

**T.C.  
MİLLÎ EĞİTİM BAKANLIĞI**

## **BİLİŞİM TEKNOLOJİLERİ**

**AÇIK KAYNAK KODLA FORM İŞLEMLERİ  
VE OTURUM YÖNETİMİ  
482BK0133**

**Ankara, 2012**

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

# İÇİNDEKİLER

AÇIKLAMALAR .....	ii
GİRİŞ .....	1
ÖĞRENME FAALİYETİ-1 .....	3
1. FORM İŞLEMLERİ.....	3
1.1. Form Oluşturma .....	4
1.2. Form Kontrol Yapılarını Kullanma.....	5
1.2.1. <input> Kullanımı .....	5
1.2.2. <textarea> Kullanımı .....	6
1.2.3. <select> ve <option> Kullanımı .....	6
1.3. POST ve GET Metodu ile Formlardan Bilgi Almak .....	7
1.3.1. GET Metodu .....	7
1.3.2. POST Metodu .....	9
1.4. Kullanıcıdan Gelen Bilgi ve Güvenlik.....	16
UYGULAMA FAALİYETİ .....	19
ÖLÇME VE DEĞERLENDİRME .....	21
ÖĞRENME FAALİYETİ-2 .....	23
2. DÜZENLİ İFADELER .....	23
2.1. Düzenli İfade Nedir?.....	23
2.1.1. Kullanım Amaçları .....	24
2.2. Söz Dizisi .....	24
2.2.1. Söz Dizisinde Kullanılan Özel Karakterler .....	25
2.2.2. Söz Dizilerinde Görünmez (non-printing) Karakterleri Yakalamak .....	29
2.2.3. Söz Dizilerinde Geniş Aralıkları İçeren Karakter Setleri .....	29
2.2.4. Söz Dizilerinde Sınırları Yakalamak .....	30
2.2.5. Yer Değiştirme .....	31
2.3. PHP’de Düzenli İfadeler .....	31
2.3.1. PHP’de Düzenli İfadeler İçin Kullanılan Fonksiyonlar .....	31
UYGULAMA FAALİYETİ .....	38
ÖLÇME VE DEĞERLENDİRME .....	40
ÖĞRENME FAALİYETİ-3 .....	42
3. OTURUM YÖNETİMİ.....	42
3.1. Çerezler .....	42
3.1.1. Çerezlerin Kullanım Yerleri .....	42
3.1.2. PHP ile Çerezler .....	43
3.2. PHP’de Session (Oturum Yönetimi).....	46
3.2.1. Session Kullanım Yerleri.....	46
UYGULAMA FAALİYETİ .....	50
ÖLÇME VE DEĞERLENDİRME .....	52
MODÜL DEĞERLENDİRME .....	54
CEVAP ANAHTARLARI .....	57
KAYNAKÇA .....	59

# AÇIKLAMALAR

<b>KOD</b>	<b>482BK0133</b>
<b>ALAN</b>	<b>Bilişim Teknolojileri</b>
<b>DAL/MESLEK</b>	<b>Web Programcılığı</b>
<b>MODÜLÜN ADI</b>	<b>Açık Kaynak Kodla Form İşlemleri Ve Oturum Yönetimi</b>
<b>MODÜLÜN TANIMI</b>	Form işlemleri, düzenli ifadeler ve oturum yönetimi işlemlerine yönelik öğrenme materyalidir.
<b>SÜRE</b>	40/32
<b>ÖN KOŞUL</b>	“Açık Kaynak Kodla Temel Uygulamalar” modülünü tamamlamış olmak.
<b>YETERLİK</b>	Form, düzenli ifadeler ve oturum yönetimi işlemlerini yapmak.
<b>MODÜLÜN AMACI</b>	<b>Genel Amaç</b> Bu modül ile gerekli ortam sağlandığında; düzenli ifadeleri kullanarak form ve oturum açma işlemlerini yapabileceksiniz. <b>Amaçlar</b> <b>1.</b> Form işlemlerini yapabileceksiniz. <b>2.</b> Düzenli ifadeleri kullanabileceksiniz. <b>3.</b> Oturum işlemlerini yapabileceksiniz.
<b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b>	<b>Ortam:</b> Uygulamaların yapılabileceği uygun laboratuvar ortamı.
<b>ÖLÇME VE DEĞERLENDİRME</b>	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığımız bilgi ve becerileri ölçerek sizi değerlendirecektir.

# GİRİŞ

## Sevgili Öğrenci,

Bu modülde üç ana başlık altında üç önemli konu öğreneceksiniz. Bu konular sırasıyla Form İşlemleri, Düzenli İfadeler ve Oturum Yönetimi'dir. Öğreneceğiniz konular PHP dili esas alınarak anlatılmış olsa da kullanılan kavramlar ve yöntemler sebebiyle PHP'yi kullanmayan internet programcıları tarafında da etüt edilmelidirler.

Modülün konusu olan başlıklar, Bilişim Teknolojileri alanında çalışmayı hedefleyenlerin üzerinde titizlikle durması gereken konulardan oluşmaktadır.

Bu modülde:

- Form İşlemleri ile günümüz internet sistemlerinde veri akışının nasıl gerçekleştiğini,
- Düzenli İfadeler ile verinin nasıl değerlendirildiğini ve istenilen biçimlerde olup olmadığını,
- Oturum Yönetimi ile internet kullanıcılarına özel sistemlerin nasıl tasarlandığını kavrayacaksınız.



# ÖĞRENME FAALİYETİ-1

## AMAÇ

Form elemanları oluşturabilecek ve bu elemanlardan gelen bilgileri PHP ortamında elde edebileceksiniz.

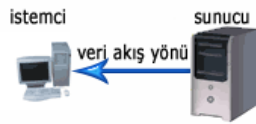
## ARAŞTIRMA

- Statik ve dinamik site arasındaki temel fark nedir?
- Form nedir? İnternet sitelerinde formlar nerelerde kullanılır?
- Form elemanları nelerdir?
- Formlarda kullanılan yöntemler nelerdir?

## 1. FORM İŞLEMLERİ

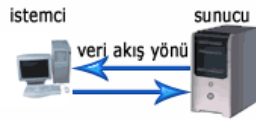
Formların internet siteleri üstündeki etkisini ve kullanım yöntemlerini anlayabilmek için öncelikle statik ve dinamik siteler arasındaki farkı iyi kavramak gerekir. O zaman öncelikle kısaca statik ve dinamik site kavramlarına bir bakalım.

Genel bir bakış açısıyla web siteleri iki kategoride incelenebilir. Bunlar, durağan ve etkileşimli web siteleridir. Durağan, diğer bir ifade ile statik web siteleri web sayfalarının hazırlandıktan sonra hiçbir değişime uğramayan ve sadece tek taraflı bilgi akışının olduğu sayfalardır. Bilgi, web sayfası sunucusundan sayfayı ziyaret eden internet tarayıcısına, diğer bir ifade ile istemciye doğrudur. Site tamamen siteyi oluşturan kişi tarafından düzenlenebilir.



Şekil 1: Durağan web sitelerinde veri akışı

İkinci web sitesi çeşidi ise, günümüzde yaygın olarak kullanılan dinamik (etkileşimli) web siteleridir. Dinamik sitelerde kullanıcı etkileşimi ön plandadır. Web sayfalarına yeni içerikler eklenebilir, var olanlar güncellenebilir ya da silinebilir. Dinamik web sitelerinde, bilgi, web sayfası sunucusundan sayfayı ziyaret eden tarayıcıya olabileceği gibi tarayıcıdan web sayfası sunucusuna doğru da olabilir.



Şekil 2: Dinamik web sitelerinde veri akışı

Dinamik bir web sitesi, her kullanıcı için ya da her ziyarette özel olarak oluşturulabilir. Dünya üzerindeki her kullanıcı, bir siteye kendi kullanıcı adıyla girdiğinde diğer kullanıcılardan farklı bir içeriğe erişebilir. Bu farklı içerik, web sitesinin renkleri, teması, müzikleri, sitedeki yetki derecesi vb. şeklinde olabilir. Günümüz internetindeki popüler siteler dinamik sitelerdir.

Dinamik web sitelerini hazırlamada en çok kullanılan yapı formlardır. HTML formları çeşitli kontrolleri içeren ve ziyaretçi tarafından değiştirilebilen yapılardır. Formların içerdiği kontrol yapıları şunlardır;

- buttonlar (button, submit, reset),
- checkbox,
- radio,
- text,
- hidden,
- password,
- textarea,
- select,
- option,
- file,
- image.

## 1.1. Form Oluşturma

Bir HTML formu `<form>` etiketi ile başlar ve `</form>` etiketi ile sonlanır. Bu etiketler arasına yerleştirilen kontrol yapılarıyla formumuzu oluşturmuş oluruz. HTML formları çeşitli görevleri yerine getirmek için kullanılabilir. Örneğin anketler, yorum bölümleri, kullanıcı üyelik bölümleri, forumlar, sosyal paylaşım sitelerindeki video paylaşımı, resim paylaşımı ya da paylaşılanların beğenilmesi, derecelendirilmesi vb.

**<form>** etiketinin üç temel özelliği form oluşturmak için önemlidir.

- **action:** Formdaki bilgilerin gönderileceği url adresi.
- **method:** Bilginin gönderilme yöntemi. POST yada GET yöntemlerine sahiptir. Değer atanmadığı durumlarda GET yöntemi kullanılır. İleri konularda kullanımlarına değinilecektir.
- **name:** Formun ismidir. Bu özelliğe değer verirken PHP'nin değişken kurallarına uymak gerekir.



PHP'de değişken ismi verirken boşluk ya da özel karakter kullanmadığımızı hatırlayınız.



### Örnek:

```
<form action= "url_adresi" method= "get name"?= "form_ismi"></form>
```

Form etiketi bir formun oluşabilmesi için önemlidir fakat form kontrol yapıları olmadan anlamlı bir form oluşturmuş olmayız.

## 1.2. Form Kontrol Yapılarını Kullanma

HTML formlarının etkin bir şekilde kullanılması kontrol yapıları ile mümkündür. Kontrol yapıları sayesinde amaca uygun formlar oluşturulabilir ve internet kullanıcılarının kullanımına sunulabilir. Kontrol yapıları sahip oldukları çeşitlilikle kullanıcıların kolay ve anlaşılabilir bir şekilde formları kullanmalarını sağlamaktadır.

Formlarda kullanılan kontrol yapıları oluşturulurken her kontrolün sahip olduğu “*name*” özelliği vardır. Name özelliği formlardan gelen bilgi elde edilirken kullanılmaktadır.

Form kontrol yapıları aşağıda kullanımları ile listelenmiştir.

### 1.2.1. <input> Kullanımı

Genel amaçlı bir form etiketidir. Sonlandırıcı etiketi yoktur. Girdi türü **type** ögesinde belirtilerek farklı girdilerin alınmasını sağlar.

Girilecek Kod	Görevi
type = "eleman türü"	Eleman türü belirtilir. text, password, checkbox, radio, submit, reset, file, hidden, image, button değerlerinden biri kullanılır. Etiket bu değerlerle nasıl kullanıldığı aşağıda örneklerle açıklanmaya çalışılacaktır.
name = "isim"	Girilen verinin hangi değişken ismi altında değerlendirileceğini belirtmek için kullanılır. Mutlaka belirtilmelidir.
value = "değer"	"radio" ve "checkbox" dışında bu parametrenin kullanımı isteğe bağlıdır. Belirtilecek "değer", eleman türüne göre değişir. Etiket varsayılan değeri olarak kullanılır.
size = "değer"	"text" ve "password" elemanlarında karakter sayısını, diğerlerinde ise piksel cinsinden genişliği belirtmek için kullanılır.
maxlength = "değer"	"text" ve "password" elemanlarında girilebilecek en çok karakter sayısını belirtmek için kullanılır. Ön tanımlı değeri sınırsızdır.
checked = "değer"	Sadece "radio" ve "checkbox" elemanlarında seçili eleman "olarak belirtmek için kullanılır. Diğerlerinde yok sayılır.
src = "dosya ismi"	"image" elemanında resim dosyasını belirtmek için kullanılır.
alt = "metin"	Fare eleman üzerindeyken balon içinde görünecek açıklama metni için kullanılır.
align = "left center right"	Elemanın form üzerinde nasıl konumlanacağını belirtmek için kullanılır.
readonly	Veri girme amacıyla kullanılmayacak elemanlar içindir.
disabled	Veri girişi engellenecek elemanlar içindir.

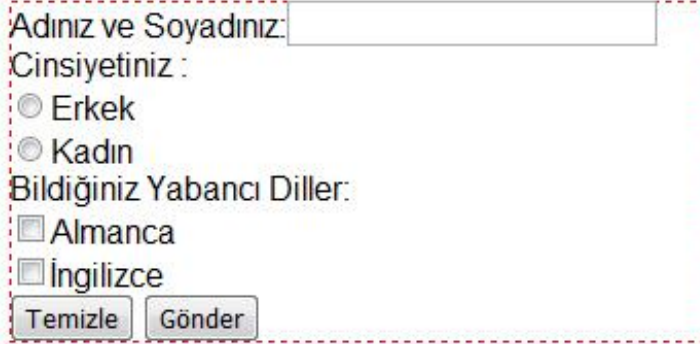
Tablo 1 <input> etiketi yardımcı kodları

## Örnek:

Örnek kaynak kodu:

```
<form action= "url_adresi" method= "get" name= "form_ismi">
Adınız ve Soyadınız:<input type="text" name="adSoyad" size="30" />
Cinsiyetiniz : <br />
<input type="radio" name="cinsiyet" value="Erkek"/>Erkek<br />
<input type="radio" name="cinsiyet" value="Kadın"/>Kadın<br />
    Bildiğiniz Yabancı Diller:<br />
<input type="checkbox" name="yDil[]" value="Almanca"/>Almanca<br />
<input type="checkbox" name="yDil[]" value="İngilizce"/>İngilizce<br />
<input type="reset" name="temize" value="Temizle"/>
<input type="submit" name="gonder" value="Gönder"/>
</form>
```

Örnek Tarayıcı Görünümü:



### 1.2.2. <textarea> Kullanımı

Formumuza uzun metinler yazmak için kullanılmaktadır. Örneğin adres bilgisi ya da yorum girişi için kullanılabilir bir form elemanıdır.

