

PHP - Notlar

Web Teknolojileri (PHP) Dersi İçin Aldığım, Final Dönemimde Bana Fazlaca Yardımcı Olan PHP Notlarımı Size De Yardımcı Olması Umuduyla Paylaşıyorum.

Bu Notlarda Dersin Öğretim Görevlisi [Mehmet Can Hanaylı](#)'nın Fazlaca Emeği Vardır. Notların Büyük Bir Kısımını Ondan Alınmıştır. İzin Verdiği Ve Destek Olduğu İçin Kendisine Teşekkür Ederim.

Konular

Genel Bilgi

GIT Tanım

Echo Kullanarak Ekrana Yazdırmak

Echo İçerisinde HTML Tagları Kullanmak

Echo Kullanarak Birleştirme Yapmak

Değişken Kullanımı

Tırnak İşaretleri Arasındaki Farklar

Temel Matematiksel İşlemler

Sık Kullanılan String Fonksiyonları

Metnin Parçalanarak Diziye Dönüştürülmesi (Explode)

Dizinin Metne Dönüştürülmesi (Implode)

İsim Soyisim Yazdırma

Diziler

Dizinin İçeriğinin Foreach Döngüsü İle Yazdırılması

Dizinin İçeriğinin For Döngüsü İle Yazdırılması

If-Else

Karşılaştırma Operatörleri

Mantıksal Operatörler

Switch Case İşlemi

Try Catch İşlemi

While Döngüsü

Do While Döngüsü

For Döngüsü

Foreach Döngüsü

Çok Boyutlu Foreach Döngüsü

While Döngüsü (Break)

Do While Döngüsü (Break)

For Döngüsü (Break)

While Döngüsü (Continue)

For Döngüsü (Continue)

Fonksiyonlar

Recursive Fonksiyon

Form İşlemleri

Genel Bilgi	GIT Tanım	Echo Kullanarak Ekranaya Yazdırmak	Echo İçerisinde HTML Tagları Kullanmak	Echo Kullanarak Birleştirme Yapmak	Değişken Kullanımı	Tırnak İşaretleri Arasındaki Farklar
İsim Soyisim Yazdırma	Diziler	Dizinin İçeriğinin Foreach Döngüsü İle Yazdırılması	Dizinin İçeriğinin For Döngüsü İle Yazdırılması	If-Else	Mantıksal Operatörler	Karşılaştırmalı Operatörler
Foreach Döngüsü	Çok Boyutlu Foreach Döngüsü	While Döngüsü (Break)	Do While Döngüsü (Break)	For Döngüsü (Break)	While Döngüsü (Continue)	For Döngüsü (Continue)

Web Teknolojileri Ve Programlama (PHP) Dersi

Notlarım

PHP'de kod yazmak için önceki gereksinimleriniz önemli değil fakat dikkat etmemiz gereken şey şu: Bir programlama dilinde web tabanlı kodlama yapacaksak öncelikle bilgisayarın bir websayfası gibi çalışmaya başlaması gerekiyor.

PHP'de yorumlama vardır. Bu yorumlamanın olabilmesi için bir servisin olması gerekiyor.

Bilgisayarlarımızda PHP kod yazabilmek için yapmamız gereken bazı temel çalışmalar var bunlardan ilki XAMPP WAMP AppServ gibi yazılımlar kullanmak. Bunlar en çok bilinen, bilgisayarın localde bir web sunucusu gibi çalışmasını sağlayan yazılımlardır.

Bu programlardan bir tanesini kurun. Ben XAMPP kurdum, sadece XAMPP yazarak websitesinden kurulumu yapabilirsiniz.

Bu programı kurduktan sonra artık masaüstünde çalışmayacağız. Programı açtıktan sonra APACHE, MySQL gibi web servisleri (modüller) göreceksiniz. Start dediğimizde bu modüller çalışmaya başlayacaktır. Çalışmıyorsa da hatada ne diyor çıktısı ne gibi sorular sorarak ilerlememiz gerekiyor.

Hata ne hatası sorusunu sormak çözüm değil, bu hatayı neyden kaynaklı veriyor çözümü nasıl olabilir sorularını sormamız gerek. Her başlatıp durdurduğunuzda PID(Process ID)'ler değişir bunun sebebi bunların unique (benzersiz) ID'ler olarak tutulmasıdır.

80 veya 443 portları web tarafında hizmet veren portlardır. SSL sertifikası aktif olduğu sürece websitesi bu portlar üzerinde çalışır.

Bilgisayarınızda örneğin ASP geliştiriyorsanız ve 80 portunda çalışıyorsanız PHP geliştirirken aynı port üzerinde çalışamazsınız. Çalışmaya çalışırsanız da PORT 80 in use by "unable to open process" şeklinde bir hata alırsınız. Eğer böyle bir hatayla karşılaşırsanız Config kısmına gelin ve Apache(http.conf)'a tıklayın. Aşağıya doğru indikten sonra Listen 80 kısmını değiştirin. Listen 8080 veya Listen 81 yapabilirsiniz.

Tarayınızda localhost:80 yazarsanız bu port default(varsayılan) olduğu için herhangi bir işlem olmaz.

Çalışan servisin config(konfigasyonunda) bir değişiklik yaparsanız ya değişiklikleri reload etmeniz gerek ki reload diye bir seçenek yok ya da restart edip(start-stop) yeniden başlatmanız gerek. Aksi takdirde error alırsınız. Gelen indexOf sayfası ise XAMPP içine koyduğunuz http docslardır. Bunlar

genel yayın dizimleridir. Programı ilk kurduğunuzda default (varsayılan) klasörlerle gelir. Bundan sonra yazmış olduğunuz tüm kodları XAMPP'in içindeki localhost içerisine atmanız gerekiyor.

GIT

Git versiyon kontrol sistemi- Git SCM Versiyon kontrollerini başlatmak için git init yazarız.

Git status çalışma dizinimizdeki değişiklikleri bize gösterir. Working directory bizim çalışma alanımız staging area takip ettiğimiz yer, local repo ise değişiklikleri nokta nokta yani zaman makinesi gibi kaydettiğimiz alandır. Bir belgeyi dosyayı ya da klasörü repoya eklemek için git add blabla.txt kullanılır örneğin. Git commit atmak için git commit -m "Belge Eklendi" yazabiliriz.

