WEB TASARIM I

Öğr. Gör. M. Mutlu YAPICI

Ankara Üniversitesi
Elmadağ Meslek Yüksekokulu

Öğr. Gör. M. Mutlu YAPICI

Ders İzlencesi

Hafta	Modüller/İçerik/Konular
1. Hafta	PHP Tanımı ve Sunucu Kurulumları
2. Hafta	PHP Yazım Notasyonu ve Değişkenler
3. Hafta	PHP de Karar kontrol yapıları ve Döngüler
4. Hafta	Dizi ve Dizi işlemleri
5. Hafta	Fonksiyon, Sınıf ve Nesne Kavramları
6. Hafta	HTML ve PHP
7. Hafta	PHP ile Veritabanı İşlemleri
8. Hafta	MYSQL, MYSQLİ
9. Hafta	AJAX
10. Hafta	
11. Hafta	
12. Hafta	
13. Hafta	
14. Hafta	

Bu Ünitede Ele Alınan Konular

- PHP dilinin genel yapısı
- Yazım notasyonu
- Yazım kuralları
- Değişken ve sabit tanımlama
- Değişken veri tipleri
- Operatörler
- Karar kontrol komutları (İf-else Switch-case)
- Döngü yapıları (For, while, do-While, foreach)

Ders Kazanımları

Bu bölümü Bitirdiğimizde,

- PHP dilinin genel yapısı
- Yazım notasyonu
- Yazım kuralları
- Değişken ve sabit tanımlama
- Değişken veri tipleri
- Operatörler
- Karar kontrol komutları (İf-else Switch-case)
- Döngü yapıları (For, while, do-While)

öğrenmiş olacaksınız.

Döngüler

Döngüler bir program içerisinde belirli işlerin defalarca yapılmasını sağlayan komut bloklarıdır. Sonsuz döngüler yapılabildiği gibi belirli kriterler sağlanana kadar devam eden döngüler de yapılabilir.

4 tip döngü vardır. Bunlar:

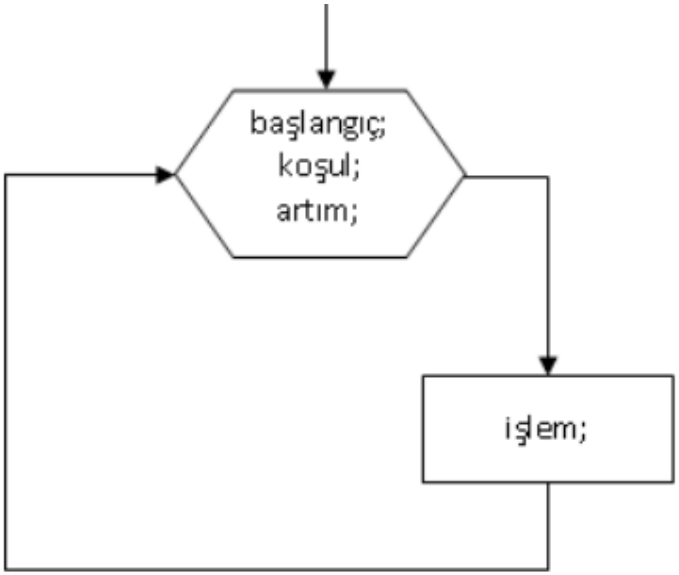
- for döngüleri
- while döngüleri
- do while döngüleri
- foreach döngüleri' dir.

For Döngüsü

Belirlenen başlangıç değerinden itibaren belirtilen koşul sağlanana kadar içine yazıldığı kod parçasını ardı ardına çalıştıran bir döngü çeşididir.

Yapısı:

```
for(başlangıç; koşul; artım)
{
    yapılacak işler;
}
```



Başlangıç, döngü kontrol değişkeni olarak da ifade edilebilir. Döngü içerisinde bir sayaç görevini görür.

Koşul, döngünün ne kadar çalışacağını denetleyen mekanizmadır. Koşul sağlanıyorken döngü çalışmaya devam eder. Koşul sağlanmadığında ise döngü durur. Koşulda genellikle başlangıç değerinin durumu denetlenir.

Artım, başlangıç değerinin döngünün her adımda artma ya da azaltma miktarını belirler. Eğer başlangıç değeri hiç değişmez ise sonsuz döngü oluşur.

