

PHP Nedir?

PHP, dünyada en çok kullanılan web programlama dillerinden biridir. Biraz daha teknik bir açıklama yapmak gerekirse şöyle söylenebilir. PHP, sunucu taraflı çalışan bir betik ve programlama dilidir. Önceleri kişisel web sitesi yapmak amacıyla Rasmus Lerdorf tarafından geliştirilen php, gördüğü büyük ilgiyle çok yaygın kullanılan bir dil haline gelmiştir. Bu nedenle Personel Home Pages ifadesini değiştirerek artık PHP Hypertext Preprocessor rekürsif ifadesini kullanmaya başlamıştır.

Personel **H**ome **P**ages : Kişisel web sayfası

PHP **H**ypertext **P**reprocessor: Üstün Yazı Önışlemcisi olarak çevrilen ve kendi tanımı içerisinde kendini barındıran (recursif) bir ifadedir.

PHP, dünyada en çok kullanılan web sunucusu olan apache üzerinde çalışan ve gösterdiği üstün performans ile web dünyasında çok önemli bir yere sahip olan bir dildir. (Bazı eklentilerle IIS üzerinde de çalışabilmektedir.)

PHP ile ayrılmaz bir bütün olan MySQL veritabanı ile etkileşimli web sayfaları yapmak oldukça kolaydır. PHP ve MySQL o kadar uyumlu çalışır ki ikisi ayrılmaz iki kardeş sayılırlar. Bu, veritabanı olarak sadece MySQL kullanılabileceği anlamına gelmiyor tabiki. PHP ile Access, Microsft SQL Server, PostgreSQL ve Oracle gibi bir çok veritabanına da bağlantı kurulabilir. Ancak en uyumlu çalıştığı veritabanı sunucusu MySQL'dir.

Gelişim sürecine göre günümüzde PHP5 versiyonu mevcuttur. Öncesinde ise PHP4 ve PHP3 versiyonları vardır. PHP5'in bunlardan en önemli farkı nesneye yönelik programlamaya imkan vermesidir.

PHP kodları çoğu zaman HTML kodları arasına gömülü olarak yazıldığından öncesinde HTML kodlarının bilinmesinde fayda vardır.

PHP ile Neler Yapılabilir?

PHP ile yapılabilecek bazı uygulamalar şunlardır

- Kullanıcı ile etkileşimli çalışan web sayfaları
- Duruma göre kişiye özel içerik görüntüleme
- Web tabanlı veritabanı uygulamaları (Forumlar, İçerik Yönetim Sistemleri, Sınav Sayfaları, Takip sistemleri, ...)
- Anketler
- E-Ticaret Siteleri
- Dosya yönetimi
- E-Posta uygulamaları
- Grafik işleme
- Matematiksel hesaplamalar

ve daha sayamadığımız bir çok uygulama PHP ile yapılabilmektedir.

Avantajları

- Çok hızlı bir dildir
- Öğrenilmesi çok kolaydır
- Rahat ve hızlı kodlanabilir. HTML arasına kolaylıkla yerleştirilebilir.
- Hep söylenegelen script dilidir ifadesinin gereği derlenmeden çalışan bir dildir.
- Ücretsiz bir dildir. Rahatlıkla internet üzerinden edinilebilir.

- Büyük bir geliştirici topluluğuna sahip olduğundan kaynak açısından oldukça zengin bir dildir
- Apache'nin yanında farklı web sunucular (IIS, PWS) üzerinde de çalışabilir.
- Windows, UNIX ve LINUX sistemlerde çalışabildiğinden platform bağımsız bir dildir
- Başta MySQL olmak üzere Access, MS SQL, PostgreSQL ve Oracle gibi bir çok veritabanı ile bağlantı kurabilmektedir.
- Kod kütüphaneleri açısından oldukça zengindir.
- Açık kaynak kodlu özgür bir dildir. Kullanıcılar tarafından geliştirilmeye açıktır.
- Açık kaynak olmasının gereği oldukça güvenli ve hızlıdır. Çünkü bir hata kendisini izleyen binlerce gözden kaçamaz.

