### WEB TABANLI PROGRAMLAMA

<u>BÖLÜM 12</u> PHP İLE FORM, DOSYA VE XML İŞLEMLERİ HTML FORMLARDAN BİLGİ ALMA, METİN DOSYALARI VE XML BELGE İŞLEME

Prof. Dr. Turgay Tugay Bilgin turgay.bilgin@btu.edu.tr



## GENEL BAKIŞ...

12.1) HTML Formlardan Bilgi Almak

12.2) GET Metodu Kullanan Formlardan Bilgi Almak

12.3) POST Metodu Kullanan Formlardan Bilgi Almak

12.4) Kendi Kendine Bilgi Gönderen Formlar

12.5) extract() Kullanarak Form İşlemleri

12.6) PHP'de Harici Kod Dosyası Kullanımı

12.6.1) include () Kullanımı

12.6.2) require() Kullanımı

12.7) PHP'de Dosyalama İşlemleri

12.7.1) Dosya Açma ve Kapatma

12.7.2) Dosyadan Okuma

12.7.3) Dosyaya Yazma

12.8) PHP ile XML Dosya Kullanımı

12.8.1) PHP ile XML Belge Oluşturma

12.8.2) PHP ile XML Belgeyi Okuma

12.9) Özet

### (12.1) HTML FORMLARDAN BİLGİ ALMAK

- Kullanıcıdan bilgi alıp sunucu üzerinde işlem yapmak için en önemli bileşenlerin HTML formları olduğunu daha önce HTML ünitesinde görmüştük. HTML form bileşenleri ve bunların kullanımı ile ilgili örnekler HTML Formları konusunda verilmişti.
- Bu ünitede, HTML formlardan alınan ve sunucuya ulaşan bilgilerin PHP ile nasıl işleneceği öğretilecektir. Ayrıca metin dosya oluşturma ve metin dosyayı okuma işlemleri de örnekle açıklanacaktır.
- Bölüm 4'de anlatılan "HTML formlar" konusundan hatırlanacağı üzere <form> elemanı kullanıcıdan aldığı bilgileri GET veya POST şeklinde iki farklı metot ile sunucuya postalayabiliyordu. PHP ile bu bilgilerin karşılanması, HTML formun hangi metot ile bilgi gönderdiğine göre değişir. Her iki metot ile gönderilen verilerin PHP ile karşılanması 12.2 ve 12.3 nolu bölümlerde verilmiştir.

### (12.2) GET METODU KULLANAN FORMLARDAN BİLGİ ALMAK

• GET metodu ile postalanan formlardaki her bir form elemanı PHP dilinde

\$\_GET['....']

ifadesi ile alınır. Burada noktalı bölüme değeri alınacak form elemanının adı yazılır. Aşağıdaki örnekte bir HTML formu ve bunun gönderdiği bilgileri ekrana görüntüleyen PHP dosyası verilmiştir.

Kod 12.1 Get metodu için form örneği



• Yukarıdaki kodu "c:\wamp\www" klasörüne **form.html** adıyla kaydediniz. Bu HTML form aracılığıyla "**goster.php**" adlı PHP dosyasına GET metoduyla bilgi gönderiyoruz.

Kod 12.2 PHP ile Get metodu örneği

```
<?php
if( $_GET['adsoyad'] || $_GET['numara'] )
{
    echo "Merhaba, ". $_GET['adsoyad']. "<br />";
    echo "Numaranız ". $_GET['numara']. " olarak girildi.";
}
?>
```

• Yukarıdaki dosyayı da goster.php adıyla kaydediniz. Web tarayıcınızı açıp,

http://localhost/form.html

adresini yazarak çalıştırdığınızda Şekil 12.1'deki çıktı görüntülenecektir. Kod 12.2'de verilen dosya ile öncelikle **adsoyad** ve **numara** isimli form alanlarının boş olup olmadığını bir **if** yapısı ile kontrol ediyoruz. Form alanları boş ise değerleri FALSE olacağından if bloğu içine girilmeyecektir. Böylece kullanıcılar formu doldurmadan formun "Gönder" butonuna basarlar ise değişkenin değeri ile ilgili PHP'nin hata mesajları ile karşılaşmayacaklardır.

Hatırlatma & form.html ve goster.php dosyalarının her ikisini de "c:\wamp\www" klasörü içine yerleştiriniz, aksi halde PHP dosyanız çalışmayacağından çıktı oluşmayacaktır.

