T.C.
MİLLÎ EĞİTİM BAKANLIĞI

DENİZCİLİK

İNTERNET PROGRAMCILIĞI-3

Ankara, 2013

- 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. FONKSİYONLAR 3
1.1. Fonksiyon Tanımlama ve Çağırma 3
1.1.1. Geriye Değer Döndüren Fonksiyonlar 5
1.1.2. Varsayılan Değer Alan Argümanlı Fonksiyonlar 7
1.1.3. Global Değişkenli Fonksiyonlar 8
1.1.4. Statik (Durağan) Değişkenli Fonksiyonlar 9
1.2. Zaman Fonksiyonları 10
1.2.1. getdate() 10
1.2.2. date() 11
1.2.3. time() 12
1.2.4. mktime 13
1.3. Mail Gönderme 13
UYGULAMA FAALİYETİ 16
ÖLÇME VE DEĞERLENDİRME 17
ÖĞRENME FAALİYETİ–2 18
2. DİZİLER 18
2.1. Dizi Oluşturma 18
2.1.1. Foreach() Fonksiyonu 20
2.2. Dizi Düzenleme 20
2.2.1. Dizileri Sıralama 21
2.2.2. İlişkili Dizileri Sıralama 21
2.2.3. Dizilere Değişken Ekleme 21
2.2.4. Dizileri Birleştirme 22
2.2.5. Dizinin İlk Elemanını Silme 22
2.2.6. Diziden Kesit Alma 23
UYGULAMA FAALİYETİ 24
ÖLÇME VE DEĞERLENDİRME 25
MODÜL DEĞERLENDİRME 26
CEVAP ANAHTARLARI 28
KAYNAKÇA 29

AÇIKLAMALAR

ALAN	**Denizcilik**
DAL/MESLEK	
MODÜLÜN ADI	**İnternet Programcılığı-3**
MODÜLÜN TANIMI	Web programlama içinde fonksiyon ve dizi işlemleri yapabilme ile ilgili öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	İnternet Programcılığı-2 modülünü başarmış olmak
YETERLİK	Programlama içinde fonksiyon ve dizi işlemleri yapmak
MODÜLÜN AMACI	**Genel Amaç** Öğrenci bu modül ile gerekli ortam sağlandığında; programlama komutları yardımıyla fonksiyon ve dizi uygulamaları hazırlayabilecektir. **Amaçlar** **1.** Programlama içinde fonksiyon kullanımını öğrenerek, uygulamalar yapabilecektir. **2.** Dizi işlemlerini gerçekleştirebilecektir.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Web programlama yazılımlarını çalıştırabilecek yeterlikte bilgisayar, internet bağlantısı
ÖLÇME VE DEĞERLENDİRME	Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. Öğretmen modül sonunda size ölçme aracı (uygulama, soru-cevap) uygulayarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Program yazarken kolaylık ve zaman tasarrufu en çok ihtiyaç duyulan unsurlardır. Fonksiyonlar da programınız yazarken önemli yerlerde kolaylık ve önemsenmeyecek kadar zaman tasarrufu sağlar.

İnternet programcılığı 1 ve 2 modüllerinde hazır fonksiyonları kullandınız. Bu modülde ise fonksiyonların nasıl oluşturulduğunu program içinde nasıl kullanıldığını ve programlama için önemli olan tarih zaman ve mail gönderme programlarını öğreneceksiniz.

Ayrıca bu modülün devamında programlarımızda sıkça kullandığımız dizi kavramını öğreneceksiniz. Dizilerin nasıl oluşturulduğunu ve nasıl düzenlendiği konusunda detaylı bilgiye sahip olacaksınız.

Unutmayınız ki zekice kod yazmak için programlama dilinin her konusuna hâkim olmak gerekir. Fonksiyonlar ve diziler de bir programı için vazgeçilmez nesnelerdir.

ÖĞRENME FAALİYETİ–1

AMAÇ

Programlama içinde fonksiyon kullanımını öğrenerek, uygulamalar yapabileceksiniz.

ARAŞTIRMA

- Fonksiyonlar hangi amaçlarla kullanılır araştırınız.
- Fonksiyon parametreleri (argümanları) hangi amaçla kullanılır araştırınız.
- Fonksiyon kullanmanın önemi ve kolaylıkları nelerdir tartışınız.

1. FONKSİYONLAR

Bir bilgisayar programının kalbi, fonksiyonlardır. Fonksiyonlar, bir kere tanımlanır ve birçok kereler, ihtiyaç olduğu noktada hizmete çağrılır. Fonksiyonlar genellikle bir iş veya hesaplama yapar ve bunun sonucu olarak ortaya yeni bir değer çıkartır. Bu değere "fonksiyondan dönen değer" denir.

Fonksiyonların önemini bir örnekle belirtelim. Örneğin internet üzerinde çalışan hesap makinesi yapmak istiyoruz. Topla, çıkar, çarp ve böl isimli 4 tane fonksiyonumuz olduğunu varsayalım. Kullanıcı tek tıklama ile isteği sayıların aritmetiksel işlemlerini yapabilmektedir. Her işlemin kendine ait fonksiyonu bulunmakta ve kullanıcı hangi işlemi seçtiyse o fonksiyon çağrılmaktadır. Fonksiyon bir kere çağrıldıktan sonra geriye değer döndürür ve bir dahaki çağrılmasına kadar program içinde görevi biter.

PHP’nin kullanıma hazır birçok fonksiyonu bulunmaktadır. echo() , print() veya printf bunların başlıcalarıdır. Bu fonksiyonların aldıkları parametre ile ekrana string yazdırmaktadır.

1.1. Fonksiyon Tanımlama ve Çağırma

PHP'de fonksiyonlar **function** komutu ile oluşturulur. Tanımladığımız fonksiyon, kendisini göreve çağıracak komuttan, yapacağı işlemde kullanmak üzere değer alacaksa, bu değerlere vereceğimiz değişken isimleri fonksiyon adının yanında parantez içinde gösterilir. Fonksiyon birden fazla değer bekleyecekse, bunların değişken adlarının arasına virgül koyarız. Fonksiyona ulaştırılan değerlere argüman denir. Fonksiyon, kendisine bir değer ulaştırılmasını beklemese bile içi boş parantez koymamız gerekir. Buna göre PHP'de fonksiyon şöyle yazılır:

```
function fonksiyonun_adı (argüman1, argüman2, ... argümanN)
{
   fonksiyonun yapacağı işe ilişkin komutlar
}
```

Fonksiyon adları, değişken adları ile aynı kurallara tabidir. Fonksiyon içinden fonksiyon çağrılabilir; fonksiyonların içinde PHP'nin kendi fonksiyonları kullanılabilir.

