

# Diziler

- Dizi, bellekte aynı isim altında toplanmış deęişkenler kümesidir.
- Eęer ki aynı türden veriler grubunu bellekte tutmak gerekirse diziler kullanılır.
- Örneęin verilerin sıralanması, ortalamasının bulunması gibi işlemlerde dizilere ihtiyaç duyulur.

# Diziler

- Aynı türden bilgilerin bir değişkenin içerisinde tutulmasıyla oluşturulur.
- Bir değişkenin dizi olup olmadığı "[" "]" simgelerinden anlaşılabilir.
- Kullanılış biçimi :

`$degisken = array(deger1, deger2,...,degerN)` veya

`$degisken[indis_no] = deger`

şeklindedir.

# Diziler

- Bir dizinin n. inci elemanına ulaşmak için dizinin adından sonra köşeli parantez içinde dizi elemanının indis numarası yazılır.
- Örnek :
- `$degisken[5] = "Ali";`
- Bu örnekte dizinin 6. elemanına "Ali" değeri atanmıştır. Eğer indis numarası yazılmazsa ilk eleman olan sıfırcı sıradaki elemana ulaşılır. Daha sonra indis numarası verilmeden ulaşılan eleman bir sonraki eleman olacaktır.
- Örnek :
- `print ($degisken[]); // Dizinin sıfırcı elemanı yazılır`
- `print ($degisken[1]); // Dizinin birinci elemanı yazılır`

# Diziler

- Diziler tanımlanırken array komutu kullanılmayabilir.
- `$liste = (deger1,deger2,...);` veya
- `$liste[] = deger1;`
- `$liste[] = deger2;`
- `$liste[] = degerN;`
- Diziler tanımlanırken indis numarası yerine isim verilebilir.
- `$liste[isim1] = deger1;`
- `$liste[isim2] = deger1;`

# Çok boyutlu diziler

- Dizi içindeki diziler olarak düşünülebilir. Yine bir örnekle açıklamak gerekirse bir otelin katları bir dizi olarak düşünülürse katlardaki odalar da bir dizi olarak düşünülebilir. Dolayısıyla burada dizi içindeki diziler söz konusudur.

# Çok boyutlu diziler

```
<?php
```

```
$otel["kat1"]["oda1"]="Ali";  
$otel["kat1"]["oda2"]="Ahmet";  
$otel["kat1"]["oda2"]="Ömer";
```

```
$otel["kat2"]["oda1"]="Hasan";  
$otel["kat2"]["oda2"]="Kenan";  
$otel["kat2"]["oda3"]="Mert";
```

```
$otel["kat3"]["oda1"]="Ayşe";  
$otel["kat3"]["oda2"]="Fatma";  
$otel["kat3"]["oda3"]="Hatice";
```

```
echo $otel["kat2"]["oda3"]; // ekrana Mert yazar  
?>
```

Otel		
Kat1	Oda1	Ali
	Oda2	Ahmet
	Oda3	Ömer
Kat2	Oda1	Hasan
	Oda2	Kenan
	Oda3	Mert
Kat3	Oda1	Ayşe
	Oda2	Fatma
	Oda3	Hatice

# Çok boyutlu diziler

```
<?php
$otel=array(
    "kat1"=>array("oda1"=>"Ali","oda2"=>"Ahmet","oda3"=>"Ömer"),
    "kat2"=>array("oda1"=>"Hasan","oda2"=>"Kenan","oda3"=>"Mert"),
    "kat3"=>array("oda1"=>"Ayşe","oda2"=>"Fatma","oda3"=>"Hatice")
);

echo $otel["kat2"]["oda3"]; // ekrana Mert yazar
?>
```

# Çok boyutlu diziler

- ```
$a = array();  
$a['renk'] = 'kırmızı';  
$a['lezzet'] = 'tatlı';  
$a['şekil'] = 'yuvarlak';  
$a['isim'] = 'elma';
```
- Yerine;
- ```
$a = array( 'renk' => 'kırmızı',  
           'lezzet' => 'tatlı',  
           'şekil' => 'yuvarlak',  
           'isim' => 'elma',  
           );
```



# Çok boyutlu diziler

- **Örnek:** Yandaki tabloda 3 farklı sınıfın 4'er öğrencisinin gösterildiğini kabul ederek bunu bir dizi olarak tanımlayalım.