6035			
		_	
1			
tp://localhost/gost	er.php?adsoya	d=Ali+KALE&n	umara=12060
	tp://localhost/gost	tp:// <mark>localhost/goster.php</mark> ?adsoya	tp:// <b>localhost</b> /goster.php?adsoyad=Ali+KALE&n

Şekil 12.1 GET metodu ile gönderilen bilgilerin PHP ile karşılanması

- **form.html** dosyası wampserver üzerinde çalıştırıldığında Adı Soyadı bölümüne Ali KALE, numara bölümüne ise 1206035 yazılmıştır. Gönder tuşuna basılınca **goster.php** araclığı ile bu bilgilerin ekranda görüntülendiği Şekil 12.1'de görülmektedir.
- Şekil 12.1'de oklar ile görüntülenen adres çubuğu değerlerine dikkat ediniz. GET metodu ile veriler iletildiğinde forma girilen tüm bilgiler web tarayıcının adres satırında URL'ye eklenerek iletilir. Bu iletim biçiminde, dosyanın adından sonra bir soru işareti (?) ile form elemanları eleman=değer şeklinde yazılırlar, her bir eleman=değer çifti arasında (&) işaret bulunur. Genel biçim aşağıda verilmiştir.

http://siteadi/phpdosya.php?eleman1=deger1&eleman2=deger2 ....

• Bu şekilde gönderilen form bilgilerini adres çubuğundaki ifadeleri okuyan herkes görebilir.

**Dikkat** *G Form işlemlerinde güvenliğe dikkat ediniz.* 

HTML formda input elemanının type niteliğini password olarak belirlemiş olsanız da GET metoduyla gönderiyorsanız şifre ifadesi URL çubuğunda açık olarak görünür. Bilgilendirme & GET metodu ile ne kadar bilgi gönderilebilir?

GET metoduyla değerler URL'ye eklenerek iletildiğinden gönderilebilecek bilgi miktarı en uzun URL karakter sınırı olan 256 karakter ile sınırlıdır. Fakat bazı web sunucu yazılımları, 4000 karaktere kadar bilgi gönderilmesine izin verebilmektedir.

### (12.3) POST METODU KULLANAN FORMLARDAN BİLGİ ALMAK

- GET yöntemi kullanılarak oluşturulan formlar çoğunlukla az sayıda eleman içeren basit formlardır. Karmaşık formlarda çoğunlukla POST metodu tercih edilir. GET metodunun aksine, POST metodunda bilgi taşınırken URL adresinin arkasına eklenmediği için adres çubuğunda görünmez. Bu yönüyle POST metodunun daha güvenilir olduğunu söyleyebiliriz. Fakat POST metoduyla gönderilmiş olsa da gönderilen veri şifrelenmediği için aradaki iletişimi dinleyen üçüncü kişiler bu bilgilere erişebilirler.
- POST metodunda taşınacak bilgi miktarı ile ilgili bir sınırlama yoktur. POST metodunun kullanımı GET ile benzerdir. Oluşturulan formun "method" özelliği POST yapılarak formun POST kullanarak veriyi göndermesi sağlanır. PHP dosyasında ise \$\_GET değişkeni yerine \$\_POST değişkeni kullanılır. Aşağıdaki örnekte bir HTML formu ve bunun gönderdiği bilgileri ekrana görüntüleyen PHP dosyası verilmiştir.

<html></html>
<body></body>
<form action="gosterpost.php" method="POST"></form>
Adı Soyadı: <input name="adsoyad" type="text"/> 
Numarası : <input name="numara" type="text"/> 
<input type="submit" value="Gönder"/>

Kod 12.3 PHP ile Post metodu için form örneği

Yukarıdaki kodu "c:\wamp\www" klasörüne formpost.html adıyla kaydediniz. Bu HTML form aracılığıyla "gosterpost.php" adlı PHP dosyasına POST metoduyla bilgi gönderiyoruz.

php<br if(\$ POST['adsovad']    \$ F	POST['numara'] )
{ {	obi[inamata])
echo "Merhaba, ". \$_POST  echo "Numaranızı ". \$ POS	'adsoyad']. " "; T['numara']. " olarak girdiniz.";
}	
(?>	

#### Kod 12.4 PHP ile Post metodu örneği

Yukarıdaki dosyayı da **gosterpost.php** adıyla kaydediniz. Web tarayıcınızı açıp,

http://localhost/formpost.html

adresini yazarak çalıştırdığınızda Şekil 12.2'deki çıktı görüntülenecektir. **gosterpost.php** dosyası ile öncelikle **adsoyad** ve **numara** isimli form alanlarının boş olup olmadığını bir **if** yapısı ile kontrol ediyoruz. Form alanları boş ise değerleri FALSE olacağından if bloğu içine girilmeyecektir. Böylece kullanıcılar formu doldurmadan formun "Gönder" butonuna basarlar ise değişkenin değeri ile ilgili hata mesajları ile karşılaşmayacaklardır.

← → Ø http://localhost/form.html
Adı Soyadı: Ali KALE Numarası : 1206035 Gönder
Merhaba, Ali KALE Numaranızı 1206035 olarak girdiniz.

- **formpost.html** dosyası wampserver üzerinde çalıştırıldığında *Adı Soyadı* bölümüne Ali KALE, *numara* bölümüne ise 1206035 yazılmıştır. Gönder tuşuna basılınca **gosterpost.php** aracılığı ile bu bilgilerin ekranda görüntülendiği Şekil 12.2'de görülmektedir.
- Şekil 12.2'de adres çubuğu değerlerine dikkat ediniz. POST metodu ile veriler iletildiğinde adres satırında URL'ye eklenen herhangi birşey olmadığı görünmektedir. Bu sebeple gönderilebilecek bilgi miktarı ile ilgili bir sınırlama bulunmamaktadır.

