

İnternet Programcılığı

Öğr. Gör. Kemal ÖZCAN



Hafta 7

Diziler ve Nesneler

Diziler ve Nesnelere

- Php'de diziler , birden çok veriyi tek bir deęişkende tutarak grup haline getirmek için kullanılırlar. Örneęin 0'dan 9'a kadar olan sayılara bir dizi diyebiliriz veya A'dan Z'ye kadar olan bütün harflere dizi diyebiliriz. Bir dizi ardışık sayı ya da harflerden oluşmak zorunda değildir. Örneęin bir sınıftaki bütün öğrencilerin bir araya getirilmesi de bir listedir(dizi). Gökhan , Emre, Kubilay , Fatih gibi öğrencilerin bir araya gelmesiyle dizi oluşur. Dizileri kullanmak için `array()` komutu kullanılır.

- 1-) Tek Boyutlu Diziler
- 2-) Çok Boyutlu Diziler

Olmak üzere 2 farklı şekilde kullanılırlar.

- **Tek Boyutlu Diziler**

Tek boyutlu dizilerin kullanımını aşağıdaki gibidir.

```
<?php  
$dizi_adi = array(değer1,değer2,...);  
?>
```

Veya

```
<?php  
$dizi=array();  
$dizi[anahtar] = değer1;  
$dizi[anahtar] = değer1;  
?>
```

Yukarıdaki şekillerde ki gibi kullanılabilirler.

Php 'de Dizi Elemanlarına Eriřim

- Tanınlı olan bir dizi deęiřkenin ierdięi elemanlara eriřmek iin kseli parantez iinde indis deęerini belirtmeliyiz.Ařaęıdaki rneęi inceleyiniz.

| rnek | Ekran ıktısı |
|--|---------------|
| <pre><?php \$hafta = array("pazartesi", "Salı", "arřamba", "Perřembe", "Cuma", "Cumartesi", "Pazar"); echo \$hafta[3]; ?></pre> | Perřembe |

- **Count()** fonksiyonu bir dizideki toplam eleman sayısını verir.
\$elemansayisi=count(\$dizi);

Php 'de Diziyeye Eleman Ekleme

Örnek

```
<?php
$dizi = array();
$dizi[0] = "Bilgisayar Programcılığı";
$dizi[1] = "Grafik Tasarım";
$dizi[2] = "Turizm Otelcilik";
$dizi[3] = "Muhasebe ve Vergi
Uygulamaları";
echo $dizi[2]."<br>";
$dizi[] = "Çocuk Gelişimi"; // diziyeye
eklenecek
echo $dizi[4];
?>
```

Ekran Çıktısı

Turizm Otelcilik
Çocuk Gelişimi

Dizilere Eleman Ekleme – Çıkartma

| Fonksiyon | Açıklama | Örnek |
|-----------------|--|---|
| array_shift() | array_shift() fonksiyonunu kullanarak dizi listesinin başında bulunan bir elemanı çıkarmak için kullanılır. | <pre><?php \$dizi=array("İnternet","Program cılık", "Bilgisayar","Web Tasarım"); echo array_shift(\$dizi); ?></pre> Ekrana İnternet yazar. |
| array_unshift() | array_unshift() fonksiyonunu kullanarak dizi listesinin başına istediğimiz sayıda yeni eleman ekleyebiliriz. | <pre><?php \$dizi=array("Programcılık", "Bilgisayar","Web Tasarım"); array_unshift(\$dizi,"İnternet"); Echo join("-", \$dizi); ?></pre> Ekrana İnternet – Programcılık-Bilgisayar –Web Tasarım yazar. |

Dizilere Eleman Ekleme – Çıkartma

| Fonksiyon | Açıklama | Örnek |
|--------------|--|--|
| array_pop() | array_pop() fonksiyonunu kullanarak bir dizideki listenin sonunda bulunan bir elemanı çıkartmak için kullanılır. | <pre><?php \$dizi=array("İnternet","Programcılık", "Bilgisayar","Web Tasarım"); echo array_pop(\$dizi); ?></pre> <p>Ekрана Web Tasarım yazar.</p> |
| array_push() | array_push() fonksiyonunu kullanarak bir dizideki listenin sonuna istediğimiz sayıda yeni eleman ekleyebiliriz. | <pre><?php \$dizi=array("İnternet","Programcılık", "Bilgisayar","Web Tasarım"); array_push(\$dizi,"Veritabanı","Donanım"); for(\$x=0;\$x>count(\$dizi);\$x++){ echo \$dizi[\$x]." "; }?></pre> <p>Ekрана İnternet Programcılık Bilgisayar Web tasarım Veritabanı Donanım yazar.</p> |

