

- Adres Satırından Bilgi Almak
- Formdan Bilgi Almak

Web programcılığında asıl olan konu form işlemleridir. Formlar, ziyaretçiden bizim istediğimiz bilgileri ve ziyaretçinin sunacağı bilgileri işleyip gerekli yerlere ulaştırmak için hazırlanmış HTML kalıplarıdır.

İnternet ilk çıktığında tamamen web sitesinden ziyaretçilere bilgi sunmak için geliştirilmişti. Ama şimdi bir web sayfasından sadece bunu bekleyemeyiz. Çünkü artık bir web sayfasında ziyaretçiden bilgi alma mecburiyeti-mizde doğmuştur. Bu bir anket, bir forum veya sanal bir alışveriş sitesi de olabilir.

Formlar ziyaretçiden bilgi toplamak için oluşturulmuştur. Bu formlardan gelen bilgiler, istenilen dosyalara GET ve POST metotları kullanılarak gönderilir. Bir PHP dosyasına dışarıdan üç farklı yöntemle bilgi gönderilebilir: Birincisi adres satırından, ikincisi formlardan, üçüncüsü de çerezlerden. Şimdi burada ilk iki yöntemi anlatacağız. Çerezlereyse ileriki konularda değineceğiz.

Adres Satırından Bilgi Almak

Adres satırından bilgi edinmek çok kullanılan ve basit olduğu için tercih edilen bir yöntemdir. İnternet'te gezinirken `deneme.php?go=123&say-ga=05` şeklinde, karışık gibi görünen, ama görüldüğü kadar karışık olmayan bir adres satırından bilgi alma işlemini basitçe anlatalım:

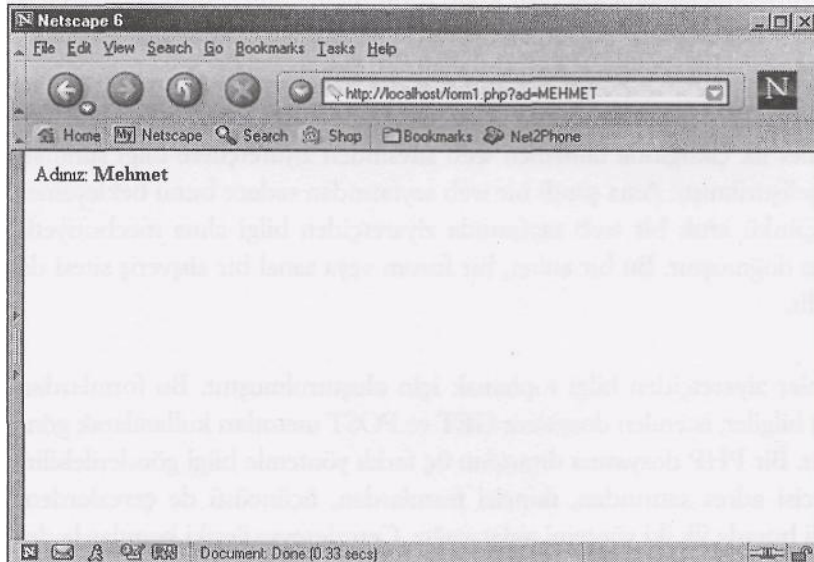
`http://www.aspet.net/git.php?go=03`

Yukarıdaki adrese tıkladığımızda tarayıcımız ilk önce internet adresine bağlanır. Sonra **git.php** dosyasının olup olmadığını sorgular. Daha sonra bu sayfaya girer ve `$go="03"`; gibi bir değişken atar. Bu dosyada, bu değişkeni kullanır. Somut bir örnek vererek işimizi kolaylaştıralım:

```
<?
```

```
print "Adınız: <b>$ad</b>"; ?>
```

Bu komut satırını `form1.php` olarak ana dizininize kaydedin. Sonra tarayıcınızı açarak adres satırına **`http://localhost/form1.php?ad=MEHMET`** yazın. Bakın karşınıza ne gelecek!



Görüldüğü gibi burada \$ad adında bir değişken oluşturduk ve adres hanesine dosyanın adını yazdıktan sonra **?ad=MEHMET** yazdığımızda programa \$ad değişkenini tanımlayıp bu değişkenin değerini sunuyoruz. Burada bir değişken sunduk. Peki birden çok değişkeni, adres satırından programa nasıl aktarırız?

