web sunucuda CGI fonksiyonunu yapmak için, web sunucudaki ayar dosyalarına bazı düzenlemeler yapılır. CGI programının kurulum yeri CGI programını icra etmek ve uzantısını belirtmek için kullanılır. Bunun için /etc/httpd/conf/httpd.conf dosyası düzenlenmelidir (aşağıdaki satırların dosyada olup olmadığını kontrol ediniz.)

- Aşağıdaki satırda görülen dizin CGI programlarının depolanması için belirtilir.

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```

/cgi-bin/ Bu dizin takma isimleri belirtir (rastgele isimler verilebilir).
"/var/www/cgi-bin/" Bu dizinde CGI programlarının bulunduğu yer belirtilir.

- CGI program için dosya tipi kayıtlıdır.

Web server, HTML ve images gibi çeşitli dosyaları işleyebilir. web server dosya uzantısından dosyanın tipine karar verir. CGI program için dosya tipi yeniden kayıt edilir.

```
AddHandler cgi-script .cgi
```

- (/var/www/cgi-bin/) nesne dizinidir ve CGI bu adreste icra edilebilir.

```
<Directory "/var/www/cgi-bin/">
```

```
Options ExecCGI  
AddHandler cgi-script .cgi
```

```
</Directory>
```

- httpd servisini yeniden başlatınız (restart).

2.2. CGI Programının Temelleri

CGI programı basitçe, web sunucu tarafından çağrılan ve çalıştırılan, web sayfası ziyaretçisine yanıt olarak belirli değerler gönderen bir programdır. Sayfa sayacı, form bilgisi tutucuları gibi uygulamalarda kullanılır. Eğer bir HTML sayfası üretecek bir CGI programı yazıyorsak, programınızda herhangi bir şey yazmadan önce,

print "content-type:text/html\n\n";

ifadesinin olması gereklidir. Bu ifade web tarayıcıya ne tür bir veri aldığını söylemek için kullanılmaktadır. Tarayıcı gönderilen dataların arasında http başlığını boş satırlarda yorumlayacaktır. Bu yüzden "\n\n" satırları gereklidir. İçerik çeşitleri aşağıda gösterilmiştir.

Content-type	{	text/html	HTML format
		text/plain	Text format
		image/gif	Picture file in GIF format

CGI programı yazarken C dili kullanılabilir. Aşağıdaki C dili ile yazılan bir cgi programı görülmektedir.

which perl

exit komutu ile çalışan program sona erdirilir.

- Yukarıdaki program vi editörü ile aşağıdaki gibi oluşturulur.

```
vi /var/www/cgi-bin/hello1.cgi
```

- "hello1.cgi" programı yukarıda gösterildiği gibi çalıştırma yetkisi olmadan yapılmıştır. Bu yüzden, öncelikle aşağıda gösterildiği gibi programa çalıştırma yetkisi verilir.

```
chmod 755 hello1.cgi
```

- Programı çalıştırma

Öncelikle, hello1.cgi dosyası (/var/www/cgi-bin/) dizinine taşınır ve aşağıda gösterildiği şekilde çalıştırılır.

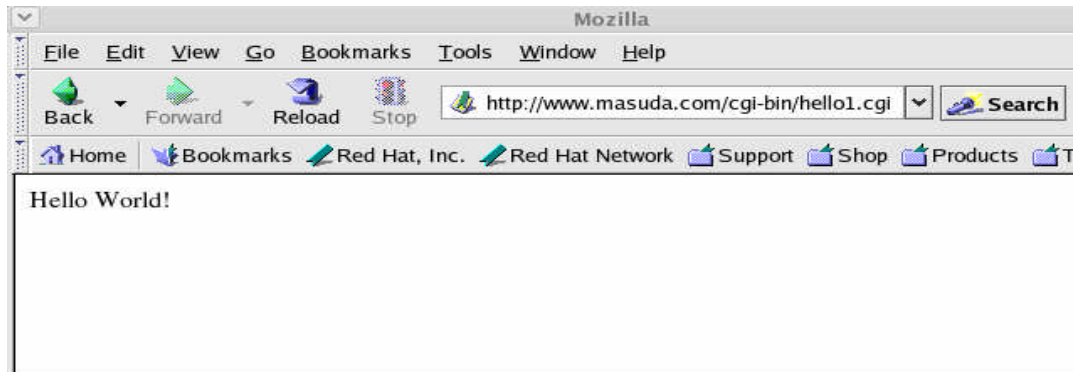
```
./hello1.cgi
```

Programın sonucu aşağıda gösterilmiştir.

```
Content-type: text/html  
Hello World!
```

- Programı tarayıcı üzerinde çalıştırmak için

Eğer sisteminizde bir DNS sunucu varsa, programa domain name yazarak, tarayıcı üzerinden erişilebilir(<http://www.yourname.com/cgi-bin/hello1.cgi>). Eğer DNS sunucu kurulu değilse, <http://www.localhost/cgi-bin/hello1.cgi> yazarak erişilebilir. Burada localhost 127.0.0.1 adresine sahip web sunucuyu ifade eder. Programın sonucu aşağıda gösterilmiştir.



Şekil 2.3: Perl dili kullanımı

2.4.1. Perl’de Değişkenlerin Kullanımı

Örnek 2.3:

```
vi /var/www/cgi-bin/islem.cgi (Dosyanın ismi ve yolu)
```

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
$a=5;
$b=$a+18;      # $b nin değeri 23
$c=9*5;      # $c nin değeri 45.
$a=$a-4;      # $a nın değeri 1.
print $a;
print $b;
print "$c<br/>";
```

Eğer değişkenler sırayla yazdırılacak olursa 12345 değeri görüntülenir.

2.4.2. HTML Etiketlerinin CGI Programında Kullanımı

Örnek 2.3:

Print fonksiyonu kullanarak “ ” karakterleri arasında html etiketlerini kullanabiliriz.