Git log ile ullanıcı detaylarını git log —oneline ile commitlerimizi görebiliriz.

Git add tüm değişiklikleri kaydeder. Reverse search ile daha önce eklediğimiz belgeleri görebiliriz. Git checkout ID kullanarak master adresini değiştirebiliriz.

Git remote add origin

Git remote -v diyerek push ve repoları yönetebiliriz.

Git push origin master yazarsak masterda yaptığımız çalışmaları origin'e iteklemiş yani göndermiş oluruz.

Dosya uzantısının php olması html kodları yazamayacağımız anlamına gelmiyor. Html uzantılı dosyada php yazabilmek için body kısmına

<?php kodlar?> yazılır.

Echo Ekran Yazdırma Komutudur.

```
<?php echo "Merhaba"; Echo "Selam"; ?>
```

Şeklinde yazılabilir.

Echo İçerisinde HTML Tagları Kullanılabilir:

```
<?php echo "Merhaba"; echo "<br>"; echo "<hr>"; echo "<h3>Adu-Bilgisayar  
Programcılığı</h3>"; echo "<h2>Adu-Bilgisayar Programcılığı</h2>"; echo "<h1>Adu-  
Bilgisayar Programcılığı</h1>"; ?>
```

Html ya da php içinde shift ve alt ve A tuşlarını kullanarak yarım satırı oluşturabiliriz. PHP'de de yorum satırı `<!-- -->` şeklindedir. `/**/` uzun açıklama satırı `//` kısa açıklama satırı-tek satır

Echo komutu:

- - Ekrana yazdırma işlemi yapar.
- - İçerisinde HTML etiketleri kullanılabilir.
- - Çift veya tek tırnak ile kullanılabilir.
- - Çift tırnak içerisinde değişken içerikleri gösterilir.
- - Nokta işareti(.) ile birleştirme işlemi yapılabilir.
- - / Kaçış karakteridir. Özel ifadelerden önce kullanılır. Özel ifadeleri pas geçmek için kullanılır. Akla gelebilecek her şey için kullanılabilir.

Echo Kullanarak Birleştirme Yapmak

```
Echo "Aydın MY0" . "Bilgisayar Programcılığı" . "<br/>";
```

Değişken Tanımlama Kuralları

Tüm programlama dillerinde dikkat edilmesi gereken şeylerdir.

1. Değişken isimleri \$ ile başlar.
2. Değişken isimleri sayısal bir ifade ile başlayamaz.
3. Değişken isimlerinde boşluk kullanılmaz. İki Kelime varsa birinci_sinif birinciSinif birinci-sinif örneklerdeki kullanımlar önerilir.
4. Değişken ve dosya isimlerinde Türkçe karakter kullanılmamalıdır.
5. Değişken isimlerinde büyük-küçük harf kullanımına duyarlıdır. \ \$sayi != \ \$Sayi
6. Değişken içerisinde eğer metinsel bir ifade varsa çift tırnak (") veya tek tırnak (') kullanılır.
7. Değişken içerisinde eğer sayısal bir ifade varsa tırnak kullanılmadan yazılır.
8. Değişken ismi, değişkenin içeriğini ifade etmelidir
9. Değişkenler ekrana echo komutu ile yazdırılır.

Bu kısımdan sonra basit bir uygulama yapıldı. index.php'den Uygulama 1'e bakmanızı tavsiye ederim.

PHP'de versiyon ve detay bilgisi phpinfo() ile öğrenilebilir. echo phpinfo();

Tırnak İşaretleri Arasındaki Farklar

```
$uni = "Adnan Menderes Üniversitesi";  
echo "Değişken İçeriği: $uni";  
echo "<br>";  
  
/* Değişken İsminin Yazdırılması */  
  
echo "1-Kazandığınız Üniversite: \$uni";  
echo "<br>";  
echo "2-Kazandığınız Üniversite:" . ' $uni';  
echo "<br>";  
echo '3-Kazandığınız Üniversite: $uni';  
  
/* Değişken İçeriğinin Yazdırılması */  
  
echo "<br>";  
echo "1-Kazandığınız Üniversite: $uni";  
echo "<br>";  
echo "2-Kazandığınız Üniversite:" . $uni;  
echo "<br>";  
echo "3-Kazandığınız Üniversite:" . "$uni";  
echo "<br>";  
echo '4-Kazandığınız Üniversite:' . "$uni";  
echo "<br>";
```

Temel Matematiksel İşlemler

Toplama İşlemi

```
$sayi1 = 10;  
$sayi2 = 20;  
echo "\$sayi1 + \$sayi2 = " . ($sayi1 + $sayi2);  
echo "<br>";  
echo "$sayi1 + $sayi2 = " . ($sayi1 + $sayi2);  
echo "<br>";
```

```
$sonuc = $sayi1 + $sayi2;  
echo "$sayi1 + $sayi2 = $sonuc";
```

Çıkarma İşlemi

```
$sayi1 = 10;  
$sayi2 = 20;  
echo "\$sayi1 - \$sayi2 = " . ($sayi1 - $sayi2);  
echo "<br>";  
echo "$sayi1 - $sayi2 = " . ($sayi1 - $sayi2);  
echo "<br>";  
$sonuc = $sayi1 - $sayi2;  
echo "$sayi1 - $sayi2 = $sonuc";
```

Çarpma İşlemi

```
$sayi1 = 10;  
$sayi2 = 20;  
echo "\$sayi1 * \$sayi2 = " . ($sayi1 * $sayi2);  
echo "<br>";  
echo "$sayi1 * $sayi2 = " . ($sayi1 * $sayi2);  
echo "<br>";  
$sonuc = $sayi1 * $sayi2;  
echo "$sayi1 * $sayi2 = $sonuc";
```