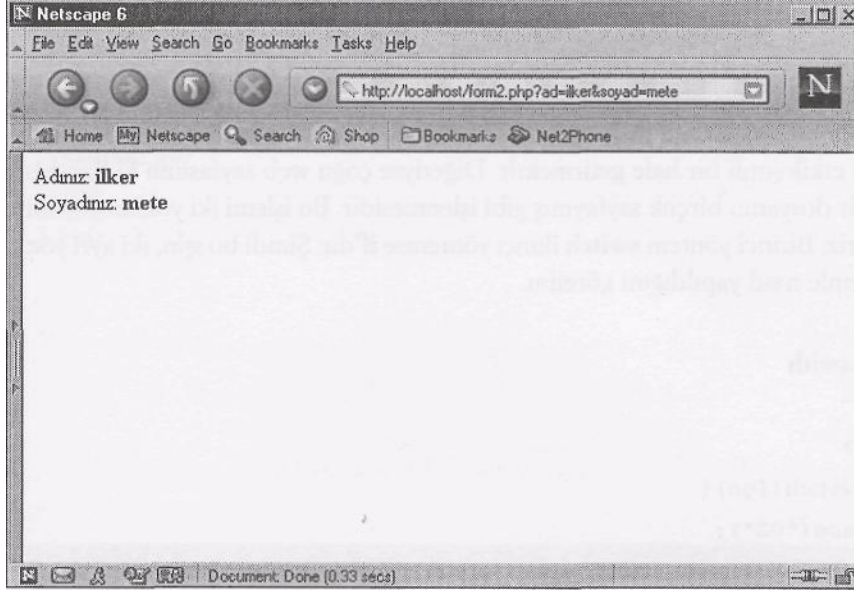
<?

```
print "Adınız: <b>$ad </b><br> Soyadınız: <b>$soyad</b>";
```

?>

Şimdi bu kodu form2.php olarak kaydedin. Sonra tarayıcınızı açarak adres hanesine şu adresi yazın.

<http://localhost/form2.php?ad=ilker&soyad=mete>



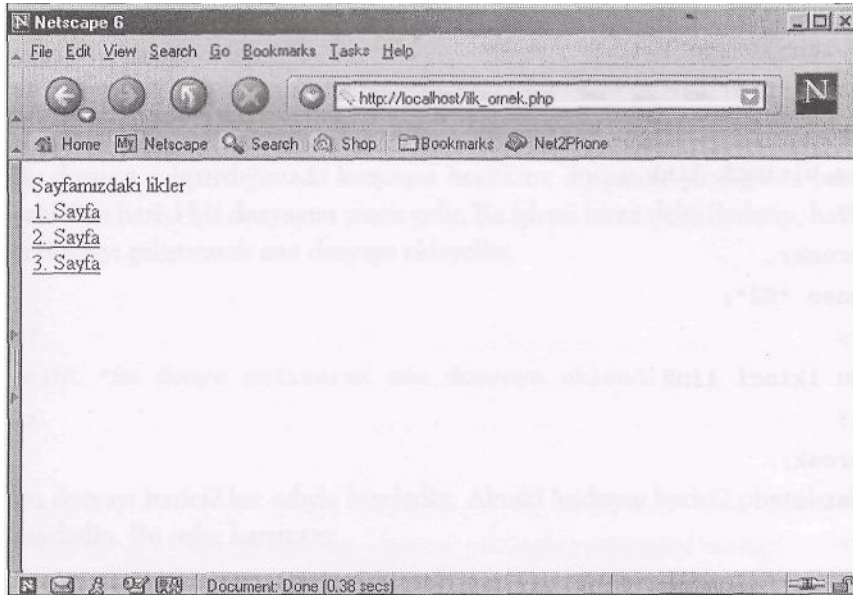
Gördüğümüz gibi **form2.php** dosyasında iki değişken oluşturduk ve bu değişkenlere adres hanesinden veri aktardık. Adres hanesindeki ilk değişkeni yazmadan önce bir soru işareti koyuyoruz. Sonra değişkeni yazıyoruz ve değişkenin değerini eşittir işareti ile yüklüyoruz. ikinci değişken oluştururken & işaretinden yararlanıyoruz. Sonra bir önceki yöntemdeki adımları aynen uyguluyoruz. İki değişken veya 50 değişken hiç fark etmez, ilk değişkeni yaparken soru işareti kullandıktan sonra, diğer değişkenlerde & işareti kullanarak değişkenleri birbirinden ayırıyoruz.