Bunların yanında söyleyebileceğimiz dezavantajları ise şunlardır.

- Yapılan basit bir hata bir çok sayfada hataya sebep olabilir.
- Ayrıca büyük programlarda cevap süresi gecikebilir
- Güncelleme işlemi web sunucusu ile beraber yapılmalıdır.

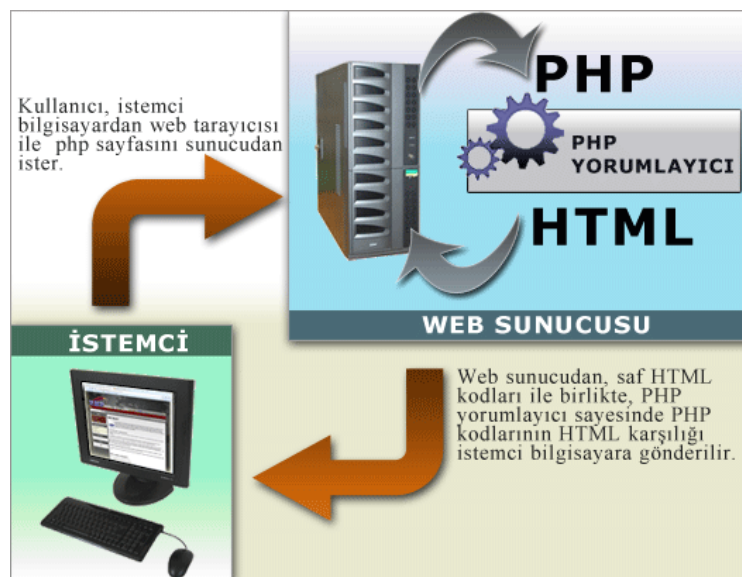
Ancak yukarıda sayılan bu kadar avantajın yanında bunlar makul kabul edilebilir.

PHP'nin Çalışması

PHP, HTML içine gömülerek yazılan ve sunucu taraflı (server side) çalışan bir dildir. Bunun anlamı şudur:

Biz, php kodlarımızı HTML etiketleri arasına yerleştiririz. Kullanıcılar ise tarayıcısının adres çubuğundan, içinde php kodları bulunan bu sayfayı istediği zaman bu istek web sunucuya iletilir. Bu isteği değerlendiren sunucu istenen dosyadaki HTML kodlarını olduğu gibi kullanıcıya gönderir ancak bizim araya eklediğimiz php kodlarına gelince bu kodları php yorumlayıcıya gönderir. Php yorumlayıcı php kodlarını yorumlayıp çalıştırdıktan sonra web sunucuya HTML karşılığını gönderir. Web sunucu php kodlarına karşılık gelen HTML kodlarını aldığı anda bunları kullanıcıya yollar.

Sonuçta kullanıcılara gönderilen sadece HTML kodlarıdır. Onlar hiçbir zaman bizim sayfaya eklediğimiz php kodlarımızı göremez. PHP kodlarının HTML karşılığını görürler.



Şekil: PHP'nin Çalışması

Hazırladığımız PHP sayfalarını çalıştırmak ve denemek için o dosyayı gerçek sunucuya göndermek ve bunu her değişiklikte tekrar tekrar yapmak çok zor ve gereksiz bir iştir. Bunun yerine hazırladığımız PHP sayfalarını kendi bilgisayarımıza kurduğumuz web sunucu yazılımı ile test eder ve son halini aldıktan sonra gerçek sunucuya göndeririz.

Kendi bilgisayarımıza kuracağımız web sunucu **localhost** olarak adlandırılır.

