

PHP

Php Nedir?

- PHP sunucu- taraflı, platformdan bağımsız ve HTML içine gömülebilir bir betik dilidir. Genel yapı ve yazılım kuralları yönünden C ve Perl dillerine çok yakın bir dildir.
- Rasmus Lerdorf tarafından hazırlanan bu dil, kendi web sitesine bağlı olan kişilerin takibini yapılması isteđi ile ortaya çıktı. Rasmus Lerdorf bu dile ilk başta "Personal Home Page" adını koydu (çünkü kendi kişisel web sayfası üzerinde kullanmıştı). Şu anda ise PHP sözcüğü, "PHP: Hypertext Preprocessor" sözcüklerinin baş harflerinden oluşan kısaltmayla tanımlanmaktadır.

Php'nin Avantajları

- Çok hızlı bir dildir.
- Rahat ve hızlı kodlama yapılabilir.
- Derlenmesine gerek yoktur.
- İnternette büyük bir geliştirici grubuna sahiptir.
- Php kodları platformdan bağımsız olduğu için Windows, Linux, MacOS gibi birbirinden farklı platformlarda çalışabilir.

Statik ve Dinamik Dil nedir?

- **Statik Dil:** Kaynak kodları kullanıcı tarafından görülebilen web programlama dilidir. Bu dil HTML olarak adlandırılmaktadır.
- **Dinamik Dil:** Kaynak kodları kullanıcı tarafından görülemeyen, önce sunucu tarafında çalışıp HTML koduna çevrilip kullanıcıya gönderilen dillerdir.
- En çok kullanılan diller PHP ve ASP.NET'dir.

PHP'nin çalışması için gerekenler

- Apache Web Sunucusu(<http://www.apache.org>)
- MySQL (Veritabanı) (www.mysql.com)
- PhpmyAdmin (Veritabanı yönetimi için)
- PHP Kütüphaneleri

Yukarıdakilerin toplu olarak kurulduğu paketler;

- **EasyPHP**
- **WampServer**
- **Xamp**
- **Apachetriad**

Php Program Yapısı

- Php kodları şu şekilde yazılır.

<?

.....

?>

Bu etiketler bazen kullanılamayabilir. Bunun için php.ini dosyası içinde short_open_tag değeri «on» yapılmalıdır.

- Yada

<?php

.....

?>

Ekrana mesaj yazdırma

- **Echo Komutu:** Genel kullanımını şöyledir:
 - Echo "mesaj";

- **Print komutu:** Genel kullanımını şöyledir:
 - `print("Hello World");`

Merhaba.php

```
<?php
```

```
echo "merhaba PHP";
```

```
?>
```

Çalıştırmak için;

Merhaba.php dosyasını Windows'ta Not Defteri'nde yazarak oluşturup;

❖ Wamp kullanıyorsanız ;
c:\Wamp\htdocs dizinine;

❖ EasyPHP kullanıyorsanız
c:\ProgramFiles\EasyPHP\htdocs
klasörüne kaydediniz.

NOT: Wamp ve EasyPHP'nin yeni versiyonlarında veya kurulum yaparken bu yollar değişmiş olabilir, genelde htdocs klasörüne kaydedilir.

Çalıştırmak için;

- <http://localhost/Merhaba.php>

ya da ;

- <http://127.0.0.1/Merhaba.php>

yazılmalıdır.

Merhaba2.php

- `<?php`
- `echo "İkinci PHP örneği";`
- `?>`

Çalıştırmak için Internet Explorer araç çubuğunda
<http://localhost/merhaba2.php>

Phpbilgi.php

- `<html>`
- `<head>`
- `<title></title>`
- `</head>`
- `<body>`
- **`<?php`**
- **`phpinfo () ;`**
`?>`
- `</body>`
- `</html>`

Çalıştırmak için Internet Explorer araç çubuğunda
<http://localhost/phpbilgi.php>

PHP'de Deęişkenler

- Deęişkenleri bir kap gibi veya bir bardak gibi dūşünebilirsiniz.
- Bardaęa suyu doldurup, boşaltabilir, sonra meyve suyu doldurabilirsiniz.
- Nasıl bir bardak birçok şeyi doldurup boşaltabiliyorsa deęişkene de aynı şekilde birçok deęer dolup boşalabilir.