<http://www.aspet.net/index.php?sayfa=01&deneme=05&go=anasayfa>

Görüldüğü gibi burada index.php dosyasına üç değişken gönderiyoruz. Bu adres şöyle de olabilir:

<http://www.aspet.net/?sayfa=01&denem=05&go=anasayfa>

Yukarıdaki kodu **ilk_ornek.php** olarak kaydedin. Bu örnek kodu çalıştırdığınızda karşınıza aşağıdaki gibi bir görüntü gelecektir.



Bu sayfadaki linkleri tıkladığınızda karşınıza sırasıyla alttaki linkler çıkacaktır.

Bu sayfanın linki ilk_ornek.php?go=02
 Bu sayfanın linki ilk_ornek.php?go=03
 Bu sayfanın linki ilk_ornek.php?go=04

Bu yöntemi kullanarak, bir dosyada birçok sayfayı barındırabilirsiniz. Ama peşinen şu hususu da belirtmek gerekir ki, bu tip sayfalar yaparken tutup da sayfanın tümünü PHP'ye yazdırmayın. Sayfaları bu şekilde yaparsanız PHP ya hata verir ya da çok yavaş çalışır. Böyle bir sayfa yapmak için PHP kodlarını, HTML kodlarının içerisine gömmek gerekir.

```
<html>
  <head>
    <title>Bu bir denemedir</title>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9"?
  </head> <body> <?
switch ( $go ) { case "01"; ?>
Bu birinci link <?
break; case "02"; ?>
Bu ikinci link <?
break; default: ?>
Bu sayfa da iki link var.Bunlar:<br>
<a href="<? echo $PHP_SELF; ?>?go=01">Birinci link</a> <a href="<? echo $PHP_SELF;
?>?go=02">ikinci link</a> <? }
?>
</body> </html>
```

Burada görüldüğü gibi PHP kodlarını HTML kodunun içerisine gömdük. Bundan sonrada, oluşturacağımız PHP kodlarını HTML kodlarının içerisine gömeceğiz.

Buraya kadar verdiğimiz bilgilerle şifreli bir link yapamayız. Mantık olarak yaparız ama pratikte zorlanırız. Böyle bir link'i yapabilmek için harici dosyalardan yararlanmalıyız. Harici dosya, herhangi bir dosyanın başka bir

dosyaya bağlanmasıdır. Bu işlemler için **include** komutu kullanılır. Şimdi alttaki kodu **harici.inc** adıyla kaydedin.

Ben harici bir dosyayım

Şimdi bu dosyayı çalıştıracak harici.php adında bir dosya oluşturalım:

```
<?
```

```
include "harici.inc"; ?>
```

Bu dosyayı çalıştırdığınızda karşınıza harici.inc dosyasına yazdığımız yazı, yani **Ben harici bir dosyayım** yazısı gelir. Bu işlemi biraz daha ilerletip, harici dosyayı çalıştırarak ana dosyaya ekleyelim.

```
<?
```

```
print "Bu dosya çalışarak ana dosyaya eklendi"; ?>
```

Bu dosyayı harici2.inc adıyla kaydedin. Alttaki koduysa harici2.php olarak kaydedin. Bu sefer karşınıza;

Bu dosya çalışarak ana dosyaya eklendi

yazısı gelecektir. Bu yöntemi her zaman her yerde kullanacağız. Bu komut bize çok kolaylık sağlar. Şimdi yukarıda verdiğimiz örneği **include** komutunu kullanarak yeniden yazalım, ilk önce harici dosyaları oluşturalım:

1.inc

Bu birinci link

2.inc

Bu ikinci link

ana.inc

Bu sayfa da iki link var.Bunlar:


```
<a href="<? echo $PHP_SELF; ?>?go=01">Birinci link</a>
```

```
<a href="<? echo $PHP_SELF; ?>?go=02">İkinci link</a>
```

harici_dosya.php

```
<?
```

```
switch ( $go ) { case "01";
    include "1.inc"; break;
    case "02";
    include "2.inc"; break; default:
    include "ana.inc"; } ?>
```

Görüldüğü gibi adres şifreleme işlemlerini yaparken **include** komutunu kullanmak işimizi kolaylaştırmıştır.