Şimdi fonksiyonlarla ilgili basit bir örnek yapalım. Gönderdiğimiz metni ekrana yazdıran PHP fonksiyonunu yazalım.

```
      <html>
      <head>
      <title>Untitled Document</title>
      <meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
      </head>

      <body>
      <?
      function yazdir($metin)
      {
      print($metin);
      }
      yazdir("Bilisim Teknolojileri");
      ?>
      </body>
      </html>
```

Örnekte yazdır isminde bir fonksiyon bulunmaktadır. Bu fonksiyona “Bilişim Teknolojileri” değerinde bir string gönderilmekte ve fonksiyon içerisinde bulunan print komutuyla parametre ekrana yazdırılmaktadır.

Fonksiyonlar tek parametre aldıkları gibi birden fazla da parametre alırlar. Şimdi iki parametre alan bir fonksiyon yazalım.

Örneğin, gireceğimiz iki sayıdan birinci sayı mı yoksa ikinci sayı mı daha büyük bu karşılaştırmayı yapan ve ekrana yazan fonksiyonu yazalım.

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
</head>
<body>
<?
function buyuksayi($sayi1,$sayi2)
{
 if ($sayi1>$sayi2)
print("BiRiNCi SAYI DAHA BÜYÜK");
else
print("iKiNCi SAYI DAHA BÜYÜK");
}
buyuksayi(25,17);
?>
</body>
</html>
```

Bu örnekte fonksiyonumuza sayi1 ve sayi2 adında iki farklı değişken yani parametre gönderdik ve fonksiyon içinde if deyimini kullanarak karşılaştırma yaptık ve ekrana büyük olan sayıyı yazdırdık. Ekran çıktımız "BİRİNCİ SAYI DAHA BÜYÜK" şeklinde olacaktır.

1.1.1. Geriye Değer Döndüren Fonksiyonlar

Buraya kadar olan örneklerde sadece hep belirli işi yapan fonksiyonları gördük. Bunların yanı sıra geriye değer döndüren fonksiyonlar bulunmaktadır. Bu tür fonksiyonlar belirli bir işlemi yapar ve programda çağrıldı yere değer döndürür.Basit bir örnekle geriye değer döndüren fonksiyonları pekiştirelim. Girdiğimiz iki sayının toplamını bulan fonksiyonu yazalım.

```
<HTML>
<HEAD>
<TITLE>PHP'de Deger Döndüren Fonksiyon</TITLE>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
</HEAD>
<BODY>
<?php
function topla ($sayi1, $sayi2)
{
        $sonuc = $sayi1 + $sayi2;
        return $sonuc;
}
$sayi1 = 12;
$sayi2 = 5;
print topla($sayi1, $sayi2);
?>
</BODY>
</HTML>
```

Yaptığımız bu örneği 4 işlem için yazarsak PHP kodumuz şu şekilde olacaktır:

```
<?php
function topla ($sayi1, $sayi2)
{
	$sonuc = $sayi1 + $sayi2;
	return $sonuc;
}
function cikart ($sayi1, $sayi2)
{
	$sonuc = $sayi1 - $sayi2;
	return $sonuc;
}
function carp ($sayi1, $sayi2)
{
	$sonuc = $sayi1 * $sayi2;
	return $sonuc;
}

function bol ($sayi1, $sayi2)
{
	$sonuc = $sayi1 / $sayi2;
	return $sonuc;
}
$sayi1 = 20;
$sayi2 = 4;
print topla($sayi1, $sayi2);
print ("<br>");
print cikart($sayi1, $sayi2);
print ("<br>");
print carp($sayi1, $sayi2);
print ("<br>");
print bol($sayi1, $sayi2);
print ("<br>");
?>
```

1.1.2. Varsayılan Değer Alan Argümanlı Fonksiyonlar

Bazı fonksiyonlarda değerleri önceden varsayılan olarak belirlenmiş parametreler bulunur. Bu fonksiyonlarda değeri girilmemiş eksik parametre veya argüman varsa programımız hata vermeden çalışır. Çünkü daha önceden belirlediğimiz varsayılan değeri alır. Aşağıdaki örneği dikkatlice inceleyiniz. Örnekte iki sayının çarpım işlemi yapılmaktadır. Ancak sayılardan herhangi biri girilmediği takdirde varsayılan değeri 1 olarak almakta ve geriye o değeriyle değer döndürmektedir.

```
<HTML>
<HEAD>
<TITLE>PHP'de varsayýlan deðeri alan Fonksiyon</TITLE>
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
</HEAD>
<BODY>
<?php
function carp ($sayi1 = 1, $sayi2 = 1)
{
        $sonuc = $sayi1 * $sayi2;
        return $sonuc;
}
$birincisayi = 20;
$ikincisayi = 4;
print carp($birincisayi, $ikincisayi);
print ("<br>");
print carp($birincisayi);
print ("<br>");
print carp();
?>
</BODY>
</HTML>
```

Browserdaki çıktımız sırasıyla 80, 20 ve 1 şeklinde olacaktır.

1.1.3. Global Değişkenli Fonksiyonlar

Fonksiyonlar sadece kendi içlerinde bulunan değişkenleri kullanabilir. Programda yer alan değişkenleri kullanamaz. Bu değişkenleri kullanabilmeleri için global ifadesini kullanmamız gerekir.