Okul	
Birinci sınıf	Ebru
	Fuat
	Gülüzar
	Emre
İkinci sınıf	Zeynep
	Haşim
	Merve
	Nihat
Üçüncü sınıf	Raşit
	Gizem
	Onur
	Ayşe

# Çok boyutlu diziler

```
<?php
```

```
$okul[0][]="Ebru";  
$okul[0][]="Fuat";  
$okul[0][]="Gülüzar";  
$okul[0][]="Emre";
```

```
$okul[1][]="Zeynep";  
$okul[1][]="Haşim";  
$okul[1][]="Merve";  
$okul[1][]="Nihat";
```

```
$okul[2][]="Raşit";  
$okul[2][]="Gizem";  
$okul[2][]="Onur";  
$okul[2][]="Ayşe";
```

```
echo $okul[1][1]; // ekrana Haşim yazar  
?>
```

# Çok boyutlu diziler

- Aynı diziyi farklı bir şekilde tanımlayalım.

```
<?php
```

```
$okul[]=array("Ebru","Fuat","Gülüzar","Emre");
```

```
$okul[]=array("Zeynep","Haşim","Merve","Nihat");
```

```
$okul[]=array("Raşit","Gizem","Onur","Ayşe");
```

```
echo $okul[1][2]; // ekrana Merve yazar
```

```
?>
```

Dikkat ederseniz burada hiçbir indis belirtilmemiştir. Dolayısıyla her dizi elemanı dizinin sonuna eklenecektir. Böylece her iki boyutun indisleri o dan başlayacaktır

# Çok boyutlu diziler

- Eğer indislerin 1 den başlamasını isterseniz şöyle bir tanımlama yapabilirsiniz.

```
<?php
```

```
$okul[1]=array(1=>"Ebru","Fuat","Gülüzar","Emre");
```

```
$okul[]=array(1=>"Zeynep","Haşim","Merve","Nihat");
```

```
$okul[]=array(1=>"Raşit","Gizem","Onur","Ayşe");
```

```
echo $okul[1][2]; // ekrana Fuat yazar
```

```
?>
```

# Çok boyutlu diziler

- Şimdi de aynı diziyi iki farklı şekilde daha tanımlayalım. Aşağıdaki tanımlamada her iki boyutun indisleri o dan başlar.

```
<?php
$okul=array(
    array("Ebru","Fuat","Gülüzar","Emre"),
    array("Zeynep","Haşim","Merve","Nihat"),
    array("Raşit","Gizem","Onur","Ayşe")
);
```

```
echo $okul[1][2]; // ekrana Merve yazar
?>
```

# Çok boyutlu diziler

- Aşağıdaki tanımlamada ise her iki boyutun indisleri 1 den başlar.

```
<?php
$okul=array(
    1=>array(1=>"Ebru","Fuat","Gülüzar","Emre"),
    array(1=>"Zeynep","Haşim","Merve","Nihat"),
    array(1=>"Raşit","Gizem","Onur","Ayşe")
);
```

```
echo $okul[1][2]; // ekrana Fuat yazar
?>
```

# Dizilerde Kullanılan Fonksiyonlar

**count():** Bir dizideki eleman sayısını döndürür.

Kullanılış biçimi :

```
count($dizi_adi);
```

Örnek :

```
<?php  
$a[0] = 1;  
$a[1] = 3;  
$a[2] = 5;  
$sonuç = count($a);  
// $sonuç == 3  
?>
```

# Dizilerde Kullanılan Fonksiyonlar

**is\_array():** Dizinin elemanına ilk ulaştığımızda değer olup olmadığını araştırır. Başka bir ifadeyle değişkenin dizi olup olmadığını kontrolünü yapar.