Lif

if deyimini kullanarak daha uzun link adresleri yapmamız mümkündür. Burada bir hususu belirtmekte yarar var: **switch** deyimini ile yaptığımız tüm şifreli link'leri if deyiminiyle de yapabiliriz. Ama if deyimini ile yaptığımız tüm şifreli linkleri switch deyimini ile yapamayız. İç içe girmiş bir switch deyiminiyle, if deyimini ile yaptığımız işlemleri yapabiliriz. Ama uzun uzadıya işlemlerden kaçınarak uzun linkleri yapmaya başlayalım. Şimdi alttaki kodu uzunlink.php olarak kaydedin.

```
<?
if ( $sayfa == "anasayfa" && $go == "01" ) {
    include "anasayfa01.ine" ; } elseif ( $sayfa == "anasayfa" && $go == "02"
) {
    include "anasayfa02.ine" ; )
elseif ( $sayfa == "anasayfa" && $go == "03" && $git == "ek-sayfa" ) {
include "anasayfaOS.ine" ; } else {
    include "anasayfa.inc" ; } ?>
```

Burada, verdiğimiz örnekteki link'leri sıralamamız gerekirse:

```
http://localhost/uzunlink.php
http://localhost/uzunlink.php?sayfa=anasayfa&go=01
http://localhost/uzunlink.php?sayfa=anasayfa&go=02
http://localhost/uzunlink.php?sayfa=anasayfa&go=03&git=eksayfa
```

Buradaki linklere tıkladığınızda PHP, kodda verdiğimiz harici dosyaları arayacak ve bulamayacağı için hata verecektir. Önemli olan, bir link'e tıkladığınızda bu linke ait harici dosyanın aranabilmesidir. Bu link adreslerinde herhangi bir yanlışlık yapıldığında otomatikman **if** deyiminde bulunan **else** bölümüne kayıtlı harici dosya çalışacaktır.

Şifrelenmiş link'li web adresleri yaparken dikkat etmemiz gereken en önemli husus; oluşturacağımız linklerde Türkçe karakter 'ç, Ç, ğ, G, ı, İ, ö, Ö, ş, Ş, ü, Ü' ve üst karakterler kullanmamaktır. Link şifrelemede tamamen İngilizce karakter ve rakamları kullanmamız gerekmektedir. Bu tip karakterleri çalıştırdığımızda browser otomatikman şifreleyecektir. Altta belli başlı karakterlerin şifrelenmiş hallerini bulunmaktadır.

ı = %FD	İ = %DD	O = %DC
ç = %E7	ö = %F6	% = %25
Ç = %C7	Ö = %D6	& = %26
9 = %FO	ü = %FC	_ = %BD
Ğ = %DO		

Formdan Bilgi Almak

HTML formlarından bilgi almak çok basittir. Herhangi bir form nesnesinden, (text, password, radio, textarea, checkbox, hidden, vb.) bilgi almak için bu form alanındaki nesnelere bir ad veririz. HTML'de bunu **name** komutu yapar. Bu komut bir değişken adı tanımlar. Gerekli yerlerde, özellikle **hidden** görünmez nesnelerde kullanacağımız bir diğer komut da **value** komutudur. Bu komut oluşturduğumuz değişkene değer atar. Eğer formda değer atamazsak **value** bölümünde yer alan bilgi değişkene atanır. Eğer **value'ye** değer atadıysak ve de form alanına bilgi girdiysek form alanına girilen bilgiye itibar edilir. Şimdi somut birer örnek verelim:

```
<input type="text" name="degisken">
```

Burada değişken adında bir değişken oluşturmuş oluyoruz. Bu değişkene değer atamak için formda çıkan alana bilgi giriyoruz.