PHP'de deęişkenler

- Deęişkenler \$ işareti ile başlar.
- Deęişken rakamla başlayamaz, deęişkenler içinde özel karakterler kullanılamaz. Ama alt çizgi (_) kullanabilirsiniz.
- Geçersiz deęişken tanımları;
\$3not, \$2x, \$a(, \$toplama?,
- Geçerli deęişken tanımları;
\$not, \$not1, \$toplama, \$notlar_toplami
- Deęişkenler içinde Türkçe karakterler kullanılmaz:
Örneęin; **ıİ , Şş , Çç , Ğğ , Öö , Üü**

Değişken tipleri

- **Tamsayı (integer)** veri tipi

Ör: \$i=400;

- **Kayan noktalı (real)** veri tipi

Ör: \$i=4.02;

tamsayi.php

- Ondalık kısmı olmayan sayılardır, pozitif ve negatif olabilirler. Örnekler: 200, -929, 283838, 2

-

`<?php`

```
$i=43;      echo ($i);
```

```
$x=-203;   echo ($x);
```

```
$y=39992; echo ($y);
```

- `?>`

Kayan noktalı (real) veri türü

- Ondalıklı kısmı bulunan sayılardır. Ondalıklı sayılara çeşitli isimlerde rastlayabilirsiniz; double, float, real gibi.

ondalik.php

- `<?php`

```
$i=23.83;      echo ($i) ;
```

```
$x=-22.7;    echo ($x) ;
```

```
$y=5432.38; echo ($y) ;
```

- `?>`

Karakter Dizileri (string) veri türü

- İsminden de anlaşılacağı gibi karakterlerden (harf, rakam, işaret) oluşan veri türüdür.

Rakam (0,1,2,3,4,5,6,7,8,9)

Harf (a,A,b,B,c,C,d,D,e,E,f,F,g,G,.....)

İşaret (+,#,%, -, &, [,], (,).....)

- String veri türü çift tırnak veya tek tırnak olarak tanımlanabilmektedir. Aradaki fark çok önemlidir.

Karakter Dizileri (string) veri türü

- a) Çift tırnak ile tanımlanan karakter dizileri
\$okul="Ankara Üniv.";
\$okul1="\$okul Bilgisayar Müh.";
echo \$okul1; // →Ankara Üniv. Bilgisayar Müh. yazar.
- a) Tek tırnak ile tanımlanan karakter dizileri
\$okul="Ankara Üniv.";
\$okul1='\$okul Bilgisayar Müh.';
echo \$okul1; // →\$okul Bilgisayar Müh. yazar.

Değişken tipleri

- **Dizi tipli (arrays)** değişkenler
- `$diziad[0]="ANKARA";`
- Not: Bu konu ilerleyen bölümlerde detaylıca incelenecektir.

degiskencift.php

Bu tanımlamada çift tırnak içinde başka bir değişken varsa değeri ekrana basılır. Örneğin aşağıdaki örnekte ekrana **Yaşasın Dünya** basılacaktır.

```
<?php
$degisken1 = "Yaşasın";
$degisken2= "$degisken1 Dünya";
echo $degisken2;
?>
```

Çalıştırmak için Internet Explorer araç çubuğunda
<http://localhost/degiskencift.php>

degiskentek.php

Bu tanımlamada tek tırnak içinde başka bir değişken varsa değeri ekrana **basılmaz**. Örneğin aşağıdaki örnekte ekrana **\$degisken1 Dünya** basılacaktır.

```
<?php
$degisken1 = "Yaşasın";
$degisken2= '$degisken1 Dünya';
echo $degisken2;
?>
```

Çalıştırmak için Internet Explorer araç çubuğunda

<http://localhost/degiskentek.php>

PHP'de Sabitler

- Bazen değerinin değiştirilmesini istemediğimiz durumlar olabilir. Örneği pi sayısı matematikte 3.14'e eşittir, bu değer değişken gibi değiştirilmemesi gerekir.
- Kullanımı Define (Sabitin ismi, değeri) şeklindedir. Örneğin;

```
Define ("pi", 3.14);
```

```
echo (pi);
```

- Burada dikkat ederseniz PHP'de değişken tanımlamak için kullanılan \$ işareti kullanılmadı.