```
vi /var/www/cgi-bin/selam.cgi
```

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "<html><head><title>Selam...</title></head>\n";
print "<body>\n";
print "<h2>Selamlar</h2>\n";
print "</body></html>\n";
```

Tarayıcıda başlık çubuğunda Selam... Sayfanın içerisinde ise “selamlar” ifadeleri gösterilecektir.

2.4.3. Özel Karakterlerin Yazdırılması

Örnek 2.4:

```
vi /var/www/cgi-bin /ozel.cgi

#!/usr/bin/perl
print "Content-type: text/html\n\n";
#Özel karakterlerin yazdırılması
$string = "Ahmet yeni gömleği için Mehmet'e 15 YTL ödedi.";
$email = "adresim@adres.com";
#string ifadelerin yazdırılması
print "$string<br />";
print "$email<br />";
print '$string ve $email<br/>';
```

Ahmet yeni gömleği için Mehmet'e 15 YTL ödedi.
adresim@adres.com
\$string ve \$email

2.4.2. For Döngüsünün Kullanımı

Örnek 2.5

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
for($i=1; $i<=10; $i++){
    print "$i";
    print "<br/>";
}
```

Bu program alt alta 1'den 10'a kadar olan sayıları yazdırır.

2.4.3. Dizi Yapısının Oluşturulması

Örnek 2.6

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
@isim=("Melih", "Kerem", "Ahmet", "Banu", "Yunus");
print " \@isim dizisinin elemanları @isim<br/>";
print "İlk eleman $isim[0] <br/>";
print "Üçüncü eleman $isim[2] <br/>";
print 'Dizde ', scalar(@isim), " kişi bulunmaktadır.<br/>";
```

@isim dizisinin elemanları Melih Kerem Ahmet Banu Yunus
İlk eleman Melih
Üçüncü eleman Ahmet
Dizide 5 kişi bulunmaktadır.

2.4.4. Foreach Döngüsünün Kullanımı

Örnek 2.7

```
vi /var/www/cgi-bin /dongu.cgi

#!/usr/bin/perl
print "Content-type: text/html\n\n";
@aylar=("Ocak", "Şubat", "Mart", "Nisan", "Mayıs", "Haziran", "Temmuz", "Ağustos");

foreach $i (@aylar) {
    print "$i<br/>";
}
```

Ocak
Şubat
Mart
Nisan
Mayıs
Haziran
Temmuz
Ağustos

2.4.5. Form Etiketinin Kullanımı

Örnek 2.8

CGI programının URL'si HTML'in FORM etiketinde tanımlandı. CGI programı, sayfada yapılan butona basıldığında çalışır.

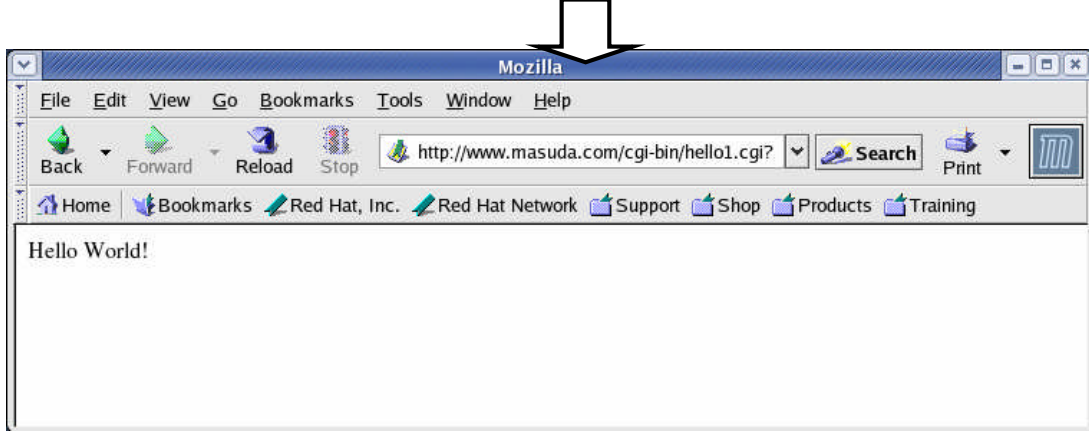
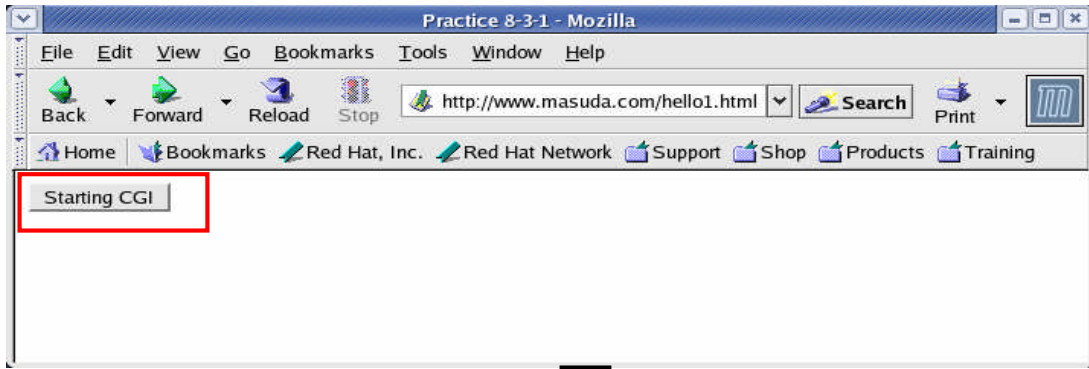
vi programı kullanarak aşağıda gösterilen dizinde bir web sayfası oluşturunuz.

```
vi /var/www/html/hello1.html
```

```
<html>
<head>
    <title>Ornek 2.3 </title>
```

```
</head>
<body>
  <form action="http://www.isminiz.com/cgi-bin/hello1.cgi">
    <input type="submit" value="Starting CGI">
  </form>
</body>
</html>
```

➤ Programın çalıştırılması



Şekil 2.4: Form üzerinden CGI programının çağırılması

Bu örnekte, “Starting CGI” butonu tıklandığında, hello1.cgi programı çalışacak ve tarayıcıda Hello World ifadesi gösterilecektir. Burada formdan submit edilen (gönderilen) değer, form etiketinin, action parametresi ile değerlendirilecek ve ilgili url’de yer alan program çalıştırılacaktır.