### (12.4) KENDİ KENDİNE BİLGİ GÖNDEREN FORMLAR

 PHP ile form uygulamalarında mutlaka HTML form içeren bir dosya ve bunu işleyen PHP dosya şeklinde iki ayrı dosya bulunmak zorunda değildir. İki ayrı dosya şeklinde bilgi gönderiminin akış diyagramı Şekil 12.3 (a)'da görülmektedir.



(a) HTML dosyadan PHP dosyaya bilgi gönderimi



(b) HTML ve PHP içeren dosyanın kendi kendine bilgi gönderimi Şekil 12.3 Kendi kendine bilgi gönderen formlar Formlar ile bilgi gönderimi sadece 1 adet php dosya ile de gerçeklenebilir. Tek dosya şeklinde bilgi gönderiminin akış diyagramı Şekil 12.3 (b)'de görülmektedir. Bu yapıda PHP dosyası içindeki HTML form, bilgileri kendi kendine postalar. Bu işleme ingilizcede SELF POST adı verilir. SELF POST yapan bir php dosyası oluşturmak için dosya içine hem formdan gelen bilgileri alan php kodu hemde HTML form kodları yazılır. HTML formun action niteliğine <?php \$\_PHP\_SELF ?> ifadesi yazılarak formun kendi kendine bilgi göndermesi sağlanır. POST metodu ile yapılan kendi kendine gönderme işlemine örnek aşağıda verilmiştir.

Bilgilendirme & SELF POST yönteminde nelere dikkat etmeliyim?

SELF POST yöntemine uygun hazırlanan dosyada, öncelikle gelen veriyi karşılayacak PHP kodları, sonrasında bunların bittiği yerden itibaren ise HTML kodları yazılır. Tersi bir yerleşim kullanırsanız form post edildikten sonra ekranda form tekrar görüntülenir.

Kod 12.5 PHP ile SELF POST örneği

```
<?php
  if( isset($ POST['adsoyad']) || isset($ POST['numara']))
     echo "Merhaba, ". $ POST['adsoyad']. "<br />";
     echo "Numaranızı ". $ POST['numara']. " olarak girdiniz.";
     exit();
?>
<html>
<body>
  <form action="<?php $ PHP SELF ?>" method="POST">
 Adı Soyadı: <input type="text" name="adsoyad" /> <br />
 Numarası : <input type="text" name="numara" /> <br />
 <input type="submit" value="Gönder"/>
 </form>
</body>
</html>
```

- Yukarıdaki örnek incelendiğinde isset fonksiyonu kullanıldığı görülmektedir. Bu fonksiyon bir değişkenin tanımlanmış olup olmadığını kontrol etmektedir. Bu örnekte kodlar ilk çalışmaya başladığında henüz HTML form oluşturma satırlarına gelinmediği için adsoyad ve numara değişkenleri tanımlanmamış durumdadır. PHP derleyicisinin bu konumda tanımsız değişkenlerden dolayı hata vermemesi için değişkenler isset fonksiyonu içine alınmıştır. Bu durumda eğer değişkenler tanımlanmamışsa isset fonksiyonu FALSE döndürür dolayısıyla if bloğu çalışmaz, doğrudan HTML form kodları çalışır. Ekranda HTML form görünür. Formu doldurduktan sonra Gönder butonuna basınca <?php \$ PHP SELF ?> ifadesinden dolayı form bilgileri bu dosyanın kendisine gönderilir. Artık **adsoyad** ve **numara** değişkenlerinin değeri var olduğundan isset fonksiyonu TRUE döndürür ve if bloğu içi çalışır. Ekrana formdan alınan bilgiler yazdırılır. If bloğunun son satırındaki exit() ifadesi ile PHP kodunun burada sonlanması sağlanır. Bu sayede ekrana tekrar HTML form görüntülenmesi önlenmiş olur.
- Yukarıdaki örnekte method="POST" yerine method="GET" yazılır ve değişkenler \$\_GET ile alınırsa GET metodu ile kendi kendine gönderme işlemi gerçekleştirilebilir.

### (12.5) EXTRACT () KULLANARAK FORM İŞLEMLERİ

- PHP Form işlemlerinde form ile gönderilen her bir alan için \$\_POST['degisken'] veya \$\_GET['degisken'] şeklinde kullanmanın programcı açısından yazımı zor olması ve kullanıcılar tarafından zor okunabilmesi gibi dezavantajları bulunmaktadır. PHP dilinde extract() metodu kullanarak POST veya GET metodu ile gelen tüm değişkenlere, gönderildiği formdaki name nitelikleri ile erişilmesi sağlanır. Bunu gösteren bir örnek aşağıda verilmiştir.
- Kod 12.6'yı "c:\wamp\www" klasörüne form.html adıyla kaydediniz. Bu HTML form aracılığıyla "goster.php" adlı PHP dosyasına GET metoduyla bilgi gönderiyoruz.

```
<html>
<body>
<form action="goster.php" method="GET">
Adi Soyadi: <input type="text" name="adsoyad" /> <br />
Numarasi : <input type="text" name="numara" /> <br />
<input type="submit" value="Gönder"/>
</form>
</body>
</html>
```