Localhost Kurulumu

Hazırladığımız PHP sayfalarını gerçek sunucuya yüklemekten önce kendi bilgisayarımızda test etmek için web sunucu yazılımını ve beraberinde gerekli diğer programları bilgisayarımıza kurmamız gerekmektedir. Kendi bilgisayarımıza kuracağımız web sunucu **localhost** olarak adlandırılır.

Kendi bilgisayarımızda php çalıştırmak için aşağıdaki programları kurmamız gerekmektedir.

- Apache (Web Sunucusu)
- PHP (PHP Dili)
- MySQL (Veritabanı Sunucusu)
- phpMyAdmin (Veritabanı sunucusunu yönetmek için kullanılan web arayüzü)

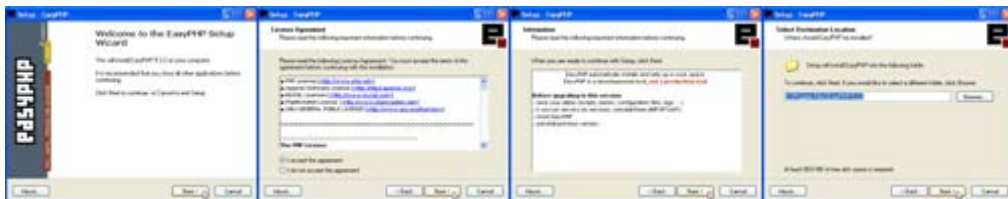
Yukarıda yazılı programları ayrı ayrı bilgisayarımıza kurduktan sonra yapılandırmasını yaparak bilgisayarınızda php çalıştırabilirsiniz. Bunun nasıl yapılacağı ile ilgili bir çok kaynak bulabilirsiniz. Ancak bunu yaparken yaşanan bir çok sorun ve zorluk yüzünden tüm bu paketleri ve yapılandırmayı kapsayan hazır kurulum paketleri (programları) daha çok kullanılmaktadır. Bu programlardan birini kurarak bir kaç dakika içinde bilgisayarınızda php yazıp çalıştırabilirsiniz.

PHP çalıştırmak için kullanılan hazır programlardan ençok kullanılanları şunlardır:

- EasyPHP
- Xampp
- AppServ
- Wamp



Bunlardan birini bilgisayarınıza kurarak php yazmaya başlayabilirsiniz. Biz bunlardan EasyPHP kurulumunu ve yapılandırmasını anlatacağız.





EasyPHP kurulumunu yaptıktan ve çalıştırdıktan sonra görev çubuğunda saatin olduğu kısma simgesi geliyor. Bu simgeye çift tıkladığımızda karşımıza gelen pencereden easyPHP'yi yönetebiliyoruz. Bazı durumlarda easyPHP'yi çalıştırdığımız halde apache yada mySQL başlatılmadığından elle bu servisleri başlatmamız gerekebilir. Bunu için de yine bu pencereyi kullanıyoruz.



Kurulumu yaptıktan sonra hazırladığımız php sayfalarını **C:\Program Files\EasyPHP-5.3.2\www** dizinine kaydetmeliyiz. Kaydettiğimiz sayfayı test etmek için kullandığımız tarayıcıyı açıp adres çubuğuna;
http://localhost/sayfaadi.php yazmak yeterlidir.

Eğer daha önceden asp yada aspx için IIS'yi kurduysanız yani localhost yüklediyseniz bizim php için yüklediğimiz localhost ile çakışmaktadır. Bunun sebebi her iki sunucu da web sayfalarını istemcilere 80 nolu porttan göndermek istemeleridir. Bu durumda sunuculardan birinin port numarasını 81 olarak değiştirmemiz gerekecektir. Sonradan kurduğumuz apache web sunucusunun portunu 81 olarak değiştirmek için aşağıdaki pencerede görünen seçeneği kullanmalısınız.