<textarea rows="5" cols="20" name="adres"></textarea> şeklinde kullanılır.

<textarea> ile kullanılan;

- rows, satır sayısını belirtir
- cols sütun sayısını belirtir.

### 1.2.3. <select> ve <option> Kullanımı

Formlarımızda seçim kutuları oluşturmak için kullanılan yapılardır. Örneğin bir şehir seçimi için kullanılabilir.<select> ve <option> kullanılırken "name" özelliğinin <select> etiketi içerisinde "value" özelliğinin ise <option> içerisinde kullanılması gerekmektedir.

```
<select name="sehir">
    <option value="Ankara">Ankara</option>
    <option value="İstanbul">İstanbul </option>
    <option value="Samsun">Samsun</option>
</select>
```

## 1.3. POST ve GET Metodu ile Formlardan Bilgi Almak

PHP'nin en güçlü özelliklerinden biri HTML formlarına yaklaşım biçimidir. Bilinmesi gereken ilk önemli durum, form içindeki tüm elemanların PHP tarafından aktif olarak kullanılabilir olmasıdır.

Formların veri aktarabilmeleri için iki ayrı yöntem vardır. Bunlar POST ve GET yöntemleridir. Dolayısıyla PHP'de formlardan gelen bilgileri almanın da iki yöntemi vardır. Bunlar;\$\_POST,\$\_GET yöntemleridir.

### 1.3.1. GET Metodu

GET metodu ile veriler iletildiğinde forma girilen tüm bilgiler adres satırında URL'ye eklenerek iletilir.

#### Örnek:

Aşağıda 'form\_get\_1.html' dosyasına dikkat edilirse herhangi bir PHP kodu içermemektedir. Bununla birlikte bu dosyada bulunan form bir PHP dosyasına yönlendirilmektedir.

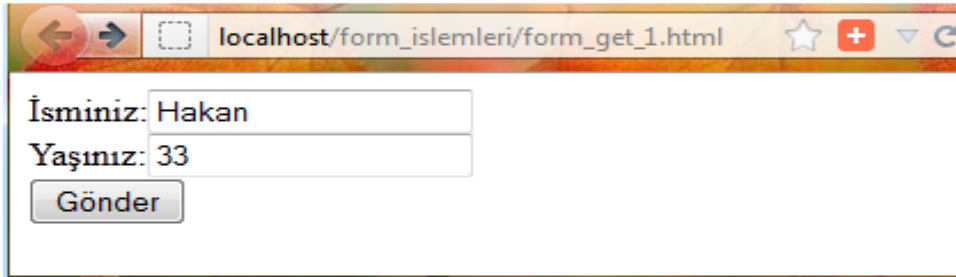
'formu\_isle\_1.php' isimli dosya ise kendisine GET metodu kullanılarak gönderilen bilgileri alıp ekrana yazmak için hazırlanmış bir PHP dosyasıdır. Formdan gelen bilgiler \$\_GET değişkeni kullanılarak alınmaktadır.

Örnekte formu\_isle\_1.php' sayfasının tarayıcı görünümüne dikkat edilecek olursa adres çubuğunda bilginin URL'ye nasıl eklendiğini görebiliriz. Burada bilgilerin eklenme işini gerçekleştiren yapı HTML <form> etiketidir. URL adresinden sonra "?" kullanılarak bilgi ekleme işlemi gerçekleştirilmiştir. Bilgiler, değişken\_ismi=değer şeklinde eklenmiş ve her yeni bilgi "&" işareti ile peşi sıra eklenmiştir.

form\_get\_1.html dosyası kaynak kodu:

```
<html>
<head>
<title>Formlarda GET kullanımı</title>
</head>
<body>
<form action="formu_isle_1.php" method="get" name="form">
  İsminiz:<input type="text" name="isim" /><br />
  Yaşınız:<input type="text" name="yas" /><br />
  <input type="submit" value="Gönder"/>
</form>
</body>
</html>
```

form\_get\_1.html dosyası tarayıcı görünümü:



localhost/form\_islemleri/form\_get\_1.html

İsminiz: Hakan  
Yaşınız: 33  
Gönder

formu\_isle\_1.php dosyası kaynak kodu:

```
<html>
<head>
<title>Formlarda GET kullanımı</title>
</head>
<body>
<?php
  echo $_GET['isim']."-".$_GET['yas'];
?>
</body>
</html>
```

formu\_isle\_1.php dosyası tarayıcı görünümü:



localhost/form\_islemleri/formu\_isle\_1.php?isim=Hakan&yas=33

İsminiz: Hakan Yaşınız: 33

HTML formlarında bu tür bilgiler `<a>...</a>` etiketi kullanılarak da oluşturulabilmektedir. Bu işlem “**URL sorgu cümlesi oluşturmak**” olarak adlandırılabilir.

Yukarıdaki örneğin `<a>`etiketi ile oluşturulması şu şekildedir;

`<a href="formu_isle_1.php?isim=Hakan&yas=33">URL Sorgu Cümlesi</a>` URL sorgu cümleleri çoğunlukla bir yazının devamını okumak için devamını okuyun linklerinde karşımıza çıkmaktadır.

### Örnek:

Formdan gönderilen iki sayının toplamını ekrana yazan örneği hazırlayalım. PHP değişkenleri olan `$_GET` ve `$_POST` değişkenleri bütün veri türlerini taşıyabilir.

form\_get\_2.html dosyası kaynak kodu:

```
<html>
<head>
<title>Form ile Toplama</title>
</head>
<body>
<form action="form_ile_topla.php" method="get" name="frmTopla">
Sayı1:<input type="text" name="sayi1" /><br />
Sayı2:<input type="text" name="sayi2" /><br />
<input type="reset" value="Temizle" /> <input type="submit" value="Gönder" />
</form>
</body>
</html>
```

form\_ile\_topla.php dosyası kaynak kodu:

```
<?php
$sayi1=$_GET['sayi1'];
$sayi2=$_GET['sayi2'];
$sonuc=$sayi1+$sayi2;
echo $sonuc;
?>
```

GET yöntemi kullanılarak oluşturulan formlar çoğunlukla basit formlardır. Karmaşık formlarda çoğunlukla POST metodu tercih edilir.

### 1.3.2. POST Metodu

POST metodu formlarda bilgiyi taşıma yöntemlerinden bir diğeridir. GET metodunun aksine bilgi taşınırken URL adresinde görünmez. Bu yönüyle POST metodunun daha güvenilir olduğunu söyleyebiliriz. Ayrıca POST metodunda bilginin taşınması esnasında bir sınırlama yoktur.

POST metodunun kullanımı GET ile bezerdir. Oluşturulan formun “method” özelliği POST yapılarak formun bu yöntemi kullanarak veriyi göndermesi sağlanır. PHP tarafında ise `$_GET` dizi değişkeni yerine `$_POST` dizi değişkeni kullanılır.

### Örnek:

Aşağıda ‘form\_post\_1.html’ dosyası ‘formu\_isle\_1.php’ isimli dosyaya yönlendirilmiştir.

‘formu\_isle\_1.php’ isimli dosya ise kendisine POST metodu kullanılarak gönderilen bilgileri alıp ekrana yazmak için hazırlanmış bir PHP dosyasıdır. Formdan gelen bilgiler `$_POST` değişkeni kullanılarak alınmaktadır.

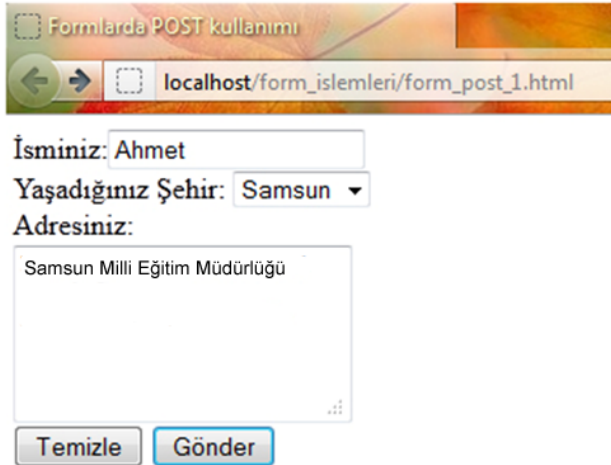
### Notice: Undefinedindex:

Eğer php sayfanızda yukarıdaki hata mesajını ve hata satırını görüyorsanız bunun anlamı şudur: \$\_GET yada \$\_POST dizi değişkenleri içine yazdığınız index isimleri bulunamadı. Formdan gelen kontrol yapılarına verilen isimleri kontrol ediniz.

form\_post\_1.html dosyası kaynak kodu:

```
<html>
<head>
<title>Formlarda POST kullanımı</title>
</head>
<body>
<form action="formu_isle_1.php" method="post" name="form">
  İsminiz:<input type="text" name="isim" /><br />
  Yaşadığınız Şehir:
  <select name="sehir">
    <option value="Ankara">Ankara</option>
    <option value="İstanbul"> İstanbul </option>
    <option value="Samsun">Samsun</option>
  </select><br />
  Adresiniz:<br />
  <textarea name="adres" rows="5" cols="20"></textarea><br />
  <input type="reset" value="Temizle"/> <input type="submit" value="Gönder"/>
</form>
</body>
</html>
```

form\_post\_1.html dosyası tarayıcı görünümü:



Formlarda POST kullanımı

localhost/form\_islemleri/form\_post\_1.html

İsminiz: Ahmet

Yaşadığınız Şehir: Samsun

Adresiniz:

Samsun Milli Eğitim Müdürlüğü

Temizle Gönder

```
<title>Formlarda POST kullanımı</title>
</head>
<body>
<?php
    echo "İsminiz: ".$_POST['isim'].'<br />';
    echo "Yaşadığınız şehir: ".$_POST['sehir'].'<br />';
    echo "Adresiniz: ".$_POST['adres'].'<br />';
?>
</body>
</html>
```

formu\_isle\_1.php dosyası tarayıcı görünümü:



İsminiz: Ahmet  
Yaşadığınız şehir: Samsun  
Adresiniz: Samsun Milli Eğitim Müdürlüğü

### Örnek:

Bu örneğimizde POST metodu kullanılarak farklı form kontrol elemanlarından bilgi elde edilecektir.

‘form\_post\_2.html’ dosyasında Cinsiyet ve Yabancı Dil sırasıyla “radio” ve “checkbox” özelliğini taşıyan form kontrolleridir.

Cinsiyet için kullanılan iki farklı “radio” özelliğindeki form kontrollerinin “name” özellikleri aynıdır. Bunun anlamı iki farklı değerden sadece bir tanesinin ‘formu\_isle\_2.php’ dosyasına iletilecektir.

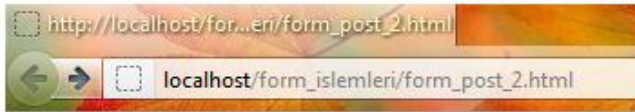
Yabancı Dil için kullanılan üç “checkbox” özelliğindeki form kontrollerinin “name” özellikleri de aynıdır. Farklı olarak “[ ]” (köşeli parantez) işaretleri eklenmiştir. Bunun anlamı, verilen isim üzerinden değerler dizi olarak gönderilecektir. ‘formu\_isle\_2.php’ dosyasında ise diğerlerinden farklı olarak foreach döngüsü kullanılarak dizi değişkeni üzerinden istenilen değerler elde edilecektir.



form\_post\_2.html dosyası kaynak kodu:

```
<form action= "formu_isle_2.php" method= "post" name= "form_ismi">
İsim ve Soyisminiz:<input type="text" name="adSoyad" size="30" /><br />
Cinsiyetiniz : <br />
<input type="radio" name="cinsiyet" value="Erkek"/>Erkek<br />
<input type="radio" name="cinsiyet" value="Kadın"/>Kadın<br />
    Bildiğiniz Yabancı Diller:<br />
<input type="checkbox" name="yDil[]" value="Almanca"/>Almanca<br />
<input type="checkbox" name="yDil[]" value="İngilizce"/>İngilizce<br />
<input type="checkbox" name="yDil[]" value="Çince"/>Çince<br />
<input type="reset" name="temize" value="Temizle"/>
<input type="submit" name="gonder" value="Gönder"/>
</form>
```

form\_post\_2.html dosyası tarayıcı görünümü:



İsim ve Soyisminiz: Elif Aydın

Cinsiyetiniz :

Erkek

Kadın

Bildiğiniz Yabancı Diller:

Almanca

İngilizce

Çince

Temizle

Gönder

‘form\_post\_2.html’ dosyasında kullanılan iki kontrol nesnesi olan “reset” ve “submit” özelliğindeki butonlara isim verilmemiştir. Bunun nedeni, PHP sayfasında elde edilmeyecek olan kontrol nesnelere isim vermeye gerek yoktur.



formu\_isle\_2.php dosyası kaynak kodu:

```
<?php
echo "İsim ve Soyisminiz: ".$_POST['adSoyad'].'<br />';
echo "Cinsiyetiniz: ".$_POST['cinsiyet'].'<br />';
echo "Bildiğiniz Yabancı Diller: ".$_POST['yDil'].'<br />';
foreach($_POST['yDil'] as $deger){
    echo $deger.'<br />';
}
?>
```

formu\_isle\_2.php dosyası tarayıcı görünümü:



İsim ve Soyisminiz: Elif Aydın  
Cinsiyetiniz: Kadın  
Bildiğiniz Yabancı Diller:  
İngilizce  
Çince

### Örnek:

Bu örneğimizde form yoluyla resim yükleme (upload) işlemi gerçekleştireceğiz. Resim yükleme işlemi için diğer formlardan farklı olarak <form> etiketinin “enctype” özelliğinin “multipart/form-data” olarak atanması gerekiyor. Bu atama ile birlikte formumuz MIME standartlarına uygun bütün dosyaların yüklemesini yapabilecektir.

