

Değişkenler

Bütün programlama dillerinde değişken kavramı vardır. Değişkenler verilerin taşınması ve tanımlanması için kullanılan bir veya birden fazla karakterle isimlendirilen kod parçalarıdır.

PHP’de değişken \$ ile başlar ve tanımlanır.

```
$meyve1 = "Elma"; //Elma değerini taşır.
```

```
$meyve2 = "Kiraz"; //Kiraz değerini taşır.
```

```
$sayi1 = 55; //55 değerini taşır.
```

PHP de değişkenler kelime,isim anlamlı veya anlamsız sözcükler olabilir. Bu sözcüklerde (, - , * + /) gibi harf olmayan karakter ve semboller yer almaz.

Tek alt çizgi _ bulunabilir ve başlayabilir.

Değişkenler sayı işe başlamaz.Sayılarla başlayan özel değişkenlerdir ileriki zamanlarda öğreneceksiniz. Büyük-Küçük harf duyarlıdır. \$a ve \$A farklıdır.

```
$123 = 60 ; // Yanlış
```

```
$iki-sayi = 1; // Yanlış
```

```
$ad soyad = "yılmaz"; // Yanlış
```

Değer Atama

Metin ve string türü değişkenler tek tırnak ya da çift tırnak içine alınır.Sayılar için gerek yoktur.

```
$Sayi = 123;
```

```
$kelime = 'kelime';
```

```
$kelime2 = "bu bir kelimedir " ;
```

Değişkenler ekrana yazdırılmak için **echo()** veya **print()** fonksiyonları kullanılabilir.

Php’deki veri türleri : Sayılar(number) , Dizge(String) , Dizi (Array) , Mantıksal(Boolean), Nesne(object).

Sayılar tam veya ondalıklı olabilir.

Nesne türü veriler sınıflardan new operatörü ile bir değişkene atama yapılması ile türetilir venesneler tek tırnak veya çift tırnak arasına alınmaz.

İşlemler

\$a = 5;

\$b = \$a + 10 ; //b değişkeninin değeri toplam 15 olacaktır.

\$c = \$b*10; // \$c değişkenin değeri 150 olacaktır.

\$a = \$a – 1 ; // \$a değişkeninin değeri 4 olacaktır.

<?php

\$x = 5;

\$y = 3;

\$kalan = \$x % \$y ;

echo \$kalan; //Ekрана 2 yazar.

?>

PHP aritmetik işlemleri soldan başlayarak önce çarpma sonra bölme, ardından toplama ve çıkarma şeklinde yapar. Parantez işlem önceliğinin sahiptir.

<?php

\$x = 6;

\$sonuc = (\$x / 3) + \$x ;

echo \$sonuc; //Ekрана 8 yazar.

?>

<?php

\$x = pow(3 , 4) ;

echo \$x ; // Ekрана 3 üssü 4 hesaplanarak yazılır.

?>

```
<?php
$х = sqrt( 64 );
echo $х;                // Ekрана 8 yazar.
?>
```

```
<?php
$х = abs( -5 );
echo $х;                // 5 yazılır.
?>
```

```
<?php
$х = pi();
echo $х;                // 3.1415926535898 yazılır.
?>
```

```
<?php
echo max(1, 30, 5, 6);  //30 yazdırılır
echo min(4, 100 , 1);   //1 yazdırılır.
?>
```

```
<?php
$х = 12;
$у = 1.45;
$sonuc = $х * $у;
echo $sonuc;            // Ekрана 17.4 yazar.
?>
```

