

PHP - Operatörer

Följande är en försvenskad version utav det ni hittar på följande länk:

https://www.w3schools.com/php/php_operators.asp

Operatörer används för att utföra processering på variabler och värden. Det finns en mängd olika typer av operatörer.

- aritmetiska operatörer
- tilldelningsoperatörer
- jämförelseoperatörer
- öknings-/minskningsoperatörer
- logiska operatörer
- strängoperatörer
- array-operatörer

Vi kommer endast gå igenom några utav dessa.

Aritmetiska Operatörer

Här kommer några exempel på aritmetiska operatörer. Prova gärna exemplen, som finns som länkar till höger i tabellen.

Operatör	Namn	Exempel	Resultat	Visa
+	addition	$\$x + \y	Summan utav $\$x$ och $\$y$.	Exempel
-	subtraktion	$\$x - \y	Differansen mellan $\$x$ och $\$y$.	Exempel
*	multiplikation	$\$x * \y	Produkten utav $\$x$ och $\$y$.	Exempel
/	division	$\$x / \y	Kvoten utav $\$x$ och $\$y$.	Exempel
%	(modulus)	$\$x \% \y	Resten utav $\$x$ dividerat med $\$y$.	Exempel
**	exponentiering	$\$x ** \y	Resultatet när $\$x$ upphöjs till $\$y$.	

Tilldelningsoperatörer

Vi har redan använt oss utav en tilldelningsoperatör - när vi tilldelar en variabel ett värde.

Exempel på tilldelning:

```
$n = 1;  
echo $n;
```

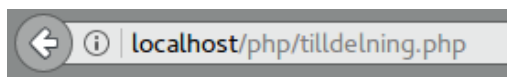


1 ←

I exemplet ovanför sparas värdet 1 i variabeln \$n och sedan skrivs innehållet i variabeln ut.

Fråga: Men, vad händer om vi tilldelar olika värden efter varandra till en och samma variabel?

```
$n = 1;  
echo $n;  
echo "<br>";  
$n = 2;  
echo $n;
```



1
2 ←

Svar: Det gamla värdet skrivs över.

(Se gärna sektionen om tilldelningsoperatörer (eng. assignment operators) i följande länk för att se fler varianter på tilldelning: https://www.w3schools.com/php/php_operators.asp)

Jämförelseoperatörer

Här följer de vanligaste jämförelseoperatörerna. Se gärna exempel för varje i länkarna till höger. Jämförelseoperatörer gör ett test där två värden jämförs. Resultatet av testet är antingen sant, eller falskt.

Operatör	Namn	Exempel	Resultat	Visa
==	lika med	$\$x == \y	Returnerar sant om \$x är lika med \$y	Exempel
!=	inte lika med	$\$x != \y	Returnerar sant om \$x inte är lika med \$y	Exempel
>	större än	$\$x > \y	Returnerar sant om \$x är större än \$y	Exempel

<	mindre än	$\$x < \y	Returnerar sant om \$x är mindre än \$y	Exempel
>=	större eller lika med	$\$x \geq \y	Returnerar sant om \$x är mindre än, eller lika med \$y.	Exempel
<=	mindre eller lika med	$\$x \leq \y	Returnerar sant om \$x är större än, eller lika med \$y.	Exempel

Logiska operatörer

Logiska operatörer utför test, liksom jämförelseoperatörer. Den returnerar antingen sant, eller falskt.

Operatör	Namn	Exempel	Resultat	Visa
and	och	$\$x \text{ and } \y	Returnerar sant om både \$x och \$y är sanna.	Exempel
&&	och	$\$x \ \&\& \ \y	(samma som ovan)	Exempel
or	eller	$\$x \text{ or } \y	Returnerar sant om \$x eller \$y är sanna.	Exempel
	eller	$\$x \ \ \y	(samma som ovan)	Exempel
xor		$\$x \ \text{xor} \ \y	Returnerar sant om \$x eller \$y är sanna, men inte både och.	Exempel
!	inte	$! \ \$x$	Returnerar sant om \$x inte är sant.	Exempel

Strängoperatörer

Man kan sammanfoga två strängar till en sträng med operatören "." (punkt).

Exempel:

```
$var1 = "Första strängen, ";
$var2 = "andra strängen.";
echo $var1 . $var2;
```

Om man har tilldelat en sträng till en variabel och vill sedan lägga till ännu en sträng till samma variabel, då kan man använda operatören ".=". I exemplet nedan blir resultatet samma som ovan, men vi sammanfogar strängarna redan under tilldelningen.

Exempel:

```
$var1 = "Första strängen, ";  
$var1 .= "andra strängen."  
echo $var1;
```

Sant / Falskt

Vad innebär egentligen sant eller falskt i datorvärlden?

I PHP är det relativt enkelt. Värdet 0, eller inget värde alls är falskt, medan allt annat är sant. I övningen nedan kommer du att märka att när något var sant, då skrivs det ut en etta. Om det var sant, då skrivs det inte ut någonting.

Övning

Prova minst två jämförelseoperatörer och två logiska operatörer genom att applicera följande kod (byt ut <operatör> med den operatör du vill använda). Prova gärna också att byta ut värdena för \$x och \$y.

Det som skrivs ut, med funktionen echo är alltså alltid resultatet. Kom ihåg att resultatet kan vara ingenting och att det också är ett resultat.

```
$x = 0;  
$y = 1;  
echo $x <operatör> $y;
```

För att ytterligare förstå operatörerna bättre, prova att tilldela strängar till variablerna \$x och \$y, eller inget alls. T.ex:

```
$x = "en sträng";  
$y = "";  
echo $x <operatör> $y;
```