**Multipurpose Internet Mail Extensions** (Çok amaçlı İnternet Posta Eklentileri); E-posta uygulamaları aracılığıyla gönderilecek olan iletiye çeşitli türdeki içeriği eklemek için kullanılan bir İnternet standardıdır.

Örneğimizdeki diğer bir fark ise “Resim Dosyası” alanı için oluşturulan “input” form kontrolünün “type” özelliği “file” olarak atanmasıdır. Bu sayede kullanıcılar tarayıcıda “Gözet” (Browse) düğmesi sayesinde forma istedikleri dosyayı gönderebileceklerdir.

Bu örnekte, resim dosyası olarak yükleme yapılacağı belirlendiği için, ‘form\_isle\_3.php’ dosyasında yüklenen resim bir ‘img’ etiketinin ‘src’ özelliğine atanarak resmin tarayıcıda gözükmesi sağlanmaktadır.

‘form\_post\_3.html’ dosyasında “Gözet” düğmesi tıklanarak bilgisayarımızda bulunan her hangi bir dosyanın forma iliştilmesi mümkün olmaktadır.

Formlara girilen bilgilerin kontrol edilmesi çok önemli bir konudur. Örneğin yukarıda Gözet düğmesi kullanılarak bir resim dosyası yerine farklı bir formatta dosya yüklemesi yapılsaydı örnek istenildiği gibi çalışmazdı. Dosyanın yüklenmesi gerçekleşirdi. Bununla

birlikte 'img' etiketine bir resim dosyası yolu verilmediğinden istenildiği gibi çalışmamış olurdu.

form\_post\_3.html dosyası kaynak kodu:

```
<form action= "formu_isle_3.php" method= "post" name= "form_ismi"
  enctype="multipart/form-data">
Resim Başlığı:<input type="text" name="baslik" size="30" /><br />
Yorumunuz:<br />
<textarea name="yorum" rows="5" cols="45"></textarea><br />
Resim Dosyası:<input type="file" name="dosya" size="30" /><br />
<input type="reset" value="Temizle"/> <input type="submit" value="Gönder"/>
</form>
```

form\_post\_3.html dosyası tarayıcı görünümü:



Örneğimizdeki dosyanın \$\_FILES özelliklerine şu kodla erişebiliriz.

```
<?php
```

```
foreach($_FILES['dosya'] as $sanahtar=>$deger)
```

```
echo "$sanahtar=>$deger<br />";
```

```
?>
```

Yukarıdaki kodun ekrandaki karşılığı şu şekilde olacaktır.

```
name=>kizkulesi.jpg
```

```
type=>image/jpeg
```

```
tmp_name=>D:\xampp\tmp\php7BA3.tmp
```

```
error=>0
```

```
size=>56647
```

Formdan gelen dosyanın sunucuya kopyalanması işlemi için ‘copy’ komutu kullanılmaktadır. Copy komutunun kullanımı şu şekildedir:

copy ( string \$kaynak , string \$hedef [, resource \$bağlam ] )

- \$kaynak dosyanın kaynak yolunu,
- \$hedef ise dosyanın kopyalanacağı adresi belirler.

formu\_isle\_3.php dosyası kaynak kodu:

```
<?php
echo "Resim Başlığı: " . $_POST['baslik'] . '<br />';
echo "Yorumunuz: " . $_POST['yorum'] . '<br />';
echo "Resim Dosyası: " . '<br />';
@copy($_FILES['dosya']['tmp_name'], $_FILES['dosya']['name']);
?>

```

formu\_isle\_3.php dosyası tarayıcı görünümü:



Resim Başlığı: Kız Kulesi

Yorumunuz: M.Ö. 24 yıllarına kadar uzanan tarihi bir geçmişe sahip olan kule, Karadeniz'in Marmara ile birleştiği yerde küçük bir ada üzerinde kurulmuştur.

Resim Dosyası:



‘form\_isle\_3.php’ dosyasında kullanılan \$\_FILES dizi değişkeni küresel bir değişkendir. Formdan gelen dosyanın elde edilmesi için kullanılmaktadır. \$\_FILES dizi değişkeninin sahip olduğu değerler şunlardır:

- name: Dosyanın ismi
- type: Dosyanın türü.
- tmp\_name: Kaynak dosyanın kopyasının yolu.
- error: Aktarım esnasındaki hata kodu.
- size: Dosyanın boyutu.

formu\_isle\_3.php dosyasının tarayıcıdan alınan kaynak görüntüsü:

```
1  
2 Resim Başlığı: Kız Kulesi<br />Yorumunuz:  
M.Ö. 24 yıllarına kadar uzanan tarihi bir  
geçmişe sahip olan kule, Karadeniz'in  
Marmara ile birleştiği yerde küçük bir ada  
üzerinde kurulmuştur.<br />Resim Dosyası:  
<br /> 
```

'form\_isle\_3.php' dosyasını görüntüleyen kaynak görüntüsü incelendiğinde resmin <img> etiketinin 'src' özelliğine bakıldığında resmin isminin olduğu görülecektir. Tarayıcıda görüntülenen resim sunucuya kopyalanan resimdir.

## 1.4. Kullanıcıdan Gelen Bilgi ve Güvenlik

PHP' de yazılmış bir uygulamanın en zayıf noktası, dilin kendisinde olan bir sorun değil, yazılan uygulamanın bazı varsayımlara dayanarak güvenliği göz önünde bulundurmamasıdır.

Kod yazarken her zaman yazdıklarınızı dikkatlice inceleyip, kullanıcıdan gelen bilgilerin doğru ve tam şekilde denetlenip onaylandığına emin olmanız gerekir. Kullanıcıdan gelen bilgiyi alıp işleme koyduğunuz her durumda kendinize şu soruları sorabilirsiniz:

- Bu betik (kod ya da script) benim istemediğim dosyalara erişim, değişiklik veya silme imkanı veriyor mu?
- Beklenmeyen bir bilgi geldiğinde de istenilen işlem yapılacak mı?
- Bu betik başka bir amaç için kullanılabilir mi?
- Bu betik, sistemde var olan başka betiklerle birlikte kötü amaçlı kullanılabilir mi?
- Yapılan işlemler daha sonra inceleme amacıyla kayda alınacak mı?

Yaptığımız projenin uygunluğuna göre kod yazmadan önce ve yazarken bu soruları sık sık kendinize sormanız, proje bittiğinde değerlendirme yapıp değişiklikler yapmanızdan çok daha faydalı olacaktır. Bu sayede proje tasarımı güvenlik düşünülerek yapılmış olacak ve daha sonra güvenlik dolayısıyla ciddi değişiklikler yapma zorunluluğunuz ortadan kalkacaktır. Güvenliğe vereceğiniz öncelikli önem, daha sonra yapacağınız ve güvenliği riske atacak değişikliklerden çok daha faydalıdır.

Belirtilen bir değişkenin değerini, kaynağı veya geçerliliği bozacağını düşündüğünüz register\_globals, magic\_quotes ve benzeri ayarları kapatmanız iyi olabilir. Hata bildirim kipi

'error\_reporting(E\_ALL)' kullanımı, aynı zamanda uyarıları da göstereceği için kodunuzda sorun olup olmadığı hakkında daha iyi bilgi verebilir.

PHP' de tip denetim yapısı katı değildir ve bu güvenlik sorunlarını tetiklemektedir. Uygulamanızda değişken tiplerinin neler olduğunu bilmek ve mantıksal olarak kullanıcıdan gelen değişkenlerin türünü bilmek birçok durumda hataları ortadan kaldırabilir.

PHP'de kullanılabilecek çok sayıda tip tanımlama fonksiyonu vardır. Bu fonksiyonlardan bazıları is\_int, is\_array, is\_bool, is\_null, is\_float, is\_numeric vb. dir.

### Örnek:

Formlardan gelen bilgiler her zaman beklenen doğru bilgiler olmayabilir. Bazen istenmeyen bilgilerde formlarla birlikte gönderilebilir. Bu bilgiler sayfalarımızın istenildiği gibi görünmesini engelleyebilir. Hatta çok daha ciddi problemler çıkmasına sebep olabilir.

form\_guvenlik.html dosyası kaynak kodu:

```
<form action="formu_denetle.php" method="post" name="form">
  İsminiz:<input type="text" name="isim" /><br />
  Yaşınız:<input type="text" name="yas" /><br />
  <input type="submit" value="Gönder"/>
</form>
```

form\_guvenlik.html dosyası tarayıcı görünümü:



formu\_denetle.php dosyası kaynak kodu:

```
<?php
echo 'İsminiz:'. $_POST['isim'].'<br />';
echo 'Yaşınız:'. $_POST['yas'].'<br />';
?>
```

formu\_denetle.php dosyası tarayıcı görüntüsü:





'*htmlspecialchars*' fonksiyonu verilen metin içerisindeki HTML etiketlerinin işaretlere dönüştürülmesini sağlar.

Aşağıdaki örnekte POST yöntemi kullanılarak gönderilen bilgiler '*htmlspecialchars*' fonksiyonu kullanılarak HTML karakter dönüşümleri yapılmaktadır.

formu\_denetle.php dosyasının htmlspecialchars ile düzenlenmiş hali

```
<?php
$_POST['isim']=htmlspecialchars($_POST['isim'], ENT_QUOTES);
$_POST['yas']=htmlspecialchars($_POST['yas'], ENT_QUOTES);

echo 'İsminiz:'. $_POST['isim'].'<br />';
echo 'Yaşınız:'. $_POST['yas'].'<br />';
?>
```

formu\_denetle.php dosyası düzenlenmiş halinin tarayıcı görüntüsü:



İsminiz:<u>ali</u>  
Yaşınız:20

HTML karakter dönüşümü yapılmış olarak görüntülenen internet sayfasının kaynak kodu aşağıda olduğu gibidir. Görüldüğü gibi HTML için özel anlamı olan '<', '>' işaretleri '*htmlspecialchars*' fonksiyonu sayesinde sırasıyla '&lt;', '&gt;' özel karakterlerine dönüştürülmüştür.

formu\_denetle.php dosyası düzenlenmiş halinin tarayıcıdaki kaynak görüntüsü:

```
İsminiz:&lt;u&gt;ali&lt;/u&gt;<br />Yaşınız:20<br />
```

Aşağıdaki örnekte foreach döngüleri ile GET ve POST metotları ile gelen bütün değişkenlerin HTML etiketlerinin dönüştürülmesi sağlanır.

```
<?php
foreach($_POST as $anahtar_post=>$deger_post)
    $_POST[$anahtar_post]=htmlspecialchars($deger_post, ENT_QUOTES);
foreach($_GET as $anahtar_get=>$deger_get)
    $_GET[$anahtar_get]=htmlspecialchars($deger_get, ENT_QUOTES);
?>
```

## UYGULAMA FAALİYETİ

Aşağıda verilen işlem basamaklarını takip ederek konuyu daha da pekiştirelim.

İşlem Basamakları	Öneriler
<ul style="list-style-type: none"><li>➤ İki sayının ve dört işlemden birinin seçilebildiği bir form oluşturunuz.</li></ul>	<ul style="list-style-type: none"><li>➤ Sayılar için “<i>text</i>”, dört işlem için “<i>radio</i>” form kontrol yapılarını kullanabilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Oluşturduğumuz formu elde edecek bir PHP sayfası hazırlayınız.</li></ul>	<ul style="list-style-type: none"><li>➤ Form oluştururken seçtiğiniz yönetime göre \$_POST veya \$_GET yöntemini seçiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Oluşturduğunuz PHP sayfasına gelen tüm form bilgilerinin HTML dönüşümlerini yapınız.</li></ul>	<ul style="list-style-type: none"><li>➤ ‘<i>foreach</i>’ ve ‘<i>htmlspecialchars</i>’ yapılarını kullanabilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Oluşturduğunuz PHP sayfasına gelen sayıların tipini kontrol ediniz.</li></ul>	<ul style="list-style-type: none"><li>➤ ‘<i>is_numeric</i>’ fonksiyonunu kullanabilirsiniz.</li></ul>
<ul style="list-style-type: none"><li>➤ Oluşturduğunuz PHP sayfasında her işlem için bir kod bloğu oluşturunuz.</li></ul>	<ul style="list-style-type: none"><li>➤ Kod blokları seçilen işleme göre if...else..if yapısı ya da switch..case yapısı kullanılarak yapılabilir.</li><li>➤ Örneğin; <pre>if(\$_POST['islem']=='Topla') echo \$_POST['sayi1']+ \$_POST['sayi2'];</pre></li></ul>

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Form oluşturabildiniz mi?		
2. Text bileşenini ekleyebildiniz mi?		
3. Radio bileşeni ekleyebildiniz mi?		
4. Form oluştururken yöntemi belirleyebildiniz mi?		
5. Form oluştururken seçtiğiniz yönteme bağlı olarak metot belirleyebildiniz mi?		
6. Oluşturduğunuz PHP sayfasına gelen tüm form bilgilerinin HTML dönüşümlerini yapabildiniz mi?		
7. Oluşturduğunuz PHP sayfasına gelen sayıların tipini kontrol edebildiniz mi?		
8. Oluşturduğunuz PHP sayfasında her işlem için bir kod bloğu oluşturabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme’ye geçiniz.