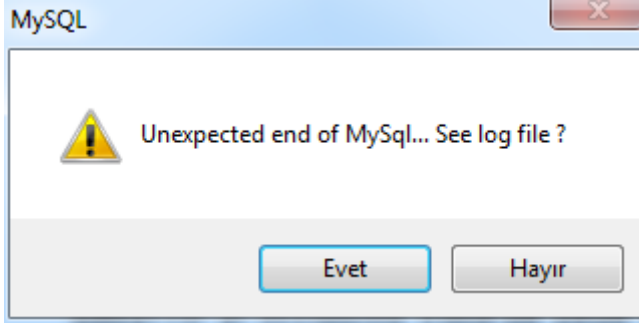
Bu seçeneğe tıkladığımızda karşımıza gelen metin dosyasında 56.satırdaki (farklı versiyonlarda satır numarası değişebilir) Listen 127.0.0.1:80 satırını Listen 127.0.0.1:81 olarak değiştirip kaydetmek yeterlidir.

```
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 127.0.0.1:81
```

Bu değişiklikten sonra kaydettiğimiz sayfaları çağırırken tarayıcımızın adres çubuğuna;
http://localhost:81/sayfaadi.php şeklinde bir adres yazmamız gerekmektedir.

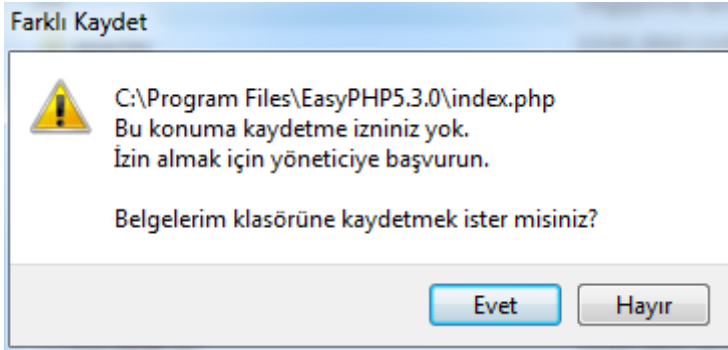
Yaşanan Bazı Sorunlar

EasyPHP'yi Windows Vista'da yada Windows 7'de kurup çalıştırdığımızda bazı bilgisayarlarda MySQL ile ilgili bir hata mesajı almakta ve MySQL veritabanı sunucusunu çalıştırılamamaktadır.



Bunun sebebi Windows Vista'da ve Windows 7'deki güvenlik önlemleridir. Bunu aşmak için EasyPHP'yi, sağ tıklayarak **yönetici olarak çalıştırmak** yeterlidir.

Diğer bir sorun da sayfaları **C:\Program Files\EasyPHP-5.3.2\www** dizinine kaydederken karşılaştığımız kaydetme izniniz yoktur sorunudur.



Bu sorunu aşmak için **www** dizinine sağ tıklayıp **Özellikler->Güvenlik->Düzenle->Users** için Tam Denetim iznini ver demek yeterlidir.

PHP kodlarını HTML kodları arasına yazdığımızı daha önce söylemiştik. PHP kodlarını HTML kodlarından ayırmak için

```
<?php
```

```
.....
```

```
?>
```

ayraçları kullanılır ve her PHP komutundan sonra **noktalı virgül (;)** kullanılır.

Aşağıda **index.php** olarak kaydettiğimiz sayfamızın kodlarını görüyorsunuz.

```
<html>
<body>
<?php
echo "Merhaba Dünya! " ;
?>
</body>
</html>
```

Hazırladığımız bu sayfayı çalıştırmak için tarayıcımızın adres çubuğuna **http://localhost/index.php** yazmak yeterlidir.

Not: Eğer yazılan adreste sayfanın adı belirtilmemişse varsayılan olarak sunucu **index.php**'yi arar.