Değişken tipini öğrenme

- **gettype() fonksiyonu:**

Belirtilen değişkenin türünü döndürür.

```
$virgullu=2.11;
```

```
Print (“türü:”.gettype($virgullu));
```

Değişken tipini değiştirme

- **settype() fonksiyonu:**

Belirtilen değişkenin türünü değiştirir.

```
$virgullu=2.11;
```

```
settype($virgullu,string);
```

```
Print (“türü:”.gettype($virgullu));
```

Değişkeni silme

- **unset()** fonksiyonu:

Belirtilen değişkeni hafızadan siler.

```
<?php  
$degisken = "Ankara Üniv.";  
unset($degisken);  
?>
```

Değişkenin değerini kontrol etme

- **empty() fonksiyonu:**

Bir değişkenin değerinin olup olmadığını kontrol eder.

```
<?php
$degisken = "";
if(empty($degisken))
{
echo "değişken değeri boş veya geçersiz değer";
}
?>
```

Değişken Değerlerinde Boşluk Silme

- **trim() fonksiyonu:**

Verinin sağında ve solunda yer alan tüm boşlukları temizlemek için kullanılır.

- `<?php`
- `$veri = " www.enverbagci.net ";`
- `echo trim($veri);`
- `?>`

Değişken Değerlerinde Boşluk Silme

- **rtrim()** fonksiyonu:

Verinin sağında yer alan tüm boşlukları temizlemek için kullanılır.

- `<?php`
- `$veri = " www.enverbagci.net ";`
- `echo rtrim($veri);`
- `?>`

Değişken Değerlerinde Boşluk Silme

- **ltrim() fonksiyonu:**

Verinin solunda yer alan tüm boşlukları temizlemek için kullanılır.

- <?php
- \$veri = " www.enverbagci.net ";
- echo ltrim(\$veri);
- ?>

Değişken Kontrol Fonksiyonları

- `is_string()`
- `is_int()`
- `is_float()`
- `strval()`
- `intval()`
- `floatval()`

is_string() fonksiyonu

- Değişken string türünde mi diye bakar. Belirtilen değişken string türündeyseniz **TRUE** aksi takdirde **FALSE** döner.
- ```
<?php
if (is_string("23")) {
 echo "dizgedir\n";
} else {
 echo "dizge değildir\n";
}
?>
```

# is\_int() fonksiyonu

- Değişken bir tamsayı mı diye bakar. Belirtilen değişkenin değeri tamsayı ise **TRUE** aksi takdirde **FALSE** döndürür.
- ```
<?php  
if (is_int(23)) {  
    echo "tamsayıdır\n";  
} else {  
    echo "tamsayı değildir\n";  
}  
?>
```

is_float() fonksiyonu

- Değişken float türünde mi diye bakar. Belirtilen değişken float türündeysen **TRUE** aksi takdirde **FALSE** döner.

```
• <?php
  if (is_float(27.25)) {
    echo "float türünde\n";
  } else {
    echo "float türünde değil\n";
  }

?>
```

strval() fonksiyonu

- Sayısal değişkeni stringe çevirir.
- <?php
- \$sayi=90;
- echo gettype(\$sayi);
- echo gettype(strval(\$sayi));
- ?>

- Ekran çıktısı: integer string

```
// Bu şekilde de yazılabilir.  
<?php  
$sayi=90;  
echo gettype($sayi);  
$sayi=strval($sayi);  
echo gettype($sayi);  
?>
```

intval() fonksiyonu

- String (alfasayısal) bir değeri sayısal bir değere dönüştürür.
- ```
<?php
$deyim = "42a55bc4t"; //sol başta rakam yoksa sıfır değeri olur.
$sayi = intval($deyim);
echo $sayi; // 42 olacaktır.
?>
```
- \$deyim=40.9 gibi float olsa sonuc 40 olurdu.