Atama ve İşlem Operatörleri

```
$numara = 5.5 ;
$Anumara = $numara + 12 ;           //Sonuc 17.5 olur.
Bu işlem aynı zamanda daha kısa olarak aşağıdaki gibi yapılabilir.
$numara = 5.5 ;
$numara += 12;                      //Sonuc aynıdır.
```

```
$string = "Merhaba";
```

```
$string .= "Nasılsın?" ; //Birleştirerek Merhaba Nasılsın? Basılacaktır.
```

```
$a = 5 ;
```

```
$a++; //değer 1 artırılır sonuç 6 olur.
```

```
$a += 10; //son değer 10 ile toplanır ve 16 olur.
```

```
$a /=2; //son değer 2 ye bölünür ve sonuç 8 olur.
```

```
<?php
```

```
$numara = 55;
```

```
$numara %=12;
```

```
echo $numara; //55 in mod 12 si alınır ve sonuç olarak 7 ekrana bastırılır.
```

```
?>
```

Artırma ve Eksiltme İşlemi

```
<?php
```

```
$a = 5 ;
```

```
echo (++$a); // Ekrana 6 yazar.
```

```
?>
```

```
<?php
```

```
$a = 5 ;
```

```
echo ($a++); // Ekrana 5 yazar,yazdıktan sonra değeri 1 artırır.
```

```
?>
```

```
<?php
```

```
$a = 5 ;
```

```
echo (--$a); // Ekrana 4 yazar.
```

```
?>
```

Range Fonksiyonu

Belirli aralıkta bir dizi oluşturmaya yarar.

```
<?php
```

```
print_r ( range(3, 10, 2 ) );      // Array ( [0] => 3 [1] => 5 [2] => 7 [3] => 9 )
```

```
?>
```

Number_format Fonksiyonu

Karmaşık sayılarda işlemler yapmak için kullanılır. Sayıyı binlik bölümlere ayırır.

```
<?php
```

```
$n = 10843;
```

```
echo number_format($n);      //10.843 olarak yazar.
```

```
?>
```

```
<?php
```

```
$n = 10843;
```

```
echo number_format($n, 2);      // 10,843.00 olarak yazar.
```

```
?>
```

```
<?php
```

```
$n = 10843;
```

```
echo number_format($n, 2, ',', '.');      // 10.843,00 olarak yazar.
```

```
?>
```

Sayıları yuvarlamak için floor() , ceil() , round() fonksiyonları kullanılır.

```
<?php
```

```
echo floor(12.456);      // 12 olarak bastırır.
```

```
echo ceil ( 12.456);      // bir üste yuvarlar 13 olarak bastırır.
```

```
echo round (12.656);      //Noktadan sonraki sayı 5 ve üstü ise bir yukarı tam sayıya çevirir.  
Değilse bir alt sayıya çevirir. Sonuç : 13
```

```
?>
```

Trigonometri fonksiyonları cos() , sin(), tan() , acos vs. gibi fonksiyonlar kullanılır.

```
<?php
```

```
echo cos(30);          //0.15425144988 bastırılır.
```

```
echo sin(30);          //-0.988031624093 bastırılır.
```

```
echo tan(30);          //-6.40533119665 bastırılır.
```

```
?>
```

Rastgele sayı üreten fonksiyonları : mt_srand() , mt_rand() , srand() , rand()

```
echo rand();           //rastgele sayı yazdırılır.
```

```
echo mt_rand();        //rastgele sayı yazdırılır.
```

Veri Türünü Öğrenme ve Değiştirme

Verilerin türünü öğrenmek için **gettype()** fonksiyonu kullanılabilir.

```
$degisken = 1;
```

```
print gettype($degisken);          //integer yazacaktır.
```

```
$metin = "merhaba";
```

```
Print gettype($metin);           //String yazacaktır.
```

Veri türü settype() fonksiyonu ile değiştirilir.Stringler başka veri türüne dönüştürülemez.Integer veri tipleri double olabilir.

```
$a = 7.8;
```

```
settype($a , integer);
```

```
echo gettype($a);
```

Settype kullanılmadan doğrudan int,float,object ve string fonksiyonları ile veri tipi değişebilir.

```
<?php
$deger = "10.55 TL" ;
$a = (int) $deger;
echo $a;
echo '<br>';
$b = (float)$deger;
echo $b;
echo '<br>';
$c = (string)$deger;
echo $c;
?>
```

Bir değişkenin taşıdığı değeri bazı durumlarda işlevi bittiğinde onu tanımsız yapmak gerekebilir.Bunun için unset() fonksiyonu kullanılır.

```
<?php
$numara = 123;
echo $numara;
unset($numara);
echo $numara;
?>
```

String Veriler

String veriler; harfler,sayılar,boşluklar,semboller vs. gibi bir veya daha çok karakterin bir araya gelmesi ile meydana gelen veri tipine denir.