```
<?php
$sayi = 25;
function yazdir ()
{
        print ($sayi);
}
yazdir();
?>
```

Yandaki örnek kodda $sayi değişkeni fonksiyon dışında olduğu için yazdir() fonksiyonu içinde herhangi bir değer yazdırılamayacaktır. Dolayısıyla ekran çıktısı boş olacaktır.

<?php
$sayi = 25;
function yazdir ()
{ global $sayi;
print ($sayi);
}
yazdir();
?>
	Bu kodda ise önemli bir fark bulunmaktadır. Fonksiyon içinde bulunan global $sayi satırı ile fonksiyon dışında bulunan $sayi değişkeni fonksiyon içinde kullanılmaktadır. Ve ekrana 25 değeri yazdırılmaktadır.

1.1.4. Statik (Durağan) Değişkenli Fonksiyonlar

Fonksiyonlarımızda kullandığımız değişkenler fonksiyonun işi bittiği zaman ilk değeri ne ise o değerini alır yani diğer bir deyişle değişken ölür. Fonksiyonun ikinci veya daha sonra çağırıldığı zamanlarda değişkenlerinin en son değerlerine ihtiyacımız olabilir. Bunun için **static** ifadesini kullanabiliriz.

```
<?php
function saydir () {
        static $sayi = 0;
        $sayi++;
        print ("<h3>Fonksiyonun tuttuğu sayı: $sayi </h3>");
        }
// Başka kodlar buraya girebilir
        print ("<h2>Fonksiyonun birinci kez çağrılması:</h2>");
        saydir();
        print ("<h2>Fonksiyonun ikinci kez çağrılması:</h2>");
        saydir();
        print ("<h2>Fonksiyonun üçüncü kez çağrılması:</h2>");
        saydir();
        print ("<h2>Fonksiyonun dördüncü kez çağrılması:</h2>");
        saydir();
?>
```

Örnekte saydir() isimli bi fonksiyonumuz var bu fonksiyonda bulunan static değişken sayesinde $sayi değişkeninin değeri her çağrıldığında bir sayı artırılmakta ve en son değeri kalıcı olmaktadır.

Ekran çıktısı

Fonksiyonun birinci kez çağrılması:

Fonksiyonun tuttuğu sayı: 1

Fonksiyonun ikinci kez çağrılması:

Fonksiyonun tuttuğu sayı: 2

Fonksiyonun üçüncü kez çağrılması:

Fonksiyonun tuttuğu sayı: 3

Fonksiyonun dördüncü kez çağrılması:

Fonksiyonun tuttuğu sayı: 4

1.2. Zaman Fonksiyonları

PHP'de tarih ve zaman bilgisini çok esnek bir biçimde kullanabilmemize izin veren önemli fonksiyonlar vardır. Şimdi bu önemli fonksiyonları sırayla inceleyelim:

1.2.1. getdate()

Bu fonksiyon çalıştırıldığında geriye bir dizi döndürür. Bu dizinin elemanlarını kullanarak o anki tarih ve zaman bilgisini programımıza rahatlıkla aktarabiliriz. Fonksiyon o anki zaman bilgisi ile ilgileniliyorsa parametresiz kullanılabilir. Ama geçmiş veya gelecekteki bir tarih için Unix Epoch biçiminde, ilgilenilen zamanı temsil eden bir parametre almalıdır.

getdate() fonksiyonun döndürdüğü dizinin indisleri ve değerleri aşağıdaki tablodaki gibidir:

Dizi indisleri	Değerleri
seconds	saniye
minutes	dakika
hours	saat
mday	ayın kaçıncı günü olduğu bilgisi (1,2,3,... 31)
wday	haftanın kaçıncı günü olduğu bilgisi (Pazar: 0, P.Tesi: 1,... C.tesi: 6)
mon	`ay (1,2,...12)`
year	`yıl (2001,2002,...)`
yday	`yılın kaçıncı günü olduğu bilgisi (1,2,3,...365)`
weekday	`haftanın hangi günü olduğu bilgisi (Sunday, Monday,... Saturday)`
month	`ay (January,February,...December)`

Tablo 1.1: getdate() fonksiyonun döndürdüğü dizinin indisleri ve değerleri

1.2.2. date()

Bu fonksiyon parametre olarak özel biçimlendirme ifadeleri alır. Genel kullanım şekli:

date("bicimlendirme parametreleri");

Özel biçimlendirme parametrelerinden en çok kullanılanları aşağıda verilen tablodaki gibidir.

Parametre	İşlevi
a	12 saat esasına dayanan Anglo-Sakson sistemlerinde öğleden önce ("am") veya öğleden sonra ("pm") işaretini sağlar.
A	Aynı işaretlerinin büyük harfle yazılmasını sağlar.
d	İki haneli gün sayısı, tek haneli günlerin önüne sıfır konur: "01"-"31"
D	Üç haneli gün adı kısaltması: "Sun","Mon", ..
F	Uzun ay adı: "January","February", ..
h	12 saatlik sistemde saat: "01"-"12"
H	24 saatlik sistemde saat: "00"-"23"
g	12 saatlik sistemde tek haneli saatlerin önüne sıfır konmadan saat: "1"-"12"
G	24 saatlik sistemde tek haneli saatlerin önüne sıfır konmadan saat: "0"-"23"
i	Dakika: "00-59"
j	Tek haneli sayıların önüne sıfır konmadan gün sayısı: "1"-"31"
l	(Küçük l harfi) Uzun gün adı: "Sunday", "Monday", ..
L	Artık yıl olup olmadığına ilişkin Boolean (doğru/yanlış) değişken. Artık yıl ise 1 değilse 0 döndürür.
m	Tek haneli sayıların önüne sıfır konarak ay sayısı: "01"-"12"
n	Tek haneli sayıların önüne sıfır konmadan ay sayısı: "1"-"12"
M	Kısaltılmış ay adı: "Jan","Feb", ..
s	Saniye: "00"-"59"
S	İngilizce "'nci" anlamında 2 karakter ek: "th","nd"
t	Belirtilen ayın toplam gün sayısı: "28"-"31"
w	Haftanın kaçıncı günü olduğu bilgisi: "0" Pazar, "6" Cumartesi
Y	Dört haneli yıl: "2000"
y	İki haneli yıl "02"
z	Yılın kaçıncı günü olduğu bilgisi: "0"-"365"

Normalde date() fonksiyonunun içine "-" işaretinde olduğu gibi yazılması üzere başka metinler de girebilirsiniz. Ama pek çok harfe özel bir anlam yüklendiği için bu tür girişimler başarılı bir çıktı vermeyecektir. Örneğin yukarıdaki örnekteki "Tarih" ifadesini date()'in içine ekleyemedik. Çünkü T, a, r, i ve h harflerinin hepsinin fonksiyon için ayrı bir anlamı var.

date() fonksiyonu, seçimlik olarak ikinci bir parametre alır. Bu parametre Unix Epoch biçiminde zaman bilgisi içermelidir. Böylece date() fonksiyonu o anki zaman göre değil belirtilen zamana göre çıktı üretecektir.

1.2.3. time()

Bu fonksiyon parametresiz çalışır ve Unix Epoch biçiminde o anki zaman bilgisini içeren bir tamsayı değer döndürür.

time() fonksiyonuna saniye cinsinden değer ekleyip çıkartarak geçmiş ve gelecek tarihlere ait Unix Epoch biçiminde değer elde edebilir, bu değeri date(), getdate() gibi fonksiyonlarda kullanabiliriz.

Örnek Kod