## ÖLÇME VE DEĞERLENDİRME

Bu faaliyet sonunda kazandıklarınızı aşağıdaki soruları cevaplandırarak ölçünüz.

**Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.**

1. ( ) POST metodunda formdaki veriler adres çubuğu üzerinden taşınır.
2. ( ) \$\_POST ve \$\_GET değişkenleri PHP'nin küresel dizi değişkenleridir.
3. ( ) Form kullanılarak resim yüklemek mümkün değildir.

**Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz**

4. Hangisi <form> etiketi oluşturulurken kullanılan bir özellik değildir?  
A) action  
B) method  
C) name  
D) type
5. Hangisi form kontrol elemanı değildir?  
A) text  
B) checkbox  
C) combobox  
D) select
6. Form kontrol yapılarının hangi özelliği PHP sayfasında taşıdıkları bilgiyi almayı sağlar?  
A) size  
B) name  
C) type  
D) hiçbirisi
7. Formlardan gelen bilgi güvenliği için aşağıdakilerden hangisi kullanılmaz?  
A) copy  
B) is\_numeric  
C) is\_array  
D) htmlspecialchars

**Aşağıdaki cümleleri dikkatlice okuyarak boş bırakılan yerlere doğru sözcüğü yazınız.**

8. .... web sayfalarına yeni içerikler eklenebilir, var olanlar güncellenebilir ya da silinebilir.
9. Durağan (statik) web sayfalarında veri akışı ..... yönlüdür.

### **DEĞERLENDİRME**

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-2

## AMAÇ

Açık kaynak kod kullanarak düzenli ifadeler konusunu öğreneceksiniz. Özek karakter gruplarını kullanarak söz dizileri (desen, örüntü, İngilizce karşılığı patterns) oluşturacak ve bu söz dizilerini PHP fonksiyonları ile birlikte kullanacaksınız

## ARAŞTIRMA

- Düzenli ifadeler nedir?
- Düzenli ifadeler hangi tür programlarda kullanılıyor?
- Düzenli ifade standartları nelerdir? Özellikle PHP'nin kullandığı standartlar nelerdir?

## 2. DÜZENLİ İFADELER

Bilgisayar dünyasının Türkçe adıyla **düzenli ifadeler**, İngilizcesi ile **regular expressions**, konusu bütün programlama dillerinde karşımıza çıkmaktadır. Her programlama dilinin düzenli ifadeler için kullandığı farklı fonksiyonları bulunmaktadır. Bununla birlikte düzenli ifadelerde kullanılan söz dizileri (desen, örüntü, İngilizce karşılığı patterns) ve bu söz dizilerini oluşturan işaret ve kurallar çok benzerdir.

### 2.1. Düzenli İfade Nedir?

Belirlenen kurallara uyan söz dizilerine düzenli ifade denir. En yaygın kullanıldığı şekliyle örnek vererek konuyu açalım. Bir form oluşturduğumuzu ve bu formda e-posta alanı olduğunu varsayalım. Form gönderildiğinde giriş yapılan e-postanın gerçek olup olmadığını düzenli ifadeleri kullanarak basit bir söz dizisi ile öğrenebiliriz.

Ya da bir metin içerisinde “PHP Matematik Fonksiyonları” ifadesini aradığımızı varsayalım. Bu ifadenin bulunması için PHP’de birçok fonksiyon (strstr, strpos vb.) kullanabiliriz. Bununla birlikte benzer ifadelerin tümünü görmek istiyorsak ne yapmalıyız?

Örneğin “PHP Metin Fonksiyonları”, “PHP Tarih ve Saat Fonksiyonları” ifadelerini ve bunlara benzeyen fakat aklımıza gelmeyen ifadelerin tamamını arama sonucunda görmek istiyorsak düzenli ifadeler kullanabiliriz. Kısacık bir söz dizisiyle istediğimizi elde etmeyi sağlar.

### 2.1.1. Kullanım Amaçları

- Nasıl yazıldığını tam olarak bilmediğimiz bir sözcüğü aratabiliriz.

Örneğin, bir makalede "entrepreneurship" sözcüğünü bulmak istiyorsak, ancak ne şekilde yazıldığından emin değilsek, ilk ve son birkaç karakterini verip aratabiliriz.

- Bir metnin içindeki, özel bir forma uyan parçaları bulabiliriz.

Örneğin, düz bir metin içindeki tüm URL'leri (İnternet adreslerini), yalnızca ana yapıyı tarif ederek, teker teker ayıklayabiliriz. (http:// ile başlayan ve arada noktalar içeren sözcükler)

- Aldığımız bir metnin, belli bir yapıya uyup uymadığını kontrol edebiliriz.

Örneğin, kullanıcılardan e-posta adresi girmelerini istediğimiz bir web sayfasında, metin kutusuna girilen yazının geçerli bir e-posta adresi olabilecek formatta olup olmadığına bakabiliriz. (@ işareti içerecek, belli karakterlerden oluşacak, ikinci blokta en az bir nokta olacak vs)

- Belli formatta aldığımız bir veriyi, istediğimiz başka bir formata çevirebiliriz.

Örneğin, bir XML belgesinden kopyaladığımız veriyi, kendi uygulamamızda kullanmamız için bir PHP dizisi (array) haline getirebiliriz.

### 2.2. Söz Dizisi

En basit düzenli ifadeler, tek bir karakterden oluşur. Harfler, rakamlar, noktalama işaretleri gibi pek çok karakter, kendisiyle eşleşen düzenli ifadelerdir. Yani, bir metinde a karakterini arıyorsak, girmemiz gereken düzenli ifade, doğal olarak a karakteridir. Ancak, bazı karakterler, düzenli ifadelerin söz dizisinde özel anlamlar ifade eder. Bu özel karakterler, \ . \* [ ] ^ \$ sembolleridir. Aradığımız karakter, bu özel karakterlerden bir tanesiye, başına \ karakteri koyarak, özel anlam içermesini engelleyebiliriz. Örneğin, bir metin içinde nokta karakterlerini bulmak istiyorsak, girmemiz gereken düzenli ifade nokta ( . ) değil ( \ . ) olmalıdır.

Düzenli ifadelerin, bazı özel sembollerle bir araya gelmesi sonucu, daha güçlü ve karmaşık düzenli ifadeler elde edilir. Aşağıdaki tablolar, düzenli ifadelerin nasıl elde edildiğini açıklar:

## 2.2.1. Söz Dizisinde Kullanılan Özel Karakterler

Özel Karakterler	Açıklama
^	Metin verilen ifade ile başlamalıdır.
\$	Metin verilen ifade ile sonlanmalıdır.
?	Metinde bir önceki karakter olabilir de olmayabilir de.
+	Metinde bir önceki karakter 1 veya daha fazla olmalıdır.
*	Metinde bir önceki karakter olmayabilir ya da 1 veya daha fazla sayıda olabilir.
()	Düzenli ifade içinde grup oluşturmak için kullanılır. ? + ve * karakterlerinin gelmesi sayesinde oluşturulan gurubun var olması ya da tekrar etmesi önem kazanır.
Özel Karakterler	Açıklama
{ }	Metin içinde bir karakter ya da gurubun tekrar sayısını belirlemek için kullanılır. Kullanım şekli {2} 2 defa {2,} iki ya da daha fazla {2,5} iki, üç, dört veya beş defa
[ ]	Bir aralığı ifade etmek için kullanılır. [a-z] a ile z harfleri arasındaki bütün harfler [A-Z] A ile Z harfleri arasındaki bütün harfler [0-9] 0 ile 9 arasındaki rakamlar [a-z A-Z 0-9] bütün küçük, büyük harfler ve rakamlar
.	Herhangi bir karakteri belirtmek için kullanılır.
\	Özel karakterleri söz dizileri içinde kullanmak için kullanılır.
	veya

Aşağıdaki tabloları bir anda anlamak mümkün olmayacaktır. Bu aşamada düzenli ifadeler için PHP’de kullanılan `preg_match` fonksiyonunun kullanımına bakıp aşağıdaki tabloda verilen örnekleri deneyerek ilerlemek işlerimizi kolaylaştıracaktır. Bu şekilde adım adım düzenli ifadeler konusunu anlamak mümkün olacaktır.

Dikkat edilmesi gereken önemli bir nokta, söz dizileri oluşturulurken iki aynı simge arasına yazılmalıdırlar. Örneğin; /Bafra/ yada -Bafra- yada @Bafra@ gibi.

Söz Dizisi	Eşleşme Sağlanır	Eşleşme Sağlanmaz
/Bafra/	<ul style="list-style-type: none"> <li>➤ Bafra'da hava çok güzel.</li> <li>➤ Bugün Bafra çok güzel.</li> </ul>	<ul style="list-style-type: none"> <li>➤ B afra</li> <li>➤ Bafira Bafay</li> </ul>
/^Samsun/	<ul style="list-style-type: none"> <li>➤ Samsun Spor Kulübü</li> <li>➤ Samsun'un geçmişi.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Karadeniz Samsun Sahilli</li> <li>➤ Atatürk'ün Şehri Samsun</li> </ul>
/^Samsun/	<ul style="list-style-type: none"> <li>➤ Atatürk'ün Şehri Samsun</li> <li>➤ Karadeniz şehri Samsun</li> </ul>	<ul style="list-style-type: none"> <li>➤ Samsun Spor Kulübü</li> <li>➤ Samsunun geçmişi.</li> </ul>
/inter?net/	<ul style="list-style-type: none"> <li>➤ internet</li> <li>➤ intenet</li> </ul>	<ul style="list-style-type: none"> <li>➤ interinet</li> <li>➤ intnet</li> <li>➤ internnet</li> </ul>
/inter+net/	<ul style="list-style-type: none"> <li>➤ internet</li> <li>➤ internnet</li> <li>➤ interrrnet</li> </ul>	<ul style="list-style-type: none"> <li>➤ inetnet</li> <li>➤ intelnet</li> <li>➤ intrnet</li> </ul>
/inter*net/	<ul style="list-style-type: none"> <li>➤ inetnet</li> <li>➤ internet</li> <li>➤ internnet</li> </ul>	<ul style="list-style-type: none"> <li>➤ intnet</li> <li>➤ intirnet</li> <li>➤ inteenet</li> </ul>
/int(er)+net/	<ul style="list-style-type: none"> <li>➤ internet</li> <li>➤ interernet</li> </ul>	<ul style="list-style-type: none"> <li>➤ intnet</li> <li>➤ intenet</li> </ul>

Söz Dizisi	Eşleşme Sağlanır	Eşleşme Sağlanmaz
/int(er)*net/	<ul style="list-style-type: none"> <li>➤ intnet</li> <li>➤ internet</li> <li>➤ interernet</li> </ul>	<ul style="list-style-type: none"> <li>➤ intenet</li> <li>➤ intrnet</li> <li>➤ intelnet</li> </ul>
/int(er){2}net/	<ul style="list-style-type: none"> <li>➤ interernet</li> </ul>	<ul style="list-style-type: none"> <li>➤ intnet</li> <li>➤ internet</li> <li>➤ intererernet</li> </ul>
/int(er){2,}net/	<ul style="list-style-type: none"> <li>➤ interernet</li> <li>➤ intererernet</li> <li>➤ interererernet</li> </ul>	<ul style="list-style-type: none"> <li>➤ intnet</li> <li>➤ internet</li> </ul>
/int(er){2,3}net/	<ul style="list-style-type: none"> <li>➤ interernet</li> <li>➤ intererernet</li> </ul>	<ul style="list-style-type: none"> <li>➤ internet</li> <li>➤ interererernet</li> </ul>
/[a-z]/ <u>İçinde bir tane küçük harf bulunanlar</u>	<ul style="list-style-type: none"> <li>➤ i</li> <li>➤ internet</li> <li>➤ İNTERNET</li> </ul>	<ul style="list-style-type: none"> <li>➤ İ</li> <li>➤ İNTERNET</li> <li>➤ 123 İNTERNET.</li> </ul>
/[abc]/ <u>İçinde a.b.c harflerinden en az bir tane harf bulunanlar</u>	<ul style="list-style-type: none"> <li>➤ a</li> <li>➤ bir</li> <li>➤ cenk</li> </ul>	<ul style="list-style-type: none"> <li>➤ d</li> <li>➤ internet</li> <li>➤ A</li> <li>➤</li> </ul>
/[0-9]/ <u>İçinde en az bir tane rakam bulunanlar</u>	<ul style="list-style-type: none"> <li>➤ 12345</li> <li>➤ 1 sayı</li> </ul>	<ul style="list-style-type: none"> <li>➤ Bir sayı</li> </ul>
/[a-zA-Z0-9]/ <u>İçinde en az bir tane harf yada rakam bulunanlar</u>	<ul style="list-style-type: none"> <li>➤ internet</li> <li>➤ 12345</li> <li>➤ İNTERNET</li> <li>➤ ---a</li> </ul>	<ul style="list-style-type: none"> <li>➤ &lt;,.; :&gt;</li> <li>➤ ...</li> <li>➤ ---</li> </ul>
/^[a-z]/ Metin küçük harfle başlamalı.	<ul style="list-style-type: none"> <li>➤ internet</li> </ul>	<ul style="list-style-type: none"> <li>➤ İnternet</li> <li>➤ 01internet</li> </ul>
/[a-z]\$/ Metin küçük harfle bitmeli.	<ul style="list-style-type: none"> <li>➤ internet</li> </ul>	<ul style="list-style-type: none"> <li>➤ interneT</li> <li>➤ internet1</li> </ul>
/^[a-z]+\$/ Metin küçük harfle başlamalı, küçük harfle devam etmeli ve küçük harfle bitmeli. Yani metin küçük harflerden oluşmalı	<ul style="list-style-type: none"> <li>➤ internet</li> <li>➤ internet sayfası</li> </ul>	<ul style="list-style-type: none"> <li>➤ İnternet</li> <li>➤ internet sayfası</li> <li>➤ 12internet</li> </ul>