String veriler tek veya çift tırnak arasına alınarak atanır.

PHP çift tırnak içindeki değişkenlerin hepsini yorumlar.

Tek tırnak içindeki hiçbirşeyi yorumlamaz.

Hiç değiştirilmeden aynen yazıldıkları gibi bırakılır.

```
<?php
```

```
$fiyat = 23.00;
```

```
$kdv = 18;
```

```
echo "Tişört $fiyat TL + $kdv dir.<br/>"; //Çift tırnak olduğu için yorumlanır.
```

```
echo 'Tişört $fiyat TL + $kdv dir.<br/>'; //Tek tırnak olduğu için yorumLANMAZ.
```

```
?>
```

Printf ve Sprintf Fonksiyonları

Hem biçimlendirme hem de dönüştürme işlemlerini yapar. Kullanım alanı oldukça geniştir. Her iki fonksiyonda aynı işi yapar. printf() fonksiyonu biçimlendirmeyi ekrana yazar. sprintf()fonksiyonu ise biçimlendirmeyi echo veya bir değişkene atayarak yine **echo** ile yazdırır.

```
<?php
```

```
$fiyat = 50.99;
```

```
$kdv = "KDV dahil";
```

```
printf ('Ürünün fiyatı %s %d TL ' , $kdv , $fiyat);
```

```
?>
```

%s ----- Veri ne olursa olsun String olarak değerlendirir.

%d ----- Sayıyı tam sayı yapar.(Virgüllü sayıları tam sayı)

%x ----- Veriyi 16lık sayı yapar. (Hexadecimal)

%o ----- Veriyi 8lik sayı yapar. (Octal)

%f ----- Sayıyı noktalı sayı yapar.

```
<?php
$s = 'merhaba';
$t = 'merhaba programcılar';
$j = 65;

printf("[%0.3s]\n", $s);      // İlk 4 karakteri elde eder.
printf("[%10s]\n", $s);      // Sola boşluk koyarak 10a tamamlar.
printf("[% -10s]\n", $s);    // Sağa boşluk koyarak 10a tamamlar.
printf("[%010s]\n", $s);     // Sola 0 koyarak 10a tamamlar.
printf("[%'#10s]\n", $s);    // Sola #işareti koyarak 10a tamamlar.
printf("[% .10s]\n", $t);    // En fazla 10 karakter
printf("[%010d]\n", $j);     // Sola 0 koyar 10 basamak yapar.

?>
```

```
<?php
$color = sprintf("%06x", mt_rand(0, 0xFFFFFF));
echo "<style>
.renk {
    width:400px;
    padding:5px;
    border:1px solid silver;
    background:#$color
}
</style>";

?>

<div class="renk">

sprintf() fonksiyonu ve X dönüştürücüsü ile random renk elde ediyoruz.

</div>
```

Kısaca sprintf ya da printf fonksiyonlarını birden fazla parametre ile kullanabiliyoruz .