Bölme İşlemi

```
$sayi1 = 10;  
$sayi2 = 20;  
echo "\$sayi1 / \$sayi2 = " . ($sayi1 / $sayi2);  
echo "<br>";  
echo "$sayi1 / $sayi2 = " . ($sayi1 / $sayi2);  
echo "<br>";  
$sonuc = $sayi1 / $sayi2;  
echo "$sayi1 / $sayi2 = $sonuc";
```

Üst Alma İşlemi

```
$x = 4;  
$y = "2";  
//$y = (int)"2"; //Dönüşüm İşlemi  
echo "Değişken Tipi: " . gettype($x) . "<br>";  
echo "Değişken Tipi: " . gettype($y) . "<br>";  
echo "$x<sup>$y</sup>= " . pow($x, $y);
```

Karekök Alma İşlemi

```
$x = 123;  
$karekok = sqrt($x);  
echo "$x sayısının karekökü: $karekok";
```

Mutlak Değer İşlemi

```
$x = (-12);  
$mutlak = abs($x);  
echo "|$x| sayısının mutlak değeri: $mutlak";
```

Taban Değişim İşlemi

```
echo "27 sayısının 2'lik tabandaki karşılığı : " . base_convert(27, 10, 2);  
echo "<br>";  
$sayi = 255;  
$taban = 10;  
$yenitaban = 2;  
$sonuc = base_convert($sayi, $taban, $yenitaban);  
echo "($sayi)<sub>$taban</sub>=($sonuc)<sub>$yenitaban</sub>";
```

Mod İşlemi

```
$x = 20;  
$y = 4;  
$mod = fmod($x, $y);  
echo "$x mod $y : $mod";
```

Girilen Sayının Tek Mi Çift Mi Olduğunu Gösterir

```
$z = 71;  
echo "<br> $z sayısı: " . (fmod($z, 2) == 0 ? "Çifttir" : "Tektir");
```

Yuvarlama İşlemleri

```
$x = 15.313;  
$y = 15.385;  
  
echo "$x Bir ondalık basamak yuvarlaması sonucu: " . round($x, 1) . "<br>";  
echo "$y Bir ondalık basamak yuvarlaması sonucu: " . round($y, 1) . "<br>";  
  
echo "$x İki ondalık basamak yuvarlaması sonucu: " . round($x, 2) . "<br>";  
echo "$y İki ondalık basamak yuvarlaması sonucu: " . round($y, 2) . "<br>";  
  
echo "<h5>Yuvarlama İşlemleri (floor(x))</h5>";
```

En Yakın En Küçük Tam Sayıya Yuvarlar.

```
$x = 15.913;  
$y = 15.213;  
echo "$x Floor yuvarlaması sonucu: " . floor($x) . "<br>";  
echo "$y Floor yuvarlaması sonucu: " . floor($y) . "<br>";  
  
echo "<h5>Yuvarlama İşlemleri (ceil(x))</h5>";
```

Kendisinden Büyük, En Küçük Tam Sayıya Yuvarlar.

```
$x = 15.913;  
$y = 15.213;  
echo "$x Ceil yuvarlaması sonucu: " . ceil($x) . "<br>";  
echo "$y Ceil yuvarlaması sonucu: " . ceil($y) . "<br>";  
  
echo "<h5>Rasgele Sayı Üretme İşlemi (rand(x,y))</h5>";  
  
echo "10-100 Arasında Rasgele Sayı Üretildi: " . rand(10, 100);  
  
echo "<br>";  
for ($i = 1; $i <= 10; $i++) {  
    echo "$i. Sayı:" . rand(0, 100) . "<br>";  
}
```

Uygulama Ödevi

Kuyruk ve Yığın Veri yapılarının php üzerinde dizileri kullanılarak örneklendirmesini gerçekleştiriniz.

Rasgele üretilen sayının 50'den büyükse Geçtiniz-Notunuz: xx 50'den küçükse Kaldınız-Notunuz: xx uygulamasını gerçekleştiriniz.

```
use function PHPSTORM_META\type;  
  
$not = rand(10, 100); if ($not >= 50) { echo "Geçtiniz-Notunuz:$not"; } else { echo  
"Kaldınız-Notunuz:$not"; }
```

Sık Kullanılan String Fonksiyonları

```
$yazi = "Aydın Adnan Menderes Üniversitesi"; echo "<br> Değişken İçeriği: $yazi (" .  
gettype`($yazi) . ")";
```

İçeriğin Büyük Harfe Dönüştürülmesi

```
echo "<br> \$yazi İçeriğinin Büyük Harfle Yazılması: " . $buyuk_yazi =  
strtoupper($yazi); echo "<br> \$yazi İçeriğinin Büyük Harfle Yazılması: " . $buyuk_yazi  
= mb_strtoupper($yazi);
```

İçeriğin Küçük Harfe Dönüştürülmesi

```
echo "<br> \$yazi İçeriğinin Küçük Harfle Yazılması: " . $kucuk_yazi =  
strtolower($buyuk_yazi); echo "<br> \$yazi İçeriğinin Küçük Harfle Yazılması: " .  
$kucuk_yazi = mb_strtolower($buyuk_yazi);
```

İçeriğin İlk Harfinin, Büyük Harfe Dönüştürülmesi

```
echo "<br> \$yazi İlk Harfinin, Büyük Harfle Yazılması: " . $ilk_yazi =  
ucfirst($kucuk_yazi);
```

İçeriğin Tüm Kelimelerinin İlk Harfinin, Büyük Harfe Dönüştürülmesi

```
echo "<br> \$yazi Tüm Kelimelerinin İlk Harfinin, Büyük Harfle Yazılması: " . $kelime =  
ucwords($kucuk_yazi);
```

İçeriğin Harf Sayısı:

```
echo "<br> \$yazi İçerisindeki Harf Sayısı: " . strlen($yazi);
```

Ascii Char Karşılığı

```
echo "<br>Karakter Karşılığı: " . chr(109); echo "<br>Karakter Karşılığı: " . chr(64);
```

Uygulama 32-127 char karşılıkları:

```
echo "<br>Ascii Karakterleri: "; for ($i = 32; $i <= 127; $i++) { echo chr($i) . "-"; }
```

Metnin Parçalanarak Diziye Dönüştürülmesi (Explode)

```
$dizi = explode(" ", $yazi); echo "<br>"; print_r($dizi);
```