Özellikle tek seçenek işaretlenmesini istediğimiz formlarda (genelde anketlerde) kullanılan **radio** form alanlarını da aynı şekilde hazırlıyoruz. Fakat burada **value** değer atama komutunu da birlikte kullanıyoruz.

```
<input type="radio" name="değişken" value="01"> •<input type="radio"
name="degisken" value="02"> <input type="radio" name="değişken" value="03">
```

Burada görüldüğü gibi **radio** düğmelerinde seçenek yapabilmemiz için değişkeni tanımladığımız **name** komutuna aynı adı vermeliyiz. Burada üç seçenekli bir **radio** düğmesi yaptık. Birinci seçilirse, gerekli programa **\$degis-ken="01"** şeklinde bir değişken ve değeri atanır. Diğer düğmele seçilirse, gerekli programa **\$degisken="02"** ve **\$degisken="03"** değişkenleri ve değerleri atanır.

Checkbox düğmeleri, birden çok seçmeli düğmeler yapmamız gerektiğinde kullanılabilen form alanlarıdır. Bu düğmenin **radio** düğmesinden farkı şudur: **radio** düğmelerinde seçmeme şansımız yoktur. Mutlaka bu düğmelerden birini seçmemiz gerekir. Ama **checkbox** düğmelerinde seçip seçmeme şansımız vardır. Bir veya birden çok düğmeyi seçme şansımız olduğunda düğmelerin adını farklı veriyoruz.

```
<input type="checkbox" name="girdil" value="01"> <input type="checkbox"
name="girdi2" value="02">
```

Formlarda çokça kullandığımız select form alanlarında da bir alan seçebilme olanağımız olduğundan **radio** düğmelerine benzemektedir. Bu düğmeleri kullanma şekilde aşağıdaki gibidir:

```
<select name="deneme"> <option value="100">al</option> <option
value="200">a2</option> </select>
```

Burada, çoklu seçenekten birini seçebilmemiz için değişkene değişik değerler atıyoruz, al seçeneğini seçili duruma getirdiğimizde **deneme** değişken! nin değeri 100 olur. Diğer seçeneği seçtiğimizde de değişkenin değeri 20i? olur.

Buraya kadar HTML'le form oluşturmayı anlattık. Şimdi de bu form bilgilerini gerekli PHP dosyasına göndereceğimiz metotları görelim. HTML'de form bilgileri bir dosyaya iki yöntemle gönderilir. Bunlar GET ve POST me-todlarıdır. Bu metotlardan istediğimizi kullanabiliriz. Ama POST metodu daha güvenilirdir. Çünkü bu yöntemde formdan alınan bilgiler tamamen şifrelenerek istenilen dosyaya gönderilir. GET metotundaysa link şifreleme de olduğu gibi form bilgilerini adres satırına yansıtarak gerekli dosyaya iletir. Form gönderme işlemi kısaca şöyledir:

```
<form action="gönderilecekdosya" method="GET/POST">
```

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

Şimdi buraya kadar öğrendiğimiz soyut bilgileri somut bir örneğe yansıtalım. Alttaki HTML kodunu **form1.htm** olarak kaydedin.

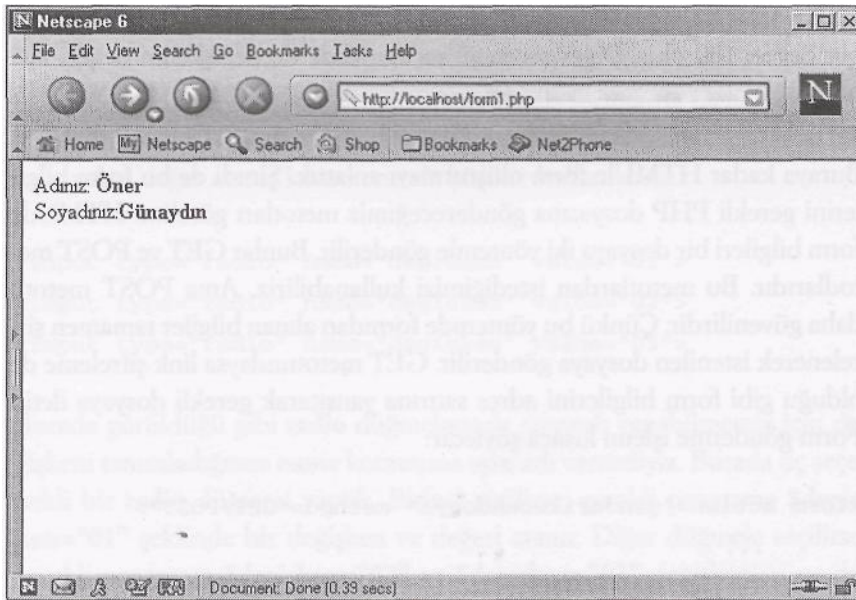
```
<form action="form1.php" method="post"> Adınız: <input type="text"
name="adi"xbr> Soyadınız: <input type="text" name="soyadi"><br>
<input type="submit" value="GÖNDER"> </form>
```