Söz Dizisi	Eşleşme Sağlanır	Eşleşme Sağlanmaz
/^[a-zA-z0-9]+\$/ Metin harf ya da sayılardan oluşmalı	<ul style="list-style-type: none"> <li>➤ İnrenet123</li> <li>➤ İNTERNET</li> </ul>	<ul style="list-style-type: none"> <li>➤ İnternet 123</li> </ul>
/^[^0-9]\$/ <i>^ işareti [] köşeli parantez içinde istenmeyen karakterler için kullanılır.</i>	<ul style="list-style-type: none"> <li>➤ Hava sıcaklığı 5 derece</li> </ul>	<ul style="list-style-type: none"> <li>➤ Hava sıcaklığı 5</li> </ul>
/^[^0-9]/	<ul style="list-style-type: none"> <li>• Beş derece</li> </ul>	<ul style="list-style-type: none"> <li>• 5 derece</li> </ul>
/^[0-9][a-z]/	<ul style="list-style-type: none"> <li>• 2x</li> </ul>	<ul style="list-style-type: none"> <li>• y2</li> </ul>
/inte.net/	<ul style="list-style-type: none"> <li>• internet</li> <li>• intelnet</li> </ul>	<ul style="list-style-type: none"> <li>• intenet</li> </ul>
/internet\./	<ul style="list-style-type: none"> <li>• internet.</li> </ul>	<ul style="list-style-type: none"> <li>• İnternet</li> </ul>
/^(Bay Bayan)/ <i>Bay veya Bayan ile başlayan metinler için kullanılır.</i>	<ul style="list-style-type: none"> <li>• Bay Ahmet.</li> <li>• Bayan Ayşe.</li> </ul>	<ul style="list-style-type: none"> <li>• Merhaba Bayan.</li> <li>• Merhaba Ahmet Bey.</li> </ul>



### 2.2.2. Söz Dizilerinde Görünmez (non-printing) Karakterleri Yakalamak

Bunu da şöyle izah edebiliriz: Örneğin bir metin yazarken satır sonuna geldiğinizde enter tuşuna bastınız ve alt satıra geçtiniz. Oysa görünürde o noktaya basılmış herhangi bir karakter yoktur. Aslında tam tersine sizin metin editörünüz o noktaya sizin göremeyeceğiniz bir karakteri, yani yeni satır (newline) karakterini basmıştır. İşte aşağıda listesini verdiğimiz buna benzer görünmez karakterleri yakalamak için yine \ ters bölü anahtarını kullanmalısınız.

Görünmez (non-printing) karakterler	Açıklaması
\n	Herhangi bir yeni satır karakteri
\r	Herhangi bir satır başı (carriage return) karakteri
\t	Herhangi bir girinti (tab) karakteri
\f	Herhangi bir kağıt çıktısı (formfeed) karakteri
\cx	Herhangi bir CTRL-X karakteri (CTRL+C ya da CTRL+V gibi)
\xFF	Herhangi bir on altılık (hexadecimal) kod. (© demek için \xA9 gibi)

### 2.2.3. Söz Dizilerinde Geniş Aralıkları İçeren Karakter Setleri

Aşağıdaki karakter setleri rakam, harf ya da beyaz boşluk(enter, tab vb.) gibi geniş karakter gruplarını tanımlamak için kullanılır.

Karakter setleri	Açıklama
\d	Herhangi bir ondalık sayı karakterlerini temsil eder.(0,1,2,3 vb. gibi)
\D	Ondalık sayı karakteri dışında herhangi bir karakteri temsil eder.
\s	Herhangi bir beyaz boşluk (whitespace) karakterini temsil eder. (\r,\n,\t,\f vb. gibi)
\S	Beyaz boşluk dışında herhangi bir karakteri temsil eder.
\w	Herhangi bir yazı karakterini temsil eder. (a'dan z'ye ve 0'dan 9'a tüm harf, rakam ve karakteri)
\W	Yazı karakteri dışında herhangi bir karakteri temsil eder.

Söz Dizisi	Eşleşme Sağlanır	Eşleşme Sağlanmaz
<code>/w/</code>	<ul style="list-style-type: none"> <li>➤ Bafra'da hava çok güzel.</li> <li>➤ Hava sıcaklığı 5 derece</li> </ul>	<ul style="list-style-type: none"> <li>➤ Yazı karakteri dışındaki herhangi bir karakteri içeren metinler</li> </ul>
<code>/^w{2,3}\$/</code>	<ul style="list-style-type: none"> <li>➤ Ay</li> <li>➤ Ali</li> <li>➤ 12</li> <li>➤ A1</li> </ul>	<ul style="list-style-type: none"> <li>➤ A</li> <li>➤ Ayşe</li> <li>➤ 1234</li> </ul>
<code>/^d{2}\$/</code> <i>Sadece iki rakamdan oluşan metinler.</i>	<ul style="list-style-type: none"> <li>➤ 12</li> <li>➤ 45</li> <li>➤ 91</li> </ul>	<ul style="list-style-type: none"> <li>➤ 123</li> <li>➤ 1a</li> <li>➤ A12</li> </ul>
<code>/^D{2,3}\$/</code> <i>Rakam olmayan 2 ya da 3 karakter.</i>	<ul style="list-style-type: none"> <li>➤ ay</li> <li>➤ ay1</li> </ul>	<ul style="list-style-type: none"> <li>➤ a1</li> <li>➤ ay2</li> <li>➤ net2</li> </ul>

#### 2.2.4. Söz Dizilerinde Sınırları Yakalamak

Düzenli ifadelerde kullanılan özel karakterlerden bazıları bir ifade ya da kelimenin sınırlarını belirlemek için olanıdır. Ama bu kullanımın diğerlerinden belirgin bir farkı vardır. O da aşağıda listesi verilen karakterlerin aslında fiziksel bir karşılıklarının olmayışıdır.

Sınır karakterleri	Açıklama
<code>\b</code>	Kelime sınırı (karakter)
<code>\B</code>	Kelime sınırı hariç herhangi bir karakter
<code>\A</code>	İfadenin başı (karakter)
<code>\Z</code>	İfadenin yada satırın sonu (karakter)
<code>\z</code>	İfadenin sonu (karakter)

Söz Dizisi	Eşleşme Sağlanır	Eşleşme Sağlanmaz
<code>/\b{3}\b/</code> <i>İçinde üç harfli kelimeleri arar.</i>	<ul style="list-style-type: none"> <li>➤ Bafra'da hava çok güzel.</li> <li>➤ Hava sıcaklığı beş derece</li> <li>➤ 123 sayma sayıları</li> </ul>	<ul style="list-style-type: none"> <li>➤ Merhaba arkadaşlar.</li> <li>➤ 1234</li> <li>➤ Abcd ab12</li> </ul>
<code>/\b{3}\b/</code> <i>İçinde üç harfli kelimeleri arar.</i>	<ul style="list-style-type: none"> <li>➤ 123 sayma sayıları</li> </ul>	<ul style="list-style-type: none"> <li>➤ Bafra'da hava çok güzel.</li> <li>➤ Hava sıcaklığı beş derece</li> </ul>
<code>/\bPHP\b/</code>	<ul style="list-style-type: none"> <li>➤ PHP öğreniyoruz.</li> <li>➤ Günümüzde PHP'nin kullanımı artmaktadır.</li> <li>➤ Merhaba PHP.</li> </ul>	<ul style="list-style-type: none"> <li>➤ PHP nin</li> </ul>

## 2.2.5. Yer Değiştirme

Bazı durumlarda arama işlemine ek olarak, bulunan söz dizisi örneğinin yer değiştirmesi de gerekebilir. Bu işlem için düzenli ifadelerin temel tanımına ait olmayıp, gelişmiş bir özellik olan bellek kullanımı gereklidir.

Bu özellik kullanıldığında ifade gücü Tip – 3 dillerinin üzerine çıkar. Arama örüntüsü içinde ‘(‘ ile başlayıp, ‘)’ ile biten parçalara yer değiştirme işleminde, \1 gibi numaralı bir yazmaç (register) ile referansta bulunulur.(13)

Söz Dizisi	Yer Değiştirme	Sonuç
/(PHP)(öğrenmek)(çok kolay)/	\3\1\2	çok kolay PHP öğrenmek
/(PHP)(öğrenmek)(çok kolay)/	\1\3	PHP çok kolay
/(<h1>)(Başlık)(</h2)/	\2	Başlık
/(PHP)(öğrenmek)(çok kolay)/	Merhaba \1	Merhaba PHP

## 2.3. PHP’de Düzenli İfadeler

PHP 5.3.0 sürümü ile birlikte düzenli ifadeler yaklaşımında değişikliğe gitmiştir. Bu sürümden önce kullandığı POSIX standardını bırakarak PCRE standardına geçmiştir.

### 2.3.1. PHP’de Düzenli İfadeler İçin Kullanılan Fonksiyonlar

PHP’de kullanılan ve bizim bu modülde üzerinde duracağımız PCRE fonksiyonları **preg\_match** ve **preg\_replace**’tir. Bütün PCRE fonksiyonları için <http://www.php.net/manual/tr/ref.pcre.php> adresini ziyaret edebilirsiniz.

#### ➤ **preg\_match ( string \$soz\_dizisi , string \$metin)**

\$soz\_dizisi ile belirtilen düzenli ifadeyi eşleştirmek için \$metin içinde arama yapar.Eşleşme var ise TRUE, yok ise FALSE değeri döner. Fonksiyona ait üç tane daha değer (parametre) vardır. Bunların kullanılması zorunlu olmadığı için burada yer vermedik.

Şimdi düzenli ifadeler konumuzdan uzaklaşmadan adım adım düzenli ifadeleri kullanarak nasıl söz dizileri oluşturacağımıza bir bakalım.

Basit bir örnekle başlayalım.

Örneğin “PHP ile Düzenli İfadeler” metni içinde “PHP” ifadesini arayalım.

preg\_match\_01.php dosyası kaynak kodu:

```
<?php
if (preg_match("/PHP/", "PHP ile Düzenli İfadeler")) {
    echo "Eşleşme sağlandı.";
} else {
    echo "Eşleşme sağlanamadı.";
}

//Ekrana "Eşleşme sağlandı." mesajı çıkar.
?>
```

Görüldüğü gibi istenilen söz dizisi metin içerisinde olduğundan *preg\_match* fonksiyonu TRUE değerini döndürür ve ekrana “Eşleşme sağlandı.” mesajı çıkar.

Söz dizisini oluştururken PHP ifadesini “/” karakteri ile başlatıp sonlandırdık. Fonksiyonumuzun doğru çalışabilmesi için bu gerekliydi.

Yukarıdaki örnekte metin içerisinde PHP ifadesi geçtiği sürece sonuç hep aynı olacak ve ekrana “Eşleşme sağlandı.” mesajı çıkacaktır. Örneğin “PHP ile Düzenli İfadeler”, “Düzenli İfadeler ve PHP”, “PHP öğreniyorum!”, “Bugünlerde PHP uygulamaları geliştiriyorum.”

Yukarıdaki örnekte aramanın büyük ve küçük harfe duyarlı olması için söz dizisini şu şekilde düzenlememiz yeterli olacaktır. **/php/i**

Şimdi bazı özel karakterler kullanarak arama sonuçlarımızı daraltalım.

**Örnek:**

Söz Dizisi	Açıklama
<code>^b(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})\b/</code>	Metin içinde IP adresi arar
<code>^b[_a-zA-Z0-9-]+\.(?[_a-zA-Z0-9-]+)*@[a-zA-Z0-9-]+\.(?[_a-zA-Z0-9-]+)\b/</code>	Metin içinde e-posta adresi
<code>/^[_a-zA-Z0-9-]+\.(?[_a-zA-Z0-9-]+)*@[a-zA-Z0-9-]+\.(?[_a-zA-Z0-9-]+)\$/</code>	Verilen metnin e-posta olup olmadığına bakar
<code>@^((https?)(ftp))://([a-zA-Z0-9]([a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?.)+[a-zA-Z]{2,6}\$@</code>	URL adresi
<code>^[+90-]\d{3}[-]\d{3}[-]\d{2}[-]\d{2}\$/</code>	Telefon numarası +90-312-111-90-00

## Örnek:

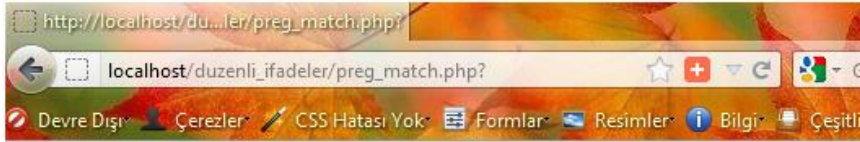
Düzenli ifadeler için pratik bir örnek yapalım. Örneğimizde bir form oluşturup, form ile birlikte hem söz dizisini hem de metni preg match fonksiyonuna gönderelim. Böylece bütün söz dizilerini tek bir form üzerinden kontrol etmemiz mümkün olacaktır.

preg\_match.php dosyası kaynak kodu:

```
<?php error_reporting(0);?>
<form action="?" method="post">
<pre>Söz Dizisi :<input type="text" name="ifade" value="<?php echo $_POST['ifade'];?>"
size="40"/></pre>
<pre>Metin :<textarea cols="30" rows="5" name="aranan"><?php echo $_POST['aranan'];
?></textarea></pre>
<input type="reset" value="Temizle" /><input type="submit" value="Gönder" />
</form>
<?php
if(!empty($_POST['aranan']) and !empty($_POST['ifade'])) {
    echo "@preg_match('$_POST[ifade]', '$_POST[aranan]')." <br>";

    if(@preg_match($_POST['ifade'], $_POST['aranan']))
        echo "Eşleşme Sağlandı.";
    else
        echo "Eşleşme Sağlanmadı.";
}
?>
```

preg\_match.php dosyası tarayıcı görünümü:



Söz Dizisi :

Metin :

@preg\_match("/^[+90[-]\d{3}[-]\d{3}[-]\d{2}[-]\d{2}\$/", '+90-362-543-00-00')  
Eşleşme Sağlandı.

echo fonksiyonu içerisine yazılmış olan \$\_POST değişkenlerinin indexlerinin tek tırnak ya da çift tırnak kullanılmadan yazıldığına dikkat edelim. Aynı zamanda echo fonksiyonunun doğru çalışabilmesi için \$\_POST değişkeni içinde tek tırnak karakteri olmamalı ya da \` şeklinde kullanılmalı.