Sabit Değerler

PHP de sabit değerler define fonksiyonu ile kullanılır.

```
<?php
define ("MESAJ", "MERHABA BU MESAJDIR");
define ("isim", "ANIL");

echo MESAJ;
echo "<br>";
echo isim;
echo "<br>";
define ("SAYI", 500);
echo SAYI*7;
?>
```

Özel Sabit Değerler

Önceden tanımlanmış bazı önemli değerler aşağıdaki gibidir.

```
<?php
echo __FILE__;      // İçinde bulunduğu program dosyasının adını yol bilgisini dahil verir.
print "<br>";

echo __LINE__;      // Bulunduğu yerin satır sayısını verir.
print "<br>\n";

echo PHP_VERSION;   // PHP nin sürüm sayısını verir.
print "<br>\n";

echo PHP_OS;        // PHP nin çalıştığı işletim sistemini verir.
print "<br>\n";

echo __DIR__;       // Çalışan PHP Dosyasının bulunduğu dizini verir.
print "<br>\n";

echo dirname(__FILE__); // Aynı dir sabiti gibi çalışır.
?>
```

Sunucu (Server) Bilgileri

Sunucu bilgileri,sunucu üzerine kurulmuş dinamik dizi değişkenlere denir. Örneğin ziyaretçinin web tarayıcısının türü, IP adresi gibi yararlı ve kullanım alanı geniş değişkenlere sahiptir.

Bir web programcısı bu değişkenlerden yararlanarak istatistik tutabilir, rapor hazırlayabilir, en önemlisi sunucuya ulaşan bütün giriş ve çıkışlar hakkında bilgi edinebilir.

Değişken	Açıklama
DOCUMENT_ROOT	Sunucunun dosya yolu bilgisini verir.
HTTP_REFERER	Dışarıdan sunucuya ait adrese erişen sunucunun adresini verir.
HTTP_USER_AGENT	Tarayıcı türü ve versiyonu verir.
HTTPS	Bir istek SSL(HTTPS Protokolü ile) geldiyse true döndürür.
PATH_INFO	Çalıştırılan PHP Programının yolu bilgisi verir.
QUERY_STRING	GET metodu ile gönderilen anahtar değer çiftlerini elde eder.
REMOTE_ADDR	Sunucuya erişen ziyaretçinin IP adresini verir.
REQUEST_METHOD	GET,POST,PUT gibi iletişim metotlarını tespit eder.
REQUEST_URL	Çalıştırılan ve çağrılan dosyanın yolunu verir.
SCRIPT_NAME	Çalıştırılan veya çağrılan dosyanın tam fiziksel yolunu verir.
SERVER_NAME	Sunucu adını verir.
SERVER_PORT	Sunucunun port adresini verir.Genelde 80 dir.

\$_SERVER global değişkeni ile sunucu bilgilerini tümünü elde edebiliriz.

```
<?php
```

```
echo $_SERVER['HTTP_USER_AGENT'];
```

```
?>
```

```
<?php
```

```
echo "IP adresin : ". $_SERVER['REMOTE_ADDR'];
```

```
?>
```

```
<?php
```

```
echo "SUNUCUNUN IP ADRESİ : " .$_SERVER['SERVER_ADDR'];
```

```
?>
```

```
<?php
```

```
echo "PROGRAMIN ADI : " .$_SERVER['SCRIPT_FILENAME'];
```

```
?>
```

```
<?php
```

```
echo "İSTEK ADRESİ : " .$_SERVER['REQUEST_URI'];
```

```
?>
```

```
<?php
```

```
echo "kullandığınız metot : " .$_SERVER['REQUEST_METHOD'];
```

```
?>
```

Tüm bilgileri elde etmek için :

```
<?php
```

```
phpinfo();
```

```
?>
```

IP ADRESİ İLE ÜLKE VE ŞEHİR BULMA

```
<?php
```

```
$ip=file_get_contents("https://en.iponmap.com".$_SERVER['REMOTE_ADDR']);
```

```
preg_match_all("'<div style=\"line-height:1.5em\">(.*)</div>'si",$ip,$m);
```

```
echo str_replace("images","https://en.iponmap.com/images",$m[1][0]);
```

```
?>
```

Siteye nerden girildi ?

`$_SERVER['http_REFERER']` değişkeni sitemize ait bir link tıklanarak siteye ulaşıldığında gelinen adresin bilgilerini bize verir.