İlişkili Diziler

- Php'de anahtar ve değerlerin bir araya geldiği dizilere ilişkili diziler denir. İlişkili dizide birbirinden virgülle ayrılmış birden çok anahtar => değer çifti olabilir. Anahtar isimleri sayı veya sözcüklerden de oluşabilir.

Anahtar dizi sözcük de olabilir

```
$bilgisayar=array('bir' => 'anakart', 'iki' => 'ram', 'üç' => 'işlemci', 'dört' => 'ekran kartı');
```

Anahtar dizi sayı da olabilir

```
$bilgisayar=array(1 => 'anakart', 2 => 'ram', 3 => 'işlemci', 4 => 'ekran kartı');
```

| Örnek | Ekran Çıktısı |
|---|------------------------|
| <pre><?php \$dizi = array("ad" => "Kemal", "soyad" => "ÖZCAN"); echo \$dizi["ad"] . " "; echo \$dizi["soyad"] . " "; ?></pre> | <pre>Kemal ÖZCAN</pre> |

- Anahtar adı bir değişken ise tırnak arasına almamıza gerek yoktur.

```
<?php
$anahtar="ad";
$dizi = array("ad"=>"Kemal","soyad" => "ÖZCAN");
echo $dizi[$anahtar] . "<br>";
echo $dizi["soyad"] . "<br>";
?>
```

Örnek

```
<?php
$dizi = array("id"=>"1","baslik" =>
"İnternet Programcılığı");
$dizi["konu"] ="Diziler";
print_r($dizi);
?>
```

Ekran Çıktısı

```
Array(
  [id]=>1
  [baslik]=> İnternet Programcılığı
  [konu]=>Diziler
)
```

- Diziye yeni bir eleman eklemek istersek [] parantez ile dizinin sonuna ekleyebiliriz.

| <u>Örnek</u> | <u>Ekran Çıktısı</u> |
|--|---|
| <pre><?php \$dizi = array("Ondokuz Mayıs Üniversitesi","Samsun MYO"); \$dizi[] = "Bilgisayar Programcılığı"; print_r(\$dizi); ?></pre> | <pre>Array([0]=>Ondokuz Mayıs Üniversitesi [1]=> Samsun MYO [2]=>Bilgisayar Programcılığı)</pre> |

- Diziyi İstedığımız sıradan başlatabiliriz

| <u>Örnek</u> | <u>Ekran Çıktısı</u> |
|--|---|
| <pre><?php \$dizi = array(5=>"Ondokuz Mayıs Üniversitesi",6=>"Samsun MYO"); \$dizi[] = "Bilgisayar Programcılığı"; print_r(\$dizi); ?></pre> | <pre>Array([5]=>Ondokuz Mayıs Üniversitesi [6]=> Samsun MYO [7]=>Bilgisayar Programcılığı)</pre> |

- Bir dizinin tamamını silmek istiyorsak veya bir elemanını silmek istiyorsak unset() fonksiyonunu kullanırız.

| Örnek | Ekran Çıktısı |
|--|--|
| <pre><?php \$dizi = array(5=>"Ondokuz Mayıs Üniversitesi",5=>"Samsun MYO"); \$dizi[] = "Bilgisayar Programcılığı"; unset(\$dizi[5]); print_r(\$dizi); ?></pre> | <pre>Array([6]=> Samsun MYO [7]=>Bilgisayar Programcılığı)</pre> |

- Bir dizinin tamamını silmek istiyorsak veya bir elemanını silmek istiyorsak unset() fonksiyonunu kullanırız.