For Döngüsü Örnekleri

Örnek 2-1: 1'den 10'a kadar olan sayıları ekrana yazdırınız.

```
$i;
for($i=1;$i<=10;$i++)
{
    Echo($i);
}
```

Yukarıdaki kodu incelediğimizde;

- **Döngü kontrol değişkenimiz olan i'ye 1 değerini atayarak başlangıç değerimizi,**
- **Döngümüzün ne zamana kadar döneceğini belirlediğimiz koşulumuzu i<=10 ifadesini,**
- **i++ ile de i değerimizi döngümüzün her dönüşünde 1 arttıracağımızı**

belirliyoruz.

For Döngüsü Örnekleri

Örnek :10'dan 0'a geriye doğru sayan ve sayıları ekrana yazdıran programı yazdırınız.

For Döngüsü Örnekleri

Örnek :Klavyeden girilen 10 sayıdan en büyüğünü bulan PHP programını yazınız.

Cevap: En büyük sayıyı bulmak için girilen sayıları bir referans değer ile karşılaştırmamız gerekir. Başta sıfır olarak aldığımız referans değer kendinden daha büyük bir değer ile karşılaştığında yeni referans değerimiz bu sayı olacaktır. İşlem sonunda referans değer yazdırılacak

For Döngüsü Örnekleri

Örnek :0'dan klavyeden girilen sayıya kadar olan sayıların toplamını ekrana yazdıran programı yazınız.

For Döngüsü Örnekleri

Örnek :Aşağıdaki çıktıyı veren programı for döngüsü ile yazınız.

For Döngüsü Örnekleri

Örnek :Aşağıdaki çıktıyı veren programı for döngüsü ile yazınız.

```
*****
*****
*****
*****
*****
```

For Döngüsü Örnekleri

Örnek :Aşağıdaki çıktıyı veren programı for döngüsü ile yazınız.

```
*****
****
***
**
*
```

For Döngüsü Örnekleri

Örnek :1 ile 50 arasındaki sayılardan asal olanlarını bulup ekrana yazan PHP programını yazınız.

Cevap: Asal sayılar sadece bire ve kendisine bölünebilen sayılardır. İlk döngümüz 1 ile 50 arasındaki sayıları elde etsin.İkinci döngü ise bunların içerisinden asal olan sayıları bulsun.

For Döngüsü Örnekleri

Örnek : $dizi=array("Ankara", "İzmir", "Samsun", "Manisa", "Antalya", "Bodrum", "İstanbul");

Yukarıdaki dizinin içeriğini ekrana alt alta yazan PHP programını yazınız.

For Döngüsü Örnekleri

Örnek : 10 dan 300'e kadar 5 in katları olan sayıların hepsini bir diziye yükleyin ve dizinin içeriğini ekrana alt alta yazan PHP programını yazınız.

For Döngüsü Örnekleri

Örnek : $dizi=array(2 => "Ankara", 4 => "İzmir", "ali" => "Samsun", 10 => "Manisa", "veli" => "Antalya", "Bodrum", "İstanbul" => 34);

Yukarıdaki dizinin içeriğini ekrana alt alta yazan PHP programını yazınız.

Foreach Döngüsü Örnekleri

Normalde dizilerin içeriklerini for döngüsü ile ekrana yazdırmak mümkündür. Ancak bir önceki örnekte verilen dizinin bu şekilde ekrana yazdırılamadığını siz de gördünüz. Eğer dizilerin ofset (indis ya da key) numaraları ardarda düzenli bir şekilde artan sayılar ise bu mümkün olabiliyor. Bir önceki örnekte elemanların ofsetleri hem düzenli değil hem de sadece sayılardan oluşmuyor, metinsel değerlerden de oluşuyor. Bu gibi durumlarda dizi içeriğini ekrana yazdırmak için for döngüsü kullanmak mümkün olmuyor. İşte bu gibi durumlarda kullanılmak üzere FOREACH döngüsü oluşturulmuştur. Ofsetlerini bilmediğiniz ya da ofsetleri düzenli olmayan dizilerde FOREACH kolaylıkla kullanılabilmektedir.

