

Diziler

İçerisinde tek bir değer depolayan değişkenler *skaler (scalar)* değişkenler olarak adlandırılırken içerisinde değerler grubu ya da serisi barındıran değişkenler *dizi (array)* olarak adlandırılmaktadır. Bir dizinin birçok elemanı olabilmektedir. Her eleman, metinsel ya da sayısal bir tek değer barındırabileceği gibi başka biri dizi de depolayabilir. Başka diziler depolayan biz diziye *çok boyutlu dizi* adı verilir.

Bir dizi array() fonksiyonunu kullanılarak belirtilebilir. Eleman olarak, birbirlerinden virgüllerle ayrılmış çok sayıda anahtar => değer çifti alabilir.

```
array(  
    anahtar1 => değer1,  
    anahtar2 => değer2,  
    anahtar3 => değer3,  
    ...  
)
```

Son dizi elemanından sonraki virgül seçimlidir ve konmayabilir. Bu genellikle tek satırlık dizilerde yapılır. Yani, array(1, 2,) yerine array(1, 2) yazmak tercih edilir. Diğer taraftan, çok satırlı dizilerde yeni elemanların sona eklenmesini kolaylaştırdığından sondaki virgölün konulması tercih edilir.

Bir anahtar ya *integer* ya da *string* türünde olabilir. Bir değer ise herhangi bir türde olabilir.

Sayısal Anahtarlı Diziler

Sayısal anahtarlı dizilerde bir sayı ile anahtar belirtmeye her zaman gerek yoktur. Aşağıdaki gibi bir tanım da sayısal anahtarlı bir dizi olarak kabul edilir ve varsayılan olarak ilk anahtar numarası 0 olarak atanır.

```
1. <?php  
2.  
3.   $arabalar = array("Volvo", "BMW", "Toyota");  
4.  
5.   $arabalar = array(0 => "Volvo", 1 => "BMW", 2=> "Toyota");  
6.  
7.   $arabalar[0] = "Volvo";  
8.   $arabalar[1] = "BMW";  
9.   $arabalar[2] = "Toyota";  
10.  
11. ?>
```

Yukarıdaki örnekte 3 farklı atama yöntemi ile değerler arabalar dizi değişkenine atanmıştır.

BELTEK-04-3-970 | İnternet Programcılığı [PHP&MySQL]

Hafta 4 – Ders 1 | Diziler

Sayısal anahtarlı dizilerde dizi elemanlarını görüntülemek için `print_r()`, `foreach()` vb. fonksiyonlara başvurulabilir.

```
1. <?php
2.
3.     $sarabalar = array("Volvo", "BMW", "Toyota");
4.
5.     print_r($sarabalar); // Dizi elemanlarını anahtarları ile beraber görüntüleyebiliriz
6.
7.     foreach($sarabalar as $saraba){ // Bu döngü eleman sayısı kadar döndürülür
8.         echo "$saraba<br>"; // Her bir döngüde dizideki bir eleman ekrana yazdırılır
9.     }
10.
11.     $selemanSayisi = count($sarabalar); // Dizideki eleman sayısı bir değişkene atanır
12.
13.     for($i=0; $i<$selemanSayisi; $i++){ // For döngüsü ile bizim anahtar değerleri döndürülebilir
14.         echo "$sarabalar[$i]<br>";
15.     }
16.
17. ?>
```

Yukarıdaki örnekte 3 farklı fonksiyon ile (`print_r`, `foreach`, `for`) bir dizideki elemanlar ekran çıktısı olarak görüntülemiştir.

Metinsel Anahtarlı Diziler

PHP'de bir dizi varsayılan olarak sayısal anahtarlı tanımlandığı için eğer metinsel anahtarlı bir diziyi ihtiyacınız varsa bu anahtarları mutlaka eleman atama esnasında belirtmeniz gerekir.

```
1. <?php
2.
3.     $yas = array("Ahmet"=>35, "Beyza"=>30, "Mehmet"=>20, "Funda"=>25);
4.
5.     $yas["Ahmet"] = 35;
6.     $yas["Beyza"] = 30;
7.     $yas["Mehmet"] = 20;
8.     $yas["Funda"] = 25;
9.
10. ?>
```

Yukarıdaki örnekte 2 farklı atama yöntemi ile değerler `$yas` dizi değişkenine isim anahtarlarıyla birlikte atanmıştır.

Metinsel anahtarlı dizilerin elemanlarını ekrana yazdırmak için sayısal anahtarlı dizilerde olduğu gibi `print_r()`, `foreach()` vb. fonksiyonlara başvurulabilir.

```
1. <?php
2.
3.     $yas = array("Ahmet"=>35, "Beyza"=>30, "Mehmet"=>20, "Funda"=>25);
4.
5.     print_r($yas); // Bu fonksiyon ile dizi elemanları anahtarları ile beraber ekrana yazdırılacaktır
6.
7.     foreach($yas as $deger){ // Bu döngünün çıktısında sadece yaş değerleri ekrana yazdırılacaktır
8.         echo $deger."<br>";
9.     }
10.
11.     foreach($yas as $isim => $deger) { // Bu döngünün çıktısında anahtar-eleman ikilisi ekrana yazdırılacaktır
12.         echo "İsim= $isim - Yaşı= $deger<br>"; // Beklenen ekran çıktılarından biri -> İsim= Funda - Yaşı= 25
13.     }
14.
15. ?>
```

Yukarıdaki örnekte dizi değişkenindeki elemanlar anahtarlarıyla birlikte ya da sadece değerleriyle ekrana yazdırılacaklardır.