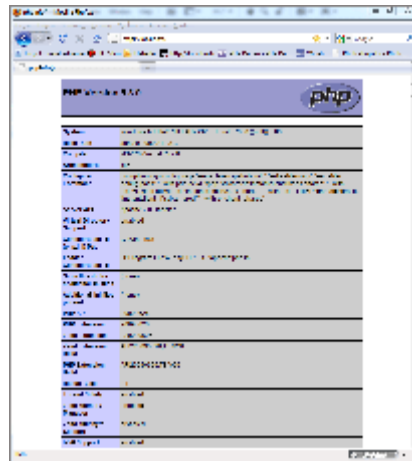
Varsa çalıştırır. Burada sayfamızın adı **index.php** olduğundan adres sadece **http://localhost/** şeklinde de yazılabilirdi. Aksi halde sayfanın adının da yazılması gerekmektedir.

Çalıştırılan sayfanın ekran görüntüsüne ve kullanıcıya gönderilen kaynak koduna dikkat ediniz.



Bilgisayarımıza kurulu olan Apache, PHP ve MySQL hakkındaki tüm bilgileri ekranda görmek için **phpinfo()** fonksiyonu kullanılır. Bu fonksiyon ile versiyon bilgileri, izin bilgileri ve diğer tüm yapılandırma bilgileri ekranda görüntülenebilir. Aşağıdaki kodu yazıp kaydederek çalıştırdığınızda sonucu sizler de ekranda görebilirsiniz.

```
<html>
<body>
<?php
phpinfo();
?>
</body>
</html>
```



PHP'de Açıklamalar

PHP kodu yazarken yazdığınız kodlar ile ilgili yada yaptığınız iş ile ilgili olarak yazmak istediğiniz açıklamalar veya hatırlatıcı notlar olabilir. Unutmayınız ki iyi kodlanmış bir sayfada açıklama satırlarını programcılar her zaman kullanmaktadır.

Tek bir satırdaki açıklamalar için # yada // , birden fazla satırı açıklama satırı yapmak için /* ve */ kullanılır. Unutmayın açıklamalar çalıştırılmaz, programda göz ardı edilir. Onlar sadece hatırlatıcı yada açıklayıcı notlar için kullanılır.

```
<?php
echo "Merhaba Dünya!"; # buradan sonrası açıklama satırıdır
$tarih="01.05.2010"; // satırın sadece bu kısmı açıklama satırıdır

# bu satırın tamamı açıklama satırıdır
// bu satırın da tamamı açıklama satırıdır

/*
Bu kısımda ise birden
fazla satır
açıklama satırı olarak
tanımlanmıştır
*/

echo "<br><b>Tarih</b>: "; /* bu araya açıklama ekledik */ echo $tarih;

// echo $sonuc;  <- kodun çalışmasını istemediğimiz için açıklama satırı
yaptık.
?>
```

Değişkenler

Bilindiği üzere değişkenler o an için değer saklamak için kullanılırlar. Bu değerler metinler, sayılar yada diziler olabilir ve program akışı içerisinde istediğiniz kadar kullanıp değerini değiştirebilirsiniz. PHP'de kullanılan değişkenlerin önünde \$ işareti kullanılır. Yeni başlayanlar için bu işareti unutmamalarını aksi halde programın istediğiniz gibi çalışmayacağını hatırlatırız.

Aşağıda tanımlanmış birkaç değişken ve ona atanmış farklı değerler görmekteyiz.

```
<?php
$metin="Merhaba Dünya!";
$sayi=27;
?>
```

Görüldüğü üzere birkaç değişken tanımlayıp onlara istediğimiz değerleri atadık. Öncesinde bunun ne tür bir değer saklayacağını söylemedik. Çünkü PHP, değişkene atanan değere göre türünü otomatik olarak kendisi ayarlamaktadır. Böyle bir işlemi kuralcı bir dilde yaptığınızda, kullandığınız değişkenin türünü belirtmediğinizden size hata verecektir. Bu yönüyle php bize oldukça esneklik kazandırmaktadır.

Bir değişkenin adını belirlerken aşağıdaki kurallara dikkat etmelisiniz.