**error\_reporting(0);** kod satırının PHP’de hata mesajlarını tarayıcıya göndermediğini hatırlayalım.

➤ **preg\_replace ( string \$soz\_dizisi , string \$yeni\_ifade, string \$metin)**

\$soz\_dizisi ile belirtilen düzenli ifadeyi eşleştirmek için \$metin içinde arama yapar. Yapılan arama sonucunda eşleşenlerin hepsi yeni ifade ile değiştirilir.

**Örnek:**

Bir ifade içinde “*Bay*” olarak geçen kelimeleri “**Bayan**” olarak değiştiren örneği yapınız.

preg\_replace01.php dosyası kaynak kodu ve tarayıcı görüntüsü.

```
<?php
$sonuc=preg_replace("/Bay/", "Bayan", "Merhaba Baylar.");
echo $sonuc;
?>
```



## Örnek:

Verilen ifade içinde http:// söz dizisini http\_ ile değiştiren örnek.

preg\_replace02.php dosyası kaynak kodu ve tarayıcı görüntüsü.

```
<?php
$soz_dizisi = '-http://-';
$yenisi = 'http_';
$metin = 'Okulumuz web site adresi http://www.bafraeml.12.tr';
echo $metin;
echo "<br/>Değişimden sonrası.<br/>";
echo preg_replace($soz_dizisi, $yenisi, $metin);
?>
```



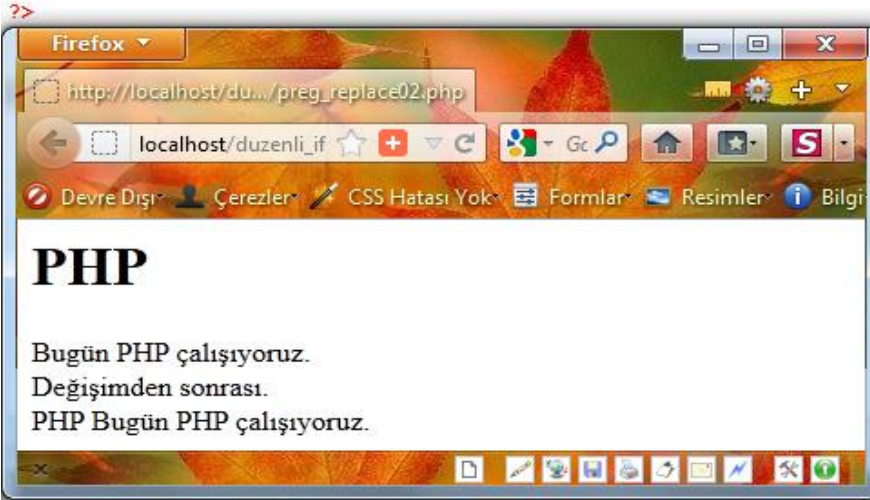


## Örnek:

'preg\_replace' fonksiyonu kullanarak bir metinde olan <h1></h1> etiketini yazıdan çıkarmak.

preg\_replace02.php dosyası kaynak kodu ve tarayıcı görüntüsü.

```
<?php
$soz_dizisi = '-(<h1> (.*) (</h1>)-i';
$yenisi = '\2';
$metin = '<h1>PHP</h1> Bugün PHP çalışıyoruz.';
echo $metin;
echo "<br/>Değişimden sonrası.<br/>";
echo preg_replace($soz_dizisi, $yenisi, $metin);
```



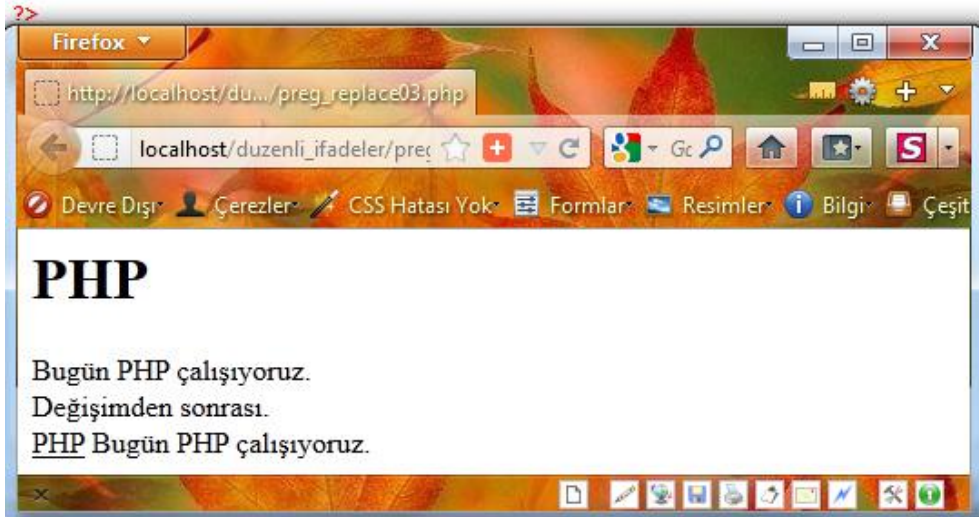


Örnek:

'**preg\_replace**' fonksiyonunu kullanarak bir metinde olan `<h1></h1>` etiketini yerine `<u></u>` etiketi eklemek

preg\_replace03.php dosyası kaynak kodu ve tarayıcı görüntüsü.

```
<?php
$soz_dizisi = '@(<h1> (.*) </h1>@i';
$yenisi = '<u>\2</u>';
$metin = '<h1>PHP</h1> Bugün PHP çalışıyoruz.';
echo $metin;
echo "<br/>Değişimden sonrası.<br/>";
echo preg_replace($soz_dizisi, $yenisi, $metin);
```



## UYGULAMA FAALİYETİ

Düzenli ifadeleri kullanarak aşağıdaki uygulamayı yapınız.

İşlem Basamakları	Öneriler
➤ Kullanıcıların yorum yapabildiği bir form oluşturunuz.	➤ Ad, Soyadı, e-posta, Yorum ve Gönder alanlarından oluşabilir.
➤ Gönderilen formdan e-postanın gerçek olup olmadığını bulan ve ekrana yazan kod satırlarını yazınız.	➤ “ <i>preg_match</i> ” fonksiyonunu kullanınız.
➤ Gönderilen formda <i>yorum</i> içinde varsa URL adresini kaldıran PHP sayfasını oluşturunuz.	➤ “ <i>preg_replace</i> ” fonksiyonunu kullanınız.

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Kullanıcıların yorum yapabildiği bir form oluşturabildiniz mi?		
2. Gönderilen formdan e-postanın gerçek olup olmadığını bulan ve ekrana yazan kod satırlarını yazabildiniz mi?		
3. Gönderilen formda <i>yorum</i> içinde varsa URL adresini kaldıran PHP sayfasını oluşturabildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme” ye geçiniz.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) `/[^0-9a-zA-Z]$/` söz dizesinin anlamı: Metin harf yada sayılardan oluşmalı
2. ( ) `/^\d{2}:\d{2}$/` söz dizesinin anlamı: Metin saat formatında olmalı.(Örn;12:15)
3. ( ) Metin içinde aranılan bir ifadeyi yenisi ile değiştirmek için `php_replace` fonksiyonu kullanılır.

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz

4. Hangisi düzenli ifade özel karakteri değildir?  
A) .  
B) ;  
C) ?  
D) \*
5. Hangisi metnin verilen ifade ile sonlandığını gösterir?  
A) \$  
B) ^  
C) +  
D) \
6. Hangisi metin içinde bir karakter yada grubun tekrar sayısını belirlemek için kullanılır?  
A) ()  
B) []  
C) {}  
D) hiçbirisi
7. Hangisi metin içinde grup oluşturmak için kullanılır?  
A) ()  
B) []  
C) {}  
D) hiçbirisi
8. Hangisi bir aralığı ifade etmek için kullanılır?  
A) ()  
B) []  
C) {}  
D) hiçbirisi

**Aşağıdaki cümleleri dikkatlice okuyarak boş bırakılan yerlere doğru sözcüğü yazınız.**

- 9.** Metinde bir önceki karakter olmayabilir ya da 1 veya daha fazla sayıda olabilir anlamına gelen sembol ..... işaretidir.
- 10.** Herhangi bir karakteri temsil eden sembol ..... işaretidir.

### **DEĞERLENDİRME**

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-3

## AMAÇ

Açık kaynak kod kullanarak oturum yönetimini etkin bir şekilde kullanabileceksiniz. Oturum esnasında kullanılan çerezler ve session kavramlarını öğrenecek ve bunların PHP dilindeki kullanımlarını uygulayabileceksiniz.

## ARAŞTIRMA

- Oturum Yönetimi nedir?
- Web geçmişini sildiğinizde silinen dosya türlerini öğreniniz.
- Çerezlerin faydalarını araştırınız..

## 3. OTURUM YÖNETİMİ

Oturum, bir kullanıcının bir web sitesini ziyaret etme sürecidir, diyebiliriz. Başka bir tanım ile, bilgisayarınızın başına geçip tarayıcınızı çalıştırdığımız, web' de sörf yaptığımız ve daha sonra tarayıcınızı kapattığımız zaman arasında kalan süreç bir oturumdur.

Oturum Yönetimi kavramının gelişmesi internet sitelerinin ziyaretçilerine daha iyi, hızlı ve özgün hizmet verebilmeleri ile ilgilidir. Amaç elde edilmiş verilerin kaybedilmeden kullanıcı yararına kullanılması. ve kullanıcıların istek ve yönelimlerini tespit ederek siteyi en etkin bir şekilde kullanmalarını sağlamaktır.

### 3.1. Çerezler

İnternet sayfalarını gezerken sıkça karşılaştığımız kavramlardan biri olan Türkçe ismiyle **çerezler** İngilizcesi ile **cookies**, kullanıcıyı ayırt etmenin en etkili yollarından biridir. Belki de bütün internet siteleri ziyaretçileri hakkında bir takım bilgiler toplarlar. Kimi zaman bu bilgileri veri tabanına kimi zaman ise kullanıcıların kendi bilgisayarlarına kaydederler.

Çerezler, internet sitelerinin kullanıcı bilgisayarına yerleştirdiği küçük boyutta dosyalardır. Tarayıcının bir sayfayı açtığı her tekrarda internet siteleri çerezlerini tekrar gönderir.

#### 3.1.1. Çerezlerin Kullanım Yerleri

Çerezler, web sayfası ile bilgisayarımız arasındaki referansımızdır. Daha önce girdiğimiz bir web sayfasına girişlerimiz çerezler sayesinde kaydedilebilir. Daha sonra aynı sayfaya girmemiz halinde girdiğimiz sayfa bizi tanır, bize kimi referans bilgileri verir, sayfayı daha sonra ziyaret ettiğimizde bu referans bilgileri web sayfasına otomatik olarak iletilir.

Çerezler çeşitli amaçlarla kullanılabilirler. Örnek verecek olursak bunlardan biri anket formlarıdır. Anket formlarında oy kullandığınızda web sitesi bu durumu bir çerezle bilgisayarınıza kaydedebilir. Siteyi tekrar ziyaret ettiğinizde anket sayfası bilgisayarınızdaki çerezleri kullanarak anket kullandığınızı tespit eder ve anketi tekrar kullanmanızı önerir.

Bir başka örnek de üyelik sistemleridir. Üyelik sistemlerinde de çerezler anket sistemlerinde olduğu gibi kullanılır. Sayfaya girişin bir kullanıcı adı ve şifre ile mümkün olduğunu varsayalım. Sayfaya üye olduktan sonra kullanıcı adı ve şifre ile giriş yapılır. Kullanıcı adı ve şifre çerez ile kaydedilir. Daha sonra bu internet sitesi aynı bilgisayardan tekrar ziyaret edildiğinde çerez bilgileri okunur. Kullanıcı adı ve şifre çerezlerden alınarak kullanıcı oturum sayfası açılır. Aslında çerezler çok daha fazla işlemde kullanılır.

### 3.1.2. PHP ile Çerezler

PHP HTTP çerezlerini destekler. Çerezler, uzak tarayıcıda veri saklamayı ve böylece geri dönen kullanıcılar hakkında veri takibi yapmayı sağlar. Çerezleri, `setcookie()` işlevleri ile atayabilirsiniz. Çerezler, HTTP başlığının birer parçasıdır, bu yüzden tarayıcıya bir veri göndermeden önce `setcookie()` işlevini çağırmanız gerekir.