Foreach Döngüsü Örnekleri

Forach döngüsünün kullanım şekli :

```
Foreach(Kullanılacak Dizi AS Sıradaki Değeri Tutacak Değişken  )
{
        .....
}
```

Örnek:

```
Foreach($dizi as $a )
{
        echo $a;
}
```

Foreach Döngüsü Örnekleri

Forach döngüsünün kullanım şekli 2 : Eğer ki aynı zamanda dizinin ofsetini de görmek istiyorsak ofseti de farklı bir değişkene yüklememiz mümkündür.

```
Foreach(Kullanılacak Dizi AS Ofset Değişkeni => Değer Değişkeni  )
{
        .....
}
```

Örnek:

```
Foreach($dizi as $ofset => $a )
{
        echo   $ofset."İndisinde Bulunan Değer = ".$a;
}
```

While Döngüsü

While döngüsü bir koşul sağlanıyorken dönmeye devam eder. Koşul yanlış (false) sonucunu verdiği zaman ise sonlandırılır.

Yapısı:

```
while(koşul)
{
    yapılacak işler;
}
.
```

Koşul, döngünün ne kadar çalışacağını denetleyen mekanizmadır. Koşul sağlanıyorken döngü çalışmaya devam eder. Koşul sağlanmadığında ise döngü durur. Koşulda genellikle başlangıç değerinin durumu denetlenir.

While Döngüsü Örnekleri

Örnek : Sıfırdan itibaren 100 e kadar olan Sayıların Toplamını aldıran program.

```
$toplam=0;
$sayi=0;

While($sayi <100)
{
        $toplam +=$sayi;
        $sayi++;
}

ECHO (“Sayıların Toplamı =”,$toplam);
```

Do - While Döngüsü

For ve while döngülerinde döngü bloklarının koşul sağlanmadığı takdirde hiç çalıştırılmama ihtimali vardır. Ancak döngünün en az bir kere çalıştırılması istenilen durumlarda do-while döngüleri kullanılırlar.
Do-While döngülerinde koşul döngü içerisindeki işlemler bir kez gerçekleştirildikten sonra kontrol edilir. Koşul doğru olduğu müddetçe de döngü içerisindeki işlemler tekrarlanmayı sürdürür.
Genel yazım şekli şöyledir.

```
do
{
        yapılacak işler;
}
while(koşul);
```

Do - While Döngüsü Örnekleri

Örnek : 1 ile 200 arasındaki sayılarda 4' e kalansız bölünebilen veya 7' ye kalansız bölünemeyen sayıların adetini bulduran programı yazınız?

Do - While Döngüsü Örnekleri

Örnek : Klavyeden girilen bir sayıyı 2 li sayı sistemine çeviren C# programını yazınız?

Dallanma - Atlama (Jump) Komutları

Programın akışı esnasında başka satırlara atlama işlemi gerçekleştiren bir takım anahtar sözcükler vardır.

Bunlar;

- **break,**
- **continue,**
- **goto,**
- **return anahtar sözcükleridir.**

Break

Break anahtar sözcüğü döngülerden çıkmak için kullanılır. Döngülerde, break anahtar sözcüğüne rastlandığı anda döngüden çıkılır ve program döngü bloğundan sonraki ilk deyimle akışına devam eder.

Break anahtar sözcüğü döngü bloklarının ya da switch bloklarının dışında kullanılamazlar.

Break Örneği

Örnek : ‘A’ harfinden başlayıp ‘Z’ye kadar devam eden bir döngü de ‘K’ harfine gelindiğinde döngüden çıkan programın kodunu yazınız.

Çözüm:

```
for ($i = 'A'; $i <= 'Z'; $i++)
{
        if ($i == 'K')
        break;
        ECHO($i);
}
ECHO("Döngüden çıkıldı...");
```

Continue

Continue ifadesi, break ifadesine benzerdir ve bir for, foreach, while ya da do...while döngüsü içinde de kullanılabilir. Ancak, döngünün dışına çıkmak yerine mevcut döngüden çıkarak bir sonraki döngüye geçişi sağlayacaktır.