Kod 12.6 PHP ile extract örneği için form oluşturulması

• Kod 12.7'yi de goster.php adıyla kaydediniz. Web tarayıcınızı açıp,

http://localhost/form.html

```
<?php
extract($_GET);

if( $adsoyad || $numara )
{
    echo "Merhaba, ". $adsoyad. "<br />";
    echo "Numaranızı ". $numara. " olarak girildi.";
}
?>
```

#### Kod 12.7 PHP ile extract örneği

adresini yazarak çalıştırdığınızda ekrana gelen formu doldurunca Şekil 12.2'deki çıktının aynısı görüntülenecektir. Kod 12.7'de verilen örnekte öncelikle **adsoyad** ve **numara** isimli form alanlarının Kod 12.6'daki formdaki input elemanlarının name niteliklerine verilen değerler ile aynı adlı değişkenler olduklarına dikkat ediniz. Bu işlem <code>extract(\$\_GET)</code>; satırı sayesinde gerçekleşmiştir. Bu satır, bütün <code>\$\_GET</code> değişkenlerini otomatik olarak PHP değişkeni haline dönüştürmüştür. Böylece kodlar çok daha kısa ve okunaklı hale gelmiştir. Aynı işlemler POST metodu ile bilgi gönderme sırasında da kullanılabilir. Aradaki tek fark, <code>extract()</code> metodunun <code>extract(\$\_POST)</code>; şeklinde kullanılması gerekliliğidir.

### (12.6) PHP'DE HARİCİ KOD DOSYASI KULLANIMI

• PHP dilinde oluşturduğunuz kodları içeren dosyaları başka PHP dosyalarına ekleyerek C/C++ dillerindeki kütüphane kullanımına benzer şekilde kullanabilirsiniz. Böylece kendi kod kütüphanelerinizi oluşturabilirsiniz. Bu iş için PHP dilinde iki farklı fonksiyon mevcuttur:

➤ include()

➤ require()

Bunların her ikisi de aynı işi yapıyor olsa da görevlerinde bazı farklılıklar mevcuttur. Bunların detaylı kullanımları 12.6.1 ve 12.6.2 bölümlerinde verilmiştir.

### (12.6.1) INCLUDE() KULLANIMI

- include() fonksiyonu parametre olarak aldığı bir dosyadaki tüm metni alır ve kendi bulunduğu satıra kopyalar. Parametre olarak aldığı dosya içinde herhangi bir hata varsa include() fonksiyonu bir uyarı oluşturur ancak PHP dosyası çalışmaya devam eder.
- Bu fonksiyon çoğunlukla web sayfalarında üst veya alt menü şeridi gibi tüm sayfalarda sabit olan içerikleri PHP koduna eklemekte kullanılır. Bu sayede sadece bir dosya üzerinde değişiklik yaparak tüm sitenin üst menü veya alt menü şeridi değiştirilebilir. Aşağıdaki örnekte include() fonksiyonu ile sayfaya üst menü eklenmiştir.

#### Kod 12.8 PHP ile include() kullanımı

```
<html>
<body bgcolor="#90c3d4">
<?php include("menu.php"); ?>
Bu bölümde site içeriği yer
alacaktır.
</body>
</html>
```

#### Kod 12.9 PHP ile include() kullanımı için örnek menü

```
height="100">
<h1> Kişisel Web Sayfam </h1>
<a href="hakkimda.html">Hakkımda</a> -
<a href="linkler.html">Linkler</a> -
<a href="linkler.html">Linkler</a> -
<a href="ozgecmis.html">Özgeçmişim</a><br />
```

- Yukarıdaki Kod 12.8 dosya içeriğini "c:\wamp\www" klasörüne anasayfa.php adıyla kaydediniz. Yanındaki dosyayı ise aynı klasöre menu.php adıyla kaydediniz.
- Web tarayıcınızı açıp http://localhost/anasayfa.php şeklinde anasayfa.php dosyasını çalıştırdığınızda Şekil 12.4'deki gibi bir çıktı elde edilir.
- Yukarıdaki örnekte **anasayfa.php** dosyasının içindeki

<?php include("menu.php"); ?>

satırı sayesinde **menu.php** dosyasının içeriği **anasayfa.php** içine kopyalanmış olur. Kolaylıkla anlaşılması açısından menü şeridi mor renk (**c390d4**), ana sayfa arka planı mavi renk (**90c3d4**) olarak renklendirilmiştir. Kitapta bu renk farkları tam olarak görülemeyebilir, örneği kendi bilgisayarınızda çalıştırarak sonucu gözlemleyiniz.