#### ➤ Setcookie fonksiyonu

**Kullanımı :** `setcookie` (string \$isim [, string \$değer [, int \$zaman\_aşımı = 0 [, string \$yol [, string \$alan [, bool \$güvenlik = false [, bool \$sadece\_http = false ]]]]] )

- **\$isim:** Çerezin ismi.
- **\$değer:** Çerezin değeri. Bu değer istemcinin bilgisayarında saklanır. Bu bakımdan çerezlerde güvenlik ihlaline konu bilgiler saklamayın. İsim değiştirilmesinde 'çerezim' ismini kullandığınız varsayımıyla, bu çerezin değerini `$_COOKIE['çerezim']` değişkeni ile alabilirsiniz.
- **\$zaman\_aşımı:** Çerezin zaman aşımına uğrama süresi. Değer bir Unix zaman damgasıdır. Başka bir deyişle `time()` işlevinden dönen değere çerezin zaman aşımına uğrayacağı süreyi saniye cinsinden eklemek suretiyle bulacağımız değerdir. Ayrıca, `mktime()` işlevini de kullanabilirsiniz. `time()+60*60*24*30` ile çerez 30 gün sonra zaman aşımına uğrar. Burada 0 atanır veya bir şey belirtilmezse, çerez oturumun sonunda (tarayıcı kapandığında) zaman aşımına uğrar.
- **\$yol:** Çerezin üzerinde etkin olacağı sunucudaki yol. '/' belirtilirse çerez alanın tamamında kullanılabilir olacaktır. Eğer '/isim1/' belirtilirse, çerez sadece /isim1/ dizininde ve /isim1/isim2/ gibi alt dizinlerde kullanılabilir olacaktır. Ön tanımlı değer çerezin atandığı içinde bulunulan dizindir.
- **\$alan:** Çerezin kullanılabilmesi için alan. Çerezi mesela.dom ve alt alanlarında kullanılabilir yapmak için değer olarak '.mesela.dom' belirtmelisiniz. Değer olarak, 'belge.mesela.dom' belirtilirse çerez sadece belge alt alanında kullanılabilir olacaktır.
- **\$güvenlik:** Çerezin istemciye güvenli bir HTTPS bağlantısı üzerinden aktarılması gerektiğini belirtmek için kullanılır. TRUE belirtildiği

takdirde, çerez sadece güvenli bağlantı mevcutsa gönderilecektir. FALSE ön tanımlı değerdir.

- **\$sadece\_http**: TRUE olduğu takdirde çerez sadece HTTP protokolü üzerinden erişilebilir olacaktır. Yani çerez, JavaScript gibi betik dilleri tarafından erişilebilir olmayacaktır. Bu ayarlama, kimlik hırsızlığı riskini etkin şekilde azaltmaya yardımcı olabilir.

### Örnek:



```
D:\xampp\htdocs\cereler_oturum_yonetimi\cookie01.php
1 <?php
2 $deger = "test deđeri.";
3 setcookie("test_cerezi", $deger);
4 ?>
5 <html>
6 <body>
7 ...
8 </body>
9 </html>

D:\xampp\htdocs\cereler_oturum_yonetimi\cookie02.php
1 <html>
2 <body>
3 <?php
4 echo $_COOKIE["test_cerezi"];
5 ?>
6 </body>
7 </html>
```



Örnekte <html> etiketinden önce setcookie kullanılmıştır. Bunun sebebi setcookie fonksiyonu ile çerez oluştururken sayfaya herhangi bir bilgi gönderimi yapılmadan önce fonksiyon kullanılmalıdır.

Yukarıdaki cookie02.php sayfası ziyaret edildiğinde ekranda “test deđeri” yazacaktır.

Yukarıdaki örnekte cookie01.php fonksiyonu şu şekilde düzenlenirse çereze zaman aşım süresi eklenmiş olur. “time()” fonksiyonunun 60\*60\*24 ile çarpılması zaman olarak bir günü ifade eder. “time()” fonksiyonu o anki zaman damgasını saniye cinsinden elde eder. Diğer ifadelerde 60 saniye, 60 dakika ve 24 saati ifade eder. Bu sayıların çarpımı bir günün saniye karşılığını verir.

**setcookie(“test\_cerezi”, \$deger, time() + (60\*60\*24))**

Çerez isimleri dizi isimleri olabilir. Böylece siz PHP betiğinizde bu çerezler için dizi kullanırken, kullanıcının sisteminde bunlar ayrı ayrı çerezler olarak saklanırlar. Tek bir çereze çok sayıda isim ve deđer atamak için “explode” işlevini kullanabilirsiniz.



## Örnek:

Şimdi de çerezleri kullanarak oturum yönetimi yapalım.

```
D:\xampp\htdocs\cereszler_oturum_yonetimi\oturum_ac.php
1 <?php
2 $kullanici='ahmet';
3 $sifre='12345';
4 setcookie("kullanici",$kullanici, time()+(60*60*24*30));
5 setcookie("sifre",$sifre, time()+(60*60*24*30));
6 ?>
7 <html>
8 <body>
9 Şifre kaydedildi!
10 </body>
11 </html>

D:\xampp\htdocs\cereszler_oturum_yonetimi\sayfa1.php
1 <?php
2 if ( $_COOKIE['kullanici']!= 'ahmet' and $_COOKIE['sifre']!= '12345' ){
3     echo "Şifreyi yanlış girdiniz..";
4     exit();
5 }
6 ?>
7 <html>
8 <body>
9 Oturum Açıldı Merhaba: <?php echo $_COOKIE['kullanici'];?>
10 </body>
11 </html>
```



Yukarıda birinci dosya olan oturum\_ac.php dosyası ile kullanıcı girişi yapılır. Bu aşamadan sonra sayfa1.php dosyası açıldığında kullanıcı adı ve şifre ile çerez karşılaştırılır. Eğer yanlış ise sayfa exit() fonksiyonu ile sonlandırılır. Değilse sayfa açılmaya devam edilir. Bu aşamada site içinde başka bir sayfaya geçilmek istenirse kullanıcı adı ve şifrenin URL sorgu cümlesi ile taşınarak (**sayfa2.php?kullanici=ahmet&sifre=12345**) her sayfanın başında çerezler ile karşılaştırılması gerekir. Karşılaştırma için kullanılacak if cümlesi ise şu şekilde olmalı.

```
<?php
if ( $_COOKIE['kullanici']!=$_GET['kullanici'] and $_COOKIE['sifre']!=
[$_GET[sifre]]){
echo "Şifreyi yanlış girdiniz...";
exit();
}
?>
```

‘**exit**’ fonksiyonu sayfanın çalışmasını sonlandırır.

Çerezlerle oturum yönetimi PHP oturum yönetimi kullanılmadan çok zayıf olacaktır. Oturum açmak için tarayıcılar zaten “Kullanıcı adı ve şifre kaydedilsin mi?” şeklinde soru sormaktadır. Bu soruya evet dediğinizde o sitenin kullanıcı adı ve şifresi çerezler yoluyla kaydedilmektedir. Bundan sonrasında PHP oturumu açılarak hem sayfalar arası geçişler çok daha kolay olur hem de çok sayıda oturumla ilgili verileri saklama imkânı olur.

## 3.2. PHP’de Session (Oturum Yönetimi)

PHP’deki oturum desteği, belli bir veriyi sonraki erişimlerden korumak için kullanılır. Böylece daha özelleştirilmiş uygulamalar oluşturabilir ve sitenizi daha fonksiyonel yapabilirsiniz.

Sitenize gelen her ziyaretçiye eşsiz bir kimlik atanır, buna oturum kimliği (**session\_id**) diyoruz. Bu kimlik istemci tarafından bir çerezde ya da doğrudan URL üzerinde saklanır.

Oturum desteği, internet sayfalarının çok sayıda (sayısı yazılımcıya bağlı) değişkeni oturuma kaydetmenize izin verir. Böylece bir oturumun başlaması ile birlikte bütün değerler sayfalar arası geçişlerde korunur.

PHP’de oturum yönetimi için kullanılan pek çok fonksiyon vardır. Bu modülde **session\_start()** ve **session\_destroy()**fonksiyonları konu edilecektir. Aynı zamanda **\$\_SESSION** küresel değişkeni de modülde konu edilecektir.

### 3.2.1. Session Kullanım Yerleri

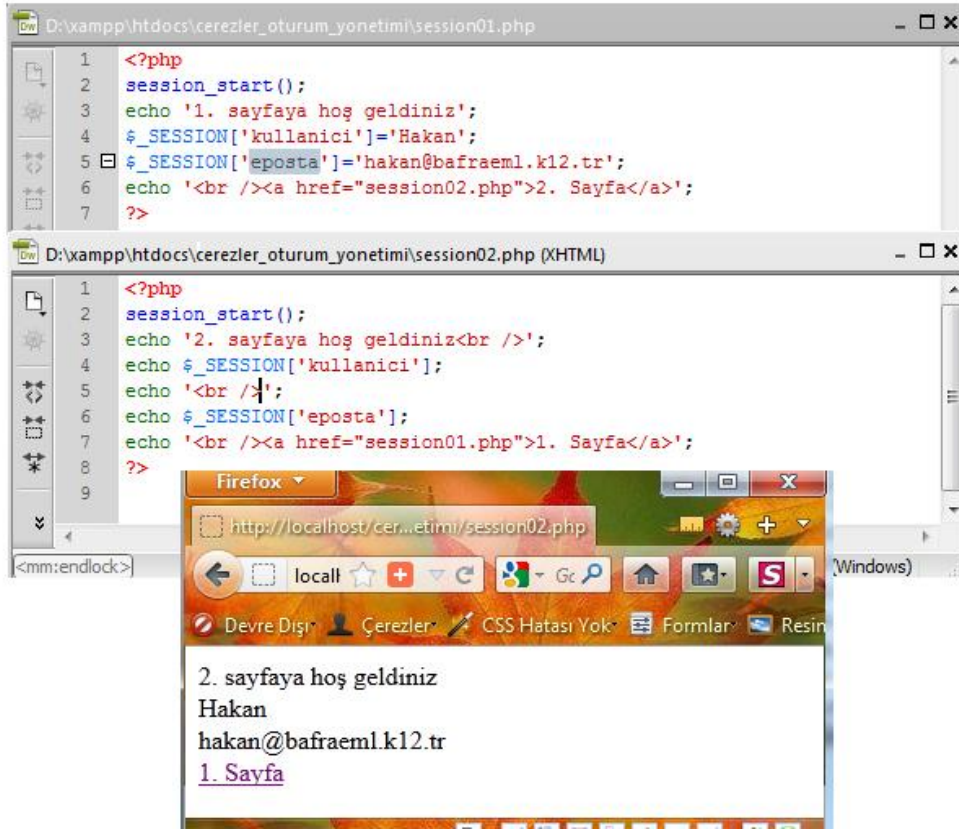
Oturumun açılması ile birlikte çok sayıda değişkeni sayfalar arasında kullanabilmek, kullanıcıların tarayıcı kapatılana kadar site üyelik bilgilerinin aktif olarak kullanılabilmesini sağlar. E-posta sistemleri, üyelik gerektiren sistemler bu yöntemi kullanırlar

E-ticaret sistemleri oturum yönetimi sayesinde müşterinin kişisel bilgilerinin yanında aktif oturumdaki alış-verişlerini oturum değişkenleri ile birlikte taşır. Buna yaygın olarak alış-veriş sepeti denmektedir.

## ➤ session\_start() fonksiyonu

session\_start() bir oturumu ya oluşturur ya da GET, POST veya çerez gibi bir istek üzerinden aktarılan oturum kimliğine istinaden eski bir oturumu geçerli oturum haline getirir. Oturum başarıyla başlatılabilmişse **TRUE** aksi takdirde **FALSE** döner.

Oturumun başlaması ile birlikte oturum değişkenlerinin tanımlanması mümkün olur. Oturum değişkenleri \$\_SESSION dizi değişkeninde tanımlanır. Örneğin; \$\_SESSION['kullanici'] = 'ahmet' gibi.



```
1 <?php
2 session_start();
3 echo '1. sayfaya hoş geldiniz';
4 $_SESSION['kullanici']='Hakan';
5 $_SESSION['eposta']='hakan@bafraeml.k12.tr';
6 echo '<br /><a href="session02.php">2. Sayfa</a>';
7 ?>
```

```
1 <?php
2 session_start();
3 echo '2. sayfaya hoş geldiniz<br />';
4 echo $_SESSION['kullanici'];
5 echo '<br />';
6 echo $_SESSION['eposta'];
7 echo '<br /><a href="session01.php">1. Sayfa</a>';
8 ?>
```

Firefox  
http://localhost/ce...etimi/session02.php  
local  
Devre Dış Çerezler CSS Hatası Yok Formlar Resin

2. sayfaya hoş geldiniz  
Hakan  
hakan@bafraeml.k12.tr  
[1. Sayfa](#)

! Oturum değişkenlerinin sayfalar arasında kaybolmadan dolaşabilmesi için session\_start fonksiyonunun sayfaya hiçbir HTML değeri gönderilmeden önce başlatılması gerekir.

'Bu uygulamalar yaygın olarak veri tabanları ile birlikte kullanılır. Üye giriş sayfasında kullanıcı adı ve şifre bölümü olur. Kullanıcının girdiği bilgiler veri tabanında doğrulanır. Eğer doğrulanma gerçekleşmişse oturum başlatılır ve oturum değişkenleri oluşturulmaya başlanır. Eğer bilgiler onaylanmaz ise kullanıcı giriş sayfasına yönlendirilir.