| Örnek | Ekran Çıktısı |
|---|--|
| <pre><?php \$dizi = array(5=>"Ondokuz Mayıs Üniversitesi",5=>"Samsun MYO"); \$dizi[] = "Bilgisayar Programcılığı"; unset(\$dizi); print_r(\$dizi); ?></pre> | <p>Ekranında ya bir hata mesajı görünür ya da boş bir ekran görünür.</p> |

İç İçe Çok Boyutlu Diziler

- Bir dizi içerisinde birden fazla dizi tanımlama işlemine çok boyutlu diziler denir.

| Örnek | Ekran Çıktısı |
|---|---|
| <pre><?php \$dizi = array("bilgisayar" => array("Gökhan","Hasan","Ahmet"), "turizm" => array("Hilal","Melike","Ayşe")); print_r(\$dizi); ?></pre> | <pre>Array ([bilgisayar] => Array ([0] => Gökhan [1] => Hasan [2] => Ahmet) [turizm] => Array ([0] => Hilal [1] => Melike [2] => Ayşe))</pre> |

- Bir dizi içerisindeki elemanları indise eşit olanları yazdırma.

| Örnek | Ekran Çıktısı |
|--|--------------------------------------|
| <pre><?php \$dizi = array("bilgisayar" => array("Gökhan","Hasan","Ahmet"), "turizm" => array("Hilal","Melike","Ayşe")); echo \$dizi["bilgisayar"][0]; echo \$dizi["bilgisayar"][1]; echo \$dizi["turizm"][0]; echo \$dizi["turizm"][1]; ?></pre> | <pre>Gökhan Hasan Hilal Melike</pre> |

Döngüler ile İç İçe Dizi Elemanlarını Listeletme

- For döngüsü ile dizi elemanlarını listeletme işlemi aşağıdaki gibidir.

| Örnek | Ekran Çıktısı |
|--|--|
| <pre><h1>Öğrenciler</h1> <?php \$ogrenciler = array(array("ad"=>"Gökhan","bolum"=>"Bilgisayar Programcılığı"), array("ad"=>"Ahmet","bolum"=>"Grafik Tasarım"), array("ad"=>"Can","bolum"=>"Seracılık")); ?> <?php for(\$i=0;\$i<count(\$ogrenciler);\$i++){?> <?php echo \$ogrenciler[\$i]["ad"]; ?> <?php echo \$ogrenciler[\$i]["bolum"]; ?> <?php } ?> </pre> | <p>Öğrenciler</p> <ul style="list-style-type: none">• Gökhan Bilgisayar Programcılığı• Ahmet Grafik Tasarım• Can Seracılık |

- Foreach Döngüsü ile dizi elemanlarını listeleme işlemi aşağıdaki gibidir.

| Örnek | Ekran Çıktısı |
|---|---|
| <pre><h1>Öğrenciler</h1> <?php \$ogrenciler = array(array("ad"=>"Gökhan", "bolum"=>"Bilgisayar Programcılığı"), array("ad"=>"Ahmet", "bolum"=>"Grafik Tasarım"), array("ad"=>"Can", "bolum"=>"Seracılık")); ?> <?php foreach(\$ogrenciler as \$sanahtar => \$deger) { ?> <?php echo \$deger["ad"]; ?> <?php echo \$deger["bolum"]; ?> <?php } ?> </pre> | <p>Öğrenciler</p> <ul style="list-style-type: none">• Gökhan Bilgisayar Programcılığı• Ahmet Grafik Tasarım• Can Seracılık |

Nesneler

- Sınıf (class) nesnelerinin temelini oluşturmaktadırlar. Nesne herhangi bir işlevi yerine getirebilecek altyapı hazırlayan ve çeşitli fonksiyonlar içeren bir yapıdır.

Sınıf Nedir ?

- Dünyadaki nesnelerin kendi aralarında gruplaşması ile meydana gelen özel bir yapıya sınıf denir. Örneğin hayvanlar, meyveler, sebzeler vb. gibi.
- Hayvanlar bir sınıfın adıdır. Hayvanlar sınıfı içinde kuş, köpek, kedi o sınıfın birer nesnelere dir. Programlama dilinde nesnelerin bir araya gelmesiyle (sınıflandırılmasıyla) sınıf (class) oluşmaktadır.