2.4.6. HTML’deki A Etiketinin kullanımı

Örnek 2.9

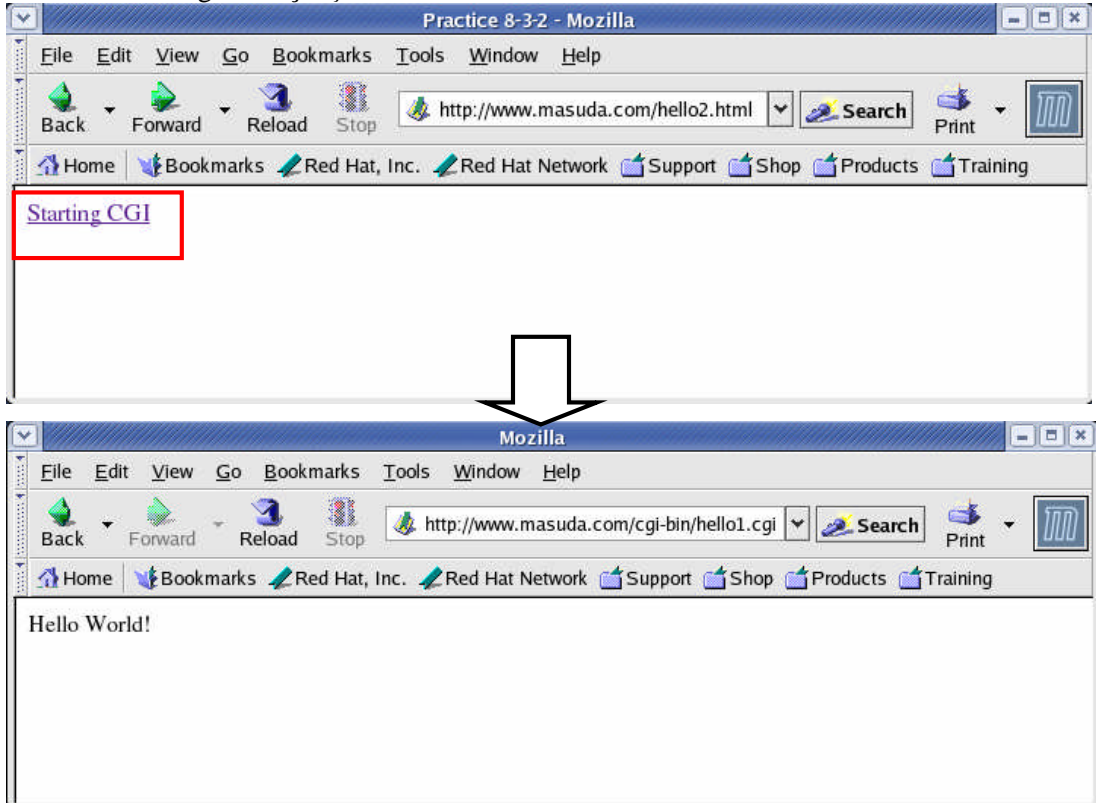
CGI programının URL’si HTML’in A etiketinde tanımlanır. CGI programı, sayfadaki linke tıklandığı zaman çalışır.

➤ **Programın yapılması**

```
vi /var/www/html/hello2.html
```

```
<html>
<head>   <title>Ornek 2.5</title>
</head>
<body>
  <A HREF="http://www.isminiz.com/cgi-bin/hello1.cgi">Starting CGI</A>
</body>
</html>
```

➤ **Programın çalıştırılması**



Şekil 2.5: Link üzerinden CGI programının çağırılması

2.4.7. HTML'deki FRAME Etiketinin Kullanımı

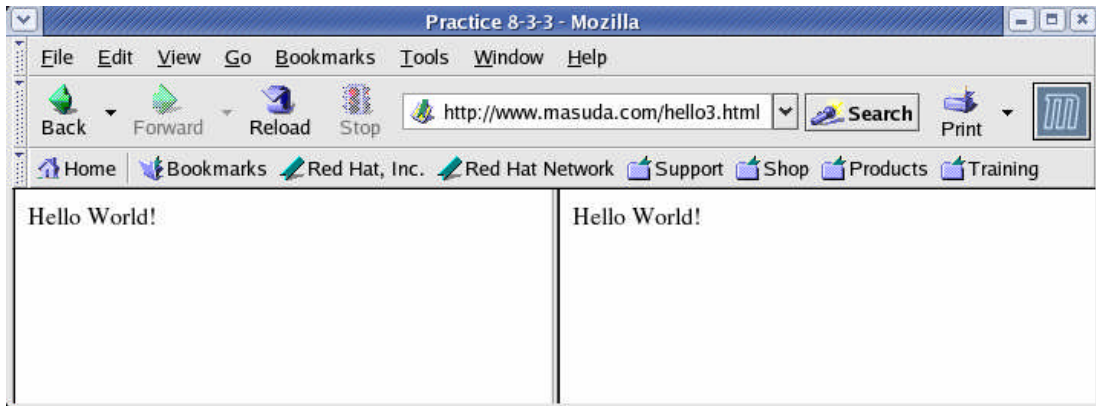
Örnek 2.10

CGI programının URL'si HTML'nin FRAME etiketinde tanımlanır. CGI programı, HTML dosyası görüntülendiğinde cgi programı çalışacaktır.

➤ Programın yapılması

```
vi /var/www/html/hello3.html
```

```
<html>
<head><title>ornek 2.10 </title></head>
<frameset cols="50%,50%">
    <frame src="http://www.isminiz.com/cgi-bin/hello1.cgi">
    <frame src="http://www.isminiz.com/cgi-bin/hello1.cgi">
</frameset>
</html>
```



Şekil 2.6: Çerçeve ile CGI programının kullanılması

Web sayfasının her bir frame'i kaynak olarak hello1.cgi programını kullanmaktadır. Bunun için gerekli yol tanımı src parametresinde gerilmiştir.