- Bir değişkenin adı harfle yada _ ile başlamalıdır.

- Değişken isimlerinde boşluk bırakılmamalıdır. Boşluğun yerine _ işareti kullanabilirsiniz.
- Bir değişkenin isminde a-z, A-Z, 0-9 ve _ karakterlerinin dışında başka bir karakter kullanamazsınız.

Not: PHP dilinde büyük küçük harf ayrımı olduğunda **\$adi** ile **\$Adi** değişkenlerinin aynı olmadığını unutmayınız.

```
<?php
$metin="Merhaba Dünya!";
$Metin="Hello World!";
$adi_soyadi="Ali KAVAK";
$sayi_1=27;
$sayi_2=15;
echo $Metin; // sayfaya Hello World! Yazar.
?>
```

Değişkenlerle İlgili Örnekler

Örnek 1: Tanımlı iki sayının toplamını bulup farklı şekillerde ekrana yazdıralım.

```
<?php
$sayil=23;
$sayi2=42;
$sonuc=$sayil+$sayi2;

echo $sonuc; // ekrana 65 yazar."
echo "<br>";
echo $sayil+$sayi2; // ekrana 65 yazar
echo "<br>";
echo "Sonuc=", $sonuc; // ekrana Sonuc=65 yazar
echo "<br>";
echo "Sonuc=" . $sonuc; // ekrana Sonuc=65 yazar
echo "<br>";
echo "Sonuc=$sonuc"; // ekrana Sonuc=65 yazar
echo "<br>";
echo "Sonuc=", $sayil+$sayi2; // ekrana Sonuc=65 yazar
echo "<br>";
echo "Sonuc=" . ($sayil+$sayi2); // ekrana Sonuc=65 yazar
echo "<br>";
echo "Sonuc=$sayil+$sayi2"; // ekrana Sonuc=23+42 yazar. Çünkü aritmetik işlemler çift tırnak içinde yapılamaz. + operatör olarak değil, ekran yazılması gereken bir karakter olarak algılanır.
echo "<br>";
echo "$sayil+$sayi2=", $sayil+$sayi2; // ekrana 23+42=65 yazar
?>
```

echo fonksiyonunda kullanılan virgül (,) ile nokta (.) arasındaki fark:

Virgül (,) parametreleri ayırmak için kullanılır.

Nokta (.) değişkenleri birleştirip tek bir değere dönüştürmek için kullanılır.

Aritmetiksel işlemler ise çift tırnak içinde gerçekleşmez. Zira aritmetiksel operatörler (+, -, *, /) ekrana yazılması gereken karakterler olarak algılanır.

Örnek 2: Tanımlı olan ad ve soyad değerlerini ekranda yan yana gösterelim.

```
<?php
$adi="Fuat";
$soyadi="Ocak";
$adisoyadi_1=$adi." ".$soyadi; // ad , boşluk ve soyad birleştirip tek
değer olarak atanıyor
$adisoyadi_2="$adi $soyadi"; // string içerisinde ad ve soyad kullanılıyor

// aşağıdaki kodların herbiri ekrana Fuat Ocak yazar
echo $adi." ".$soyadi; // echo ya tek parametre veriliyor
echo "<br>";
echo "$adi $soyadi"; // echo ya tek parametre veriliyor
echo "<br>";
echo $adisoyadi_1;
echo "<br>";
echo $adisoyadi_2;
echo "<br>";
echo $adi," ",$soyadi; // echo ya 3 parametre veriliyor
?>
```

Örnek 3: Tanımlı olan 3 sayıyı toplam değişkenine katalım.

```
<?php
$toplam=11;
$sayi1=5;
$sayi2=7;
$sayi3=14;
$toplam+=$sayi1; // toplam değişkenine sayi1 katılıyor
$toplam+=$sayi2; // toplam değişkenine sayi2 katılıyor
$toplam+=$sayi3; // toplam değişkenine sayi3 katılıyor
echo "Toplam=$toplam"; // ekrana Toplam=37 yazar
?>
```