Sınıf Nedir ?

- Gerçek dünyadaki sınıf ve nesnelere gibi tekrar kullanılabilir yazılım parçalarındandır.
- Sınıflar nesnelere ait bir çok özelliği barındırabilirler.
- Sınıflar geliştirilmeye açıktırlar. Bir çok özellik , eleman ve fonksiyon barındırabilir.
- Sınıf içinde paketlenen fonksiyonların nasıl kullanılacağını belirtmesine soyutlama (Abstraction) denir.
- Paketleme ve soyutlama işlemi bir nesneyi belli etmek için yeterli 2 tane özelliklerden biridir.

Nesne Nedir ?

Sınıfın bir örneğidir. Sınıfa ait özellikleri taşımaktadır.

- Her nesnenin kendine ait özellikleri(Attributes, Properties) vardır.
- Her nesnenin kendine özgü davranışları(Behaviors,Actions) vardır.
- Her yeni nesne mirasla(inheritance) bir üst(parent) sınıfının ait nesnelerin özellik ve yöntemlerini devralabilir.

Sınıfın genel oluşturulma biçimi aşağıdaki gibidir.

```
Class sınıf_adi{  
    //Tanımlanmış özellikler, sabitlerin ekleneceği yerdir.  
    ...  
    //Diğer bütün metotların ekleneceği alandır.  
    ...  
}
```

- Var anahtar sözcüğünü kullanarak bir sınıf tanımı içinde değişkenler tanımlanabilir.
- Aşağıdaki kodlarda degisken1 degisken2 diye iki niteliği olan, tablo1 diye bir sınıf oluşturulmaktadır.

```
Class tablo{  
var $degisken1;  
var $degisken2;  
}
```

İşlemleri sınıf tanımı içinde fonksiyonlar tanımlayarak oluşturuyoruz.

```
Class tablo  
{  
    Function islem1()  
    {  
    }  
    Function islem2($param1,$param2)  
    {  
    }  
}
```

- Önceki sayfadaki nesnede , işlem1() fonksiyonu hiç parametre almazken işlem2() iki tane parametre alır.
- Yeni bir nesne oluşturmak için **new** kelimesi kullanılır. Bunu yaptığımızda nesnenizin hangi sınıfın örneği olacağını belirlemeniz ve yapılandırıcının gerektirdiği parametreleri sağlamanız gerekir.

| Örnek | Ekran Çıktısı |
|---|--|
| <pre><?php class tablo1 { Function construct(\$param) { echo "Yapılandırıcı \$param parametreyle çağrıldı"; } }\$a=new tablo1; \$a->construct("Birinci"); echo" "; \$b=new tablo1; \$b->construct("İkinci"); echo" "; \$c=new tablo1; c->construct("Üçüncü"); ?></pre> | <p>Yapılandırıcı Birinci parametreyle çağrıldı Yapılandırıcı İkinci parametreyle çağrıldı Yapılandırıcı Üçüncü parametreyle çağrıldı</p> |

- **This** = Sınıf içindeki özelliklere erişmek için özellik adına **\$this** değişkeni bir -> işareti ile eklenir. Burada **this** erişim metodudur ve yabancı nesne olarak tanımlanır. Bu bilgilere göre özelliğe erişmek için **\$this->\$isim** şeklinde oluşturulan method içine yazabiliriz. This ile tanımlanmayan değişkenler sınıfın her yerinde kullanılamaz.

```
Class musteri{  
public $isim;  
    function isim_sec($ad)  
    {  
        $this->$isim=$ad;  
    }  
    ...  
}
```

Örnek

```
<?php class musteri{
public $isim;
function isim_sec($ad){
    $this->isim=$ad;
}
function isim_al(){
    return $this->isim;
}
}
$a= new musteri();
$a->isim_sec("Ondokuz Mayıs
Üniversitesi");
echo $a->isim_al();
echo "<br>";
$b= new musteri();
$b->isim_sec("Samsun MYO");
echo $b->isim_al();
?>
```

Ekran Çıktısı

Ondokuz Mayıs Üniversitesi
Samsun MYO



Ders Sonu

**Haftaya
Dosyalama İşlemleri**