Kullanılış biçimi :

```
is_array( $dizi_adi );
```

Örnek :

- ```
<?php
$dizi = array('bu', 'bir', 'dizidir');

echo is_array($dizi) ? 'Dizi' : 'Dizi değil';

?>
```



# Dizilerde Kullanılan Fonksiyonlar

- **Foreach:** döngü kullanmadan dizi elemanlarına ulaşılmasını sağlar.

- foreach (dizi\_ifadesi as \$değer) deyim

- **<?php**

```
$dizi = array(1, 2, 3, 4);  
foreach ($dizi as $deger) {  
    echo "Değer: $deger<br />";  
}  
?>
```

# Dizilerde Kullanılan Fonksiyonlar

**array\_merge():** İki dizinin elamanlarını birleştirerek yeni bir dizi oluşturur.

Kullanış biçimi:

```
$yeni_dizi = array_merge ( $ilk_dizi, $ikinci_dizi );
```

Örnek :

- ```
<?php
$ilk = "Ankara", "İstanbul";
$son = "İzmir", "Adana";
$sonuc = array_merge($ilk, $son);
print_r($sonuc);
?>
```

```
Çıktısı:
Array (
    [0] => Ankara
    [1] => İstanbul
    [2] => İzmir
    [3] => Adana
)
```

# Dizilerde Kullanılan Fonksiyonlar

- **array\_push()**: Mevcut dizinin sonuna yeni değer ekler.

- ```
<?php
$küme = array("elma", "armut");
array_push($küme, "muz", "portakal");
print_r($küme);
?>
```

```
Çıktısı:
Array (
    [0] => elma
    [1] => armut
    [2] => muz
    [3] => portakal
)
```

- Not: print\_r — Ekrana bir değişkenin düzgün bir şekilde okunabilecek gösterimini basar.

# Dizilerde Kullanılan Fonksiyonlar

- **array\_shift():** Dizinin ilk elemanını siler.

- ```
<?php
$harfler = array("a", "b", "c", "d");
array_shift($harfler);
print_r($harfler);
?>
```

```
Çıktısı:
Array (
    [0] => b
    [1] => c
    [2] => d
)
```

# Dizilerde Kullanılan Fonksiyonlar

- **array\_unshift()**: Dizinin ilk başına yeni değer ekler.

- `<?php`

```
$harfler = array("a", "b", "c", "d");
```

```
array_unshift($harfler,"e","f");
```

```
print_r($harfler);
```

```
?>
```

Çıktısı:

Array (

[0] => e

[1] => f

[2] => a

[3] => b

[4] => c

[5] => d )

# Dizilerde Kullanılan Fonksiyonlar

- **array\_sum()**: Dizinin değerlerinin toplamını bulur.

```
<?php
$a = array(2, 4, 6, 8);
echo "sum(a) = " . array_sum($a) . "\n";

$b = array("a" => 1.2, "b" => 2.3, "c" => 3.4);
echo "sum(b) = " . array_sum($b) . "\n";
?>
```

Çıktısı:  
sum(a) = 20  
sum(b) = 6.9

# Dizilerde Kullanılan Fonksiyonlar

- **in\_array**: Bir verinin dizi içerisinde olup olmadığını kontrol eder. Varsa TRUE yoksa FALSE değeri döndürür.

- `<?php`

```
$os = array("Mac", "NT", "Irix", "Linux");
```

```
if (in_array("Irix", $os)) {  
    echo "Irix var";  
}
```

```
}
```

```
if (in_array("mac", $os)) {  
    echo "Mac var";  
}
```

```
}
```

```
?>
```

# Dizilerde Kullanılan Fonksiyonlar

- **array\_search()**: Bir dizi içerisinde arama yapar. Aranılan bilgi bulunursa index numarasını döndürür. Yoksa boş döndürür.

```
<?php
```

```
$array=array(0=>'blue',1=>'red',2=>'green',3=>'red');
```

```
$key=array_search('green',$array); // $key = 2;
```

```
echo $key;
```

```
$key=array_search('red',$array); // $key = 1;
```

```
echo $key;
```

```
?>
```