Örnek 4: Farklı türdeki tanımlı değişkenleri birleştirip sayfaya yazalım.

```
<?php
$boyut=7;
$renk="green";
$acilis="<marquee><i>";
$kapanis="</i></marquee>";
$metin="Nasipse gelir hintten yemenden nasip değilse ne gelir elden!";

$tam_metin="<font size='$boyut'
color='$renk'>".$acilis.$metin.$kapanis."</font>";
echo $tam_metin;
?>
```

Bu kodu çalıştırdığımızda ekranda kayan bir yazı görmekteyiz. Burada şunu görmekteyiz: echo ile sayfaya yazdırılan değerler aslında sayfanın kaynak koduna yazdırılıyor. Bunun anlamı şudur: Eğer echo ile sayfaya html kodları yazdırılırsa bunlar tarayıcı tarafından yorumlanarak ekranda gösterilir.

Dikkat edilmesi gereken bir diğer nokta ise çift tırnaklar içinde yazılan html parametrelerine atanan değerler tek tırnak içinde yazılmaktadır. Zira bu şekilde tırnakların karışması engellenmiş oluyor. Bunu sağlamanın öteki yolu ise tek tırnaklar yerine önüne \ koymak şartıyla çift tırnak kullanmaktır.

Aşağıdaki koda dikkat ediniz.

```
$tam_metin="<font size=\"\$boyut\"  
color=\"\$renk\">\". $acilis.$metin.$kapanis.\"</font>\";
```

Örnek 5: Yarıçapı tanımlanmış bir dairenin alanını ve çevresini sabit tanımlı pi değeri ile hesaplayıp ekrana yazdıralım.

```
<?php  
define("pi",3);  
$r=4;  
$alan=pi*$r*$r;  
$cevre=2*pi*$r;  
echo "<strong>Alan=</strong>$alan<br>";  
echo "<strong>Çevre=</strong>$cevre";  
?>
```

Dikkat edilirse sabiti kullanırken önüne \$ koymadık.

Örnek 6: Tanımlı olan sayı büyüklüğünde yine tanımlı olan başlığı ekranda gösterelim.

```
<?php  
$sayi=3;  
$metin="Bir musibet bin nasihattan iyidir.";  
echo "<h$sayi>$metin</h$sayi>";  
$sayi=5;  
echo "<h$sayi>$metin</h$sayi>";  
$sayi=1;  
echo "<h$sayi>$metin</h$sayi>";  
?>
```

Yukarıdaki kodları çalıştırdıktan sonra sayfanın kaynak koduna dikkat ediniz.

Eğer echo ile kaynak koda yazdırdığınız ifadelerin kaynak kodda alt alta görünmesini istiyorsanız yazdırılan değerın sonun `\n` (new line-yeni satır) ifadesini ekleyiniz. Aynı örneği aşağıdaki gibi çalıştırıp sayfa kaynağına bakınız.

```
<?php  
$sayi=3;  
$metin="Bir musibet bin nasihattan iyidir.";  
echo "<h$sayi>$metin</h$sayi>\n";  
$sayi=5;  
echo "<h$sayi>$metin</h$sayi>\n";  
$sayi=1;  
echo "<h$sayi>$metin</h$sayi>\n";  
?>
```

Operatörler

Aritmetik İşlem Operatörleri

Matematiksel hesaplamaları yapmak için kullanılan operatörlerdir.

Operatör	Açıklama
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
%	Mod Alma
++	Bir Arttırma
--	Bir Azaltma

% operatörü bir sayının başka bir sayıya bölümünden kalanı vermektedir.