```
<?
$bugun=getdate();
print $bugun[mday]."-".$bugun[mon]."-".$bugun[year];
   //20-4-2007
print "<br>";

print date("d-m-Y");
   //20-4-2007
print "<br>";

$bugun=getdate(time());
print $bugun[mday]."-".$bugun[mon]."-".$bugun[year];
   //20-4-2007
print "<br>";

print date("d-m-Y",time());
   //20-4-2007
print "<br>";

$dun=getdate(time()-24*60*60);
print $dun[mday]."-".$dun[mon]."-".$dun[year];
   //19-4-2007
print "<br>";

print date("d-m-Y",time()-24*60*60);
   //19-4-2007
print "<br>";

$yarin=getdate(time()+24*60*60);
print $yarin[mday]."-".$yarin[mon]."-".$yarin[year];
   //21-4-2007
print "<br>";

print date("d-m-Y",time()+24*60*60);
   //21-4-2007
?>
```

1.2.4. mktime()

Bu fonksiyon parametre olarak aldığı zaman bilgilerine (saat, dakika, saniye, ay, gün, yıl) göre Unix Epoch biçiminde bir değer üretir. Parametresiz kullanıldığında o anki zaman için değer döndürür. time() fonksiyonunda sadece saniye cinsinden zaman üzerinde ileri geri oynanabildiği için mktime() fonksiyonu bu yönden daha kullanışlıdır.

Örnek Kod

```
<?
// Bundan iki gün bir ay üç yıl önce'sine yolculuk yapalım..
$zaman=mktime(0,0,0,date("m")-1,date("d")-2,date("Y")-3);
print "Tarih:".date("d-m-Y",$zaman);

?>
```

1.3. Mail Gönderme

Hazırladığımız web sitelerinden kullanıcılara otomatik mail atmak isteyebiliriz. Veya üyelik sistemi oluşturduğum uz bir sitede kullanıcıların üyeliklerini aktif hale getirmek için mail adreslerine aktivasyon kodu göndermemiz gerekebilir. Bu sıkça rastlanan durumlardan bir tanesidir. Bunun için kullanıcılara otomatik mail atmamız gerekir.

Php ile e-postalar yollamak gayet basit ve tek bir fonksiyon yani **mail()** ile gerçekleştirilebiliyor. Bu fonksiyonu en basit ve genel olarak şu şekilde kullanabiliyoruz:

Mail() fonksiyonun kullanımı şu şekildedir:

mail("gönderilecek e-posta adresi","e-postanın konusu","e-postanın içeriği")

Yukarıdaki ile bu işlemi en basit yol ile gerçekleştiriyoruz. Yollanacak e-postanın HTML içerikli olmasını, gönderen e-posta adresinin görüntülenmesini, e-posta içeriğinin dil kodunun tanımlanması, e-posta Cc ve Bcc alıcılarının belirlenmesi için *headers* olarak adlandırılan bilgilerinde gönderilmesi gerekmektedir. O zaman üstteki fonksiyonu şununla değiştiriyoruz.

`mail`**(**$alici_eposta, $eposta_konusu, $eposta_mesaji, $headers**)**

$alici_eposta : E-postayı yollayacağımız kişinin e-posta adresi.
$eposta_konusu : Yollayacağımız e-postanın konusudur.
$eposta_mesaji : Yollayacağımız e-postanın içeriğidir.
$headers: Gelişmiş e-posta tanımlamalarının tümünü ya da birkaçını içeren değişkendir ve kullanılma zorunluluğu yoktur.

$headers değişkenindeki tanımlamalar

From: E-postanın hangi e-posta adresinden yollandığıdır.
Reply-To: Kişi kendisine yollanan e-postayı yanıtlamak istediğinde, burada tanımlayacağınız e-posta adresine mesajını gönderebilir.
X-Mailer: Maili yollatan yazılımdır.
MIME-Version: E-posta MIME sürümüdür.
Content-type: İçerik türü ve karakter setlerinin tanımlamalarını içerir.
To: E-postanın gönderileceği adres(ler)dir.
Cc: E-postanın kopyasının gönderileceği görünür adres(ler)dir.
Bcc:E-postanın kopyasının gönderileceği gizli adres(ler)dir.

Şimdi birkaç adet uygulama yapalım.

Mail Gönderme Örneği

```
<?
$alici_eposta = "murat_dinc@eposta.com";
$eposta_konusu = "Tebrik";
$eposta_mesaji = "Bayramınızı en içten dileklerimle kutlarım";
mail($alici_eposta, $eposta_konusu, $eposta_mesaji);
?>
```

Yukarıdaki satırlar ile "Tebrik" konulu ve "Bayramınızı en içten dileklerimle kutlarım." mesajını içeren e-postayı "murat_dinc@eposta.com" adresine gönderdik.

Mail Gönderme Örneği