# floatval() fonksiyonu

- String (alfasayısal) bir değeri float bir değere dönüştürür.
- <?php
- \$deyim = "60.34asd";
- \$sayi = floatval( \$deyim );
- echo \$sayi; // 60.34 olacaktır.
- ?>

# PHP'de Operatörler (1)

- + Toplama Operatörü

```
$a=2 ; $a=$a +2 ; echo $a ; (Sonuç :4)
```

- - Çıkarma Operatörü

```
$a=6 ; $a=$a -3 ; echo $a ; (Sonuç :3)
```

- \* Çarpma Operatörü

```
$a=4 ; $a=$a *2 ; echo $a ; (Sonuç :8)
```

- / Bölme Operatörü

```
$a=2 ; $a=$a /2 ; echo $a ; (Sonuç :1)
```

- % Mod alma Operatörü (iki sayının bölümünden kalan)

```
$a=8 ; $a=$a %3 ; echo $a ; (Sonuç :2)
```

# PHP'de Operatörler (2)

- += operatörünün kullanımı;

```
$a = $a + 1; $a+=1;
```

- -= operatörünün kullanımı;

```
$a = $a -1; $a-=1 ;
```

- \*= operatörünün kullanımı;

```
$a = $a * 2; $a*=2;
```

- /= operatörünün kullanımı;

```
$a = $a / 2; $a/=2;
```

# PHP'de Operatörler (3)

- ++ Operatörü

```
$a=2;
```

```
echo ++$a; (Sonuç:3)
```

```
$a=2;
```

```
echo $a++; (Sonuç:2)
```

# PHP'de Operatörler (4)

- -- Operatörü

```
$a=2;
```

```
echo --$a; (Sonuç:1)
```

```
$a=2;
```

```
echo $a--; (Sonuç:2)
```

# PHP'de Operatörler (5)

- **Atama Operatörü**

```
$ad="Ali"
```

```
$a=2;
```

```
$toplam=$a+$b;
```

# PHP'de Operatörler (6)

- String Birleştirme Operatörü

```
$ad="Ali"
```

```
$soyad="Ak";
```

```
$adsoyad=$ad.$soyad;
```

Ekrana AliAk Yazar.

# PHP'de Operatörler (7)

- **Mantıksal Operatörler**
- And Kullanımı: &&
- Or Kullanımı: ||
- Not Kullanımı: !
  
- Bu operatörler genellikle Şartlı yapılarda kullanılır.
- Örnekleri şartlı yapılar konusunda verilecektir.

# PHP'de Operatörler (8)

- **Karşılaştırma Operatörleri**
- == Eşittir
- != Eşit değil
- < Küçüktür
- > Büyüktür
- <= Küçük eşittir
- >= Büyük eşittir.
  
- Bu operatörler de yine bir önceki slaytta belirtildiği gibi karşılaştırma işlemlerinde kullanılır.

# Toplam.php

```
<?php
$sayi1=5;
$sayi2=3;
$toplam=$sayi1 + $sayi2;
echo "Toplam sonucu:". $toplam;
?>
```

- Çalıştırmak için Internet Explorer araç çubuğunda <http://localhost/toplam.php>

# İslem.php

```
<?php
$sayi1=5;
$sayi2=3;
$bolsonuc=$sayi1/$sayi2;
$cikarsonuc=$sayi1 - $sayi2;
$modsonuc= $sayi1 % $sayi2;
$carpsonuc = $sayi1 * $sayi2;

printf("bölüm sonucu:%d" . $bolsonuc . "
");
printf("çıkarma sonucu:%d" . $cikarsonuc . "
");
printf("mod sonucu:%d" . $modsonuc . "
");
printf("çarpım sonucu:%d" . $carpsonuc . "
");
?>
```

Çalıştırmak için Internet Explorer araç çubuğunda  
<http://localhost/islem.php>