# Dizilerde Kullanılan Fonksiyonlar

- **array\_slice()**: Dizinin içersinden belirli bir bölümü alır.
- Kullanımı: array\_slice(dizi,başlangıç,adet)

```
<?php
```

```
$input = array("a", "b", "c", "d", "e");
```

```
$output = array_slice($input, 2); // çıktısı: "c", "d", "e"
```

```
print_r($output);
```

```
$output = array_slice($input, -2, 1); // çıktısı: "d"
```

```
print_r($output);
```

```
$output = array_slice($input, 0, 3); // çıktısı: "a", "b", "c"
```

```
print_r($output);
```

```
?>
```

# Dizilerde Kullanılan Fonksiyonlar

- sort: Dizi elemanlarını küçükten büyüğe sıralar.

```
<?php
```

```
$fruits = array("lemon", "orange", "banana", "apple");
```

```
sort($fruits);
```

```
foreach ($fruits as $key => $val) {
```

```
    echo "fruits[" . $key . "] = " . $val . "\n";
```

```
}
```

```
?>
```

Çıktısı

fruits[0] = apple

fruits[1] = banana

fruits[2] = lemon

fruits[3] = orange

# Dizilerde Kullanılan Fonksiyonlar

- `rsort()`: Dizi elemanlarını büyükten küçüğe sıralar.
- ```
<?php
$meyveler = array("portakal", "armut", "muz", "kayısı");
rsort($meyveler);
foreach ($meyveler as $indis => $meyve) {
    echo "$indis = $meyve\n";
}
?>
```

Çıktısı  
0 = portakal  
1 = muz  
2 = kayısı  
3 = armut

# Program Kontrol Deyimleri

- Şarta başlı işlemler;
  - IF-ELSE Yapısı,
  - Switch-Case Yapısı,
- Döngüler;
  - For,
  - While,
  - Do-while.
  - foreach

# Kontrol Yapıları

```
<?  
  
switch ( $no ) {  
  
    case 1 ;  
        echo ("1 nolu sınıf");  
        break;  
  
    case 2 ;  
        echo ("2 nolu sınıf");  
        break;  
  
    default:  
        echo ("3 nolu sınıf");  
}  
?>
```

```
<?  
  
if($user_name=="") {  
    echo "Username girmediniz."  
}  
  
else if($user_name!="ugur") {  
    echo "Yanlış username";  
}  
  
else {  
    echo "Merhaba $user_name";  
}  
?>
```

# sart.php

```
<?php
$not1=40; $not2=50;
if ($not1>$not2)
echo "not1 değeri not2 den büyük";
elseif ($not1<$not2)
echo "not1 değeri not2 den küçük";
else
echo "not1 ve not2 eşit";
?>
```

# sart2.php

```
<?php
```

```
$yas=18; //yaş değerini 19 ve 30 olarak da deneyin
```

```
if ($yas<18)
```

```
echo "siteye girmeniz yasak!";
```

```
else
```

```
echo "Siteye Hoş geldiniz";
```

```
?>
```

# Sart3.php

```
<?php
```

```
$not1=90; (Not1=50, Not2=50 ve Not1=40, Not2=30 olarak deneyin)
```

```
$not2=40;
```

```
$ortalama=($not1+$not2)/2;
```

```
echo "Not ortalaması=". $ortalama;
```

```
if ($ortalama>50)
```

```
echo "geçtiniz";
```

```
elseif ($ortalama<50)
```

```
echo "kaldınız";
```

```
else echo "Sınırdasınız";
```

```
?>
```



# sartsayi.php

```
<?php
$i=1;
if ($i == 0) {
    echo "i eşit 0";
} elseif ($i == 1) {
    echo "i eşit 1";
} elseif ($i == 2) {
    echo "i eşit 2";
}
?>
```

# sartmeyve.php

```
<?php
$i="portakal"; //elma ve muz olarak de deneyiniz.
switch ($i) {
    case "elma":
        echo "i deęeri elma";
        break;
    case "portakal":
        echo "i deęeri portakal";
        break;
    case "muz":
        echo "i deęeri muz";
        break;
}
?>
```