Uygulama Paragraftaki Kelime Ve Cümle Sayılarını alt alta yazdırınız.

```
$metin = "Lorem ipsum dolor sit amet consectetur adipisicing elit. Velit rerum voluptas odio amet fugiat nam numquam. Soluta fugiat quam perspiciatis, maiores ad non amet inventore officiis quibusdam totam ducimus adipisci. Lorem ipsum dolor sit amet consectetur adipisicing elit. Harum illo, repellendus maxime ut dolores quam esse quas ipsum eveniet quasi. Voluptatem, delectus aliquam praesentium atque quo adipisci illum voluptatibus minus?";
```

```
$kelimelerDizisi = explode(" ", $metin); $cumlelerDizisi = explode(".", $metin);
```

```
echo "<br>Kelime Sayısı: " . count($kelimelerDizisi); echo "<br>Cümle Sayısı: " . count``($cumlelerDizisi);
```

Uygulama Veri tabanından gelen 2021-12-25 değerini 25-12-2021 şeklinde ekrana yazdırınız.

```
echo "<br>"; $tarih = "2021-12-25"; $yenitarih = explode("-", $tarih); echo  
$yenitarih[2] . "-" . $yenitarih[1] . "-" . $yenitarih[0];
```

Dizinin Metne Dönüştürülmesi (Implode)

```
echo "<br>"; $aylar = ["Ocak", "Şubat", "Mart", "Nisan", "Mayıs"]; print_r($aylar);  
$aylarString = implode(" ", $aylar); echo "<br>String: $aylarString";
```

str_split(): Verilen bir metnin belirten sayı kadar ayrılarak bir dizi içeriğine aktarılması. Eğer sayı belirtilmezse metni harf harf böler.

```
echo "<br>"; $iban = "TR0010002000300040005000"; $yeniIban = str_split($iban, 4);  
echo "<pre>"; print_r($yeniIban); echo "</pre>";  
  
echo "<br>İmplode IBan: " . implode("-", $yeniIban);  
  
echo "<br>Foreach Iban: "; foreach ($yeniIban as $parca) { echo $parca . " "; }  
  
echo "<br>For Iban: "; for ($i = 0; $i <div count($yeniIban); $i++) { echo  
$yeniIban[$i] . " "; }
```

Php'de Diziler

1. Dizi oluşturmak için array() fonksiyonu veya [] kullanılabilir.
2. Dizi içerisinde farklı türden birçok değer bulunabilir.(int, string, double, obje)
(1,2,3,"Ali","Ahmet","125,20")

3. "echo \$diziAdi" ile dizi içeriği ekrana yazdırılmaz. Array to String Conversion hatası verir. Fakat dizinin örneğin 3 indisine sahip elemanı elemanı "echo \$diziAdi[3]" şeklinde yazdırılabilir.
4. Dizinin içerisindeki string ifadeler çift tırnak("") içerisinde yazılır.
5. Dizinin içerisindeki int ifadeler direk yazılabilir.
6. Dizinin elemanları birbirinden virgül(,) ile ayrılır.
7. Dizi içerisinde indisler varsayılan olarak 0'dan başlar.
8. Diziler içerisinde diziler tanımlanabilir.

```
$dizi1 = array(1, 2, 3, "Buse Nur", "Ocak", "Şubat"); $dizi2 = [1, 2, 3, "Buse Nur"];  
  
echo "<pre>"; print_r($dizi1); echo "</pre>";  
  
echo "Dizinin 1. elemanı: " . $dizi1[1] . "<br>"; echo "Dizinin 1. elemanın Türü: " .  
gettype``($dizi1[1]) . "<br>";  
  
echo "Dizinin 5. elemanı: " . $dizi1[4] . "<br>"; echo "Dizinin 5. elemanın Türü: " .  
gettype``($dizi1[4]) . "<br>";  
  
echo "Dizinin 0. İndisi: <b> $dizi1[0]</b><br>"; echo "Dizinin 1. İndisi: <b>  
$dizi1[1]</b><br>"; echo "Dizinin 2. İndisi: <b> $dizi1[2]</b><br>"; echo "Dizinin 3.  
İndisi: <b> $dizi1[3]</b><br>"; echo "Dizinin 4. İndisi: <b> $dizi1[4]</b><br>"; echo  
"Dizinin 5. İndisi: <b> $dizi1[5]</b><br>";
```

Dizinin içeriğinin foreach döngüsü ile yazdırılması.

```
echo "<h4>Dizinin İçeriğinin Foreach Döngüsü İle ``Yazdırılması-1 </h4>"; foreach  
($dizi1 as $eleman) { echo "$eleman <br>"; }
```

```
echo "<h4>Dizinin İçeriğinin Foreach Döngüsü İle ``Yazdırılması-2 </h4>"; foreach  
($dizi1 as $anahtar => $deger) { echo " $anahtar – $deger <br>"; }
```

Dizinin içeriğinin for döngüsü ile yazdırılması.