Şekil 12.4. include() fonksiyonu örneği

### (12.6.2) REQUIRE() KULLANIMI

 require() fonksiyonu da parametre olarak aldığı bir dosyadaki tüm metni alır ve kendi bulunduğu satıra kopyalar. Fakat, parametre olarak aldığı dosya içinde herhangi bir hata varsa require() fonksiyonu ölümcül hata (fatal error) mesajı oluşturur ve PHP dosyası çalışması durur.

**Bilgilendirme** *G* include() *ve* require() *arasında ne kadar fark var*?

include() ve require() fonksiyonlarının tek farkı hata oluştuğu sıradaki davranışlarıdır. Eğer harici olarak eklenen dosyadaki hatalarda ana programın çalışması durdursun istiyorsanız require(), çalışmaya devam etsin istiyorsanız include() kullanabilirsiniz. Kod 12.10 PHP ile require() kullanımı

```
<html>
<body>
<?php require("olmayandosya.php"); ?>
Bu satır hiçbir zaman görüntülenemeyecek.
</body>
</html>
```

• Yukarıdaki dosya içeriğini "c:\wamp\www" klasörüne **anasayfa.php** adıyla kaydediniz. Bu örnekte **olmayandosya.php** adlı bir dosya bilgisayarımızda bulunmuyor, fakat biz require() ile bunu eklemeye çalışıyoruz. Ekleme işleminde require() kullandığımız için dosyayı bulamayınca ölümcül hata oluşacak ve bu satırdan sonraki hiçbir satır çalıştırılmayarak ekranda, Şekil 12.5'de görülen hata oluşacaktır.



Şekil 12.5. require() fonksiyonu örneği

• Yukarıdaki örnekte **olmayandosya.php** adlı bir dosya oluşturursanız hiçbir hata oluşmadan kodunuz include() fonksiyonunda anlatıldığı gibi çalışacaktır.

## (12.7) PHP'DE DOSYALAMA İŞLEMLERİ

- PHP dilinde, web sunucunun hard diskinde dosya oluşturabilir, içine bilgi kaydedebilir ve sonra bu bilgileri tekrar okuyarak ekrana yazdırabilirsiniz. Dosyalama işlemleri C dilindeki dosya işlemlerine çok benzerdir. Bu bölümde aşağıdaki dosya işlemleri anlatılacaktır:
  - → Dosya açma ve kapatma,
  - $\rightarrow$  Dosyadan okuma,
  - → Dosyaya yazma
  - → Dosyayı kapatma

• 12.7.1 Dosya Açma ve Kapatma : Dosya işlemleri için öncelikle fopen() fonksiyonu ile dosya açmak gereklidir. fopen() fonksiyonunun kullanım şekli aşağıda verilmiştir. Bu fonksiyon ilk parametre olarak açılacak dosyanın yolu ve adını, ikinci parametre olarak ise dosya açma modu parametresini alır. Açılan dosya bir değişken adına eşleştirilmelidir. Bu değişkene dosya değişkeni denir. Bu satırdan sonraki tüm dosya işlemlerinde bu değişken adı kullanılarak dosyaya erişilecektir. Aşağıdaki örnek kullanımda dosya değişkeni olarak \$file adlı değişken kullanılmıştır.

• Dosya açma modları bir sonraki slayttaki tabloda verilmiştir. İhtiyaca göre bunlardan sadece biri kullanılabilir.

Dosya açma modu	Tanımı
<u>r</u>	Dosyayı sadece okumak için açmayı sağlar. Dosya işaretçisini dosyanın başına konumlandırır.
<u>r</u> +	Dosyayı okumak ve yazmak için açmayı sağlar. Dosya işaretçisini dosyanın başına konumlandırır.
W	Dosyayı sadece yazmak için açmayı sağlar. Dosya işaretçisini dosyanın başına konumlandırır. Eğer dosya mevcut ise içindeki <u>hersey</u> silinir. Dosya yok ise oluşturulur.
<u>w</u> +	Dosyayı okumak ve yazmak için açmayı sağlar. Dosya işaretçisini dosyanın başına konumlandırır. Eğer dosya mevcut ise içindeki <u>herşey</u> silinir. Dosya yok ise oluşturulur.
a	Dosyaya sadece ekleme yapmak için açmayı sağlar. Dosya işaretçisini dosyanın <b>sonuna</b> konumlandırır. Dosya yok ise oluşturulur.
<u>a</u> +	Dosyaya ekleme yapmak ve okumak için açmayı sağlar. Dosya işaretçisini dosyanın <b>sonuna</b> konumlandırır. Dosya yok ise oluşturulur.

Tablo 12.1 Dosya açma modları

- Bir dosya fopen() ile açılmaya çalışıldığında eğer dosya yok ise fopen() fonksiyonu FALSE değeri döndürür, eğer dosya var ise bir dosya işaretçisi (file pointer) döndürür. Bu değer bir değişkene atanarak (yukarıdaki örnekte \$file adlı değişken) daha sonra bu dosya ile ilgili tüm işlemlerde kullanılır.
- Bir dosyayı açıp gerekli okuma veya yazma işlemleri tamamlandıktan sonra dosya mutlaka kapatılmalıdır. Dosya kapatılmadığı durumda başka bir uyguma bu dosyaya erişemeyecektir. Dosya kapatmak için fclose() fonksiyonu kullanılır. fclose() fonksiyonu kapatılacak dosyayı parametre olarak alır. Kullanımı aşağıdaki gibidir.

fclose( \$file );

12.7.2 Dosyadan Okuma : Dosya fopen() ile açıldıktan sonra içindeki veriyi okumak için fread() fonksiyonu kullanılır. Bu fonksiyonun iki parametresi bulunur. Kullanımı aşağıdaki gibidir.