Bu form alanından alınan bilgileri göndereceğimiz **form1.php** dosyasını hazırlayalım:

```
<?
echo "Adınız: <b>$adi </b>xbr>\n"; echo "Soyadınız:<b> Ssoyadi
</b>\n"; ?>
```

Bu form dosyasını çalıştırıp form alanını doldurduktan sonra gönder düğmesine tıkladığınızda aşağıdaki

gibi bir sayfayla karşılaşacaksınız.



Görüldüğü gibi bu formdaki bilgiler **POST** metoduyla ile form1.php dosyasına gönderilmiştir. Şimdi **GET** ve **POST** metodunun farkına varabilmek için form1.htm dosyasını açın ve form etiketinde bulunan **POST** yerine **GET** yazarak gönder butonuna basın. Sonuç yine aynı olacaktır. Farklı olan tek şey adres satırının değişik olmasıdır.

`http://localhost/form1.php?adi=%D6ner&soyadi=G%FCnayd%FDn`

İki yöntem arasında bir fark yoktur. Ama İnternet'te kullanacağımız dosyaların güvenli olmasını açısından; örneğin üyelik sisteminin bulunduğu bir sitede login olmak için girilen bir form alanına tutup da **GET** metoduyla yaparsak kötü niyetli kişiler bu linke tıklayarak siteye giriş yapabilir. Bu gibi durumlarda **POST** metodunu kullanmalıyız. Bazı durumlarda da **GET** metodunu kullanmamız gerekebiliyor. Ama genelde **POST** metodu kullanılır. Şimdi daha kapsamlı bir örnek verelim: Alttaki örneği **form2.htm** olarak kaydedin.

```
<html>
<head>
<title>Bu bir denemedir</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-9"> </head> <body>
<form action="form2.php" method="get"> Ad ve soyadınız: <input type="text"
name="ad"> E-Mail : <input type="text" name="mail"> En sevdiğiniz meyve
nedir? <select name="meyve"> <option value="muz">MUZ <option value="nar">NAR • <option
value="çilek">Cilek </select> Cinsiyetiniz?
Erkek : <input type="radio" name="cinsiyet" value="erkek"> Kadın : <input type="radio"
name="cinsiyet" value="kadın"> Hiç kivi tadına baktınız mı? <input type="checkbox"
name="kivi" value="kivi">
<br>
<input type="submit" value="GÖNDER">
</form>
</body>
</html>
```

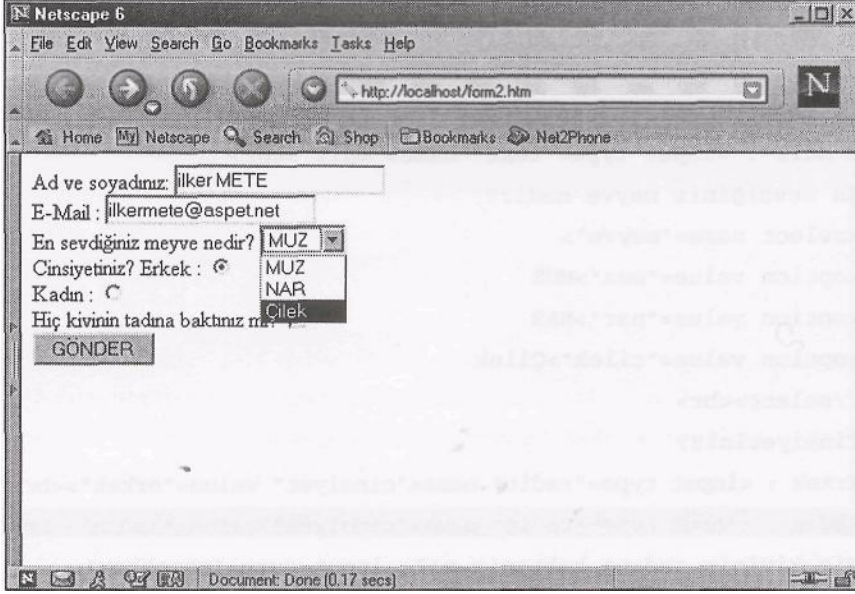
Bu HTML kodunu çalıştırıp gönder butonuna tıkladığınızda bilgilerin derlenip gönderileceği dosya olan **form2.php** dosyasının kodları altta verilmiştir.