```
<? $headers = "From: alisakarya@eposta.com" . "\r\n" .
        "Reply-To: cevap@eposta.com" . "\r\n" .
        "X-Mailer: PHP/" . phpversion();
$alici_eposta = "eminerol@eposta.com";
$eposta_konusu = "Deneme";
$eposta_mesaji = "Bu bir deneme e-postası mesajidir.";
mail($alici_eposta, $eposta_konusu, $eposta_mesaji, $headers);      ?>
```

Bu örneğin birinci örnekten tek farkı "headers" bilgilerini barındırıyor olması. Böylece e-postamız "eminerol@eposta.com" adresinden yollanmış olacak ve kişi yanıtlamak istediğinde "cevap@eposta.com" e-posta adresi gösterilecek. Ayrıca e-postamızı "PHP/5.2.1" ile gönderdiğimizi belirtmiş olacağız. "5.2.1" yerine sunucunuzdaki PHP sürümü yazacaktır.

Mail Gönderme Örneği

```
<?
$headers  = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=iso-8859-9' . "\r\n";
$headers .= 'To: Alıcı 1 <alici1@eposta.com>, Alıcı 2
<alici2@eposta.com>' . "\r\n";
$headers .= 'From: Yollayan Kişiyim <yollayan@eposta.com>' . "\r\n";
$headers .= 'Reply-To: Yanit E-Postasi <yanit@eposta.com>' . "\r\n";
$headers .= 'X-Mailer: PHP/' . phpversion() . "\r\n";
$headers .= 'Cc: acikkopya@eposta.com' . "\r\n";
$headers .= 'Bcc: gizlikopya@eposta.com' . "\r\n";
$alici_eposta = 'alici1@eposta.com';
$eposta_konusu = 'Deneme HTML e-postası yolluyoruz';
$eposta_mesaji = '<b><i>Bu kalın ve italik olan bir HTML formatlı
yazıdır.</i></b>';
mail($alici_eposta, $eposta_konusu, $eposta_mesaji, $headers);
?>
```

Bu örnekte ise "Deneme HTML e-postası yolluyoruz" konulu ve içeriği "Bu kalın ve italik olan bir HTML formatlı yazıdır." olan e-postayı HTML formatlı ve "iso-8859-9" karakter setine uyumlu (Türkçe karakterler) olarak, "alici1@eposta.com" ve "alici2@eposta.com" adreslerine gönderdik. Ayrıca aynı e-postanın bir kopyasını görünür olarak "acikkopya@eposta.com" adresine ve bir diğer kopyasını da bu kez gizli olarak "gizlikopya@eposta.com" adresine gönderdik.

Dikkat edecek olursak yollayan kişiyi *Yollayan Kişiyim < yollayan@eposta.com>* tarzında yazdık. Yani başta isim ve sonrasında e-posta girdik. Yollanacak kişinin ismini başına yazabilir ve e-postasını ise <> karakterleri arasına tanımlayabiliriz.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
➢ Girdiğiniz ismin yanına “hoş geldiniz” yazan fonksiyonu oluşturunuz ve program içinde kullanınız	➢ Örneğin “Ali” ismini fonksiyona gönderdiniz. Ekranda ”hoş geldiniz Ali” yazacak
➢ İki sayıyı toplayan fonksiyonuz yazınız. Ve program içinde çağırıp kullanınız.	➢ Geriye değer döndüren fonksiyonları kullanabilirsiniz
➢ İki kenar uzunluğu girilen dikdörtgenin alanını hesaplatan fonksiyonu yazınız. Fakat kenar uzunlukları eksik girildiğinde varsayılan olarak 1 değerini almanızı sağlayınız	➢ Varsayılan değeri alan fonksiyonları kullanabilirsiniz
➢ Bugünün tarihini zaman fonksiyonlarını kullanarak yazdırınız	
➢ Sistem saatinden o anın saati zaman fonksiyonlarını kullanarak ekrana yazdırınız	
➢ 1 yıl 2 ay 4 gün öncesinin tarihini ekrana yazdırınız.	
➢ Kime gönderildiği kimden gönderildiği ve e-posta mesajını içeren maili gönderen programı yazınız	
➢ Bir önceki uygulamayı geliştirerek **From,Reply-To, X-Mailer, MIME-Version, Content-type, To, Cc, Bcc,** özelliklerini içeren mail fonksiyonunu kullanınız	

ÖLÇME VE DEĞERLENDİRME

Bu faaliyet kapsamında kazandığınız bilgileri, aşağıdaki soruları cevaplayarak belirleyiniz.

1. Aşağıdakilerden hangisi PHP fonksiyon oluşturma deyimidir?
A) function
B) static
C) fonksiyon
D) global

2. Aşağıdakilerden hangisi fonksiyonda değer döndüren deyimdir?
A) back
B) next
C) return
D) function

3. Sistemden tarih ve saat fonksiyonunu almaya yarayan ve geriye dizi döndüren fonksiyon nedir?
A) getdate()
B) mktime()
C) time()
D) mkdate()

4. Aşağıdakilerden hangisi mail gönderme fonksiyonudur?
A) mail()
B) email()
C) sendmail()
D) sendemail()

5. Yollanacak e-postanın HTML içerikli olmasını, gönderen e-posta adresinin görüntülenmesini, e-posta içeriğinin dil kodunun tanımlanması, e-posta Cc ve Bcc alıcılarının belirlenmesi için adlandırılan bilgilere ne denir?
A) BB B) CC C) headers D) From

DEĞERLENDİRME

Sorulara verdiğiniz cevaplar ile cevap anahtarını karşılaştırınız. Cevaplarınız doğru ise bir sonraki faaliyete geçiniz. Yanlış cevap verdiyseniz öğrenme faaliyetinin ilgili bölümüne dönerek konuyu tekrar ediniz.

ÖĞRENME FAALİYETİ–2

AMAÇ

Dizi işlemlerini gerçekleştirebileceksiniz.

ARAŞTIRMA

- Programlamada dizilerin kullanım amaçlarını araştırınız.
- Dizilerin boyutları hakkında bilgi ediniz.
- Dizilerde "indis" ifadesi neyi temsil eder araştırınız.

2. DİZİLER

Dizilerle ilgili örneklere geçmeden önce aşağıdaki tabloyu inceleyelim. Tabloda dizi kavramının mantığı açıklanmaktadır. Tablonun tamamını bir dizi olarak düşünün ve bu tablonun her bir sütununu da dizinin değişkeni olarak düşünün.

Öğrenci NO	Adı Soyadı	Sınav-1	Sınav-2	Ortalama
1	Emin EROL	..	..	..
2	Murat DİNÇ	..	..	..
3	Ali SAKA	..	..	..
4	Fatih ULUKAYA	..	..	..

2.1. Dizi Oluşturma

Dizileri oluşturmadaki temel fonksiyonumuz array() fonksiyonudur. Diziler içindeki elemanlara değer vermek amacıyla ise "=>" işareti kullanılır. Örneğin tablodaki verileri diziye aktarmak istersek şu php kodlarını kullanırız:

```
<?php
$ogrenciler = array (
array ( adi => "Emin , soyadi => " EROL", sinav1 =>"", sinav2 =>"", not =>""),
array ( adi =>  Murat", soyadi => "DİNÇ", sinav1 =>"", sinav2 =>"", not =>""),
array ( adi => " Ali ", soyadi => " SAKA", sinav1 =>"", sinav2 =>"", not =>""),
array ( adi => "Fatih", soyadi => "ULUKAYA", sinav1 =>"", sinav2 =>"", not =>""));
?>
```

Bu komut topluluğundaki $ogrenciler dizinin ismini adi, soyadi, sinav1, sinav2 ve not ifadeleri ise dizinin endekslerini belirtir. Burada dikkat edilmesi gereken noktalardan bir tanesi eğer endeks alanımız birden fazla kelimeden oluşuyorsa bunları çift tırnak içine almalıyız. Üstteki komutlara alternatif olarak şu komutları da kullanabiliriz:

```
<?php
    $ogrenciler[0][adi] = "Emin";
    $ogrenciler[0][soyadi] = "EROL";
    $ogrenciler[0][sinav1] = "";
    $ogrenciler[0][sinav2] = "";
    $ogrenciler[0][not] = "";

    // Buraya başka kodlar girecek
    print $ogrenciler[0][adi];
?>
```

Bir dizi değişkende kaç boyut olacaksa, o kadar iç içe array() öğesi oluşturabiliriz. Buna göre tek boyutlu bir dizi değişken sadece bir array() komutu ile ve sadece değerler verilerek oluşturulabilir. Diyelim ki yukarıdaki öğrenci listemiz sadece öğrencilerin isimlerinden oluşacak. Bu durumda $ogrenciler değişkenine ilişkin satırı şöyle yazabilirdik:

```
$ogrenciler = array ("Emin", "Murat", "Ali", "Fatih");
```

PHP, böyle tek boyutlu bir dizinin örneğin birinci elemanını, "$ogrenciler[0]" adıyla bilir. Böyle bir tek-boyutlu diziyi oluşturmak için PHP bize başka bir kolaylık da sağlar: array() komutunu kullanmadan, doğruca dizinin öğelerine değer vermemiz mümkündür.

```
<?php
    $ogrenciler[] = "Emin";
    $ogrenciler[] = "Murat";
    $ogrenciler[] = "Ali";
    $ogrenciler[] = "Fatih";
    print $ogrenciler[0];
?>
```

Böyle sırayla dizi değişken oluşturur veya oluşturulmuş bir dizi değişkene ek yaparken, değişkenin sıra numarasını yazmazsak, PHP bunları kendisi sıralar. Yukarıdaki kodun da Browser penceresine "Emin" yazdırması gerekir. Mevcut tek-boyutlu bir dizi değişkene ek yaptığımızda, yeni değerin dizinin en altına eklenmesini istiyorsak, sıra numarası yazmamıza gerek yoktur. Mevcut değerlerden birini değiştirmek istiyorsak, o değerin sıra numarasını yazmamız gerekir. Bunu denemek için yukarıdaki kodu şöyle değiştirelim:

```
<?php
    $ogrenciler[] = "Emin";
    $ogrenciler[] = "Murat";
    $ogrenciler[] = "Ali";
    $ogrenciler[] = "Fatih";
        // Buraya başka kodlar girecek
    $ogrenciler[0] = "Volkan";
    $ogrenciler[15] = "Basri";
```

```
    print ("Dizideki 1'nci isim: $ogrenciler[0] <br>");
    print ("Dizideki 2'nci isim: $ogrenciler[1] <br>");
    print ("Dizideki 3'ünci isim: $ogrenciler[2] <br>");
    print ("Dizideki 4'üncü isim: $ogrenciler[3] <br>");
    print ("Dizideki 5'inci isim: $ogrenciler[4] <br>");
    print ("Dizideki 6'ncı isim: $ogrenciler[5] <br>");

    print ("Dizideki 16'ncı isim: $ogrenciler[15] <br>");
?>
```

Bu programın Browser penceresine göndereceği sırada, birinci öğrenci ($ogrenci[0]) olarak bu kez Emin değil Volkan yazdığını göreceğiz. Bunun sebebi, diziyi oluşturan ilk grup deyimden sonra,