\$okunan\_veri = fread( \$file, \$dosyaboyutu );

fread() fonksiyonunun ilk parametresi dosya işaretçisi, ikinci parametresi ise dosyadan ne kadar bilgi okunacağını ifade eden miktar değişkenidir. Eğer dosyanın tamamını okumak istiyorsanız miktar olarak dosya boyutu verilebilir.

- Dosya boyutunu öğrenmek için PHP'nin filesize() fonksiyonu kullanılır. Bu fonksiyon parametre olarak aldığı dosyanın bayt cinsinden boyutunu döndürür. Dosya okuma işlemi için aşağıdaki adımlar sırasıyla takip edilmelidir:
  - 1) Dosya fopen () fonksiyonu ile açılır,
  - 2) filesize() fonksiyonu ile dosya boyutu öğrenilir,
  - 3) Boyutu öğrenilen dosyadan fread() fonksiyonu ile veri okunur,
  - 4) Dosya fclose() fonksiyonu ile kapatılır.

```
<html>
<body>
<?php
$dosyaadi = "istiklal.txt";
$file = fopen( $dosyaadi, "r" );
if( $file == false )
  echo ( "Hata: Dosya açılamıyor!" );
  exit();
$dosyaboyu = filesize( $dosyaadi );
$dosyametni = fread( $file, $dosyaboyu );
fclose( $file );
echo nl2br($dosyametni);
?>
</body>
</html>
```

Kod 12.11 PHP ile dosyadan okuma işlemi

- Aşağıdaki örneği dosyaoku.php adıyla kaydediniz. Örnekte, içinde
  istiklal marşı yazılı olan istiklal.txt adlı dosya okunuyor ve içeriği
  ekrana yazdırılıyor. Eğer dosya yok ise veya erişilemiyor ise \$file
  değişkeni FALSE değeri alacağından bu durumda kullanıcıya mesaj
  verilerek PHP kodu sonlandırılıyor.
- Bu kodun çalışabilmesi için önceden istiklal.txt dosyasının içine istiklal marşı yazılmış olarak hazır edilmeli ve dosyaoku.php dosyası ve istiklal.txt dosyasının her ikisi de "c:\wamp\www" klasörü içinde bulunmalıdır.

Bilgilendirme & Metin dosyaların yazdırılması sırasında tüm içeriğin tek satırda görüntülenmesi problemi nasıl çözülür?

Metin dosyalar içindeki alt satıra geçirme işareti olan "\n" karakteri HTML dilinde işe yaramadığından dolayı <u>echo</u> <u>\$dosyametni</u> şeklinde yazarsanız <u>tüm</u> <u>dosya içeriği tek satır olarak</u> ekranda görüntülenecektir. Bu sorunu çözebilmek için Metin <u>dosyalardaki "</u>\n" işaretini <<u>br</u> /> şekline dönüştüren nl2br() fonksiyonunu kullanabilirsiniz. Kod 12.10'daki örnekte bu fonksiyon kullanılmıştır.

2

12.7.3 Dosyaya Yazma: Dosyaya yazmak için fwrite() fonksiyonu kullanılır. Bu fonksiyon iki parametreye ihtiyaç duyar: bunlardan ilki bir dosya işaretçisi ve ikincisi yazılacak metindir. fwrite() fonksiyonu isteğe bağlı (opsiyonel) üçüncü bir parametre olarak yazılacak veri miktarını alabilir. Eğer veri miktarı belirtildiği takdirde verilen değere ulaşıldığında yazma işlemi sonlandırılır. Aşağıdaki örnekte **yenidosya.txt** adlı bir dosyaya bilgi yazma örneği görülmektedir.

```
<?php
$dosyaadi = "yenidosya.txt";
$file = fopen($dosyaadi, "w");
if( $file == false ) {
  echo ( "Hata: Dosya acılamıvor!" );
  exit();
fwrite( $file, "Bu metin dosya içine yazılacaktır.\n" );
fclose( $file );
if( file exists($dosyaadi)) {
  echo ("Dosya başarıyla oluşturuldu.");
```

Kod 12.12 PHP ile dosyaya yazma işlemi

• Bu kodun çalışabilmesi için yukarıdaki örneği **dosyayaz.php** adıyla "c:\wamp\www" klasörü içine kaydediniz. Kodu çalıştırdıktan sonra **yenidosya.txt** adlı metin dosya "c:\wamp\www" klasörü içinde oluşacaktır. Kod çalıştıktan sonra oluşan metin dosya Şekil 12.6'da görülmektedir



Şekil 12.6. Dosya yazma örneğinde oluşan yenidosya

Yukarıdaki dosya yazma örneğinde fwrite() ile içine bilgi yazılan dosyanın oluşup oluşmadığı file\_exist() fonksiyonu ile kontrol edilmiştir.
Eğer dosya oluşmamış ise kullanıcıya hata bildirimi görüntülenir. Dosya oluşmaması nedenleri arasında hard diskte yer kalmaması veya dosyanın oluşturulacağı klasörde PHP yorumlayıcısının dosya oluşturma yetkisinin olmaması sayılabilir.

