

PHP'DE ÖZEL OPERATÖRLER

Operatör gruplarında, birde herhangi bir gruba dahil olmayan özel amaçla kullanılan operatörler mevcuttur. Özel operatörleri inceleyecek olur isek;

Operatör	Tanım	Açıklama	Örnek
.	Birleştirme operatörü	İki veya daha fazla değeri birleştirmek için kullanılır.	<pre>\$a = "Volkan " . "Alakent"; Sonuç : \$a = "Volkan Alakent";</pre> <pre>\$a = "Volkan "; \$b = \$a . "Alakent"; Sonuç : \$b = "Volkan Alakent";</pre> <pre>\$a = "Volkan "; \$b = "Alakent"; \$c = \$a . \$b; Sonuç : \$c = "Volkan Alakent";</pre>
,	Virgül / ekleme operatörü	Birbiri ardına sırayla eklenecek işlemleri veya değerleri bildirmek için kullanılır.	<pre>\$a = array("Elma", "Ispanak");</pre>
`	Çalıştırma operatörü	Değerlerin komut olarak algılanması ve çalıştırılması için kullanılır.	<pre>`ping -n 3 "www.google.com.tr"; Sonuç : Pinging www.google.com.tr [172.217.16.99] with 32 bytes of data: Reply from 172.217.16.99: bytes=32 time=50ms TTL=53 Reply from 172.217.16.99: bytes=32 time=52ms TTL=53 Reply from 172.217.16.99: bytes=32 time=50ms TTL=53 Ping statistics for 172.217.16.99: Packets: Sent = 3, Received = 3, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 50ms, Maximum = 52ms, Average = 50ms</pre>
\	Kaçırma operatörü	Kaçırma operatörü kullanım esnasında karakterin önüne yazıldığında ilgili karaktere işlev kazandırılması için kullanılır. Ayrıca bazı durumlarda ilgili karakterin kodlama karakteri olarak algılanmadan metin halinde okunmasını sağlar.	<pre>\ifade \$a = "Volkan Alakent"; \$b = "\$a"; Sonuç : \$b = "Volkan Alakent";</pre> <pre>\$a = "Volkan Alakent"; \$b = "\\$a"; Sonuç : \$b = \$a;</pre> <pre>\ (tek tırnak) \" (çift tırnak) \\ (ters slash)</pre>
@	Hata denetim operatörü	İfade sonucunda üretilebilecek olası bir hata kodu iletilsinin yok sayılması için kullanılır.	<pre>@mysqli_query(); @file(); @include();</pre>

Operatör	Tanım	Açıklama	Örnek
new	Yeni operatörü	Yeni bir object (<i>nesne</i>) oluşturmak için kullanılır.	<code>\$a = new Object();</code>
this	Bu operatörü	Oluşturulmuş olan bir object'i (<i>nesneyi</i>) işaret etmek için kullanılır.	<pre> class Uyeler{ public \$isim = "Onur "; public \$soyisim = "Tatlı"; function Bilgi(\$uyeninadi="Serkan ", \$uyeninsoyadi="Çelik"){ \$this->isim = \$uyeninadi; \$this->soyisim = \$uyeninsoyadi; } } \$Kisi = new Uyeler(); echo \$Kisi->isim . \$Kisi->soyisim . "
"; \$Kisi->Bilgi(); echo \$Kisi->isim . \$Kisi->soyisim . "
"; \$Kisi->Bilgi("Volkan ", "Alakent"); echo \$Kisi->isim . \$Kisi->soyisim . "
"; Sonuç : Onur Tatlı Serkan Çelik Volkan Alakent </pre>
instanceof	Örneği operatörü	Herhangi bir PHP değişkeninin belli bir sınıfın nesnel örneği olup olmadığını sınamak için kullanılır. instanceof operatörü kullanımına tabi tutulan değişken bir class (<i>sınıf</i>) örneği ise sonuç true (<i>doğru / olumlu / geçerli / başarılı</i>) değeri döner, bir class (<i>sınıf</i>) örneği değil ise sonuç false (<i>yanlış / olumsuz / geçersiz / başarısız</i>) değeri döner.	<pre> class Uyeler{ public \$isimsoyisim = "Volkan Alakent"; } \$Kisi = new Uyeler(); var_dump(\$Kisi instanceof Uyeler); Sonuç : true </pre> <hr/> <pre> \$Kisi = "Volkan Alakent"; var_dump(\$Kisi instanceof Uyeler); Sonuç : false </pre>

Operatör	Tanım	Açıklama	Örnek
?:	Koşullu operatör	Koşullu operatör aynı zamanda ternary (üçlü) operatörü olarak da adlandırılır. Program akışında bir şarta bağlı olarak iki ifadeden birisi çalıştırılmak istendiği durumlarda kullanılır. Serbest olarak çalışabildiği gibi, aynı zamanda değişkene değer ataması işlemlerinde de kullanılabilir. Koşul sonucu true (doğru / olumlu / geçerli / başarılı) değeri döner ise, ? (soru işareti) alanından sonra belirtilen ifade çalışır. Koşul sonucu false (yanlış / olumsuz / geçersiz / başarısız) değeri döner ise, : (iki nokta üst üste) alanından sonra belirtilen ifade çalışır.	(koşul) ? ifade1 : ifade2; \$a = "Volkan"; \$b = (\$a=="Volkan") ? "Evet" : "Hayır"; \$c = \$b; Sonuç : \$c = "Evet"; \$a = "Volkan"; \$b = (\$a=="Hakan") ? "Evet" : "Hayır"; \$c = \$b; Sonuç : \$c = "Hayır";
??	Null birleştiricisi / operatörü	Null birleştiricisi / operatörü, ternary (üçlü) operatörüne çok benzer ve aynı şekilde program akışında bir şarta bağlı olarak iki ifadeden birisi çalıştırılmak istendiği durumlarda kullanılır. Fakat bu operatördeki şart "Null"dır. (boş / değeri olmayan) isset() fonksiyonu ile birlikte üçlü olarak kullanımı kod olarak kısaltmak için eklenmiştir. Koşul sonucunda geçerli bir değer döner ise, belirtilen değer aynen geçerli olur. Koşul sonucu Null (boş / değeri olmayan) değeri döner ise, ?? (iki soru işareti) alanından sonra belirtilen değer geçerli olur.	\$a = isset(\$_POST["isim"]) ? \$_POST["isim"] : "İsim yok"; \$a = \$_POST["isim"] ?? "İsim yok";
<=>	Spaceship operatörü	Spaceship (Uzay gemisi) operatörü aslında bir karşılaştırma operatörüdür. Fakat karşılaştırma sonucundaki değer boolean (mantıksal) veri türünde dönmediği için özel operatörler arasında yer alır. Karşılaştırma sonucunda, ilk değer ikinci değere eşit ise 0, ilk değer ikinci değerden küçük ise -1, ilk değer ikinci değerden büyük ise 1 değeri döndürür.	8<=>8 Sonuç : 0 5<=>8 Sonuç : -1 8<=>5 Sonuç : 1