**FOREACH DÖNGÜSÜ**

Bir dizinin her elemanı için aynı işlemin yapılmasını sağlar. Sırasıyla dizi elemana ulaşır.

**Kullanımı:**

foreach($dizi ismi as $degiskenadi)

{

echo$degiskenadi;

}

**Örnek:**

<?php

$sayilar=array(10,11,12,15,17,16,35);

foreach($sayilar as $x)

{

echo $x;

}

?>

**Continue :**

Döngü yapısının içerisindeki işlemleri atlatarak işleme geçici olarak ara vererek bir sonraki adımdan devam etmesini sağlar.

**Örnek:**

for($i=0;$i<10;$i++)

{

if($i==5)

{

continue;

}

echo $i."<br>";

}

**Break :**

Fo-Foreach-while-do while-switchyapılarında kullanılır. Döngü yapısının içerisindeki diğer işlemlerin tümü atlanarak döngünün dışındaki ilk adımdan devam eder.

**Örnek :**

for($i=0;$i<10;$i++)

{

if($i==5)

{

break;

}

echo $i."<br>";

}

**FONKSİYONLAR**

İstediğimiz anda çağırabileceğimiz kod bloklarıdır.

**Kullanımı :**

Function ad ($parametreler)

{

Fonksiyon içinde yapılacaklar;

}

**Örnek : ( Parametre almayan ve değer döndürmeyen fonksiyon)**

function topla(){

$a=5;

$b=7;

echo $a+$b;}

topla();

echo”devam ediyo”;

**Örnek : (Parametre alan ve değer döndürmeyen fonsiyon)**

function topla($x, $y)

{

echo $x+$y;

}

$a=5;

$b=7;

topla($a,$b);

**Örnek : (Parametre almayan ve değer döndüren fonsiyon)**

function topla()

{

$a=5;

$b=7;

return $a+$b;

}

$deger=topla();

echo $deger;

**Örnek : (Parametre alan geriye değer döndüren fonksiyon)**

function topla($x, $y)

{

return $x+$y;

}

$a=5;

$b=7;

$deger=topla($a,$b);

echo $deger;

**Örnek:**  Üç değişken içerisnde yer alan metinleri bir cümle haline getiren kodları tasarlayın. Parametre alan ve değer döndürmyen bir fonsiyon ile bu işlemi yapın.

function yaz($x, $y, $z)

{

echo $x. " ".$y." ".$z;

}

$a="Ali";

$b="Ömer";

$c="EbuBekir";

yaz($a,$b,$c);

**Örnek:** Kendisine gönderilen yapılacak işleme(+,-,\*,/) göre yine kendisine gönderilen iki sayı ve +,/,\*,/ işlemlerine tabi tutan bir fonksiyon oluşturun. Parametre alan ve değer döndüren bir fonksiyon ile bu işlemi yapın.

if ($z=="+")

$sonuc=$x+$y;

else if($z=="-")

$sonuc=$x-$y;

else if($z=="\*")

$sonuc=$x\*$y;

else if($z=="/")

$sonuc=$x/$y;

return $sonuc;

}

$islem="+";

$a=5;

$b=7;

echo islem\_yap($a,$b,$islem);

**Fonksiyonlarda Dizi**

function islem($x,$y)

{ $snc[0]=$x+$y;

$snc[1]=$x-$y;

return $snc;}

$d=islem(10,20);

echo $d[0];

echo$d[1];

**Fonsiyonlar – Global:**

$x=5;

function islem($a)

{

global $x;

$sonuc=$x\*$a;

return $sonuc;

}

echo islem(10);

**Fonksiyona Varsayılan değer atama**

Fonksiyonun içerisine parametre olarak bir değer gönderilmediği zaman onun yerine geçebilecek bir değer almasını şu şekilde sağlarız.

function islem($a=5){

$sonuc=3\*$a;

return $sonuc;}

echo islem(10);

echo islem();

**Örnek:** Gönderilen iki sayıdan büyük olanı ekrana yazdıran yöntemi yazınız.

function buyuk($x,$y)

{

if($x>$y)

echo $x." Büyük ";

else if ($x<$y)

echo $y." büyük ";

else

echo " iki sayı eşit ";

}

$a=5;

$b=7;

buyuk($a,$b);

**Örnek:** Faktöriyelini alan fonksiyon yazın.

function buyuk($x,$y)

{

if($x>$y)

echo $x." Büyük ";

else if ($x<$y)

echo $y." büyük ";

else

echo " iki sayı eşit ";

}

$a=5;

$b=7;

buyuk($a,$b);

**Örnek:** Fonksiyona gönderilen iki sayıyı paremetre olarak alani bu iki sayıdan birincisinin, ikinci sayıya kuvvetini hesaplayan metotu tanımlayın ve programınızda kullanın.

function power($x,$y)

{

$carp=1;

for($i=1;$i<=$y;$i++)

{$carp=$carp\*$x;}

return $carp;

}

$a=3;

$b=2;

echo power($a,$b);