Örnek :

```
$i = 1;
$k = 100;
while ($i < $k)
{
        $i++;
        if(($i%5) ==0)
        continue;
        Echo($i." , ");

}
```

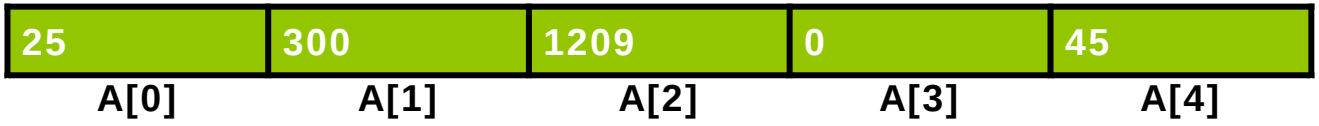
Diziler (Arrays)

Değişkenleri öğrenirken gördük ki her değişkene sadece bir değer atayabiliriz. Bazı durumlarda aynı tipteki değişkenleri bir arada tutma ihtiyacı duyabiliriz. PHP bize farklı tipteki değişkenleri bile tek bir adla saklayabileceğimiz **dizileri (Array)** sunmaktadır.

Diziler bir programlama dilindeki en önemli veri yapılarından biridir. Bellekte ardışık olarak yer kaplayan veri kümesine array(dizi) adı verilir.

Diziler yapılarına göre ikiye ayrılır.

- Statik Diziler
- Dinamik Diziler



Neden Dizilere İhtiyaç Var?

Diziler sayesinden aynı türdeki işlemler için birden fazla değişken tanımlamaktan kurtuluruz. Tek bir değişkenle 10 larca 100lerce veriyi ayrı ayrı saklama imkanı sunar diziler.
Örneğin: Sınıfımızda 180 kişinin isimlerini ve programlama dersi notlarını programa girip ilk üç kişiyi, notlarının ortalamasını ve kaç kişinin kalıp kaç kişinin geçtiğini bulmak istiyoruz.

Bunun için aşağıdaki gibi her bir öğrencinin adı soyadı ve notu için bir değişken tanımlamanın saçma ve kullanışsız olduğu aşikardır.

- **string** isim1,isim2,isim3,isim4……isim180;
- **int** not1,not2,not3,not4,not5……..not180;

Bunun yerine sadece her bir tür için iki adet 180 elemanlı dizi değişkeni tanımlamak daha kullanışlı ve kolay olacaktır.

$isimler=array( );

Neden Dizilere İhtiyaç Var?

Dizileri dosyalardan veri çekerken ve özellikle veritabanı işlemlerinde oldukça fazla kullanacağız. Bu sebeple dizi kavramını ve dizilerde kullandığımız fonksiyonları iyi bir biçimde öğrenmemiz önem arz etmektedir.

Diziler

İstersek herhangi bir dosya içerisinden verileri okuyup direk bir dizi değişkenine satır satır yükleyebiliriz. Bu dizileri kullandığımız en uygun yerlerden biridir.

```
<?Php
$dizi=file("sehirler.txt");
print_r($dizi); ///Tüm şehirleri dizi şeklinde yazar

?>
```

Dizilerin eleman sayısını Count() fonksiyonu ile öğrenebiliriz.

Diziler

Asıl dizi oluşturmak için kullanacağımız fonksiyon ise ARRAY() dir.

```
<?Php
$dizi=array( "Ankara", "İzmir", "Samsun", "Manisa");
print_r($dizi); ///Tüm şehirleri dizi şeklinde yazar

?>
```

Dizilerin eleman sayısını Count() fonksiyonu ile öğrenebiliriz.

Dizilere Veri Ekleme ve Çıkarma

Dizilere veri ekleme için ofsetini boş bırakıp aktarma işlemi yapmak yeterlidir. Bu şekilde dizinin sonuna 1 eleman daha eklemiş oluruz.

```
<?Php
$dizi=array( "Ankara", "İzmir", "Samsun", "Manisa");
$dizi[ ]="Antakya";
print_r($dizi); ///Tüm şehirleri dizi şeklinde yazar

?>
```

Dizilere Veri Ekleme ve Çıkarma

Dizilerden veri silmek içinse UNSET fonksiyonunu kullanırız.

```
<?Php
$dizi=array( "Ankara", "İzmir", "Samsun", "Manisa");
unset($dizi[2]); //ikinci 0fsete sahip samsunu siler
print_r($dizi); ///Tüm şehirleri dizi şeklinde yazar

?>
unset($dizi); //tüm elemanları siler diziyi yok eder
```

Dizilere Veri Ekleme ve Çıkarma

Dizilere veri eklemek ve çıkartmak için farklı fonksiyonlar da kullanılmaktadır.