<?

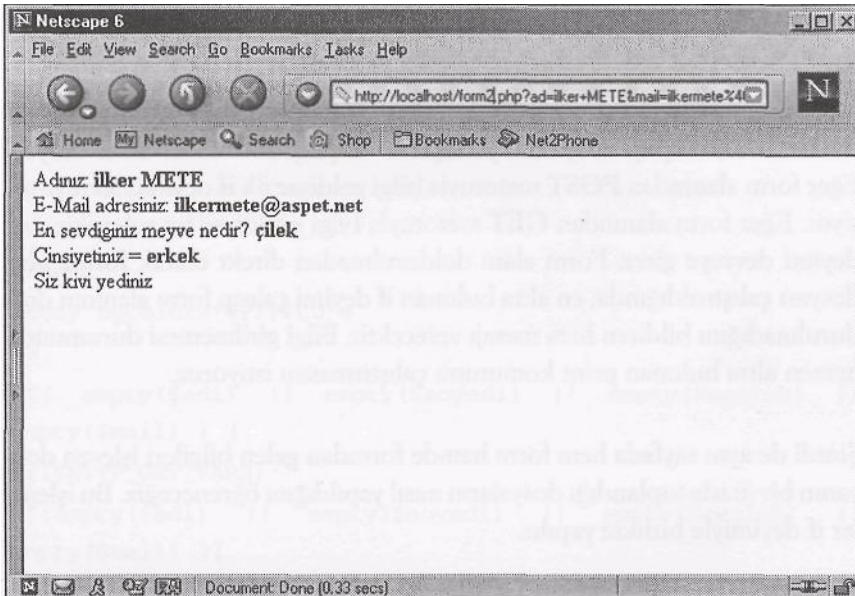
```

echo "Adınız: <b>$ad</b>";
echo "E-Mail adresiniz: <b>$mail</b>";
echo "En sevdiğiniz meyve nedir? <b>$meyve</b>";
echo "Cinsiyetiniz = <b>$cinsiyet</b>";
if (isset ($kivi) ) {
    print "Siz kivi yediniz"; } ?>

```



HTML kodumuz yukarıdaki resimde görüldüğü gibi bir sonuç çıkartmakta. Bu form alanını doldurup gönder butonuna tıkladığımızda bilgiler GET metoduyla toplanıp form2.php dosyasına gönderilir. Form2.php dosyasında, gelen bu bilgileri yakalamak için değişken isimlerini yazdırıyoruz. HTML kodunda **checkbox** kullandığımız için ziyaretçi bu bölümü doldur -mayabilir. Bu yüzden bu form alanının dolu olup olmadığını sınamak için bir if deyimi kullandık. Bu if deyiminde, farklı olarak **isset** komutunu kullandık. Bu komut bir değişkenin veri içerip içermediğini sınar. Eğer doğruysa, yani form alanı doldurulmuş ise dönen değer **TRUE**, form alanı dol-durulmamışsa **FALSE** sonucu gelecektir. Ayrıca, formdan gelen bilgiler GET metoduyla geldiği için adres satırı şifrelenmiştir.



Öyle form alanları yaparız ki bu form alanında tanımlanan değişkenleri satır satır yazmak bazen sıkıntılara yol açar. Bizi bu dertten, PHP'nin **\$HTTP_GET_VARS** ve **\$HTTP_POST_VARS** dizi değişkenleri kurtarır. Bu dizi değişkenleri, formadan gelen bilgileri derleyip toparlayarak, bir döngü yardımıyla tutulan değişken ve değerlerini ekrana basar. Şimdi yukarıda oluşturduğumuz formdaki bilgileri ekrana basalım.

```
<?
if (isset ( $HTTP_POST_VARS ) ) {
    foreach ( $HTTP_POST_VARS as $anahtar => $degeri ) { print "$anahtar = $degeri
        <br>\n";
    } } if ( isset($HTTP_GET_VARS) ) {
    foreach ( $HTTP_GET_VARS as $anahtar => $degeri ) { print "$anahtar = $degeri
        <br>\n";
    } }
if ( empty ( $HTTP_GET_VARS ) || empty ( $HTTP_POST_VARS ) ){ print "form alanından GET veya
POST metoduyla bilgi gönderilmedi"; } ?>
```