For şuradan başla, kontrol et, şuraya kadar gelsin mantığında çalışarak 3 değer alır.

```
for ($i = 0; $i < count($dizi1); $i++) { echo "$i: $dizi1[$i] <br>"; }
```

```
$sayilar = array(1, 2, 3, 4, 5);
```

```
$sayilar = array( array(1, 2, 3, 4, 5), array(10.5, 3.2, 1.7) );
```

```
echo "<pre>"; print_r($sayilar); echo "</pre>";
```

```
echo "Sayılar Dizisinin İçerisindeki İkinci Dizinin İkinci Elemanı: " . $sayilar[1][1]  
 . "<br>";
```

```
echo "Sayılar Dizisinin İçerisindeki Birinci Dizinin Üçüncü Elemanı: " . $sayilar[0][2]  
 . "<br>";
```

```
echo "<br>**\*\***\*\***\*\***\*\***\*\***\*\***\*\***\*\***<br>";
```

```
$sayilar2 = array( "Integer" => array(1, 2, 3, 4, 5), "Double" => array(10.5, 3.2,  
1.7) );
```

```
echo "<pre>"; print_r($sayilar2); echo "</pre>";
```

```
echo "Sayılar2 Dizisinin İçerisindeki İkinci ``Dizinin İkinci Elemanı: " .  
$sayilar2["Integer"] [1] . "<br>";
```

```
echo "Sayılar2 Dizisinin İçerisindeki Birinci Dizinin Üçüncü Elemanı: " .  
$sayilar2["Double"] [2] . "<br>";
```

```
$bilgiler = array( "id" => 1, "adi" => "Ahmet", "soyadi" => "Erimez", "gsm" =>
"5343333333", "cinsiyet" => "Erkek", "yas" => 30 );

echo "$bilgiler[adi] $bilgiler[soyadi] Hoşgeldiniz.``<br>"; echo $bilgiler["adi"] . "
" . $bilgiler["soyadi"] . " Hoşgeldiniz.";`

echo "<pre>"; print_r($bilgiler); echo "</pre>";

foreach ($bilgiler as $bilgi) { echo $bilgi . " - "; }

echo "<br><br><br><br>"; $bilgi = str_split($bilgiler["gsm"],3); foreach ($bilgi as
$value) { echo $value . " "; }
```

Php'de Diziler Devam

```
$isimler = array(); $isimler[] = "Mehmet"; $isimler[] = "Canan"; $isimler[0] =
"Kemal"; $isimler[] = "Nur";

echo "<pre>"; print_r($isimler); echo "</pre><hr>";

$isimler2 = array(); $isimler2["isim1"] = "Mehmet"; $isimler2["isim2"] = "Canan";
$isimler2["isim3"] = "Kemal"; $isimler2[] = "Nur"; $isimler2[] = "Gülcan";

echo "<pre>"; print_r($isimler2); echo "</pre>";
```

Uygulama Çıktı : Havvanur isimli öğrencinin web programlama, veri

tabanı, mobil programlama dersleri vardır.

```
$ogrenciDetay = array( "id" => 1, "adi" => "Havva", "soyadi" => "Yıldız", "bolum"
=> "Bilgisayar", "yas" => 20, "dersler" => array( "ders1" => "Web Programlama",
"ders2" => "Veri Tabanı", "ders3" => "Mobil Programlama" ), "sınıf" => 2,
"memleket" => "İzmir" );

echo "<pre>"; print_r($ogrenciDetay); echo "</pre> ";

echo "$ogrenciDetay[adi] $ogrenciDetay[soyadi] `` isimli öğrencinin " .
$ogrenciDetay["dersler"]["ders1"] . $ogrenciDetay["dersler"]["ders2"] .
$ogrenciDetay["dersler"]["ders3"] . " dersleri vardır. ";

echo "<hr>";
```

Uygulama: Aşağıdaki dizinin her elamını bir öğrenci detayına karşılık gelmektedir. Bu öğrencilere ait detayları aşağıdaki şekilde satır satır bir tablo oluşturup içerisine yazdırınız.

#id #adi #soyadi #bölüm #dersler #sınıf 1 havva Yıldız Bilgisayar Web Programlama 2 Veri Tabanı Mobil Programlama

```
$ogrenciDetaylari = array( "ogr1" => array( "id" => 1, "adi" => "Havva", "soyadi"
=> "Yıldız", "bolum" => "Bilgisayar", "yas" => 20, "dersler" => array( "ders1" =>
"Web Programlama", "ders2" => "Veri Tabanı", "ders3" => "Mobil Programlama" ),
"sınıf" => 2, "memleket" => "İzmir" ),

"ogr2" => array( "id" => 2, "adi" => "Kemal", "soyadi" => "Yıldız", "bolum" =>
"Bilgisayar", "yas" => 20, "dersler" => array( "ders1" => "Donanım", "ders2" =>
"Sunucu", "ders3" => "Mobil Programlama" ), "sınıf" => 2, "memleket" => "İzmir" ),
```

```
"ogr3" => array( "id" => 3, "adi" => "Kerim", "soyadi" => "Yıldız", "bolum" =>
"Matematik", "yas" => 25, "dersler" => array( "ders1" => "Cebir", "ders2" =>
"Homotopi Teori", "ders3" => "Homolojik Cebir" ), "sınıf" => 4, "memleket" =>
"Afyon" ) );

echo "<pre>"; print_r($ogrenciDetaylari); echo "</pre>";
```

Koşul İfadeleri (If-Else)

Koşul ifadeleri programlamanın bel kemiklerinden biridir. Nesne tabanlı programlama prensiplerine uygun bir programlama gerçekleştiriyoruz. Şuan uygun kodlarla uygun blokları öğrenmeye çalışıyoruz.

İf-Else Bloklarında öncelikle koşul belirtilir ve sonraki işlemler koşula göre devam eder.

Örneğin kullanıcı başarılı giriş yaparsa bu çalışsın başarısız işlem yaparsa bu kodlar çalışsın gibi durumlarda kullanılır.

Karşılaştırma Operatörleri

1. == (\$a == \$b) Eşit ise. (Genellikle Matematiksel bir karşılaştırma yapılır.)
2. === (\$a === \$b) Denk ise. (Genellikle String bir karşılaştırma yapılır.)
3. != (\$a != \$b) Eşit değil ise. (Genellikle Matematiksel bir karşılaştırma yapılır.)
4. !== (\$a !== \$b) Denk değil ise. (Genellikle String bir karşılaştırma yapılır.)
5. > (\$a > \$b) Büyük ise.
6. >= (\$a >= \$b) Büyük veya eşit ise.
7. < (\$a < \$b) Küçük ise.

8. \leq ($\$a \leq \b) Küçük veya eşit ise.

Mantıksal Operatörler

1. $!$ (olumsuzsa) ($!\$a$) $\$a$ 'nın değeri False ise True döner. Aksini yapar olumsuz mu diye sorar.
2. $\&\&$ - AND (ve) ($\$a \&\& \b) $\$a$ ve $\$b$ 'nin değeri olumlu ise True döner, iki koşul da sağlanmak zorundadır. Koşullardan bir tanesi false dönerse genel değer False olur.
3. $\|\|$ - OR (veya) ($\$a \|\| \b) $\$a$ 'nın veya $\$b$ 'nin herhangi birisinin değeri olumlu ise sonuç True olarak döner. Herhangi birisinin koşulu sağlanması genel dönüşü True yapar.