2.4.8.Sprintf Fonksiyonu

Örnek 2.11:

```
vi /var/www/cgi-bin /hello.cgi
```

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
$n=12;
$s="Hello";
print sprintf("%d%07d%s",$n,$n,$s);
print "\n";
exit;
```

120000012Hello

Perl'de değişkenler \$ karakteri ile tanımlanırlar. Print komutu ile kullanılan sprintf fonksiyonu formatlı yazım için kullanılır.

- %d : Desimal sayıların gösteriminde kullanılır.
- %04d : 0 sayısal değerin başına eklenir ve desimal sayı 4 dijite olarak gösterilir.
- %s : string karakter gösterilir.

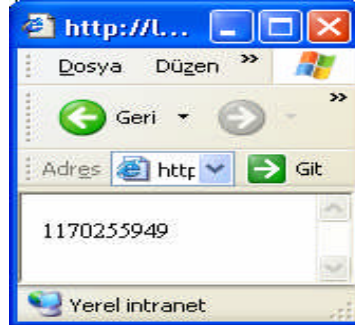
2.4.9. Time Fonksiyonu

1 Ocak 1900'dan itibaren geçen zamanı sayısal olarak ifade eder.

Örnek 2.12:

```
vi /var/www/cgi-bin /time1.cgi

#!/usr/bin/perl
print "Content-type: text/html\n\n";
print time;
print "\n";
exit;
```



Şekil 2.7: Time fonksiyonu

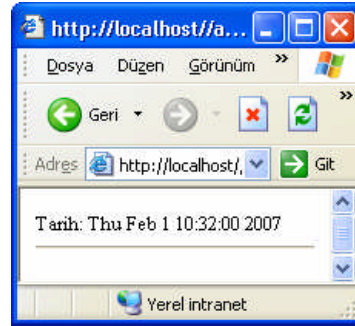
2.4.10. Localtime Fonksiyonu

Time fonksiyonu ile elde edilen zamanı değerini şimdiki zamana çevirir.

Örnek 2.13:

```
vi /var/www/cgi-bin /time2.cgi

#!/usr/bin/perl
print "Content-type: text/html\n\n";
$tarih = localtime;
print "Tarih: $tarih<hr><center>";
exit;
```

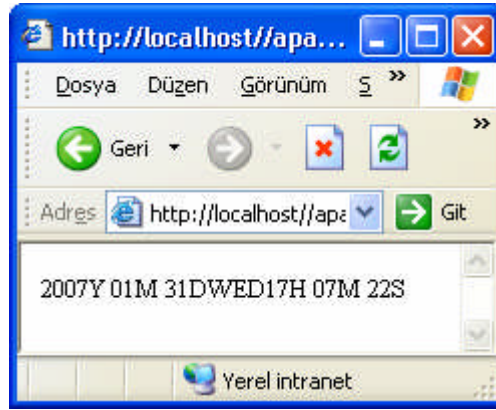


Şekil 2.8: Localtime fonksiyonu

Örnek 2.14:

```
vi /var/www/cgi-bin /time3.cgi
```

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
($sec,$min,$hour,$day,$mon,$year,$wday)=localtime(time);
$yyyymmdd=sprintf("%04dY %02dM %02dD",$year+1900,$mon+1,$day);
$we=sprintf("%s",("SUN","MON","TUE","WED","THU","FRI","SAT")[$wday]);
$hhmmss=sprintf("%02dH %02dM %02dS",$hour,$min,$sec);
print $yyyymmdd.$we.$hhmmss;
print "\n";
exit;
```



Şekil 2.9: Localtime ve time fonksiyonu

Örnek 2.15:

```
vi /var/www/cgi-bin /time4.cgi
#!/usr/bin/perl
print "Content-type: text/html\n\n";
($sec,$min,$hour,$day,$mon,$year,$wday)=localtime(time);
print "sec   :". $sec. "\n";
print "min   :". $min. "\n";
print "hour  :". $hour. "\n";
print "day   :". $day. "\n";
print "mon   :". $mon. "\n";
print "year  :". $year. "\n";
print "wday  :". $wday. "\n";
exit;
```

saniye :53 dakika :0 saat :11 gün :18 ay :11 yıl :106 hgün :1

2.4.11. Ana Programdan Alt program Çağırma

Örnek 2.12:

➤ Ana programın Yapılması

```
vi /var/www/cgi-bin/main.cgi
#!/usr/bin/perl
require "./sub_gettime.cgi";
print &gettime();
print "\n";
exit;
```

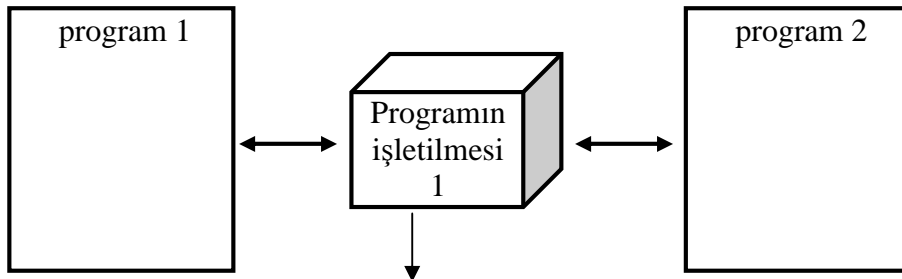
➤ Alt programın yapılması

```
vi /var/www/cgi-bin/sub_gettime.cgi
```

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
sub gettime{
my ($yyymmdd,$we,$hmmss);
($sec,$min,$hour,$day,$mon,$year,$wday)=localtime(time);
$yyymmdd=sprintf(" Y %02d M %02d D ",$year+1900,$mon+1,$day);
$we=sprintf("%s",("SUN","MON","TUE","WED","THU","FRI","SAT")[$wday]);
$hmmss=sprintf(" %02d H %02d M %02d S",$hour,$min,$sec);
return $yyymmdd,$we,$hmmss;
}
1
```

Y 2006 M 12 D MON 11 H 04 M 09 S

Alt program birden fazla program tarafından kullanılabilmesi için çalışmalarımızda kolaylık sağlar.



Bu alt programdır. Başka programlar tarafından çağırılması mümkündür.

Şekil 2.10: Alt programın çağırılması

```
my($yyymmdd,$we,$hhmmss);
```

“my” ifadesiyle bildirilen deęişken sadece alt programda etkilidir. return \$yyymmdd,\$we,\$hhmmss; ifadesiyle deęer ana fonksiyona geri döndürülür.

1 deęeri ile alt programın sonu gösterilir.