Bu kodu **form2.php** dosyasına yerleştirin ve çalıştırın. Bakın neler oluyor. Eğer form alanından **POST** metoduyla bilgi geldiyse ilk **if** deyimi devreye giriyor. Eğer form alanından **GET** metoduyla bilgi girildiyse bu sefer ikinci **if** deyimi devreye girer. Form alanı doldurulmadan direkt olarak **form2.php** dosyası çalıştırıldığında, en altta bulunan if deyimi çalışıp form alanının dol-durulmadığını bildiren hata mesajı verecektir. Bilgi girilmemesi durumunda hemen altta bulunan print komutunu çalıştırmasını istiyoruz.

Şimdi de aynı sayfada hem form hemde formdan gelen bilgileri işleyen dosyanın bir arada toplandığı dosyaların nasıl yapıldığını öğreneceğiz. Bu işlemler if deyimiyle birlikte yapılır.

Herhangi bir programı yapacaksak önceden bir plan oluşturmamız gerekmektedir. Bunun için öncelikle bu tip bir dosyanın nasıl çalışabileceğini planlayalım:

1. Çalıştıracığımız dosya ilk çalıştığında form alanı çıkacak.
2. Form alanında hiçbir bilgi doldurulmadan gönder butonuna basıldığında hata uyarısı versin.

Form doldurulup gönder butonuna basıldığında yapılacak işlemleri hazırlayalım.

Şimdi bu işlemleri bir if deyimiyle, kapsamlı bir örnekte yapmaya çalışalım:

```
<?
function uyari(){
print '
<script language="JavaScript">
alert("Formu eksik doldurdunuz")
</script>';
}
?>
<html>
<head>
<title>Çok fonksiyonlu bir form</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso=8859-9">
```

```
<meta http-equiv="Content-Type" content="text/html;
charset=iso=windows-1254">
</head>
<body bgcolor="#FFFFCO">
<?
if( empty($adi) II empty($soyadi) II empty($meslek) M
empty($mail) ) {
if($HTTP_GET_VARS):
if(empty($adi) II empty($soyadi) II empty($meslek) ||
empty($mail) ){
uyari();
}
endif ;
?>
<form action="<? echo $PHP_SELF; ?>" method="GET">
Adiniz: <input type="text" name="adi"xbr>
Soyadiniz: -cinput type="text" name="soyadi"><br>
Mesleğiniz: <input type="text" name="meslek"><br>
E-Mail : <input type="text" name="mail"xbr>
<input type="submit" value="Gönder">

</form>
<?
}else{
foreach( $HTTP_GET_VARS as $anahtar=>$deger){
print "$anahtar = <t>$deger</bxbr>\n" ;
}
}
?>
</body>
</html>
```

Burada ilk başta bir fonksiyon tanımlıyoruz ve içine bir JavaScript kodu yerleştiriyoruz. Bu JavaScript kodu formun eksik doldurulması durumunda bizi uyaracaktır. Sayfanın baş kısmını oluşturduktan sonra, bir if deyimiyle altta oluşturduğumuz formdaki değişkenlerden hepsinin olup olmadığını sınıyoruz. Bunun hemen altına bir if deyimini daha oluşturuyoruz ve bu deyimle, **\$HTTP_GET_VARS** değişkeniyle herhangi bir değişken gelip gelmediğini sınıyoruz. Eğer geldiyse yine bir if deyimiyle, bu gelen değişkenlerin formdaki değişkenlerden olup olmadığını sınıyoruz. Bunun sebebi; eğer kullanıcı form alanında herhangi bir alanı doldurmazsa JavaScript koduyla onu uyarmaktır. Eğer bütün form alanı dolduruldu ise ilk if deyiminin else bölümü çalışır ve burada bulunan döngü gelen değişkenleri ekrana listeler.