



# Introduction to PHP





## 4. Manejo de operadores



- **Operadores Aritméticos**
- **Operadores de cadenas**
- **Operadores de asignación**
- **Operadores de comparación**
- **Operadores lógicos**
- **Operadores de bits**



## Operadores y expresiones.

Los scripts en PHP tienen datos, sentencias y ***expresiones***. Las expresiones son normalmente expresiones o ecuaciones matemáticas como 10+4.

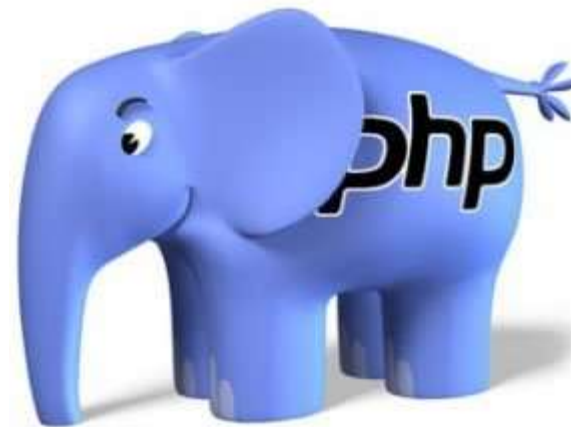
**PHP** soporta un conjunto potente de operadores *unarios*, *binarios* y de otros *tipos*.

Sintaxis: **\$<variable> = <expresión>**

¿Qué puede contener una expresión?

1. **Valores constantes** (como 3.141592)
2. **Variables simples** (como \$x)
3. **Funciones** (como pow(\$x,2))





## Operadores aritméticos.

Éstos operadores siguen las reglas algebraicas tradicionales de jerarquía o prioridad.

`/*Ejemplo:*/`

Consideremos la expresión:  $4+6*2$

¿Cuál sería su valor?

20   O   16

Respuesta: 16, ya que primero se ejecuta la multiplicación y posteriormente la suma.





## Operadores aritméticos.

Operador	Tipos enteros	Tipos flotantes	Ejemplos
+	Suma	Suma	$\$x + \$y$
-	Resta	Resta	$\$f - \$g$
*	Producto	Producto	$\$m * \$z$
/	División entera: cociente	División en coma flotante	$\$b / \$e$
%	División entera: resto		$\$d / \$t$

Debemos considerar que los operadores en ocasiones trabajan diferente según los tipos de datos de los operandos.





## Ejercicios:

1. Ejercicio de prueba de jerarquía.
2. Linealizar las siguientes fórmulas matemáticas.

$$\text{var} = \frac{a + b}{c + d}$$

$$\text{var} = a + \frac{b}{c} + d$$

$$\text{var} = \frac{a + b}{c} + d$$

3. Hacer un programa que transforme una cantidad de segundos a minutos y segundos. Ejemplo: 130 segundos son equivalentes a 2 minutos con 10 segundos.





## Operadores para cadenas.

Existen dos operadores para datos tipo string.

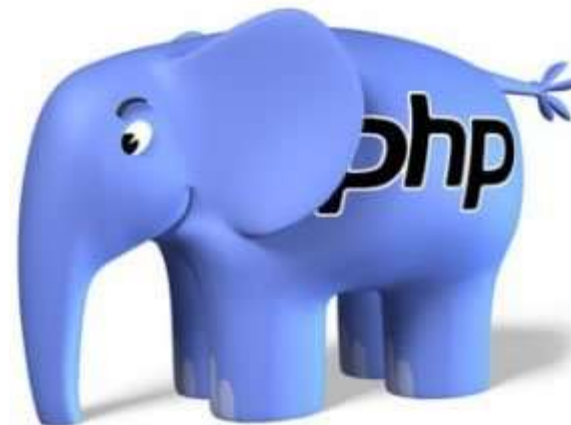
- 1) Operador de concatenación ('.').
- 2) operador de asignación sobre concatenación ('.=')

```
<?php
$a = "Hello ";
$b = $a . "World!"; // ahora $b contiene "Hello World!"

$a = "Hello ";
$a .= "World!"; // ahora $a contiene "Hello World!"
?>
```





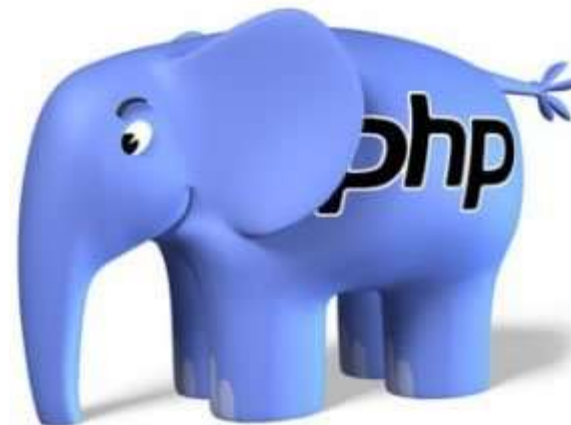


## Operadores asignación.

Sirven para abreviar notación matemática, por ejemplo si queremos si queremos sumar **\$x+1** y guardar el valor en **\$x** podemos escribir **\$x+=1**.

