

PHP'de Kullanılan Operatörler

PHP'de veya başka dillerde kullandığımız pek çok operatör çeşidimiz var. Sadece syntax olarak birbirlerinden farklı durumlar ama mantık olarak birbirlerinin neredeyse aynısılar.

Aritmetik operatörler gayet basittir. İlkokuldan beri öğrendiğimiz matematiksel işlemleri içerir, bu operatörlere matematiksel operatörlerde diyebiliriz. Bu operatörler ile istediğimiz değerlerin toplamını, farkını, bölümünü, çarpımını hatta modunu bile bulabiliriz. Bu operatörler neymiş bakacak olursak;

Operatör	İsim	İşlem
+	Toplama	$\$Sayi1 + \$Sayi2$
-	Çıkartma	$\$Sayi1 - \$Sayi2$
*	Çarpma	$\$Sayi1 * \$Sayi2$
/	Bölme	$\$Sayi1 / \$Sayi2$
%	Modül	$\$Sayi1 \% \$Sayi2$

Birden fazla harfi, kelimeyi, cümleyi arka arkaya eklemek için yani birleştirmek için bu nokta (.) **karakter katarı operatörünü kullanırız.**

PHP'de **çift tırnak ile tek tırnak** arasında bazı kullanım farklılıkları vardır. Örneğin tek tırnak ile yazacağımız değerlerin içerisine değişken yazamayız. Çünkü tek tırnak ile yazdığımız taktirde yazdığımız değişkene değişken değil metin gözüyle bakacaktır. Ama bu bahsettiğimiz olay yalnızca tek tırnak için geçerli. Çift tırnakta ise durum farklı, Çift tırnak kullanmış olsaydık değişkenin değerini yazacaktı. Yani karakter katarı operatörünü kullanmadan da değişkenleri çift tırnak içinde birleştirebiliyoruz.

Kaçış operatörüne gelecek olursak, Kaçış operatörü ters bölüm işaretidir "\". Buradaki amaç bazen dolar işaretini kendi başına kullanmamız gereken durumlarda PHP'nin dolar işaretini değişken sanmamasını sağlamaktır. Mesela ekrana fiyat: 100\$ yazdırmak istediğimizde eğer çift tırnak kullanırsak PHP hata verir çünkü çift tırnak içerisinde dolar işareti özel karakter ve değişken ifadesi olarak algılanır. Bu gibi durumlarda tek tırnak ya da kaçış operatörü kullanılır.

Atama operatörleri ile değişkenlere değer atarız. Temel atama operatörü eşittir. Tek bir eşittir işareti istenilen değişkene istenilen değeri atamamızı sağlar. Eşittir dışında pek çok atama operatörümüzde bulunmakta. Tablo halinde gösterecek olursak. Bu operatörler birleşik atama operatörleridir. Bu operatörler daha kısa kodlar yazmamıza da yardımcı olmaktadır. Bu operatörler neymiş bakacak olursak;

Operatör	Kullanılışı	İşlem
+=	\$Sayi1 += \$Sayi2	\$Sayi1 = \$Sayi1 + \$Sayi2
-=	\$Sayi1 -= \$Sayi2	\$Sayi1 = \$Sayi1 - \$Sayi2
*=	\$Sayi1 *= \$Sayi2	\$Sayi1 = \$Sayi1 * \$Sayi2
/=	\$Sayi1 /= \$Sayi2	\$Sayi1 = \$Sayi1 / \$Sayi2
%=	\$Sayi1 %= \$Sayi2	\$Sayi1 = \$Sayi1 % \$Sayi2
.=	\$Sayi1 .= \$Sayi2	\$Sayi1 = \$Sayi1 . \$Sayi2

Bir değişkendeki sayısal **değeri bir artırmak için** çift artı (++) operatörünü, **bir değer eksiltmek için** de çift eksi (- -) operatörünü kullanırız.

Karşılaştırma operatörleri ile adı üstünde birden fazla değerin kendi aralarında karşılaştırmasını yapacağız. Mesela \$Sayi1 değişkeninin değeri ile \$Sayi2 değişkeninin değeri birbirine eşit mi? ya da daha mı büyük yoksa daha mı küçük gibi karşılaştırmaları bu operatörler ile yapacağız. Karşılaştırma operatörlerini bir tablo üzerinde gösterecek olursak;

Operatör	İsim	Kullanılışı
==	Eşittir	\$Sayi1 == \$Sayi2
===	Denktir	\$Sayi1 === \$Sayi2
!=	Eşit Değildir	\$Sayi1 != \$Sayi2
!==	Denk Değildir	\$Sayi1 !== \$Sayi2
<>	Eşit Değildir	\$Sayi1 <> \$Sayi2
<	Küçüktür	\$Sayi1 < \$Sayi2
>	Büyüktür	\$Sayi1 > \$Sayi2
<=	Küçük Eşittir	\$Sayi1 <= \$Sayi2
>=	Büyük Eşittir	\$Sayi1 >= \$Sayi2

Aslında karşılaştırma operatörlerini daha çok soru sormak gibi değerlendirebiliriz. Bu ifadeleri koşul gerektirecek ifadelerde yani şöyle ise şöyle olsun şu durumda ise şöyle olsun şeklinde kullanacağız. Örneğin 2 değişkenimiz olsun biri a ve diğeri b bu değişkenlerimize göre bir koşul düzenleyeceğiz. Koşul durumunu eşit mi olarak almak istiyorsak eşittir operatörünü, büyüktür olarak almak istiyorsak büyüktür operatörünü ya da tabloda bulunan diğer operatörleri kullanacağız.

Mantıksal operatörlerde ise; birden fazla karşılaştırmayı doğrudur ya da yanlıştır olarak değerlendirip birleştirebiliriz. Örneğin bir değişkenin hem 0'dan büyük mü hem de 100'den küçük mü diye karşılaştırmasını mantıksal operatörler sayesinde yaparız.

Operatör	İsim	Kullanılışı
&&	VE	\$Sayi1 && \$Sayi2
	VEYA	\$Sayi1 \$Sayi2

Hata bastırma operatörüne gelecek olursak, Hata bastırma operatörü (@) et işaretidir. Her deyimin önünde kullanılabilir. Hata oluşturacak yerlerin başına bunu koyduğunuzda ekrana bir hata çıktısı verdirmemeye yarar. Hatayı görmezden gelmeye ve sonucu olumsuz yapmamaya yarar. Diyelim bir bölme işlemi yaptırıyoruz, bu bölme işlemine belki 0 rakamını da yazabiliriz. 0 yazdığımızda PHP bir rakamı 0'a bölemeyeceği için hata verecektir. Bu durumda bu operatörü kullanarak hatayı bastırabiliriz.