



EJERCICIOS DE CONVERSIÓN/COERCIÓN

null

1. Dadas las siguiente expresiones en PHP, para cada una aplicando las reglas de orden de agrupación (precedencia y asociatividad) y orden de evaluación:

- pon paréntesis de forma que, sin modificar su funcionalidad, haga claro en qué orden se evalúa.
- indica el valor resultante de la expresión y de las variables, si las hay

```
a) $a1 = "hola";  
unset ($a1);  
  
b) $a2 = null;  
$b2a = --$a2;  
$a2 -= 1;  
  
c) $a3 = null - 1;  
  
d) $a4 = NULL;  
$b4a = ( $a4 == $a4['lo']['que']['sea']);  
$b4b = ( $a4 === $a4['lo']['que']['sea']);  
  
e) $a5 = (null - null) * 10;  
  
f) $a6 = null;  
$lista['nombre'] = $a6['nombre'];  
$lista['edad'] = $a6['años'];  
var_dump($lista);  
  
g) $a7 = null == "";  
  
h) $a8 = null == [];  
  
i) $a9 = null == array();
```

Boolean

2. Dadas las siguiente expresiones en PHP, para cada una aplicando las reglas de orden de agrupación (precedencia y asociatividad) y orden de evaluación:

- pon paréntesis de forma que, sin modificar su funcionalidad, haga claro en qué orden se evalúa.
- indica el valor resultante de la expresión y de las variables, si las hay

```
a) $a1 = true + true + true;  
  
b) $a2 = 3 * true;  
  
c) $a3 = (bool) 0;  
$a3 = (bool) -0;  
$a3 = (bool) +0;  
$a3 = (bool) 0.0;  
$a3 = (bool) NAN;  
  
d) $a4 = true / false;  
  
e) $a5;  
$b5a = $a5 == false;  
$b5b = $a5 == true;  
  
f) $a6 = (bool) [];  
  
g) $a7 = (bool) "";  
  
h) $a8 = (bool) "0";
```



```
i) $a9 = (bool) "-0";  
j) $a10 = (bool) "+0";  
k) $a11 = (bool) "0000";  
l) $a12 = (bool) "0E10";  
m) $a13 = (bool) "0.0";  
n) $a14 = (bool) "false";  
o) $a15 = (bool) 0.000000000000000000000001;  
p) $a16 = (bool) ~0;  
q) $a17 = (bool) new stdClass;
```

int

3. Dadas las siguiente expresiones en PHP, para cada una aplicando las reglas de orden de agrupación (precedencia y asociatividad) y orden de evaluación:

- pon paréntesis de forma que, sin modificar su funcionalidad, haga claro en qué orden se evalúa.
- indica el valor resultante de la expresión y de las variables, si las hay

```
a) $a1 = is_integer(PHP_INT_MAX);  
b) $a2 = is_integer(PHP_INT_MAX + 1);  
c) $a3 = is_float(PHP_INT_MAX + 1);  
d) $a4 = is_integer(PHP_INT_MIN - 1);  
e) $a5 = is_float(PHP_INT_MIN - 1);  
f) $a6 = is_integer((PHP_INT_MAX + 1) - 1);  
g) $a7 = is_integer((PHP_INT_MAX - 1) + 1);  
h) echo "En esta plataforma un entero en PHP tiene " . PHP_INT_SIZE * 8 . " bits";
```

float

4. Dadas las siguiente expresiones en PHP, para cada una aplicando las reglas de orden de agrupación (precedencia y asociatividad) y orden de evaluación:

- pon paréntesis de forma que, sin modificar su funcionalidad, haga claro en qué orden se evalúa.
- indica el valor resultante de la expresión y de las variables, si las hay

```
a) echo PHP_FLOAT_EPSILON . PHP_EOL;  
b) echo PHP_FLOAT_MAX;  
c) echo PHP_FLOAT_MIN;  
d) $a1 = 3.00000001;  
$b1 = 3.00000021;  
$miPrecision = 0.00001;  
echo abs($a1 - $b1) < $miPrecision ? "Son iguales" : "Son distintos";
```