Array_Shift($dizi); Fonksiyonu dizinin ilk elemanını diziden çıkartır. İlk elemanını almak istediğimiz ve aldığımız elemanın silinmesini istediğimiz dizilerde kullanılır.

```
<?Php
$dizi=array( "Ankara", "İzmir", "Samsun", "Manisa");
$a = Array_Shift($dizi); //ilk elemanı siler $a ya aktarır
print_r($dizi); ///Tüm şehirleri dizi şeklinde yazar

?>
```

Döngüyle kullansak tüm elemanları silerek alabiliriz.

Dizilere Veri Ekleme ve Çıkarma

Dizilere veri eklemek ve çıkartmak için farklı fonksiyonlar da kullanılmaktadır.

Array_unShift($dizi, **"Amasra", "İzmit"....**); Fonksiyonu dizinin başına bir yada daha fazla eleman ekler.

```
<?Php
$dizi=array( "Ankara", "İzmir", "Samsun", "Manisa");
$a = Array_Shift($dizi, "Amasra", "İzmit"); //elemanları ekler
print_r($dizi); ///Tüm şehirleri dizi şeklinde yazar

?>
```

Dizilere Veri Ekleme ve Çıkarma

Dizilere veri eklemek ve çıkartmak için farklı fonksiyonlar da kullanılmaktadır.

Array_Pop($dizi); Fonksiyonu dizinin son elemanını diziden çıkartır. Son elemanını almak istediğimiz ve aldığımız elemanın silinmesini istediğimiz dizilerde kullanılır.

```
<?Php
$dizi=array( "Ankara", "İzmir", "Samsun", "Manisa");
$a = Array_Pop($dizi); //son elemanı siler $a ya aktarır
print_r($dizi); ///Tüm şehirleri dizi şeklinde yazar

?>
```

Döngüyle kullansak tüm elemanları silerek alabiliriz.

Dizilere Veri Ekleme ve Çıkarma

Dizilere veri eklemek ve çıkartmak için farklı fonksiyonlar da kullanılmaktadır.

Array_push($dizi, **"Amasra", "İzmit"....**); Fonksiyonu dizinin sonuna bir yada daha fazla eleman ekler.

```
<?Php
$dizi=array( "Ankara", "İzmir", "Samsun", "Manisa");
$a = Array_push($dizi, "Amasra", "İzmit"); //elemanları sona ekler
print_r($dizi); ///Tüm şehirleri dizi şeklinde yazar

?>
```

Bazı Dizi Özellikleri

Diziler, .NET Framework içinde tanımlı Array sınıfı temsil eder. Tüm diziler Array sınıfında tanımlı özellikleri ve metotları kullanırlar. Bu metotlardan ve özelliklerden en sık kullanılanları şunlardır;

- **count($dizi),**
- **implode("-", $dizi),**
- **explode("-", $cumle),**
- **İs_array($dizi);**
- **List($veri1, $veri2, $veri3, $veri4,….)**
- **each($dizi)**
- **extract($dizi)**
- **Array_sum($dizi)**

Dizilerle işlemler

Bir dizideki eleman sayızını Coun($dizi) fonksiyonu ile alabiliriz.

```
<?Php
$dizi=array( "Ankara", "İzmir", "Samsun", "Manisa");
 Echo Count($dizi); // 4

?>
```

Dizilerle işlemler

Bir Dizideki verileri belirli bir ayraç kullanarak tek bir değişkene yüklemek yada direk ekrana yazdırmak için implode(ayraç, dizi) fonksiyonu kullanılır.

```
<?Php
$dizi=array( "Ankara", "İzmir", "Samsun", "Manisa");
$a = implode("-",$dizi); //Tüm elemanları aralarına – ekleyerek birleştirir
echo($a);
?>
```

Dizilerle işlemler

Bir cümledeki kelimeleri diziye parçalamak için explode(ayraç, kelime) fonksiyonu kullanılır.

```
<?Php
$veri= "merhaba arkadaşlar ne mutlu türküm diyene");
$dizi = explode(" ",$veri); //Tüm kelimelri bir dizi elemanına ayırır
Print_r($dizi);
?>
```

Dizilerle işlemler

Bir değişken yapısının Dizi olup olmadığını is_array($dizi) fonksiyonu ile alabiliriz. Eğer dizi ise true değil ise false değeri döner.

```
<?Php
$d =array( "Ankara", "İzmir", "Samsun", "Manisa");
İf(is_array($d))
        print_r($d); // dizi içeriğini yazar
else
        Echo ($d); // değişken içeriğini yazar

?>
```