Not:

1. Elseif sürekli tekrar eden sorgu bloğu açar.
2. Else son kapanış koşuludur. (Herhangi bir şart belirtilmez. Hiçbir önkoşul gerçekleşmediğinde çalışacak son koşulu ifade eder.)

```
<?php
```

```
if (condition) { # code... } elseif (condition) { # code... } elseif (condition) { # code... } elseif (condition) { # code... } else { # code... }
```

```
`$sayi1 = 123;`  
`$sayi2 = 345;`  
`$metin = "Adü";`
```

Değer Varsa

```
if ($sayi1) {  
    echo "<br>If bloğu çalıştı : $sayi1";  
} else {  
    echo "<br>Else bloğu çalıştı : $sayi1";  
}
```

Değer Yoksa

```
if (!$sayi1) {  
    echo "<br>If bloğu çalıştı : $sayi1";  
} else {  
    echo "<br>Else bloğu çalıştı";  
}
```

Eşitlik Varsa

```
if ($sayi1 == $sayi2) {  
    echo "<br>Sayılar Eşit: $sayi1 = $sayi2";  
} else {  
    echo "<br>Sayılar Eşit Değil: $sayi1 != $sayi2";  
}
```

Eşitlik Yoksa Varsa

```
if ($sayi1 != $sayi2) {  
    echo "<br>Sayılar Eşit Değil: $sayi1 != $sayi2";  
} else {  
    echo "<br>Sayılar Eşit: $sayi1 = $sayi2";  
}
```

Denklik Varsa

```
if ("ogrenci" === "ogrenci") {  
    echo "<br>Girilen değerler aynı";  
} else {  
    echo "<br>Girilen değerler aynı değil.";  
}
```

Denklik Varsa

```
if ($metin === "Adü") {  
    echo "<br>Girilen değerler aynı : $metin";  
} else {  
    echo "<br>Girilen değerler aynı değil.";  
}
```

And Örneği

```
if (($sayi1 < $sayi2) and ("ogrenci" === "ogrenci")) {  
    echo "<br>Koşullar Sağlandı";  
} else {  
    echo "<br>Koşullar sağlanmadı.";  
}
```

OR Örneği

```
if (($sayi1 < $sayi2) or (15 > 25)) {  
    echo "<br>Koşullar Sağlandı";  
} else {  
    echo "<br>Koşullar sağlanmadı.";  
}
```

```
if ($sayi1 > $sayi2) {  
    echo "<br> \\\$sayi1 Değişkeni \\\$sayi2 değişkeninden büyüktür. $sayi1>$sayi2";  
} else {  
    echo "<br> \\\$sayi1 Değişkeni \\\$sayi2 değişkeninden küçüktür. $sayi1<$sayi2";  
}
```

```
if ($sayi1 >= $sayi2) {  
    echo "<br> \\\$sayi1 Değişkeni \\\$sayi2 değişkeninden büyüktür veya eşittir. $sayi1>=  
} else {  
    echo "<br> \\\$sayi1 Değişkeni \\\$sayi2 değişkeninden küçüktür. $sayi1<$sayi2";  
}
```

Uygulama:

1-Rasgele 0-100 arasında üretilen değerin tek veya çift olduğunu ekrana yazdırınız.

2-Rasgele 0-100 arasında üretilen değerin 50'den küçük olduğu durumlarda kaldınız, büyükse geçtiniz yazdırınız.

3-Doğum yılına göre 18 yaşından büyük olanların ehliyet alabileceğini kontrol eden if bloğunu yazınız.

Switch Case

switch : Koşul başlat.

case : Koşul Kontrol et.

break : Koşul kontrolünü sonlandır.

default : Koşul olumsuzlukları sonucu. Yani koşulun gerçekleşmediği durum.

Kullanım-1 switch (variable) { case 'value': # code... break; default: # code... break; }

Kullanım-2 switch (variable): case 'value': # code... break; default: # code... break; endswitch;

Switch Case İşlemi

```
<?php
```

```
$deger = 25; switch ($deger) { case ($deger < 10): echo "Sayı 10'den küçük."; break;
```

```
`case ($deger < 20):`  
    `echo "Sayı 20'den küçük.";`  
    `break;`  
`case ($deger < 30):`  
    `echo "Sayı 30'dan küçük.";`  
    `break;`  
`default:`  
    `echo "Varsayılan işlem, case'ler `gerçekleşmedi.";`  
    `break;`
```

```
}
```

```
echo "<hr>";
```

```
$ay = "Ocak"; switch ($ay) { case ($ay === "Ocak"): echo "$ay 1. Aydır."; break;  
case ($ay === "Şubat"): echo "$ay 2. Aydır."; break; case ($ay === "Mart"): echo  
"$ay 3. Aydır."; break; case ($ay === "Nisan"): echo "$ay 4. Aydır."; break; case
```

```
($ay == "Mayıs"): echo "$ay 5. Aydır."; break; default: echo " ($ay) - Listede böyle bir ay yok."; break; }
```

```
echo "<hr>";
```

```
$saat = 19; switch ($saat) { case ($saat >= 0) and ($saat <= 6): switch ($saat) { case ($saat == 1): echo "Saat Şuanda: $saat"; break; case ($saat == 2): echo "Saat Şuanda: $saat"; break; case ($saat == 3): echo "Saat Şuanda: $saat"; break; case ($saat == 4): echo "Saat Şuanda: $saat"; break; case ($saat == 5): echo "Saat Şuanda: $saat"; break; case ($saat == 6): echo "Saat Şuanda: $saat"; break; default: echo "Yanlış değer girdiniz."; break; } break;
```

```
case ($saat > 6) and ($saat <= 12): echo "Saat 6 ile 12 arasında mesai yok."; break;
```

```
`case ($saat > 12) and ($saat <= 18):`  
    `break;`
```

```
`default:`  
    ` echo " ($saat) - Default çalıştı.";`  
    ` break;`
```

```
}
```

Uygulama

- Switch case ile mesai saatlerine göre ekrana bir çıktı üreten kod bloğunu yazınız?

try = Hata oluşturabilecek kodlar için kullanılır.

catch = Hata oluştuğunda çalıştırılacak kodlar için kullanılır.

finally = Hata oluşsada oluşmasada çalıştırılacak kodlar için kullanılır.

throw = Hata oluşturabilecek kodlar için istisnai durumlarda kullanılır.