Örneğin ziyaretçiler en çok Google arama motorunu kullanır.

Aynı zamanda ziyaretçi hangi kelimeleri kullanarak arama yapmış gibi bilgileri de verir.

Index.php sayfasına yerleştirebilirsiniz :

```
<?php
if (strpos($_SERVER[HTTP_REFERER], "google") == true) {
    $url=parse_url(urldecode($_SERVER['HTTP_REFERER']));
    parse_str($url["fragment"],$sorgu);
    $kelime=$sorgu["q"]."\r\n";
    file_put_contents("kelimeler.txt",$kelime,FILE_APPEND);
?>
```

kelimeler.txt dosyasına bu kelimeleri bulup kaydedecektir.Dosyayı zamanla silmek gerekir.O yüzden veritabanı tercih edilir.

SÜPER GLOBAL DEĞİŞKENLER

Formdan bilgi alma,dosya yükleme,sistem bilgisi alma,ziyaretçi hakkında bilgi toplama,oturum ve çerez atama gibi birçok veriyi almamızı ya da aktarmamızı sağlar.

\$GLOBALS = Bir değişkeni her yerde kullanabilmeyi sağlayan süper global denenişkenlerdendir.Kullanılmak istenen bir değişkenin adı tanımlanarak kullanılır. Örneğin : \$veri = 6; değişkenin değerine ulaşmak için \$GLOBALS['veri'] şeklinde tanımlanması gerekir.

\$_POST = Formun gönderme metodu eğer method="POST" olaraan tanımlanmış ise bu formdan gelen değişkenlerin taşıdığı değeri elde etmemizi sağlar.

\$_GET = Formun gönderme metodu eğer method="GET" olaraan tanımlanmış ise bu formdan gelen değişkenlerin taşıdığı değeri elde etmemizi sağlar.

\$_COOKIE = Atanmış çerez bilgilerini okumak için kullanılan global bir değişkendir.

\$_SESSION = Oturum açmak için kullanılan global bir değişkendir.

\$_REQUEST = **\$_POST**, **\$_GET**, ve **\$_COOKIE** Global değişkenlerin değerlerin tek başına alabilen global değişkendir.

\$_FILES = Dosya yükleme değişkenidir. Her tür dosya bu değişken tanımlanarak sunucuya yüklenir.

\$_SERVER = Sunucu bilgilerini verilen anahtar bilgiye göre elde eder.IP adresi, gelineen adres, sunucu türü, tarayıcı tipi vs.

\$_ENV = Sistem değişkenlerini tutan bir global değişkendir. Dosya yolu, temp dizini, platform özelliği vb.

ÇEREZ OKUMA (COOKIES)

\$_COOKIE Değişkeni atanmış bir çerezi okumak için kullanılır. Çerezler genelde üyelik ve alışveriş sistemlerinde sıkça kullanılır.

Bir siteye adı ve şifre ile giriş yapıldığında bir sonraki gelişinizde sizi otomatik olarak kabul eden sistemler çerezlerle yapılır.

Kullanıcı bilgileri (şifreler hariç) küçük bir **text** dosyasında kullanıcının bilgisayarında saklanır ve üye bilgileri bu dosyadan okunarak kullanıcıya sunulur.

Alışveriş sitelerinde sepet bilgilerini ve kullanıcının incelediği bilgileri çerez olarak saklanır ve alışveriş süresince takip edilir , kullanılır.

setcookie (anahtar, değer, süre, yol, domain, https, http);

Çerez ataması öncesinde hiçbir şey ekrana yazdırılmaz. Aksi takdirde çerez ataması yapılmaz ve PHP Header üst bilgi uyarısı verir.

anahtar : Çerezin ismi

Değer : Çerezin değeri. Bu değer istemcinin bilgisayarında saklanır. Güvenlik nedeniyle şifre vb. gibi özel bilgiler değer olarak **verilmemelidir**.