```
require "./sub_gettime.cgi";
```

Alt programın çalıştırılacağı belirtilir.

```
print &gettime();
```

gettime alt programı çağırılır. Deęer alt programdan çıkış olarak döner.

2.2.12. Dosyalarla Çalışma

Örnek 2.16:

Bu bölümde bir metin dosyasından deęer okuma ve yazmanın metodu anlatılmıştır. Okunan ve yazılan dosya birçok CGI program tarafından kullanılır.

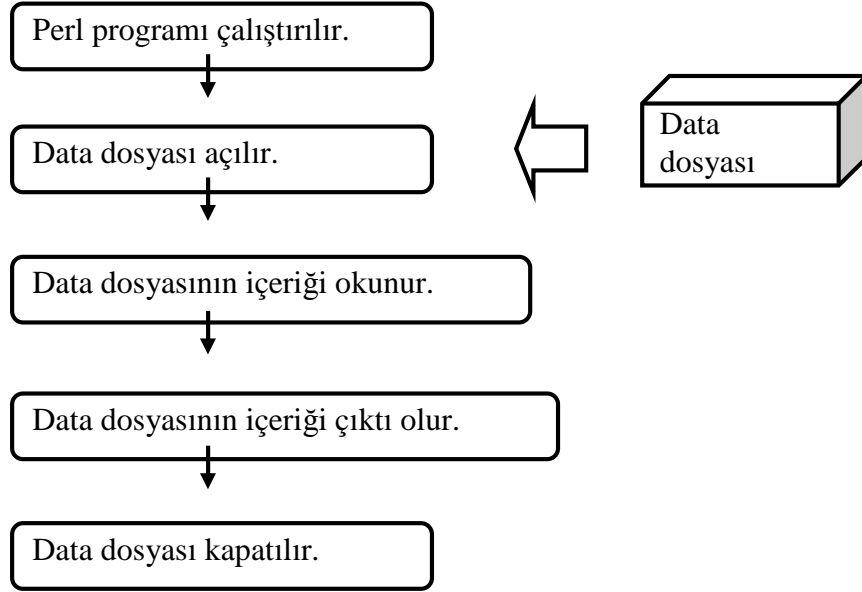
➤ Data dosyası oluşturma

```
vi /var/www/cgi-bin/sample_data.txt
```

Yukarıda gösterilen yerde bir dosya açarak aşağıdaki satırları giriniz.

```
1111  
2222  
3333
```

- Programın yapılması
Program aşağıda gösterildiği gibi işlenir.



Program dosyada bulunan ilk satırı okur.
vi /var/www/cgi-bin/read1.cgi

```
#!/usr/bin/perl
$in_file="./sample_data.txt";
open(IN_FILE,"$in_file");
$record=<IN_FILE>;
print $record;
close(IN_FILE);
exit;
```

İlk satırda veri dosyası \$in_file değişkeni ile ilişkilendirilir. Daha sonra open fonksiyonu kullanılarak ilişkilendirilen bu dosya işlemek için açılır.

Dosyanın Açılması:

```
open(file handle,file name);
```

```
open(IN_FILE,"$in_file");
```

Değişkene aktarım:

```
Variable identifier=<file handle>
```

```
$record=<IN_FILE>;
```

Dosyadan okunan ilk satırdaki değer değişkene aktarılır. Burada, "1111" ifadesi \$record değişkenine depolanır.

```
close(IN_FILE);
```

Data dosyası kapatılır.

- Programın çalıştırılması
chmod 755 sample_data.txt
chmod 755 read1.cgi
./read1.cgi

Open fonksiyonu çalıştırıldığı zaman, aşağıdaki değerler geri döndürülür.

```
{  
  Dosya açıldığı zaman :True  
  Dosya açılmadığı zaman :False
```

Böylece, program eklenen open fonksiyonundan değer döndüğüne karar verir.

2.4.13. Dosya Açılmadığı Zaman Hata Mesajının Görüntülenmesi

- Programın yapılması

```
vi /var/www/cgi-bin/read2.cgi
```

```
#!/usr/bin/perl  
$in_file="./sample_data.txt";  
if(!open(IN_FILE,$in_file)){  
  print "To fail for open file\n";  
  exit;  
}  
$record=<IN_FILE>;  
print $record;  
close(IN_FILE);  
exit;
```

Program, bitirilen deneme dosyasını okur. Eğer dosya açılmazsa aşağıdaki işlem basamakları eklenir. Dosya açılmadığı zaman "To fail for open file" mesajı görüntülenir ve program bitirilir.

```
if(!open(IN_FILE,"$in_file")){
    print "To fail for open file\n";
    exit;
}
```

"0" değeri dosya açılmadığı zaman geri döndürülür. Çünkü, sonucun olumlu olması "!" ifadesiyle değer tersine işlenerek bulunur. Bu yüzden "if" ile başlayan cümle icra edilir.

➤ Programın çalıştırılması
chmod 755 read2.cgi
./read2.cgi

Sonra, "To fail for open file" mesajı görüntülenir. Dosya ismini aşağıdaki gibi değiştiriniz. Programı çalıştırınız.