```
$ogrenciler[0] = "Volkan";
```

satırı ile birinci elemanın değerini değiştirmiş olduk. 15'nci elemana atama yapmakla, PHP'nin $ogrenciler dizisinde 6, 7, 8, 9,.. 14'e kadar boş elemanlar oluşturmasına sebep olduk.

2.1.1. Foreach() Fonksiyonu

Dizileri programlarınızda kullanırken siz en büyük kolaylık verecek fonksiyon foreach() fonksiyonudur. Özellikle veritabanı uygulamalarında for döngüsü yerine kullanılan foreach fonksiyonu dizinin eleman sayısını kendisi otomatik algılar. Böylece sizin herhangi bir for döngüsü ve sayaç kullanmanıza gerek kalmaz. Aşağıdaki örneği dikkatle inceleyiniz.

<?php $ogrenciler[] = "Emin"; $ogrenciler[] = "Murat"; $ogrenciler[] = "Ali"; $ogrenciler[] = "Fatih"; foreach ($ogrenciler as $anahtar=>$deger) { print ("$anahtar = $deger "); } ?>	0 = Emin 1 = Murat 2 = Ali 3 = Fatih

2.2. Dizi Düzenleme

Dizileri program içinde kullanırken çeşitli düzenlemelere ihtiyacınız olabilir. Bunlar sıralama, birleştirme, ekleme, silme vb işlemler olabilir. Şimdi bu işlemleri sırayla inceleyelim:

2.2.1. Dizileri Sıralama

Bir dizinin içindeki değerleri alfabetik veya küçükten büyüğe doğru sıralamak için sort() fonksiyonunu kullanırız.

Örnek Kod
`$birinci_dizi=array("Özbay","Muharrem","Hasan","Şahika","Altun", "Taç","Civelek","Tabak");` sort($birinci_dizi);

PHP, dizideki bütün değerleri A'dan Z'ye sıraya sokacaktır. sort() fonksiyonu dizinin içeriğini değiştirir. Bir diziyi Z'den A'ya veya büyükten küçüğe doğru sıralamak için de rsort() fonksiyonunu kullanabilirsiniz. Bir noktada dikkatli olmak gerekir. bu fonksiyonu ilişkili (değerlerin anahtarı olarak endeks adı bulunan) dizide kullanırsanız, PHP, anahtar değerlerini (endeks adlarını) atar, yerine 0'dan itibaren rakam koyar. Bunu önlemek için, ilişkili dizileri asort() veya ksort() fonksiyonu ile sıralamak gerekir.

2.2.2. İlişkili Dizileri Sıralama

İlişkili dizilerin diğer dizi değişkenlere göre farkı, değerlerinin bir de adı bulunmasıdır. Değerlerin adlarına anahtar denir. Bir ilişkili diziyi değerlerine göre sıralamak için asort() fonksiyonu kullanılır.

Örnek Kod
$birinci_dizi=array(ogr_01=>"Özbay",ogr_02=>"Muharrem",ogr_013>"Hasan ",ogr_04=>"Şahika"); asort($birinci_dizi);

PHP, bu diziyi değerler itibariyle alfabetik sıraya sokacaktır. Eğer sıranın değere göre değil de değerlerin anahtarına (burada ogr_01, ogr_02 olan kelimeler) göre yapılmasını istiyorsak, ksort() fonksiyonunu kullanırız.

Örnek Kod
$birinci_dizi = array(ogr_01=>"Özbay", ogr_02=>"Muharrem", ogr_013>"Hasan", ogr_04=>"Şahika"); ksort($birinci_dizi);

PHP, bu diziyi anahtarlara göre alfabetik sıraya sokacaktır.

2.2.3. Dizilere Değişken Ekleme

Bir diziye yeni değişkenler eklemek için, array_push() fonksiyonuna mevcut dizinin adını ve yeni değerleri yazarız.

Örnek Kod
`$birinci_dizi = array ( "Özbay" , "Muharrem" , "Hasan" , "Şahika" );` `$yeni = array_push ($birinci_dizi, "Altun" , "Taç" , "Civelek" , "Tabak" );`

Burada $yeni adlı değişken sadece $birinci_dizi adlı dizinin yeni eleman sayısını tutar. array_push(), kendisine adını verdiğimiz dizinin içeriğini değiştirir. Yukarıdaki örnekte içine yeni değerler yazılan dizinin elemanlarını görüntülemek için şöyle bir kod yazabiliriz:

Örnek Kod
`print ("\$birinci_dizi adlı dizide $yeni_dizi adet değişken var ");` `foreach ( $birinci_dizi as $ogrenci )` `{` `  print ("$ogrenci   ");` `}`

2.2.4. Dizileri Birleştirme

İki veya daha fazla dizinin bütün elemanlarını birleştirerek, ortaya yeni bir dizi çıkartır.

Örnek Kod
`$birinci_dizi = array ( "Özbay" , "Muharrem" , "Hasan" , "Şahika" );` `$ikinci_dizi = array ( "Altun" , "Taç" , "Civelek" , "Tabak" );` `$yeni_dizi = array_merge ( $birinci_dizi, $ikinci_dizi );` Bu kod ile oluşturulan $yeni_dizi isimli dizi değişkenin hangi elemanlara sahip olduğunu, şöyle bir kodla görebilirsiniz: `foreach ( $yeni_dizi as $yeni_eleman )` `{` `  print (" $yeni_eleman  ");` `}`

İkinci dizinin bütün elemanları, birinci dizinin elemanlarının arkasına eklenmiş olmalı. array_merge() işlemi, çok boyutlu ilişkili dizilere de uygulanabilir; PHP iki dizideki uyumlu-uyumsuz, yani birinde olan diğerinde olmayan bütün anahtar+değer çiftlerini yeni dizide de oluştur. (array_merge() işleminden sonra birleştirilen dizilerin değişmeden kaldığına dikkat edin.)

2.2.5. Dizinin İlk Elemanını Silme

Bir dizi-değişkenin ilk elemanını tümüyle silmek için array_shift() fonksiyonunu kullanırız. Bu fonksiyona sadece birinci elemanı silinecek dizinin adını vermek yeter.

Örnek Kod

```
$birinci_dizi = array("Özbay","Muharrem","Hasan","Şahika");
$silinen = array_shift($birinci_dizi);
```

array_shift(), adını verdiğiniz dizinin içeriğini değiştirir; buradaki örnekte, $silinen adlı değişken dizinin silinen birinci elemanın değerini tutar.

2.2.6. Diziden Kesit Alma

Bir dizi-değişkenin bütün elemanları yerine bir kesitini kullanmak istiyorsak, bunu array_slice() fonksiyonu ile yapabiliriz. Bu fonksiyona kesit alınacak dizinin adı, kesitin başladığı yer ve kaç adet değişken alınacağı argüman olarak verilir.

Örnek Kod