++ operatörü bir değişkenin değeri 1 arttırmakta, -- ise 1 azaltmaktadır. Ancak bunun kullanımına dikkat ediniz. Zira \$a++ ile ++\$a farklı sonuçlar verebilir. Çünkü \$a++ ifadesi önce \$a değişkenini kullan sonra 1 arttır anlamına gelmektedir. ++\$a ifadesi ise önce 1 arttır sonra kullan anlamına gelmektedir.

```
<?php
$a=15;
$b=6;

$c=$a++; // $c=15 değerini alır sonra $a=16 olur
$d=++$a; // $a=16+1=17 oldu ve $d=17 değerini aldı.

$e=$a%$b; // $a=17 nin $b=6 ya bölümünden kalanı hesaplar. $e=5 değerini alır

$f=$b--; // $f=6 değerini alır sonra $b=6-1=5 olur

$g=--$b; // $b=5-1=4 oldu ve $g=4 değerini aldı.

$h=++$a-$b++; // $a=17+1=18 oldu, $h=18-4=14 değerini aldı ve $b=4+1=5 oldu

$i=($a--)+(--$b); // $b=5-1=4 oldu, $i=18+4=22 değerini aldı ve $a=18-1=17 oldu

echo "a=$a b=$b c=$c d=$d e=$e f=$f g=$g h=$h i=$i";
?>
```

Atama Operatörleri

Operatör	Örnek Kullanım	Açık Kullanımı
=	<code>\$a=3;</code>	<code>\$a=3;</code>
+=	<code>\$a+= \$b;</code>	<code>\$a=\$a+\$b;</code>
-=	<code>\$a-= \$b;</code>	<code>\$a=\$a-\$b;</code>
=	<code>\$a= \$b;</code>	<code>\$a=\$a*\$b;</code>
/=	<code>\$a/= \$b;</code>	<code>\$a=\$a/\$b;</code>
.=	<code>\$a.= \$b;</code>	<code>\$a=\$a.\$b;</code>
%=	<code>\$a%= \$b;</code>	<code>\$a=\$a%\$b;</code>

```
<?php
$a=15;
$b=6;

$a+= $b; // $a=15+6=21
$a-= $b; // $a=21-6=15
$a*= $b; // $a=15*6=90
$a/= $b; // $a=90/6=15
$a.= $b; // $a=15.6=156
$a%= $b; // $a=156%6=0
echo "a=$a b=$b";
?>
```

PHP'de nokta (.) değişkenleri birleştirip tek bir değer oluşturmak için kullanılır. Virgül (,) ise bir fonksiyona gönderilen parametreleri ayırmak için kullanılır. Nokta (.) PHP'de ondalıklı sayılarda ondalık kısmını ayırmak için de kullanılır.

```
<?php
$adi="Ahmet";
$soyadi="Erkişi";

// $adi ve $soyadi değişkenleri arasına boşluk eklenerek birleştirilip tek
bir değer oluşturuluyor bu değer $adi_soyadi değişkenine atanıyor.
$adi_soyadi=$adi." ".$soyadi;

$boyu=187;
$kilosu=89.5; // Ondalıklı bir sayı ataması yapılıyor

// Burada echo fonksiyonuna ekrana yazması için iki parametre verilmiştir.
$adi_soyadi ve <br>
echo $adi_soyadi,"<br>";

// Burada echo fonksiyonuna tek parametre verilmiştir. Çünkü tüm değerler
birleştirilip tek bir değer olarak echo fonksiyonuna gönderiliyor.
echo "Boy: ".$boyu."<br>Kilosu: ".$kilosu;
?>
```

Karşılaştırma Operatörleri

Operatör	Açıklama
==	Eşittir
<	Küçüktür
>	Büyüktür
<=	Küçük Eşittir
>=	Büyük Eşittir
!=	Eşit Değildir
<>	Eşit Değildir (Farklıdır)

Mantıksal Operatörler

Operatör	Açıklama
&&	ve
	veya
!	Değil

Not: && yerine **and** sözcüğü, || yerine **or** sözcüğü de kullanılabilir.