```
$in_file="./sample_data.tx";
```

2.4.14. Dosyada Yer Alan Tüm Kayıtların Görüntülenmesi

➤ Programın yapılması

```
vi /var/www/cgi-bin/read3.cgi
```

```
#!/usr/bin/perl
$in_file="./sample_data.txt";
if(!open(IN_FILE,"$in_file")){
    print "To fail for open file\n";
    exit;
}
@record=<IN_FILE>;
print @record;
close(IN_FILE);
exit;
```

```
@record=<IN_FILE>;  
print @record;
```

"@ variable identifier" ifadesi belirtildiği zaman, tanımlanan değişkenler düzenlenir. Bu yüzden, dosyadaki tüm kayıtları sırayla okuyabilmek için "@record" ifadesi kullanılır.

- Programın çalıştırılması

```
chmod 755 read3.cgi  
./read3.cgi
```

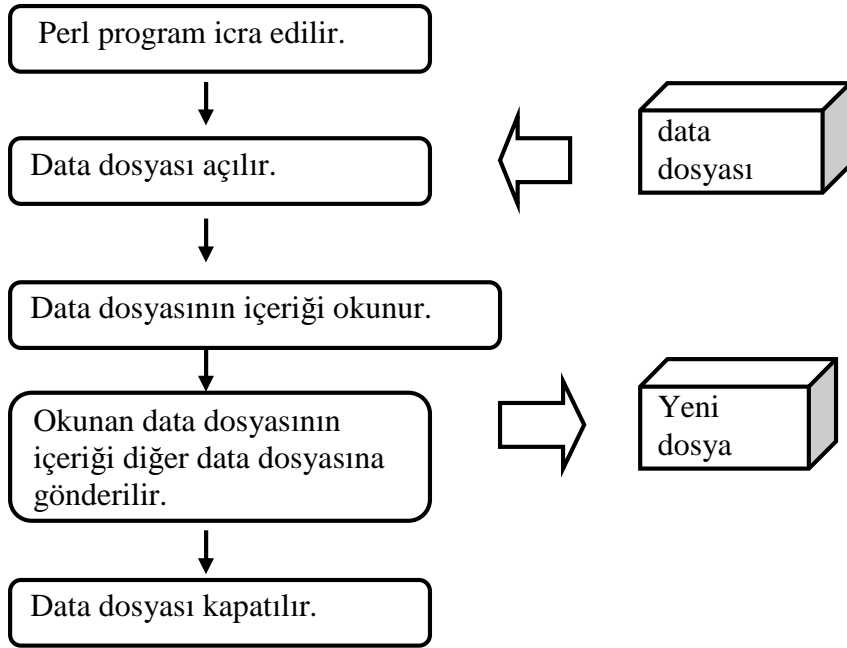


Şekil 2.11: Dosyadan Kayıtların Okunması

2.4.15. Okuma Dosyası İçin Program ve Kayıt

Bu uygulamada, program dosyayı okur ve yapılan diğer bir data dosyasına yazar.

- Programın yapılması
Program aşağıdaki işlem basamaklarından oluşur.



Örnek 2.17

Hazırlık

Dosya hedefe kopyalanarak hazırlanır. Bununla birlikte, aşağıda görüldüğü gibi boş bir dosya daha oluşturulur.

```
vi /var/www/cgi-bin/out_data.txt  
chmod 755 out_data.txt
```

- Programın yapılması:
vi /var/www/cgi-bin/read4.cgi

```
#!/usr/bin/perl  
$in_file="./sample_data.txt";  
$out_file="./out_data.txt";  
if(!open(IN_FILE,$in_file)){
```



```

print "To fail for open IN file\n";
    exit;
}
if(!open(OUT_FILE,">$out_file")){
    print "To fail for open OUT file\n";
    exit;
}
@record=<IN_FILE>;
print OUT_FILE @record;
close(IN_FILE);
close(OUT_FILE);
exit;

```

```

if(!open(OUT_FILE,">$out_file")){
    print "To fail for open OUT file\n";
    exit;
}

```

Burada, dosya hedefe kopyalandığında açılır. ">" ifadesi çağrılan bir işlem modudur. Aşağıda bir çok işlem modu gösterilmiştir. Açılan dosya sadece çıktı içindir. ">" ifadesi işlem modu olarak kullanılır.

Dosyanın kullanımı	İşlem mode	Örnek kullanımlar
Sadece giriş için	Boşluk belirtilir.	open (file handle,"\$filename")
Sadece çıkış için	<	open (file handle,"<\$filename")
Giriş ve çıkış için	+<	open (file handle,"+<\$filename")
Sadece çıkış için (replace mode)	>	open (file handle,">\$filename")
Giriş ve çıkış için (replace mode)	+>	open (file handle,"+>\$filename")
Sadece çıkış için (addition mode)	>>	open (file handle,">>\$filename")

Şekil 2.12: Dosya açılış modları

```

print File handle Output character string;

```

```

print OUT_FILE @record;

```

Tüm data sırayla çıkış olarak hedefe kopyalanan dosyada set edilir.

- Programın çalıştırılması
chmod 755 read4.cgi
./read4.cgi
- Dosyanın içeriğini kontrol ediniz.
cat out_data.txt
- İşlem modunu değiştiriniz
İşlem modunu “+>” ve “>>” gibi değiştiriniz ve sonuçları kontrol ediniz.

```

if(!open(OUT_FILE,">$out_file")){
    print "To fail for open OUT file\n";
    exit;
}

```

2.4.16. Erişim Sayıcı Program Yapma

Örnek 2.18

- Programın yapılması
CGI program için hazırlık: Bir sayıcı dosyası yapınız. Bu dosya sadece bir datadır.
vi /var/www/cgi-bin/count.log
Dosyanın içine aşağıdaki değer girilir.

0

Dosyanın izinlerini aşağıdaki gibi değiştirilir.
chmod 666 count.log çalıştırma yetkisi gerekli değildir.
vi /var/www/cgi-bin/count01.cgi

```
#!/usr/bin/perl
$filename="/count.log";
open(FILE,"+<$filename");
flock(FILE,2);
chop($count=<FILE>);
$count++;
seek(FILE,0,0);
print FILE "$count\n";
close(FILE);
flock(FILE,8);
print "Content-type:text/html\n\n";
print "<HTML>";
printf("%04d",$count);
print "</HTML>";
exit;
```

```
open(FILE,"+<$filename");
```

Okunabilen ve yazılabilen modda açar. (+<)

Flock fonksiyonu:

{	flock(**,1)	Genel kilitleme (Sadece okuma kabul edilir.)
	flock(**,2)	Özel kilitleme (Okuma ve yazma mümkün değildir.)
	flock(**,8)	Açma

```
flock(FILE,2);
```

Özel kilitleme

```
chop($count=<FILE>);
```

“count.log” dosyasındaki data okunur ve “count” değişkenine depolanır.

Sadece datanın bir satırı okunur çünkü \$ işaretiyle belirtilmiştir. @ ifadesi belirtildiği zaman okunan datanın hepsi sırayla depolanır.

Okunan datanın sonunda bir line-feed karakteri vardır. Bu yüzden, line-feed karakteri, kullanılan ayırıcı(chop) ile silinir.