Símbolo	Uso	Descripción
=	<code>\$x=\$y</code>	Asigna el valor de \$x a \$y.
+=	<code>\$x+= \$y</code>	Suma \$y a \$x y asigna el resultado a la variable \$x.
-=	<code>\$x-= \$y</code>	Resta \$y de \$x y asigna el resultado a la variable \$x.
*=	<code>\$x*= \$y</code>	Multiplica \$x por \$y y asigna el resultado a la variable \$x.
/=	<code>\$x/= \$y</code>	Divide \$x entre \$y y asigna el resultado a la variable \$x.
%=	<code>\$x%= \$y</code>	Divide \$x entre \$y de forma entera y asigna el resto a \$x.





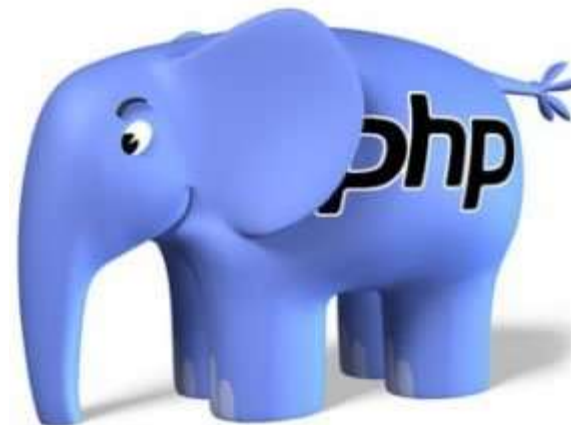
## Operadores de incremento.

Una de las características más útiles que incorpora **PHP** son los operadores especiales **++** y **--**, incrementación y decrementación respectivamente.

**++** suma 1 y **--** resta 1 cada vez que se aplica a una variable.

Incrementación	Decrementación
<code>++\$n</code>	<code>--\$n</code>
<code>\$n+=1</code>	<code>\$n-=1</code>
<code>\$n=\$n+1</code>	<code>\$n=\$n-1</code>





## Operadores de incremento.

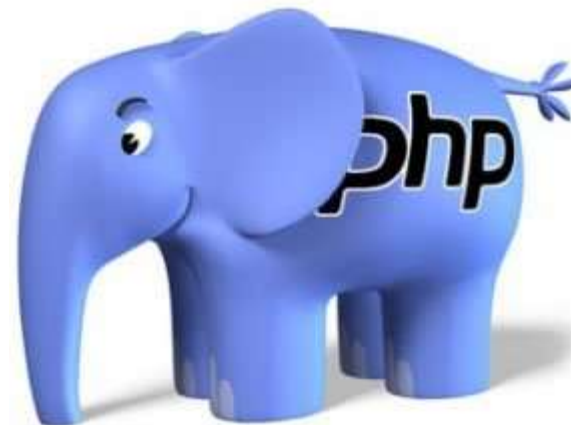
Éstos operadores tienen la propiedad que permite usarlos como prefijos o sufijo (dando un resultado diferente).

```
++$n; //Prefijo  
$n++; //Sufijo  
--$n; //Prefijo  
$n--; //Sufijo
```

Si los operadores ++ y -- están de prefijos el incremento o decremento se realiza antes de la asignación, pero si éstos están de sufijos el incremento o decremento se hace después de la asignación.

Hagamos algunas pruebas...





## Operadores de comparación.

Los operadores de comparación también llamados relacionales normalmente se usan dentro de sentencias de selección o de repetición como **while**, **switch** o **for**.

Sintaxis: <expresión\_1> <operador\_relacional> <expresión\_2>

*/\*Ejemplos:\*/*

$3 < 8$

$\$x + \$y \neq 0$

$\$m < 100$

$\$p \geq 5$

Hagamos algunas pruebas...





## Operadores de comparación.

Los operadores de comparación también llamados relacionales normalmente se usan dentro de sentencias de selección o de repetición como **while**, **switch** o **for**.