Çok Boyutlu Diziler

Çok boyutlu bir dizi, içerisinde bir ya da birden çok dizi barındıran dizilerdir. Aşağıdaki örnekte bilgiler adında bir diziye içerisinde 3 eleman barındıran 4 tane dizi atanmaktadır.

```
1. <?php
2.
3.     $bilgiler = array(
4.         array("Ahmet", 35, "İstanbul"),
5.         array("Beyza", 30, "İzmir"),
6.         array("Mehmet", 20, "Antalya"),
7.         array("Funda", 25, "Ankara")
8.     );
9.
10. ?>
```

Çok boyutlu dizilerin elemanlarını ekrana yazdırmak için sayısal anahtarlı ya da metinsel anahtarlı dizilerde olduğu gibi print_r(), foreach() vb. fonksiyonlara başvurulabilir. Ancak çok boyutlu dizilerde, herhangi bir anahtarın içerisindeki belirli bir elemana doğrudan ulaşmak isteniyorsa o zaman o elemanın konumunu belirten özel bir anahtar ikilisi -dizinin boyunu göre anahtar sayısı artırılmalıdır.

Yukarıdaki örnekte bilgiler dizisinin son anahtarındaki kayıttaki yer alan yaş bilgisine ulaşmak isteniyorsa o zaman şöyle bir kod satırı yazılmalıdır.

```
echo $bilgiler[3][1];
```

BELTEK-04-3-970 | İnternet Programcılığı [PHP&MySQL]

Hafta 4 – Ders 1 | Diziler

Ş bilgiler[3][1] -> önce 3. anahtarda bulunan diziyeye gidecek sonrasında da o dizinin 1. anahtarını ekrana yazdıracaktır. Bu dizinin varsayılan başlangıç anahtarı 0 olduğu için beklenen ekran çıktısı 25 olacaktır.

```
1. <?php
2.
3.   $bilgiler = array(
4.       array("Ahmet", 35, "İstanbul"),
5.       array("Beyza", 30, "İzmir"),
6.       array("Mehmet", 20, "Antalya"),
7.       array("Funda", 25, "Ankara")
8.   );
9.
10.  print_r($bilgiler); // Dizideki tüm elemanlar anahtarları ile beraber ekrana yazdırılacaktır
11.
12.  print_r($bilgiler[0]); // Dizi içerisindeki ilk dizideki tüm elemanlar anahtarları ile beraber ekrana
    yazdırılacaktır
13.
14.  print_r($bilgiler[3]); // Dizi içerisindeki son dizideki tüm elemanlar anahtarları ile beraber ekrana
    yazdırılacaktır
15.
16.  echo "<br>";
17.
18.  foreach($bilgiler as $bilgi){ // Dizi içerisindeki her bir eleman $bilgi değişkenine aktarılacaktır
19.      echo "$bilgi<br>"; // $bilgi değişkeninde yine bir dizi olacağı için echo komutu ile bu elemanları
    yazdırmada hata alınacaktır
20.  }
21.
22.  foreach($bilgiler as $bilgi){ // Dizi içerisindeki her bir eleman $bilgi değişkenine aktarılacaktır
23.      foreach($bilgi as $yeniBilgi){ // $bilgi dizi değişkenindeki her bir eleman $yeniBilgi değişkenine
    aktarılacaktır
24.          echo "$yeniBilgi<br>"; // Dizi içerisindeki her eleman tek tek yeni satırda olmak üzere ekrana
    yazdırılacaktır
25.      }
26.  }
27.
28.  echo $bilgiler[0][0]; // Dizinin ilk dizisinin ilk elemanı -> Ahmet
29.
30.  echo $bilgiler[0][1]; // Dizinin ilk dizisinin ikinci elemanı -> 35
31.
32.  echo $bilgiler[0][2]; // Dizinin ilk dizisinin üçüncü elemanı -> İstanbul
33.
34.  echo $bilgiler[3][2]; // Dizinin dördüncü dizisinin üçüncü elemanı -> Ankara
35.
36. ?>
```

Yukarıdaki örnekte oluşturulan biz çoklu dizinin elemanlarına ulaşmada kullanılan farklı yöntemler listelenmiştir.

Dizilerde Sıralama

Fonksiyonlar varsayılan olarak kendi kapsamlarında tanımlanan değişkenleri ya da parametre ile gelen değişkenleri tanırlar. Bu kapsamlar dışında tanımlanan değişkenler ise tanınmazlar. Aynı zamanda kendi kapsamında tanımlanan değişkenleri de fonksiyon dışında kullanırlamazlar.

Fonksiyon	Görev
sort()	Dizi elemanlarını alfabetik olarak sıralar
rsort()	Dizi elemanlarını ters yönde alfabetik olarak sıralar
asort()	Dizi elemanlarını değerlerine göre alfabetik olarak sıralar
ksort()	Dizi elemanlarını anahtarlarına göre alfabetik olarak sıralar
arsort()	Dizi elemanlarını değerlerine göre ters yönde alfabetik olarak sıralar
krsort()	Dizi elemanlarını anahtarlarına göre ters yönde alfabetik olarak sıralar
shuffle()	Dizi elemanlarını rastgele karıştırarak sıralar
array_reverse()	Dizi elemanlarını ters olarak sıralar

Kaynaklar

PHP Kılavuzu - <http://php.net/manual/tr/index.php>

PHP Tutorials - <https://www.w3schools.com/php/default.asp>

Thomson, L. & Welling, L. (2016) *PHP and MySQL Web Development* (5th ed. pp. 75-100). Addison-Wesley Professional.