```
$count++;
```

Okunan dataya bir eklenir. Kısaca, sayıcının değeri bir artırılır.

```
seek(FILE,0,0);
```

seek (file handle,offset,source)

Source: 0 : Dosyanın başı
1: Dosya işaretçisinin şimdiki pozisyonu
2: Dosyanın sonu

Offset : Kaynaktan byte numarası

Dosya İşaretçisi dosyanın başına taşınmıştır.

```
print FILE "$count\n"
```

Sayıcının değerleri “count.log” dosyasına yazılır.

```
close(FILE);
```

Dosya kapatılır.

```
flock(FILE,8);
```

Dosyanın kilitlendiği yayımlanır.

```
print "Content-type:text/html\n\n";
```

Data, html data da tanımlandığı gibi tarayıcıya geri döndürülür.

```
print "<HTML>";  
printf("%04d", $count);  
print "</HTML>";
```

Tarayıcı üzerinde gösterilen sayıcıya ait değerdir. “printf” ifadesinin formatı C dilinde olduğu gibi aynıdır.

➤ Programın çalıştırılması.

```
chmod 755 count01.cgi  
./count01.cgi
```

Çıktının aşağıda gösterildiği gibi olduğu kontrol edilir.

```
0001
```

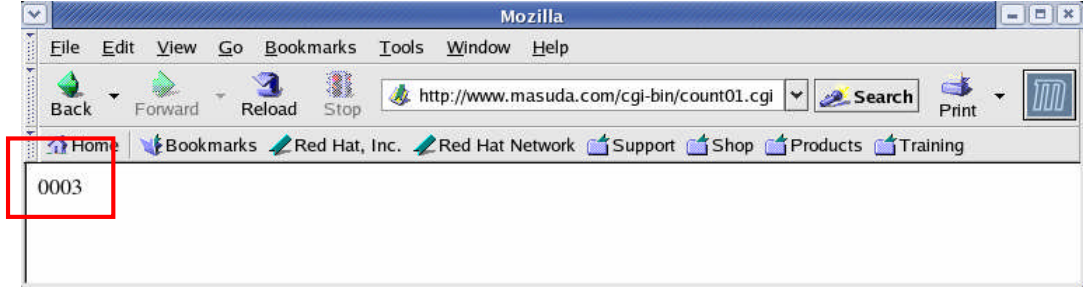
Program tekrar çalıştırılarak sayıcı değerinin aşağıda gösterildiği gibi artıp artmadığı kontrol edilir.