### (12.8) PHP İLE XML DOSYA KULLANIMI

 XML belgelerin temelde düz metin dosyalar olduğu 6. Bölümde anlatılmıştı. Fakat içlerine bilgi girişi için hiyerarşik bir veri yapısı kullanılıyor olması XML belgelerin sıradan düz metin dosyalar gibi değil ağaç yapısı şeklinde işlenmesi ihtiyacı doğurmuştur. Bu bölümde iki ayrı alt başlık olarak PHP ile XML dosyadan bilgi okumak ve XML dosya oluşturma işlemleri anlatılmıştır.

### (12.8.1) PHP İLE XML BELGE OLUŞTURMA

- XML belge oluşturmak için öncelikle XML belgede bulunacak veriler için bir hiyerarşik ağaç yapısı belirlenmelidir. Bu örnekte oluşturulacak belge için Şekil 12.7'de görülen ağaç yapısı kullanılacaktır.
- PHP ile XML işlemleri için SimpleXML adlı PHP eklentisi kullanılmaktadır. Bu eklenti PHP ile birlikte standart olarak gelmektedir. Öncelikle aşağıdaki gibi yeni bir SimpleXML nesnesi oluşturuluyor ve bu sırada XML dosyanın başlangıç satırı ve kök düğümü (okul) tanımlanıyor ve \$xml adlı değişkene atanıyor.

```
$xml = new SimpleXMLElement('<?xml version="1.0"
encoding="utf-8"?><okul></okul>');
```

• Bundan sonra, eldeki XML dosyaya bir öğrenci düğümü eklenmelidir. Bunun için \$xml adlı değişkene addchild metodu ile ogrenci adlı düğüm ekleniyor ve ad karışıklığı olmaması için sonraki satırlarda kullanmak üzere \$data olarak bir değişkene kaydediliyor.

\$data = \$xml->addChild('ogrenci');

- Bundan sonraki satırlarda öğrenci düğümüne ad, soyad, sınıf ve numara düğümleri ekleniyor ve bunların her biri için sırasıyla Erkin, Beyaz, 3, '150012001' değerleri veriliyor.
- Örnek XML dosyada 2 adet öğrenci bilgileri depolanacağından addchild metoduyla ikinci öğrenci ve buna karşılık gelen alt düğümler benzer şekilde ekleniyor. Son aşamada ise tüm XML kodlarının yazılacağı veriler.xml dosyası oluşturuluyor.



Şekil 12.7. PHP ile oluşturulacak XML

belge için örnek ağaç yapısı

3

### \$xml->asXML()

• Yukarıdaki satır ile \$xml değişkeni XML dosya biçimine çevrilerek fwrite() komutu ile dosyaya yazdırılıyor ve fclose() metoduyla dosya kapatılıyor. Oluşturulan uygulamanın tamamı aşağıdaki örnekte görülmektedir.

#### Kod 12.13 PHP ile XML oluşturma

```
<?php
$xml = new SimpleXMLElement('<?xml
version="1.0"
encoding="utf-8"?><okul></okul>');
//birinci ogrenci
$data = $xml->addChild('ogrenci');
$data->addChild('ad', 'Erkin');
$data->addChild('soyad', 'Beyaz');
$data->addChild('sinif', '3');
$data->addChild('numara', '150012001');
```

```
//ikinci ogrenci
$data = $xml->addChild('ogrenci');
$data->addChild('ad', 'Selim');
$data->addChild('soyad', 'Koyuncu');
$data->addChild('sinif', '2');
$data->addChild('numara', '130300212');
//veriler.xml adlı dosyaya yazdırılıyor
$file = fopen( "veriler.xml", "w" );
fwrite( $file,$xml->asXML());
fclose( $file );
?> 3
```

 Yukarıdaki kodları xmlyaz.php adıyla "c:\wamp\www" klasörü içine kaydediniz.
 Sonra web tarayıcınızda http://localhost/xmlyaz.php adresini yazarak çalıştırdığınızda "c:\wamp\www" klasörü içinde Kod 12.14'de içeriği verilmiş olan veriler.xml dosyasının oluştuğu görülecektir.

**Bilgilendirme** *Sco Bu kodu çalıştırınca neden ekranda hiçbir şey görüntülenmiyor*? Bu kod ile oluşan XML dosya fwrite() ile diske kaydedildiğinden ekranda herhangi bir şey görüntülenmeyecektir. Ekranda görüntülenmesi için fwrite() ile başlayan satır yerine

echo \$xml->asXML();

yazarak XML belgenin ekranda görüntülenmesi sağalabilir.

