



EJERCICIOS DE EXPRESIONES 3 (COMPARACIONES DE IGUALDAD)

Igualdad estricta ===

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 = (NULL === NULL['lo']['que']['sea']);

b) \$a2 = null;
\$b2 = null;
\$c2 = \$a2 ?? \$b2 ?? "hola";
\$r2 = \$c2 === "hola" ;

c) \$a3 = null;
\$b3 = null;
\$r3 = "hola" ?? \$b2 === null ;

d) \$a4 = null;
\$b4 = null;
\$r4 = \$a4 ?? "hola" === "hola" ?? \$b4;

e) \$a5 = null;
\$b5 = null;
\$r5 = (\$a5 ?? "hola") === "hola" ?? \$b5;

f) \$r6 = "" === [];

g) \$r7 = "0" === 0;

h) \$r8 = "" === " ";

i) \$r9 = "" === " " . "";

j) \$r10 = true === 1;

k) \$r11 = true === "1";

l) \$r12 = true === "php";

m) \$r13 = 0 === "";

Igualdad no estricta ==

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 = (NULL == NULL['lo']['que']['sea']);

b) \$a2 = null;
\$b2 = null;
\$c2 = \$a2 ?? \$b2 ?? "hola";
\$r2 = \$c2 == "hola" ;

c) \$a3 = null;
\$b3 = null;
\$r3 = "hola" ?? \$b2 == null ;

d) \$a4 = null;
\$b4 = null;



```
$r4 = $a4 ?? "hola" == "hola" ?? $b4;

e) $a5 = null;
$b5 = null;
$r5 = ($a5 ?? "hola") == "hola" ?? $b5;

f) $r6 = "" == [];

g) $r7 = "0" == 0;

h) $r8 = "" == " ";

i) $r9 = "" == "" . "";

j) $r10 = true == 1;

k) $r11 = true == "1";

l) $r12 = true == "php";

m) $r13 = 0 == "";
```

Igualdad nave espacial <=>

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) $r1 = 1 <=> 3;

b) $r2 = 1 <=> -1;

c) $r3 = 3 <=> 3;

d) $r4 = null <=> null;

e) $r5 = "" <=> "";

f) $r6 = "abc" <=> "Abc";

g) $r7 = "Abc" <=> "abc";

h) $r8 = "" <=> [];

i) $r9 = [] <=> "";

j) $r10 = (1 <=> 2) <=> "";

k) $r11 = false <=> true;

l) $r12 = false <=> null;

m) $r13 = "0" <=> null;
// pero ojo: "0" == null es false

n) $r14 = PHP_INT_MAX <=> new stdClass;

o) $r15 = [3, 4] <=> new stdClass;

p) $r16 = [3, 4] <=> [4, 3];

q) $r17 = [3, 3] <=> [3, 3];
```



```
r) $r18 = NAN <=> NAN;
```

Valores falsy y valores truthy

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) $r1 = -0 ? "verdadero" : "falso";  
b) $r2 = [] ? "verdadero" : "falso";  
c) $r3 = "" ? "verdadero" : "falso";  
d) $r4 = "0" ? "verdadero" : "falso";  
e) $r5 = "00" ? "verdadero" : "falso";  
f) $r6 = null ? "verdadero" : "falso";  
g) $a7;  
$r7 = $a7 ? "verdadero" : "falso";  
h) $a8;  
$r8 = $a8 == null ? "verdadero" : "falso";  
i) $r9 = null == "0" ? "verdadero" : "falso";  
j) $r11 = array() == null ? "verdadero" : "falso";  
k) $r12 = array() == "0" ? "verdadero" : "falso";
```