Exception = Tüm istisnalar için önceden tanımlı olan temel sınıfı çağırmak için kullanılırlar.

getMessage = Tüm istisnalar için oluşturulacak olan iletilerin atandığı veri tutuculardır.

```
try { Kod Blokları throw new Exception("Hata iletisi", 1);
```

```
    } catch (Exception İleti Ataması) {
```

```
    }finally{
```

```
}
```

Try Catch İşlemi

```
<?php
```

```
$deger1 = 25; $deger2 = 0;
```

```
try { if ($deger2 == 0) { throw new Exception("Sıfıra bölme işlemi  
`gerçekleştirilemez", 1); } else { echo "Bölme işleminin sonucu: " . $deger1 /  
$deger2; } } catch (Exception $sonuc) { echo $sonuc->getMessage(); }
```

```
echo "<hr>"; $isim = "Mehmet"; try { if ($isim == "Mehmet") { echo "Merhaba $isim,  
nasılsın"; } elseif ($isim == "Buse") { throw new Exception("Sen Buse'sin Mehmet  
Değilsin.", 1)`; } else { throw new Exception("Sen Kimsin...", 1); }
```

```
echo "<br>Throw oluşursa burası çalışmaz....";
```

```
    } catch (Exception $sonuc) { echo $sonuc->getMessage(); } finally { echo "<br>  
<br>Finally bloğu çalıştı."; }
```

?>

While Döngüsü

```
<?php
$basla = 1;
while ($basla <= 10) {
    echo "While Tur: $basla <br>";
    $basla++;
}

echo "Başla Değişkeninin Değeri: $basla";
?>
```

Do While Döngüsü

```
<?php
$basla = 1;
do {
    echo "Do-While Tur: $basla <br>";
    $basla++;
} while ($basla <= 10);
echo "Başla Değişkeninin Değeri: $basla";

?>
```

For Döngüsü

```
for ($i = 1; $i <= 10; $i++) { echo "For Tur: $i <br>"; } ?>
```

Foreach Döngüsü

```
$renkler = ["mavi", "sarı", "yeşil", "mor", "pembe"]; foreach ($renkler as $renk) {  
echo "$renk <br>"; } echo "<hr>"; foreach ($renkler as $anahtar => $deger) { echo  
"$anahtar > $deger <br>"; } ?>
```

Çok Boyutlu Foreach Döngüsü

```
"Havvanur", "soyisim" => "Yıldız", "bolum" => "Bilgisayar Prog.", "yas" => "20", "memleket" =>  
"İzmir", "dersler" => array( "Web Programlama", "Veri Tabanı", "Mobil Programlama" ), "numara"  
=> 123 );  
  
echo "<pre>";  
print_r($ogrenciBilgi);  
echo "</pre>";  
  
echo (is_array($ogrenciBilgi) == 1) ? "\$ogrenciBilgi bir dizidir" : "\$ogrenciBilgi b  
  
echo "<br><br><br><br>";  
foreach ($ogrenciBilgi as $ogrenciKey => $ogrenci) {  
  
    if (is_array($ogrenci)) {  
  
        foreach ($ogrenci as $dersKey => $ders) {  
            echo "Alt Dizi: $dersKey - $ders <br>";  
        }  
    } else {  
        echo "Ana Dizi: $ogrenciKey - $ogrenci <br>";  
    }  
}
```

?>

While Döngüsü (Break)

```
<?php
$basla = 1;
while ($basla <= 10) {
    echo "While Tur: $basla <br>";
    if ($basla == 5) {
        echo "5 bulundu. Döngüden çıkılıyor(while).";
        break;
    }
    $basla++;
}

echo "<br>Başla Değişkeninin Değeri: $basla";
?>
```

Do While Döngüsü (Break)

```
<?php
$basla = 1;
do {
    echo "Do-While Tur: $basla <br>";
    if ($basla == 5) {
        echo "5 bulundu. Döngüden çıkılıyor(do-while).";
        break;
    }
    $basla++;
} while ($basla <= 10);
```

```
echo "<br>Başla Değişkeninin Değeri: $basla";

?>
```

For Döngüsü (Break)

`for ($i = 1; $i <= 10; $i++) { echo "For Tur: $i
"; if ($i == 5) { echo "5 bulundu. Döngüden çıkılıyor(for)."; break; } } ?>` Continue döngünün bir bölümünün atlanarak devam edilmesini sağlar.

While Döngüsü (Continue)

```
<?php
$basla = 0;
while ($basla <= 10) {
    $basla++;
    if ($basla > 3 and $basla < 8) {
        continue;
    }
    echo "While Tur: $basla <br>";
}

echo "<br>Başla Değişkeninin Değeri: $basla";
?>
```

For Döngüsü (Continue)

```
for ($i = 1; $i <= 10; $i++) { if ($i > 3 and $i < 8) { continue; } echo "For Tur: $i  
<br>"; } ?>
```

Fonksiyonlar

Herhangi bir programlamada belirli bi veya daha fazla işi gerçekleştirmek üzere önceden hazırlanan küçük programlara verilen isimdir. Fonksiyonlar çağırılmadan çalışmazlar.

Kurallar:

1. Fonksiyon isimleri anlamlı olmalıdır.
2. Alfanoerik değler veya alt çizgi kullanılabilir.
3. Fonksiyon isimleri rakam ile başlayamaz.
4. Fonksiyon isimleri harf ve _ ile başlamalıdır.
5. Boşluk, Türkçe Karakterler ve özel karakterler kullanılamaz.