Kod 12.14 PHP ile oluşurulan XML dosya

<?xml version="1.0" encoding="UTF-8" ?> <okul> <ogrenci> <ad>Erkin</ad> <soyad>Beyaz</soyad> <sinif>3</sinif> <numara>1203098</numara> </ogrenci> <ogrenci> <ad>**Tunc**</ad> <soyad>Karalioğlu</soyad> <sinif>1</sinif> <numara>1503044</numara> </ogrenci> </okul>

Bilgilendirme & Çok satır içeren XML belgeler nasıl oluşturabilirim?

Bu örnekte sadece 2 öğrenci bilgileri tutan XML dosya oluşturulmuştur. Daha çok sayıda öğrenci tutmak isterseniz PHP döngü komutları ile SimpleXML eklentisini birlikte kullanabilirsiniz.

### (12.8.2) PHP İLE XML BELGEYİ OKUMA

• XML belgeden bilgi okumayı anlatabilmek için önceki örnekte oluşturulan **veriler.xml** dosyası kullanılacaktır. SimpleXML eklentisinin sağladığı

simplexml load file()

metodu ile XML belgesi kolaylıkla okunabilmektedir. Bu metot, kök düğüm içinde okuduğu her alt düğüme, bir dizi elemanı şeklinde erişmeyi sağlar. Aşağıdaki örnekte ilk öğrenciye ogrenci[0] ikinci öğrenciye ise ogrenci[1] şeklinde erişilebilmektedir. Düğümlerin alt düğümlerine ise "->" operatörü yardımıyla erişilmektedir.

Kod 12.15 PHP ile XML belgeden veri okuma

```
<?php
$xml=simplexml load file("veriler.xml");
//ogrenci 1 verileri
echo $xml->ogrenci[0]->ad . "<br>";
echo $xml->ogrenci[0]->soyad . "<br>";
echo $xml->ogrenci[0]->sinif . "<br>";
echo $xml->ogrenci[0]->numara . "<br>";
//ogrenci 2 verileri
echo $xml->ogrenci[1]->ad . "<br>";
echo $xml->ogrenci[1]->soyad . "<br>";
echo $xml->ogrenci[1]->sinif . "<br>";
echo $xml->ogrenci[1]->numara . "<br>";
?>
```

Yukarıdaki kodları xmloku.php adıyla "c:\wamp\www" klasörü içine kaydediniz. Sonrasında web tarayıcınızda http://localhost/xmloku.php adresini yazarak çalıştırdığınızda XML belgedeki veriler ekranda görüntülenecektir. Ekran çıktısı Şekil 12.8'deki gibi olacaktır. Bu örnekte sadece 2 öğrenci bilgileri tutan XML dosyadan veri okunduğu için bilgiler ayrı ayrı echo() komutları ile ekrana yazdırılmıştır. Daha çok veri içeren XML dosyalardan bilgi okuduğunuzda bunları ekrana yazdırırken PHP döngülerini kullanırsanız kodlarınız hem daha kısa hem daha okunaklı olacaktır.



Şekil 12.8. PHP ile XML belgedeki verilen okunması

# (12.9) ÖZET

PHP ile HTML form bilgilerinin işlenmesi formun hangi metot ile bilgi gönderdiğine göre değişir. GET metodu • ile postalanan formlardaki her bir form elemanı PHP dilinde \$ GET['..'] ifadesi ile alınır. GET metoduyla değerler bu şekilde URL'ye eklenerek iletildiğinden gönderilebilecek bilgi miktarı en uzun URL karakter sınırı olan 256 karakter ile sınırlıdır. Karmaşık formlarda çoğunlukla POST metodu tercih edilir. GET metodunun aksine, POST metodunda bilgi taşınırken URL adresinin arkasına eklenmediği için adres çubuğunda görünmez. POST metodu ile gönderilen bilgiler işlenirken \$ GET değişkeni yerine \$ POST değişkeni kullanılır. Kendi kendine bilgi gönderen bir PHP dosyası oluşturmak için dosya içine hem formdan gelen bilgileri alan PHP kodu hem de HTML form kodları yazılır. HTML formun action niteliğine <?php \$ PHP SELF ?> ifadesi yazılarak formun kendi kendine bilgi göndermesi sağlanır. PHP' de harici kod dosyası kullanmak için include(), require() olmak üzere iki farklı fonksiyon mevcuttur. Eğer harici olarak eklenen dosyadaki hatalarda ana programın çalışması durdursun istiyorsanız require(), çalışmaya devam etsin istiyorsanız include() kullanabilirsiniz. PHP' de dosya işlemleri için öncelikle fopen() fonksiyonu ile dosya açmak gereklidir. Veriyi okumak için fread() fonksiyonu kullanılır. Dosyaya yazmak için fwrite() fonksiyonu kullanılır. PHP ile XML işlemleri için SimpleXML adlı PHP eklentisi kullanılmaktadır. 4