# Sart5.php

```
<?php
switch ($i) {
case 0:
case 1:
case 2:
    echo "i degeri 3 den kucuk ama negatif degil";
    break;
case 3:
    echo "i degeri 3";
}
?>
```

# Sart6.php

```
<?php
$dondurma= 'sade' ;
switch ($dondurma)
{
    case 'sade' ;
    case 'visne' ;
    case 'limon' ;
        echo 'iyi secim' ;
    break ;
    default :
        echo 'Lutfen yeni bir secim yapiniz...' ;
    break ;
}
?>
```

# Döngüler (For Döngüsü)

- Döngüler bir işi daha kısa sürede ve daha az kod yazarak halletmemizi sağlayan yapılardır. Bütün programlama dillerinde döngüler vardır. PHP'de tıpkı C,Java dillerinde olduğu gibi üç tip döngü çeşidi vardır. For, Do-While ve While döngüleri. Örneğin ekrana 1'den 100'e kadar sayıları yazmak isteyelim. Döngüler olmasa her sayıyı teker yazmamız gerekir, oysa döngüyle aşağıda görüldüğü gibi birkaç satırda bu işi halledebiliriz.

- `<?php`
- `for ($i=1;$i<100;$i++)`
- `{`
- `printf("$i<br>");`
- `}`
- `?>`

# While Döngüsü

- `<?php`
- `$i=1;`
- `while ($i<100)`
- `{`
- `printf("$i<br>");`
- `$i++;`
- `}`
- `?>`

# For Örneği

- 100'den geriye 4'er 4'er yazınız.
- `<?php`
- `for ($i=100; $i>1; $i-=4)`
- `{`
- `printf("$i<br>");`
- `}`
- `?>`

# Do-While Döngüsü

- `<?php`
- `$i=1;`
- `do`
- `{`
- `printf("$i<br>");`
- `$i++;`
- `}while ($i<100)`
  
- `?>`



# For ile İşlemler

- **0,2,4,6,8...100** şeklinde çift sayıları yazdırınız.
- `<?php`
- `for ($i=0; $i<=100; $i+=2)`
- `printf("$i<br>");`
- `?>`

# Do-While ile İşlemler

- **0,2,4,6,8...100** şeklinde çift sayıları yazdırınız.
- `<?php`
- `$i=0;`
- `do`
- `{`
- `printf("$i<br>");`
- `$i=$i+2;`
- `}while ($i<=100);`
- `?>`

# While ile İşlemler

- **0,2,4,6,8...100 şeklinde çift sayıları yazdırınız.**



```
<?php
```

- `$i=0;`

- `while ($i<=100)`

- `{`

- `printf("$i<br>");`

- `$i=$i+2;`

- `}`

- `?>`

# For ile İşlem

- **1+3+5...+99 toplamını bulunuz?**
- `<?php`
- `$toplam=0;`
- `for ($i=1;$i<100;$i+=2)`
- `$toplam=$toplam + $i;`
- `printf("For ile 1 den 100 e kadar  
Toplam:$toplam<br>");`
- `?>`

# While ile İşlem

- **1+3+5...+99 toplamını bulunuz?**
- `<?php`
- `$toplam=0;`
- `$i=1;`
- `while ($i<100)`
- `{`
- `$toplam=$toplam + $i;`
- `$i+=2;`
- `}`
- `printf("While ile 1 den 100 e kadar  
Toplam:$toplam<br>");`
- `?>`

# Do-While ile İşlem

- **1+3+5...+99 toplamını bulunuz?**

- `<?php`

- `$toplam=0;`

- `$i=1;`

- `do`

- `{`

- `$toplam=$toplam + $i;`

- `$i+=2;`

- `}while ($i<100);`

- `printf("DO-While ile 1 den 100 e kadar toplam:$toplam<br>");`

- `?>`

# foreach():

- Bir array deki bütün değerleri herhangi bir değişkeni arttırmaya gerek duymadan görüntüler.

```
<?
    $dersler[] = "Matematik";
    $dersler[] = "Türkçe";
    $dersler[] = "Bilgisayar";

    foreach($dersler as $ders_adi) {
echo "$ders_adi <br>";
    }
?>
```