Dizilerle işlemler

Bir dizi içerisindeki tüm değerleri istersek tek tek değişkenlere aktarabiliriz. Bu işlem için List($veri1, $veri2, $veri3, $veri4,….) fonksiyonunu kullanabiliriz.

```
<?Php
$d =array( "Ankara", "İzmir", "Samsun", "Manisa");

List($veri1, $veri2, $veri3, $veri4)=$dizi;
        Echo ($veri1." , ". $veri2." , ". $veri3." , ". $veri4); //
değişken içeriğini yazar

?>
```

Dizilerle işlemler

Bir dizi içerisindeki eleman gruplarını ofsetleri ile birlikte alabiliriz. Bu işlem için each($dizi) fonksiyonunu kullanabiliriz. Eğer bu fonksiyonu while içerisinde kullanırsak tüm elemanları sırasıyla almış oluruz.

```
<?Php
$d = array( "Ankara", "İzmir", "Samsun", "Manisa");
$a= each($dizi);
Print_r($a);
?>
```

Eğer list ile birlikte kullanırsak tüm elemanları ofsetleri ile birlikte almış oluruz.

```
<?Php
$d = array( "Ankara", "İzmir", "Samsun", "Manisa");
While (list($k,$v)=each($dizi))
{
        echo ($k." indisinde = $v verisi bulunuyor <br/>");
}
?>
```

Dizilerle işlemler

Bir dizi içerisindeki tüm anahtarları değişkenlere dönüştürüp içerisindeki değerleri de bu değişkenlere yükleyebiliriz. Bunun için extract($dizi) fonksiyonunu kullanırız.

```
<?Php
$d=array("adi"=>"Mutlu", "soyadi"=>"Yapıcı", "ders"=>"Web
Programlama 1");
extract($d);
echo "Adı = $adi Soyadı = $soyadi Dersi = $ders";
        ?>
```

Dizilerle işlemler

Extract fonksiyonu dizi ofsetlerini değişken olarak kullandığı için dizi ofsetleri değişken tanımlama kurallarına uygun olmalıdır. Eğer ofsetleri biz tanımlamaz isek hepsi 0 dan başlayarak rakamlar olur bu durumda extract ı direk kullanamayız çünkü değişkenler rakamla başlamaz. Bu gibi durumlarda değişkenlere ön ek koyabiliriz. Extract($dizi, EXTR_PREFIX_SAME,'ek_'); şeklinde kullanılır

EXTR_PREFIX_SAME → Eğer daha tanımlanmış aynı isimde değişken varsa dizininkine ek yapar
EXTR_PREFIX_ALL → Tüm dizi ofsetlerine ek yapar
EXTR_PREFIX_INVALID → Eğer dizi ofseti değişken kuralına uymuyorsa ek yapar

```
<?Php
$d=array("Mutlu", "Yapıcı","Web Programlama 1");
extract($d, EXTR_PREFIX_INVALID, 'ek');
echo "Adı = $ek_0 Soyadı = $ek_1 Dersi = $ek_2 ";
        ?>
```

Dizilerle işlemler

Bir dizi içerisindeki rakamları toplamak için

array_sum($dizi)

Fonksiyonu kullanılır.

```
<?Php
$d=array("adi"=>"Mutlu",34, "soyadi"=>"Yapıcı",22,
"ders"=>"Web Programlama 1",55 );
echo array_sum($d);//111
        ?>
```

Dizilerde ARAMA işlemleri

Bir dizi içerisinde arama yapmak için birden fazla fonksiyon mevcuttur.

- **Array_key_exsist();**
- **İn_array();**
- **Array_search();**

Dizilerde ARAMA işlemleri

Bir dizi içerisindeki ofsetlerde arama işlemi yapmak için array_key_exists('değer',$dizi) Fonksiyonu kullanılır.

```
<?Php
$d=array("adi"=>"Mutlu",34, "soyadi"=>"Yapıcı",22,
"ders"=>"Web Programlama 1",55 );
if( array_key_exists("adi",$d))
        echo "var";
else
        Echo "yok";

        ?>
```

