

En el siguiente código identificar:

- Variables y su tipo
- Operadores
- Funciones y sus parámetros
- Estructuras de control
- Salida por pantalla

`ejercicio_1.php`

```
1 <?php
2 function doble($i)
3     return $i*2;
4 }
5 $a = TRUE;
6 $b = "xyz";
7 $c = 'xyz';
8 $d = 12;
9 echo gettype($a);
10 echo gettype($b);
11 echo gettype($c);
12 echo gettype($d);
13 if(is_int($d)) {
14     $d += 4;
15 }
16 if(is_string($a)) {
17     echo "Cadena: $a";
18 }
19 $d = $a ? ++$d : $d*3;
20 $f = doble($d++);
21 $g = $f += 10;
22 echo $a, $b, $c, $d, $f, $g;
23 ?>
```

En el código podemos ver las siguientes variables

- \$a, de tipo boolean
- \$b, de tipo string
- \$c, de tipo string
- \$d, de tipo integer
- \$f, de tipo integer
- \$g, de tipo integer

Se utilizan los siguientes operadores

- “\*”: operador binario aritmético. Devuelve el producto de dos números. Se utiliza en las líneas 3 y 19.
- “=”: operador binario de asignación. Asigna el contenido del segundo operando al primer operando. Se utiliza en las líneas 5, 6, 7, 8, 19, 20 y 21.
- “+=”: operador binario de asignación. Realiza la suma entre el primer y el segundo operando y almacena el resultado en el primer operando. Se utiliza en las líneas 14 y 21.
- “?:”: operador ternario de decisión. Se evalúa el primer operando; si este es TRUE, devuelve el segundo operando; sino, devuelve el tercer operando. Se utiliza en la línea 19.
- “++”: operador unario aritmético. Si el operador se encuentra a la izquierda del operando, incrementa el valor del operando en uno y lo devuelve; si el operador se encuentra a la derecha del operando, primero devuelve el valor del operando y luego lo incrementa en uno. Se utiliza en la línea 19 como pre-incremento y en la línea 20 como post-incremento.

Se define la función doble, que toma un parámetro en una variable \$i y devuelve el producto entre el valor de \$i y 2. Esta función se utiliza en la línea 20.

También se utilizan las funciones is\_int, que toma un parámetro y devuelve TRUE si el tipo de dato de dicho parámetro es integer o FALSE en caso contrario; y is\_string, que toma también un parámetro, pero en este caso devuelve TRUE o FALSE dependiendo de si el tipo de dato del parámetro es string. Estas funciones son utilizadas en las líneas 13 y 16 respectivamente.

Se utiliza la estructura de control if en las líneas 13 y 16. Esta permite decidir si se ejecutará o no un conjunto de instrucciones dependiendo del valor de una condición.

Las líneas 9, 10, 11 y 12 deberían mostrar los valores “boolean”, “string”, “string” e “integer” respectivamente. Luego, como la condición del if en la línea 16 tiene el valor FALSE, la instrucción echo de la línea 17 no se ejecutará. Por último la instrucción en la línea 22, que muestra el valor de cada variable, debería mostrar “1” (valor numérico de la variable booleana \$a), “xyz”, “xyz”, “18” (valor de la variable \$d luego de las instrucciones 14, 19 y 20), “44” y “44” ( $2 * 17 + 10$ ).

Entonces, la salida completa por pantalla debería ser “booleanstringstringintegerlxyzxyzl84444”.