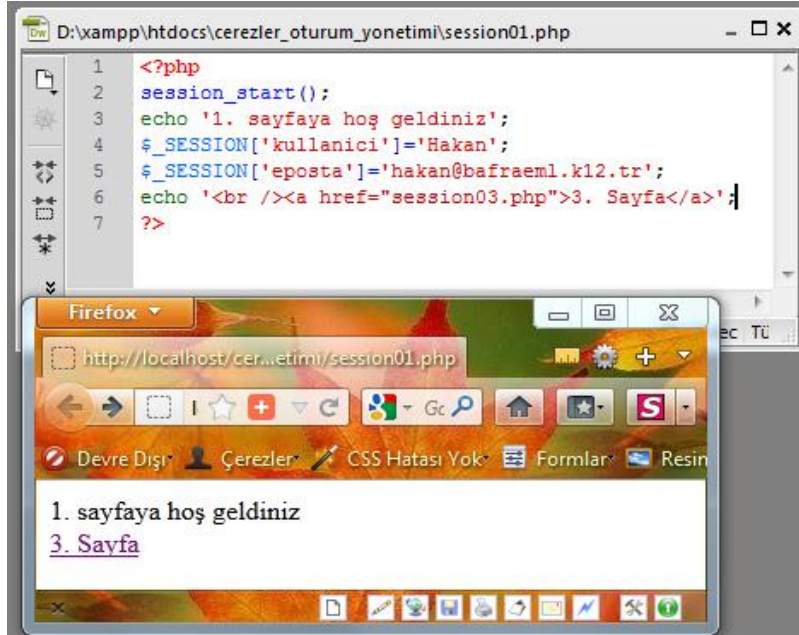
➤ **session\_destroy() fonksiyonu**

session\_destroy() işlevi geçerli oturumla ilişkilendirilmiş tüm veriyi yok eder. Kullanımı için aşağıdaki örneği inceleyiniz.


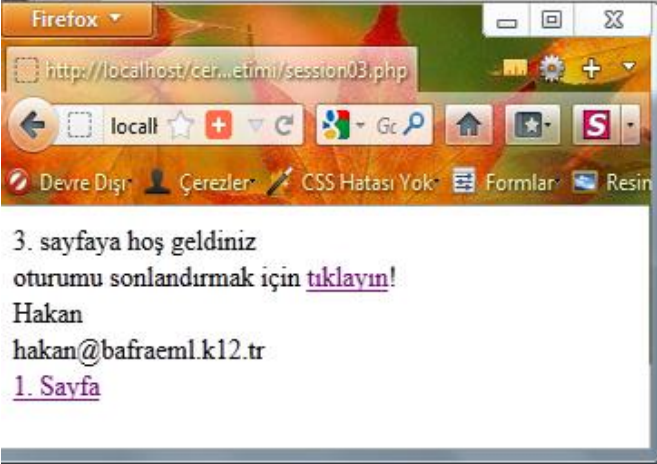
**Örnek:**

session01.php dosyası ile oturum açılmış ve kullanıcı ile eposta adlı iki küresel oturum değişkeni tanımlanmıştır.

3.Sayfa linki ile birlikte sayfa session03.php'ye yönlendirilmiştir. Bu sayfada yer alan oturum sonlandırma linkinde oturum sonlandırma işlemi için URL cümlesi oluşturulmuştur. Bu linke tıklanınca \$\_GET üstünden sayfaya oturumu kapatma değeri gönderilmiş, session03.php dosyası 3. satırında yer alan **session\_destroy** fonksiyonu ile oturum sonlandırılmış ve bütün oturum değişkenleri silinmiştir.



```
D:\xampp\htdocs\cerrezler_oturum_yonetimi\session03.php (XHTML)
1 <?php
2 error_reporting(0);
3 if($_GET['cikis']==1)
4     session_destroy();
5 else
6     session_start();
7 echo '3. sayfaya hoş geldiniz<br />';
8 if(!empty($_SESSION['eposta']))
9     |echo 'oturumu sonlandırmak için <a href="session03.php?cikis=1">
    tıklayın</a>!<br />';
10
11 echo $_SESSION['kullanici'];
12 echo '<br />';
13 echo $_SESSION['eposta'];
14 echo '<br /><a href="session01.php">1. Sayfa</a>';
15 ?>
```



! Oturum devam ederken bir oturum değişkenini silmek için unset() fonksiyonu kullanılır. Örneğin; unset(\$\_SESSION['eposta']) gibi.

## UYGULAMA FAALİYETİ

Oturum yönetimi işlemi için aşağıdaki uygulamayı yapınız.

İşlem Basamakları	Öneriler
➤ Bir üye kayıt sayfası oluşturunuz. Formu uye_kayit.php sayfasına yönlendirin	➤ Ad, soyad, e-posta, şifre, mezuniyet, meslek alanlarından oluşturulabilir.
➤ uye_kayit.php sayfasında formdan gelen bilgileri çerezler oluşturarak tutunuz.	➤ <i>setcookie</i> fonksiyonunu kullanabilirsiniz. .
➤ Bir üye giriş sayfası oluşturunuz	➤ E-posta ve şifre alanlarından oluşabilir.
➤ Üye giriş sayfasını index.php sayfasına yönlendiriniz. Kullanıcı doğrulaması yapınız.	➤ index.php sayfasında \$_COOKIE kullanılarak karşılaştırma yapılabilir.
➤ index.php'de yapılan karşılaştırma doğru ise session ile oturum açınız.	➤ Karşılaştırma doğru ise session_start fonksiyonu kullanılabilir. Ekrana kullanıcı bilgileri yazdırılabilir.

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız beceriler için **Evet**, kazanamadığınız beceriler için **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Üye kayıt sayfası oluşturabildiniz mi?		
2. Üye kayıt sayfasında ki formu uye_kayit.php sayfasına yönlendirebildiniz mi?		
3. Üye giriş sayfası oluşturabildiniz mi?		
4. Üye giriş sayfasını index.php sayfasına yönlendirebildiniz mi?		
5. Kullanıcı doğrulaması yapabildiniz mi?		
6. Karşılaştırma sonucuna bağlı olarak oturum durumunu belirleyebildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme ”ye geçiniz.



## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

1. ( ) setcookie fonksiyonu ile çerez oluştururken sayfaya herhangi bir HTML bilgi gönderimi yapılmadan önce setcookie fonksiyonu kullanılmalıdır.
2. ( ) Çerezler kullanılarak oturum yönetimi gerçekleştirilemez.
3. ( ) E-posta ve e-ticaret sistemlerinde session oturum yönetimi kullanılır.

Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

4. Hangisi setcookie fonksiyonunun aldığı değerlerden değildir?  
A) isim  
B) zaman aşımı  
C) değer  
D) metod
5. time()+60\*60\*24\*30 zaman bildirimini neyi temsil eder?  
A) 30 gün  
B) 3 gün  
C) 1 gün  
D) hiçbirisi
6. PHP'de session oturum açmak için hangi fonksiyon kullanılır?  
A) session\_destroy  
B) session\_start  
C) setcookie  
D) hiçbirisi
7. Hangisi bir cookie değişkenidir?  
A) \$\_SESSION  
B) \$\_GET  
C) \$\_POST  
D) \$\_COOKIE



8. Hangisi bir session deęiřkenidir?

- A) \$\_SESSION
- B) \$\_GET
- C) \$\_POST
- D) \$\_COOKIE

Ařaęıdaki cümleleri dikkatlice okuyarak boş bırakılan yerlere doęru sözcüęü yazınız.

9. .... , kullanıcıyı ayırt etmenin en etkili yollarından biridir.

10. PHP’de session oturum deęiřkenlerini ..... dizi deęiřkenleri tařır.

## DEęERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılařtırınız. Yanlıř cevap verdięiniz ya da cevap verirken tereddüt ettięiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doęru ise “Modül Deęerlendirme”ye geçiniz.

# MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Dinamik bir web site veri akışı nasıldır?  
A) istemciden-sunucuya, sunucudan-istemciye  
B) istemciden-sunucuya  
C) sunucudan-istemciye  
D) hiçbiri
2. <form> etiketinin üç önemli özelliği nedir?  
A) name, size, method  
B) name, action, method  
C) id, action, method  
D) hiçbiri
3. Hangisi <input> etiketinin type özelliğinin alabildiği bir değerdir?  
A) textarea  
B) select  
C) radio  
D) option
4. Hangisi doğru bir URL sorgu cümlesidir?  
A) site.php&ad=ali&soyad=ay  
B) site.php?ad=ali&soyad=ay  
C) site.php%ad=ali&soyad=ay  
D) site.php?ad-ali&soyad-ay
5. PHP, form kontrol neslerinin (form etiketlerinin) hangi özelliğini kullanarak taşıdıkları değeri elde eder?  
A) id  
B) value  
C) size  
D) name
6. Bir formdan gelen resmi elde etmek için hangi küresel değişken kullanılır?  
A) \$\_FILES  
B) \$\_POST  
C) \$\_GET  
D) hiçbiri
7. 'htmlspecialchars' fonksiyonu aşağıdaki konulardan hangisiyle en çok ilgilidir?  
A) veri tipi  
B) veri güvenlik  
C) veri silme  
D) hiçbiri

8. Regular expression ifadesinin Türkçe karşılığı nedir?  
A) özel karakterler  
B) metin fonksiyonları  
C) görünmez karakterler  
D) düzenli ifadeler
9. Hangisi görünmez karakterler için kullanılmaz?  
A) Cx  
B) \t  
C) \d  
D) \n
10. Hangisi geniş aralıkları içeren karakterler için kullanılmaz?  
A) \b  
B) \d  
C) \w  
D) \s
11. Hangisi sınırları yakalamak için kullanılır?  
A) \W  
B) \S  
C) \D  
D) \A
12. /(PHP)(Öğrenmeyi)(Seviyorum)/ söz dizisini ekrana Seviyorum PHP Öğrenmeyi şeklinde ekrana yazan yer değiştirme söz dizisi nasıldır?  
A) \1\2\3  
B) \1\3\2  
C) \3\1\2  
D) \2\1\3
13. /^[^0-9]\$/ söz dizisi hangi metin ile eşleşme sağlar?  
A) Yaşım 15  
B) 15 yaşındayım  
C) Samsun 5 derece  
D) Samsunun plaka kodu 55
14. @[a-zA-Z0-9]+@\$ söz dizisi hangi metin ile eşleşme sağlamaz?  
A) ab-c  
B) ab1c  
C) 123-a  
D) hiçbirisi
15. /^php\$/i söz dizisi hangi metin ile eşleşme sağlar?  
A) Merhaba PHP  
B) PHP  
C) php çok kolay  
D) p.h.p.

16. session\_start fonksiyonu kaç deęer(parametre) alır?  
A) 0  
B) 1  
C) 2  
D) 3

**Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.**

17. ( ) POST yöntemi GET yönteminden daha güvenli değildir.
18. ( ) Form kontrol neslerinin (form etiketlerinin) dizi deęer taşıması için kullanılan semboller [] köşeli parantez sembolleridir.
19. ( ) **Multipurpose Internet Mail Extensions** (Çok amaçlı İnternet Posta Eklentileri); E-posta uygulamaları aracılığıyla gönderilecek olan iletiye çeşitli türdeki içerięi eklemek için kullanılan bir İnternet standardıdır.
20. ( ) PHP' de tip denetim yapısı katı değildir. Bu durum PHP'nin daha güvenli olmasını sağlar.
21. ( ) Oturum sonlandırmak için session\_start fonksiyonu kullanılır.
22. ( ) **setcookie** fonksiyonu ile bir İnternet sayfada birden çok deęer oluşturulamaz.
23. ( ) **setcookie** fonksiyonu belli bir alt dizin (subdomain) için kullanılabilir.
24. ( ) Oturum Yönetimi kavramının gelişmesi, İnternet sitelerinin ziyaretçilerine daha iyi, hızlı ve özgün hizmet verebilmeleri ile ilgilidir.
25. ( ) Sitenize gelen her ziyaretçiye eşsiz bir kimlik atanır, buna oturum kimlięi (**session\_id**) denir.

## DEęERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öęretmeninize başvurunuz.

# CEVAP ANAHTARLARI

## ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	Yanlış
4	D
5	C
6	B
7	A
8	Dinamik
9	Tek

## ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	Doğru
4	B
5	B
6	C
7	A
8	B
9	*
10	.

## ÖĞRENME FAALİYETİ-3'ÜN CEVAP ANAHTARI

1	Doğru
2	Yanlış
3	Doğru
4	D
5	A
6	B
7	D
8	A
9	cookie
10	\$_SESSION

## MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

1	A
2	B
3	C
4	B
5	D
6	A
7	B
8	D
9	C
10	A
11	D
12	C
13	D
14	A
15	B
16	A
17	Yanlış
18	Doğru
19	Doğru
20	Yanlış
21	Yanlış
22	Yanlış
23	Doğru
24	Doğru
25	Doğru

## KAYNAKÇA

- ÇAYCI Özgür, **PHP ve MYSQL**, Seçkin, Ankara, 2003
- [http://tr.wikipedia.org/wiki/Web\\_sitesi](http://tr.wikipedia.org/wiki/Web_sitesi)(3.01.2012)
- (<http://www.w3.org/TR/html4/interact/forms.html#h-17.2>) (3.01.2012)
- ([http://www.belgeler.org/hpm/html-php-mysql-giris\\_prg-html\\_form.html](http://www.belgeler.org/hpm/html-php-mysql-giris_prg-html_form.html)) (5.01.2012)
- <http://php.net/manual/tr/tutorial.forms.php>(5.01.2012)
- <http://php.net/manual/tr/features.file-upload.post-method.php>(7.01.2012)
- <http://www.php.net/manual/tr/function.copy.php>(7.01.2012)
- <http://www.php.net/manual/tr/security.variables.php>(9.01.2012)
- <http://www.bilgiguvenligi.gov.tr/teknik-yazilar-kategorisi/php-icin-basit-guvenlik-onlemleri.html>(9.01.2012)
- <http://tr2.php.net/manual/tr/ref.var.php>(9.01.2012)
- <http://e-bergi.com/2008/Mayis/Duzenli-Ifadeler-1>(15.01.2012)
- <http://e-bergi.com/2008/Mayis/Duzenli-Ifadeler-1>(15.01.2012)
- <http://www.ceviz.net/programlama/duzenli-ifadeler-giris>(17.01.2012)
- <http://yilmazkilicaslan.trakya.edu.tr/teaching.html>(18.01.2012)
- <http://www.php.net/manual/tr/features.cookies.php> (20.01.2012)
- <http://www.php.net/manual/tr/function.setcookie.php>(21.01.2012)