Dizilerde ARAMA işlemleri

Bir dizi içerisindeki değerler içimde arama işlemi yapmak için in_array('değer',$dizi) Fonksiyonu kullanılır. Eğer değer varsa true yoksa false değeri döner.

```
<?Php
$d=array("adi"=>"Mutlu",34, "soyadi"=>"Yapıcı",22,
"ders"=>"Web Programlama 1",55 );
if( in_array("Mutlu",$d))
        echo "var";
else
        Echo "yok";

        ?>
```

Dizilerde ARAMA işlemleri

Bir dizi içerisindeki değerler içimde arama işlemi yapmak için in_array('değer',$dizi) Fonksiyonu kullanılır. Eğer değer varsa true yoksa false değeri döner.

```
<?Php
$d=array("adi"=>"Mutlu",34, "soyadi"=>"Yapıcı",22,
"ders"=>"Web Programlama 1",55 );
if( in_array("Mutlu",$d))
        echo "var";
else
        Echo "yok";

        ?>
```

Dizilerde ARAMA işlemleri

Bir dizi içerisindeki değerler içimde arama işlemi yapmak için array_search('değer',$dizi) Fonksiyonuda kullanılır. Bu fonksiyonun farkı eğer değer varsa ofset, yoksa false değeri döner.

```
<?Php
$d=array("adi"=>"Mutlu",34, "soyadi"=>"Yapıcı",22,
"ders"=>"Web Programlama 1",55 );
if( !array_searc("Mutlu",$d))
        echo "yok";
else
        Echo array_searc("Mutlu",$d);

        ?>
```

Dizilerde Değişiklik işlemleri

Dizilerde elemanlar arasında değişiklik yapmak, belirli noktalara yeni eleman eklemek veya eleman okumak için PHP de bir çok fonksiyon geliştirilmiştir. Bunlardan bazıları ;

1. Array_Splice();
2. Array_slice();
3. Array_unique();
4. Array_flip();
5. Array_merge();

Dizilerde Değişiklik işlemleri

Array_splice() fonksiyonu ile bir dizi içerisinden belirli bir indisten itibaren istenilen sayıda veriyi silmek veya araya veri eklemek mümkün olmaktadır.

Array_Splice(dizi , başlangıç , eleman sayısı, yeni elemanlar);

```
<?Php
$d =array( "Ankara", "İzmir", "Samsun", "Manisa");
array_splice($d,1,2);///ofset 1 den itibaren 2 tane silindi
print_r($d);///ekrana yazdıralım
        ?>
array_splice($d,1,2,'Sakarya');///ofset 1 den itibaren 2 tane silindi sakarya eklendi
```

Dizilerde Değişiklik işlemleri

Array_slice() fonksiyonu ile bir dizi içerisinden belirli bir indisten itibaren istenilen sayıda veriyi almak mümkün olmaktadır. Dizinin kesitini alabiliriz.

Array_Slice(dizi , başlangıç , eleman sayısı);

```
<?Php
$d =array( "Ankara", "İzmir", "Samsun", "Manisa");
print_r(array_slice($d,1,2));///ofset 1 den itibaren 2 tanesi alındı
print_r($d);///ekrana yazdıralım
        ?>
```

Dizilerde Değişiklik işlemleri

Array_unique() fonksiyonu ile bir dizi içerisindeki aynı elemanları silerek her elemandan bir tane kalmasını sağlayabiliriz.

Array_unique(dizi);

```
<?Php
$d =array( "Ankara", "İzmir", "Ankara", "Samsun", "İzmir", "Ankara", "İzmir", "Manisa");
print_r(array_unique($d));///aynı olan elemanlar silinir sadece biri alınır
print_r($d);///tüm diziyi ekrana yazdıralım
        ?>
```

Dizilerde Değişiklik işlemleri

Array_flip() fonksiyonu ile bir dizinin ofsetleri ile değerlerini yer değiştiririz.

Array_flip(dizi);

```
<?Php
$d =array( "Ankara", "İzmir", "Ankara", "Samsun", "İzmir",
"Ankara", "İzmir", "Manisa");
print_r(array_flip($d));///ofsetlerle değerler yer değiştirir
print_r($d);///tüm diziyi ekrana yazdıralım
        ?>
```

Dizilerde Değişiklik işlemleri

Array_merge() fonksiyonu ile birden fazla dizinin birleştirilerek tek dizi haline getirilmesi sağlanır.