6. Php tarafından kullanılan isimler fonksiyon ismi olarak kullanılamaz.

7. Fonksiyon isimleri küçük/büyük harf duyarlı değildir.

8. Fonksiyonlar kapsama/etki alanına kurallara tabidir.

Yapısı:

```
function isim(Parametre veya Parametreler){  
    Kod Blokları....  
}
```

```
<?php  
$universite = "Aydın Adnan Menderes Üniversitesi";  
  
function bilgiver()  
{  
    echo "Merhaba Bilgiver fonksiyonu çalıştı."  
}  
  
bilgiver();
```

```
echo "<br>";  
function detayli_bilgiver($bilgi)  
{  
    echo "$bilgi'nde okuyan öğrencilerin bilgisi."  
}  
  
detayli_bilgiver($universite);
```

```
echo "<br>";  
function kutuphane()
```

```
{
    return "Adnan Menderes Kütüphanesi";
}

$deger = kutuphane();

echo $deger;


echo "<br>";
function kutuphane_kitap_sayisiver()
{
    $kitap_sayisi = 1500;
    return $kitap_sayisi;
    //return 1500;
}

echo kutuphane_kitap_sayisiver();


echo "<br>";
function kutuphane_kitap_sayisini_arttir($yeni_kitap_sayisi)
{
    $sonuc = 1500 + $yeni_kitap_sayisi;
    return $sonuc;
}


echo "Yeni Kitap Sayısı:" . kutuphane_kitap_sayisini_arttir(2500);

echo "<br>";

$isimler = [
    "Ali",
    "Ayşe",
    function () {
        echo "Dizi içerisindeki fonksiyon çalıştı.";
    }
];

$isimler[2]();
```

```
function ogrenci_bilgisini_gonder($isim, $soyisim, $gsm)
{
    $ogrenci_adi = $isim;
    $ogrenci_soyadi = $soyisim;
    $ogrenci_gsm = $gsm;
    $ogrenci_durum = "Mezun";

    return array($ogrenci_adi, $ogrenci_soyadi, $ogrenci_gsm, $ogrenci_durum);
}
```

```
$veri = ogrenci_bilgisini_gonder("Ali", "Kaçar", 112);
echo "<pre>";
print_r($veri);
echo "</pre>";
```

```
function ogrenci_bilgisini_gonder2($isim, $soyisim, $gsm, $durum = "Öğrenci")
{
    $ogrenci_adi = $isim;
    $ogrenci_soyadi = $soyisim;
    $ogrenci_gsm = $gsm;
    $ogrenci_durum = $durum;

    return array($ogrenci_adi, $ogrenci_soyadi, $ogrenci_gsm, $ogrenci_durum);
}
```

```
$veri = ogrenci_bilgisini_gonder2("Ali", "Kaçar", 112);
echo "<pre>";
print_r($veri);
echo "</pre>";
```

```
function not_hesapla($vize, $final)
{
    return ($vize * 0.3) + ($final * 0.7);
}
echo "<br>";
echo "Öğrencinin Not: " . not_hesapla(80, 100);
```

Recursive Fonksiyon

```
echo "Recursive Fonksiyon";

function islem_yap($sayi)
{
    if ($sayi <= 25) {
        echo $sayi . "<br>";
        islem_yap($sayi + 1);
    }
}

echo "<br>";

islem_yap(1);

/* Recursive olarak faktöryel hesabı yapan bir fonksiyon yazınız. */

?>
```

Form-1

```
<hr>

<form action="" method="GET">
    Adınız Soyadınız: <input type="text" name="adsoyad"><br>
    Email: <input type="text" name="email"><br>
    Telefon: <input type="text" name="tel"><br>
    Cinsiyet <input type="radio" name="cinsiyet">Erkek <input type="radio" name="cinsi

    Yaşınız: <select name="yas">
        <option value="20">20</option>
        <option value="21">21</option>
        <option value="22">22</option>
        <option value="23">23</option>
        <option value="24">24</option>
    </select>

    <input type="submit" value="Formu Gönder" name="form1">
```

```
</form>
```

Form-2

```
<hr>
```

```
<form action="#" method="POST">
```

```
    Adınız Soyadınız: <input type="text" name="adsoyad"><br>
```

```
    Email: <input type="text" name="email"><br>
```

```
    Dersler:<br>
```

```
    <input type="checkbox" name="dersler[]" value="web"> Web<br>
```

```
    <input type="checkbox" name="dersler[]" value="php"> Php<br>
```

```
    <input type="checkbox" name="dersler[]" value="sql"> Sql<br>
```

```
    <input type="submit" value="Formu Gönder" name="form2">
```

```
</form>
```

```
<?php
```

```
if (isset($_POST["form2"])) { $adsoyad = $_POST["adsoyad"]; $dersler=
$_POST["dersler"];
```

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

```
}
```

Form İşlemleri

<hr>

Aynı sayfada birden fazla form gönderilecek ise mutlaka formdaki butona name değeri ve

Form-1

```
<form action="" method="GET"> Adınız Soyadınız: <input type="text" name="adsoyad">
<br> Email: <input type="text" name="email"><br> Telefon: <input type="text"
name="tel"><br> Cinsiyet <input type="radio" name="cinsiyet">Erkek <input
type="radio" name="cinsiyet">Kadın <br>
```

```
Yaşınız: <select name="yas">
  <option value="20">20</option>
  <option value="21">21</option>
  <option value="22">22</option>
  <option value="23">23</option>
  <option value="24">24</option>
</select>
```

```
<input type="submit" value="Formu Gönder" name="form1">
```

```
</form>
```

Form-2

```
<form action="#" method="POST"> Adınız Soyadınız: <input type="text" name="adsoyad">
<br> Email: <input type="text" name="email"><br>
```

Dersler:


```
<input type="checkbox" name="dersler[]" value="web"> Web<br>
<input type="checkbox" name="dersler[]" value="php"> Php<br>
<input type="checkbox" name="dersler[]" value="sql"> Sql<br>
```

```
<input type="submit" value="Formu Gönder" name="form2">

</form>

<?php

if (isset($_POST["form2"])) { $adsoyad = $_POST["adsoyad"]; $dersler=
$_POST["dersler"];

foreach ($dersler as $ders) { echo "$ders <br>"; } }

?>
```