Sintaxis: <expresión\_1> <operador\_relacional> <expresión\_2>

*/\*Ejemplos:\*/*

$3 < 8$

$\$x + \$y \neq 0$

$\$m < 100$

$\$p \geq 5$

Hagamos algunas pruebas...





## Operadores de comparación.

### Operadores de comparación

Ejemplo	Nombre	Resultado
<code>\$a == \$b</code>	Igual	true si \$a es igual a \$b después de la manipulación de tipos.
<code>\$a === \$b</code>	Idéntico	true si \$a es igual a \$b, y son del mismo tipo.
<code>\$a != \$b</code>	Diferente	true si \$a no es igual a \$b después de la manipulación de tipos.
<code>\$a &lt;&gt; \$b</code>	Diferente	true si \$a no es igual a \$b después de la manipulación de tipos.
<code>\$a !== \$b</code>	No idéntico	true si \$a no es igual a \$b, o si no son del mismo tipo.
<code>\$a &lt; \$b</code>	Menor que	true si \$a es estrictamente menor que \$b.



## Más operadores de comparación.

### Operadores de comparación

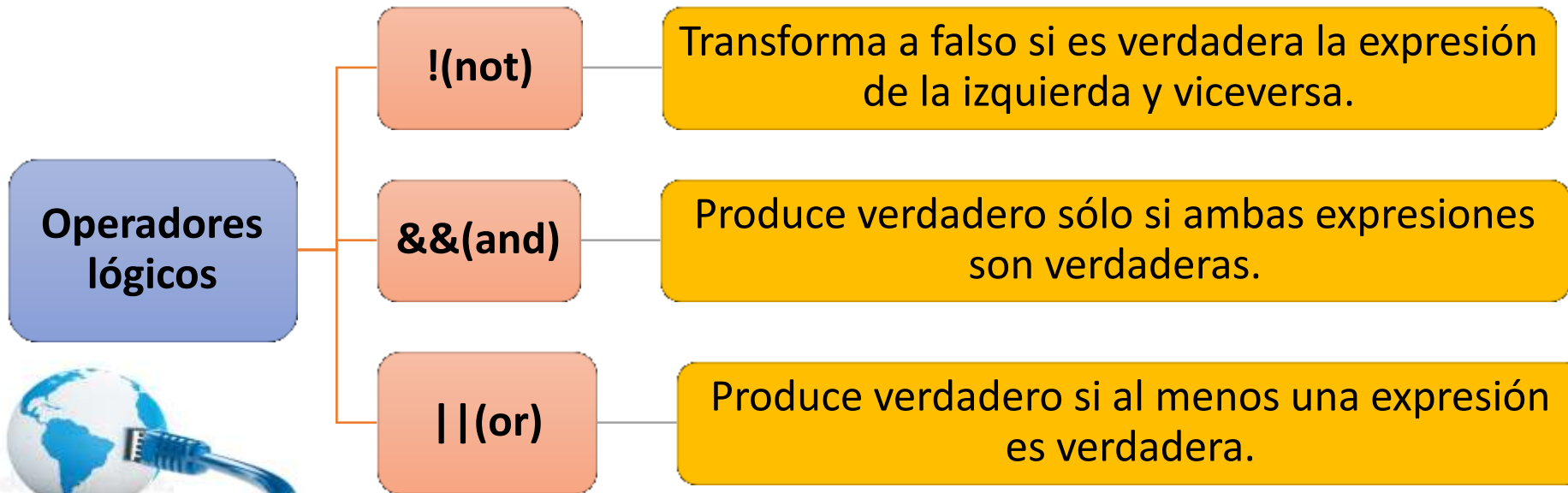
Ejemplo	Nombre	Resultado
$\$a > \$b$	Mayor que	true si $\$a$ es estrictamente mayor que $\$b$ .
$\$a \leq \$b$	Menor o igual que	true si $\$a$ es menor o igual que $\$b$ .
$\$a \geq \$b$	Mayor o igual que	true si $\$a$ es mayor o igual que $\$b$ .
$\$a \leq \geq \$b$	Nave espacial	Un integer menor que, igual a, o mayor que cero cuando $\$a$ es respectivamente menor que, igual a, o mayor que $\$b$ . Disponible a partir de PHP 7.
$\$a ?? \$b ?? \$c$	Fusión de null	El primer operando de izquierda a derecha que exista y no sea null. null si no hay valores definidos y no son null. Disponible a partir de PHP 7.



## Operadores lógicos.

Los operadores lógicos también se denominan *operadores booleanos* en honor de George Boole, creador del álgebra de Boole.

**Sintaxis:** <expresión\_1> <operador\_lógico> <expresión\_2>



Hagamos sus tablas de verdad...