Süre : Çerezin zaman aşımına uğrama süresi. Bu süreyi saniye cinsinden ekleyerek verilen değerdir. **time()+60*60*24*30** ile çerez 30 gün sonra zaman aşımına uğrar. Eğer 0 veya bir şey belirtilmez ise çerez oturum kapandığında(tarayıcı kapandığında) zaman aşımına uğrar.

Yol : Çerezin üzerinde etkin olacağı yol. '/' belirtilirse çerez alanın tamamında kullanılabilir olacaktır. Eğer '/müzik' denirse sadece o dizinde '/müzik/mp3/' gibi alt dizinlerde çalışır.

Domain : Çerezin kullanılabileceği alan adını tanımlar. yahoo.com ve alt alanlarında kullanılabilir yapmak için değer olarak '.yahoo.com' belirtmelisiniz. Değer 'mail.yahoo.com' yazılırsa sadece mail alt alanlarında kullanılabilir.

Https : Çerezin istemciye güvenli bir https bağlantısı üzerinden mi aktarılması gerektiğini belirtmek için kullanılır. TRUE belirtildiği takdirde, çerez sadece güvenli bağlantıya gönderilir. FALSE ön tanımlı değerdir.

Http: TRUE olduğu takdirde çerez sadece http protokolü üzerinden erişilebilir olacaktır. Yani çerez, JavaScript gibi betik dilleri tarafından erişilebilir olmayacaktır. Bu ayarlama XSS, saldırılarıyla kimlik hırsızlığı riskini etkin şekilde azaltmaya(tüm tarayıcılar tarafından desteklenmese de) yardımcı olabilir.

Süre ayarlamasında strptime('+30days'); fonksiyonu ile 30 gün için daha kolay bir şekilde yapılabilir.

strptime("+30 seconds")

strptime("+1 hours")

strptime("+1 week 2 days 4 hours 2 seconds")

1 saatliğine çalışan bir çerez.php oluşturun.

```
<?php
$deger = 'Deneme içerik';
//Çerezin adı Test, 1 saat süre ile çalışsın
setcookie("Test",$deger, time()+3600);
//Çerez için değer tanımlanması
$uye = 'Mustafa Bayram';
setcookie("ad",$uye,strtotime("+1hours"));
//Çerezi okuyacak dosyaya link verelim
echo ' <a href="oku.php">Çerezleri Oku</a> ';
echo ' <a href="Sil.php">Çerezleri SİL</a> ';
?>
```

Oku.php

```
<?php
error_reporting(0); //Uyarıları gizle
//Test adlı çerezi oku
echo "Test : " .$_COOKIE["Test"];
echo '<br>';
//ad adlı çerezi okuyalım
echo "ad : " .$_COOKIE["ad"];
?>
```

Sil.php

```
<?php
setcookie("Test","", time()-3600);
setcookie("ad","",strtotime("-1hours"));
//Çerezi okuyacak dosyaya link verelim
echo ' <a href="oku.php">Çerezleri Oku</a> ';
?>
```

HATA ALABİLECEK KOD

```
<!DOCTYPE html>

<html>

<head>

<title>

Çerez Ataması

</title>

</head>

<body>

<?php

$deger = 'Deneme içerik';

setcookie("Test",$deger,time()+3600);

/* 1 saatlik atandı */

echo '<a href="oku.php">Çerezi oku</a>';

?>

</body>

</html>
```

Eğer HTML içinde çerez yazılacaksa programın başına **ob_start()** ve en sonuna **ob_end_flush()** fonksiyonlarını yazmak gerekir.

```
<?php ob_start();?>

<!DOCTYPE html>

<html>

<head>

<title>

Çerez Ataması

</title>

</head>

<body>

<?php

$deger = 'Deneme içerik';
```



```
setcookie("Test",$deger,time()+3600);  
/* 1 saatlik atandı */  
echo '<a href="oku.php">Çerezi oku</a>';  
?>  
</body>  
</html>  
<?php ob_end_flush();?>
```