```
$birinci_dizi=array("Özbay","Muharrem","Hasan","Şahika","Altun","T
aç","Civelek","Tabak");
$kesit=array_slice($birinci_dizi,3,4);
```

Burada, PHP'ye $kesit adlı yeni dizi değişkene, $birinci_dizi adlı dizinin 3'ncü değerinden itibaren (3 dâhil) dört değeri yerleştirmesini bildiriyoruz. array_slice(), adını verdiğimiz değişkenin içeriğine dokunmaz; yeni dizi değişken oluşturulur.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
➢ 5 arkadaşınızın isimlerini içeren "ogrenci" adında bir dizi oluşturunuz.	➢ Tek boyutlu dizi oluşturabilirsiniz
➢ Daha sonra dizideki elemanların değerlerini ekrana yazdırınız.	➢ Foreeach fonksiyonunu kullanabilirsiniz
➢ Oluşturduğunuz diziyi alfabetik olarak sıralayınız ve ekrana yazdırınız.	➢ Sort ( ) fonksiyonunu kullanabilirsiniz
➢ Bu diziye iki başka iki arkadaşınızın ismini de ekleyiniz.	➢ array_push() fonksiyonunu kullanabilirsiniz
➢ 3 öğretmeninizin isminden oluşan "ogretmen" adında bir dizi oluşturunuz.	➢ Tek boyutlu dizi oluşturabilirsiniz
➢ "Ogrenci" ve "ogretmen" isimli dizileri birleştirerek "okul" isminde bir dizi oluşturunuz.	➢ array_merge() fonksiyonunu kullanabilirsiniz
➢ Oluşturduğunuz "okul" isimli dizinin ilk elemanını siliniz.	➢ array_shift() fonksiyonunu kullanabilirsiniz
➢ Okul isimli dizinin içinden 3. ve 4. elemanları alarak "sınıf" isimli bir dizi oluşturunuz.	➢ array_slice() fonksiyonunu kullanabilirsiniz

ÖLÇME VE DEĞERLENDİRME

Bu faaliyet kapsamında kazandığınız bilgileri, aşağıdaki soruları cevaplayarak belirleyiniz.

1. Aşağıdakilerden hangisi dizi sıralama fonksiyonudur?
A) array()
B) sort ()
C) array_slice ()
D)array_shift()

2. Aşağıdakilerden hangisi dizilere eleman ekleme fonksiyonudur?
A) foreeach()
B) array_slice()
C) array_shift ()
D) array_push()

3. Aşağıdakilerden hangisi dizinin ilk elemanını silme fonksiyonudur?
A) foreeach()
B) array_push()
C) array_shift ()
D) array_slice()

4. Aşağıdakilerden hangisi dizilerde kesit alma fonksiyonudur?
A) foreach ()
B) array_push()
C) array_shift ()
D) array_slice()

5. Aşağıdakilerden hangisi dizileri birleştirme fonksiyonudur?
A)array_shift ()
B) array_push()
C) array_merge()
D) array_slice()

Sorulara verdiğiniz cevaplar ile cevap anahtarını karşılaştırınız. Cevaplarınız doğru ise bir sonraki faaliyete geçiniz. Yanlış cevap verdiyseniz öğrenme faaliyetinin ilgili bölümüne dönerek konuyu tekrar ediniz.

MODÜL DEĞERLENDİRME

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

1. ( )PHP'de fonksiyonlar function komutu ile oluşturulur.
2. ( )Fonksiyondan geriye değer return ifadesiyle döndürülür.
3. ( )Time() fonksiyonu sadece bir parametre alır.
4. ( )Mail gönderme fonksiyonu sendmail() dir.
5. ( )foreach fonksiyonu dizinin eleman sayısını kendisi otomatik algılar.
6. ( )Bir dizi değişkenin ilk elemanını tümüyle silmek için array_shift() fonksiyonunu kullanırız.
7. ( )Bir dizi değişkenin bütün elemanları yerine bir kesitini kullanmak istiyorsak, bunu array_sort() fonksiyonu ile yapabiliriz.

KONTROL LİSTESİ

Değerlendirme Ölçütleri	Evet	Hayır
➢ Girdiğiniz ismin yanına "hoş geldiniz" yazan fonksiyonu oluşturup ve program içinde kullandınız mı?		
➢ İki sayıyı toplayan fonksiyonuz yazıp, program içinde çağırıp kullandınız mı?.		
➢ İki kenar uzunluğu girilen dikdörtgenin alanını hesaplatan fonksiyonu yazdınız mı?(Kenar uzunlukları eksik girildiğinde varsayılan olarak 1 değerini almasını sağladınız mı?)		
➢ Bugünün tarihini zaman fonksiyonlarını kullanarak yazdınız mı?		
➢ Sistem saatinden o anın saati zaman fonksiyonlarını kullanarak ekrana yazdınız mı?		
➢ 1 yıl 2 ay 4 gün öncesinin tarihini ekrana yazdınız mı?		
➢ Kime gönderildiği, kimden gönderildiği ve e-posta mesajını içeren maili gönderen programı yazdınız mı?		
➢ .Bir önceki uygulamayı geliştirerek **From, Reply-To, X-Mailer, MIME-Version, Content-type, To, Cc, Bcc,** özelliklerini içeren mail fonksiyonunu kullandınız mı?		
➢ 5 arkadaşınızın isimlerini içeren "ogrenci" adında bir dizi oluşturdunuz mu?		
➢ Daha sonra dizideki elemanların değerlerini ekrana yazdırdınız mı?		
➢ Oluşturduğunuz diziyi alfabetik olarak sıralayıp, ekrana yazdırdınız mı?		
➢ Bu diziye başka iki arkadaşınızın ismini de eklediniz mi?		
➢ 3 öğretmeninizin isminden oluşan "ogretmen" adında bir dizi oluşturdunuz mu?		
➢ "Ogrenci" ve "ogretmen" isimli dizileri birleştirerek "okul" isminde bir dizi oluşturdunuz mu?		
➢ Oluşturduğunuz "okul" isimli dizinin ilk elemanını sildiniz mi?		
➢ Okul isimli dizinin içinden 3. ve 4. elemanlarını alarak "sınıf" isimli bir dizi oluşturdunuz mu?		

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ -1'İN CEVAP ANAHTARI

1	A
2	C
3	A
4	A
5	C

ÖĞRENME FAALİYETİ -2'NİN CEVAP ANAHTARI

1	B
2	D
3	C
4	D
5	C

MODÜL DEĞERLENDİRME CEVAP ANAHTARI

1	Doğru
2	Doğru
3	Yanlış
4	Yanlış
5	Doğru
6	Doğru
7	Yanlış

KAYNAKÇA

- www.php.org 10.12.2012 14:23