```
0002
```

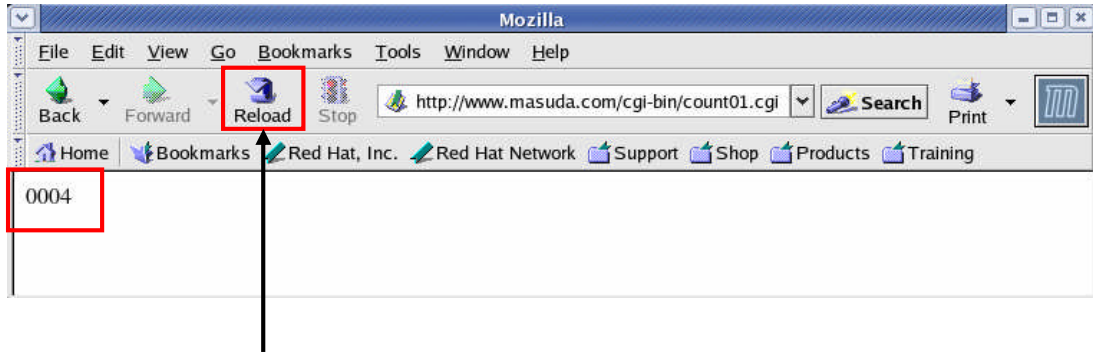
Counter dosyasının içeriği kontrol edilir.
cat count.log

Tarayıcı üzerinde icra sonuçları kontrol edilir.

<http://www.isminiz.com/cgi-bin/count01.cgi>



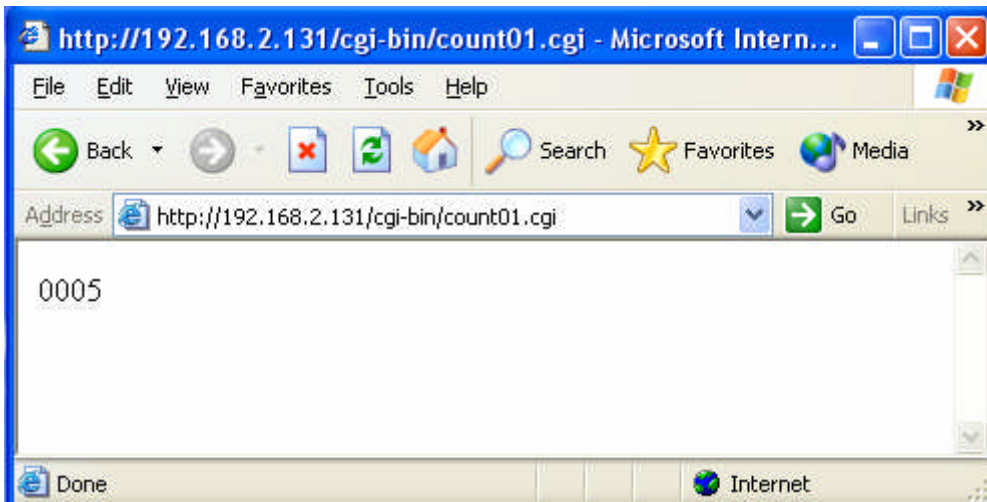
Şekil 2.13: Sayfa sayacı



Şekil 2.14: Sayfa sayacının artırılması

Reload butonuna basınız ve sayıcı değerinin artıp artmadığını kontrol ediniz

Diğer bilgisayarlardan sayıcı değerinin artıp artmadığını kontrol ediniz.



Şekil 2.15: Sayfa sayacına başka bilgisayardan erişme

UYGULAMA FAALİYETİ

Aşağıdaki sorulara ilişkin uygulama faaliyetini yapınız.

- 1'den 100'e kadar olan sayıların toplamını görüntüleyen cgi programını C dili komutlarını kullanarak yazınız.
- HTML etiketlerini kullanarak, adınızı ve soyadınızı; tarayıcıda arka plan rengi mavi, karakterlerin rengi kırmızı olacak şekilde görüntülenmesini sağlayacak cgi programını yazınız.
- Perl dili komutlarını kullanarak, daha önce oluşturmuş olduğunuz bir metin dosyasının içeriğini okuyarak, tarayıcınızda görüntüleyiniz.

İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ Yazmış olduğunuz C programının kaynak kodunu, uzantısı cgi olacak şekilde derleyiniz. Ve cgi dosyasını ilgili klasörün içine kopyalayınız.➤ Htm etiketlerini, print komutuyla birlikte "" işaretleri arasında kullanarak cgi programını oluşturunuz.➤ Perl dili ile C dili temel olarak birbirlerine benzemektedir. C dili komut yapısını gözden geçiriniz.➤ Tarayıcı kullanarak oluşturduğunuz sayfaları görüntüleyiniz.	<ul style="list-style-type: none">➤ Yazmış olduğunuz HTML etiketlerini kullanırken kullanacağınız "" işareti bazı metin editörlerinde farklı yorumlanmaktadır. Bu yüzden, kodları bir yerden başka bir yere kopyalıyorsanız, kodunuza doğru karakterlerin yerleştirildiğinden emin olun.➤ İç içe geçen etiketler kullanırken, etiketlerin uygun sırada yerleştirildiğinden emin olun.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları cevaplayarak bu faaliyette kazandığınız bilgileri ölçünüz.

OBJEKTİF TEST (ÖLÇME SORULARI)

1. Perl ve CGI birer programlama dilidir. (D / Y)
2. Perl derleyicisi Linux işletim sisteminde yer almaktadır. (D / Y)
3. CGI programları, tarayıcı tabanlı çalışır. (D / Y)
4. Perl dilinde kullanılan chop fonksiyonun görevi nedir?
A) Bir stringin son karakterini siler.
B) Bir string ifadedeki karakter sayısını bulur.
C) String bir ifadeyi sayısal hale dönüştürür.
D) İki string ifadeyi birleştirir.
5. Perl dilinde komut satırları birbirinden hangi karakterle ayrılır?
A) : B) , C) ; D) /
6. Aşağıdakilerden hangisi perl dilinde geçerli bir değişken ismidir?
A) \$\$a B) a C) \$a D) A
7. Ana programdan, bir alt program çağırarak istendiğinde kullanılan komut aşağıdakilerden hangisidir?
A) load B) include C) require D) get
8. Aşağıdakilerden hangisi doğru bir dizi tanımlamasıdır?
A) #dizi B) \$dizi C) %dizi D) @dizi
9. Aşağıdakilerden hangisi perl dilinde formatlı yazım için kullanılır?
A) print B) sprintf C) printf D) sprint
10. Perl dilinde özel karakterlerin görüntülenmesi için kullanılan karakter hangisidir?
A) ' B) # C) | D) \

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrar inceleyiniz.

MODÜL DEĞERLENDİRME

UYGULAMALI TEST (YETERLİK ÖLÇME)

Modülde yaptığınız uygulamaları tekrar yapınız. Yaptığınız bu uygulamaları aşağıdaki tabloya göre değerlendiriniz.

AÇIKLAMA: Aşağıda listelenen ölçütleri uyguladıysanız Evet sütununa, uygulamadıysanız Hayır sütununa X işareti yazınız.		
DEĞERLENDİRME ÖLÇÜTLERİ	Evet	Hayır
Linuxta vi editörü ya da gedit programını kullanarak html dosyası oluşturabildiniz mi?		
Html komutlarını kullanarak, satırlarda farklı sayıda hücrelerin bulunduğu tabloyu oluşturabildiniz mi?		
Web sayfanıza resim ekleyebildiniz mi?		
Eklediğiniz resme link verip bir üst klasörde bulunan başka bir sayfanın açılmasını sağlayabildiniz mi?		
Genişlik ve yükseklik değerlerinin girildiği bir tablonun istediğiniz hücrelerine resim ekleyebildiniz mi?		
Sunucu tabanlı bir program yazmak için C dilini kullanabildiniz mi?		
CGI programı yazarken HTML etiketlerini kullanabildiniz mi?		
Yazdığımız CGI programını web sunucunuzda deneyebildiniz mi?		
Perl dili kullanarak sabit diskinizde var olan bir metin dosyasını okuyabildiniz mi?		

DEĞERLENDİRME

Hayır cevaplarınız var ise ilgili uygulama faaliyetini tekrar ediniz. Cevaplarınızın tümü evet ise bir sonraki modüle geçebilirsiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1 CEVAP ANAHTARI

1	B
2	D
3	C
4	B
5	A
6	C
7	D
8	C
9	D
10	B

ÖĞRENME FAALİYETİ-2 CEVAP ANAHTARI

1	YANLIŞ
2	DOĞRU
3	YANLIŞ
4	A
5	C
6	C
7	C
8	D
9	B
10	D

KAYNAKÇA

- MASUDA Yoichi, İbrahim APA, **Bilgisayar Ağları**, M.E.B – JICA, Ağustos, 2005.
- www.w3schools.com