Array_merge(dizi1, dizi2, dizi3,…);

```
<?Php
$d = array( "Ankara", "İzmir", "Ankara",);
$d1= array("Samsun", "İzmir");
$d2= array("Ankara", "İzmir", "Manisa");
print_r( array_merge($d,$d1,$d2)); ;///Dizileri birleştirir
print_r($d);///tüm diziyi ekrana yazdıralım
        ?>
```

Dizilerde Sıralama işlemleri

Dizilerde elemanlar arasında sıralama yapmak için PHP de bir çok fonksiyon geliştirilmiştir. Bunlardan bazıları ;

1. ksort();
2. krsort();
3. asort();
4. Array_reverse();
5. Array_map();

Dizilerde Sıralama işlemleri

ksort() fonksiyonu ile dizi elemanlarını ofsetlerine (keylere) göre küçükten büyüğe sıralayabiliriz.

```
ksort(dizi);
```

```
<?Php
$d = array( 2=>"Ankara", 0=>"İzmir", 1=>"Antalya",4=>"Samsun",
3=>"İzmit");
ksort($d);///Dizileri anahtara göre küçükten büyüğe sıralar
print_r($d);///tüm diziyi ekrana yazdıralım
        ?>
```

Dizilerde Sıralama işlemleri

krsort() fonksiyonu ksort fonksiyonunun tersini yapar yani keye göre büyükten küçüğe sıralar.

krsort(dizi);

```
<?Php
$d = array( 2=>"Ankara", 0=>"İzmir", 1=>"Antalya",4=>"Samsun", 3=>"İzmit");
krsort($d) ;///Dizileri anahtara göre büyükten küçüğe sıralar
print_r($d);///tüm diziyi ekrana yazdıralım
        ?>
```

Dizilerde Sıralama işlemleri

asort() fonksiyonu ile dizi elemanlarını değerlerine göre küçükten büyüğe sıralayabiliriz.

asort(dizi);

```
<?Php
$d = array( 2=>"Ankara", 0=>"İzmir", 1=>"Antalya",4=>"Samsun",
3=>"İzmit");
asort($d);///Dizileri değerlerine göre küçükten büyüğe sıralar
print_r($d);///tüm diziyi ekrana yazdıralım
        ?>
```

Dizilerde Sıralama işlemleri

Array_reverse() fonksiyonu ile dizi elemanlarını tersine çevirebiliriz yani tersi sırada sıralanır.

Array_revers(dizi);

```
<?Php
$d = array( 2=>"Ankara", 0=>"İzmir", 1=>"Antalya",4=>"Samsun",
3=>"İzmit");
Array_reverse($d);///Dizileri değerlerini tersine sıralar
print_r($d);///tüm diziyi ekrana yazdıralım
        ?>
```

Dizilerde Sıralama işlemleri

Array_map() fonksiyonu ile eşit sayıda elemana sahip birden fazla diziyi bir fonksiyon içerisinden geçirip birleştirerek yeni bir dizi oluşturabiliriz. Dizilerin eşit sayıda elemana sahip olması önemlidir.

Array_map(fonksiyon,dizi1,dizi2,dizi3…..);

```
<?Php
        function bolgeler($s,$b,$p)
        {
                return "$s Şehri $b Bölgesinde Bulunur ve Plakası $p dir";
        }
$d = array( 2=>"Ankara", 0=>"İzmir", 1=>"Antalya",);
$d1= array("İç Anadolu", "Eğe","Ak Deniz");
$d2= array("06", "35", "07");
print_r( array_map("bolgeler",$d,$d1,$d2));
print_r($d);///tüm diziyi ekrana yazdıralım
        ?>
```

Dizilerde Sıralama işlemleri

Array_map() fonksiyonunun çok sık kullanıldığı diğer bir alan ise formlardan gelen GET ve POST verilerini belirli fonksiyonlardan geçirdiğimiz yerlerdir. Eğer GET ve POST verilerini belirli fonksiyonlardan geçirmek istersek array_map fonsiyonu ile birlikte kullana biliriz. Örneğin Gelen verilerdeki boşukları kaldırmak için "trim()" fonksiyonundan geçirebiliriz.

Array_map(fonksiyon,dizi1,dizi2,dizi3…..);

```
<?Php
print_r( array_map("trim",$_GET));
        ?>
```

KAYNAKLAR

- İnternet ortamı
- PHP ve AJAX Haydar TUNA
- A'dan Z'ye PHP Rıza ÇELİK