

Práctica 4: PHP

(variables, tipos, operadores, expresiones, estructuras de control, arrays y funciones)

Integrantes:

Barroso Bollero Agustin - 52818

Caracchi Victoria - 53482

Ponce Lautaro - 52898

Reschini Enrico - 52973

Ejercicio 1

- Las variables y sus tipos son: a (boolean), b (string), c (string), d (integer), f, g e i(integer).
- Los operadores son: *, =, +=, ? y ++.
- Las funciones son: doble, gettype, is_int, is_string y echodoble, gettype, is_int, is_string y echo.
- La única estructura de control que aparece es if.
- La salida por pantalla es: booleanstringstringinteger1xyzxyz184444.

Ejercicio 2

- a) En este apartado, los 3 códigos devuelven (o imprimen) lo mismo: los números del 1 al 10. Más allá de eso, los primeros 2 códigos usan un while que comienza con \$i = 1, donde se imprime el valor de \$i y luego se incrementa, mientras que en el tercer código el valor de \$i se incrementa antes de ser impreso, utilizándose el bucle do-while.
- b) En este segundo apartado, nuevamente estamos analizando 4 fragmentos de código que devuelven lo mismo: los números del 1 al 10. En todos los códigos se usa el bucle for, sin embargo en los 2 códigos de la izquierda se utiliza la forma “estándar” del mismo, mientras que en los otros 2 códigos dichos bucles dependen de la condición interna y contienen un ‘break’ para salir de ellos.
- c) Por último, en este inciso tenemos 2 códigos que son funcionalmente equivalentes. Ambos realizan la misma operación, comparando el valor de la variable \$i y ejecutando un bloque de código en función de su valor (0, 1 o 2). Sin embargo uno de los códigos simplemente realiza una serie de condiciones anidadas (if) para reconocer que cartel imprimir, mientras que el otro usa la estructura switch. Esta última mencionada, es mucho más apropiada cuando estamos trabajando con una cantidad considerable de valores a comparar y cuando tenemos la necesidad de realizar un programa que sea capaz de imprimir una de muchas salidas posibles y distintas dependiendo de ciertas condiciones.

Ejercicio 3

a) Este apartado muestra un código PHP dentro de un documento HTML, el cual crea una tabla con un ancho del 90% de la página con un borde de un píxel. Se definen dos variables: \$row, que tiene el valor de 5, y \$col que tiene el valor de 2. Esto significa que la tabla tendrá 5 filas y 2 columnas. Luego, a través de dos for anidados, se crean las filas y las columnas de la tabla. El primer bucle recorre las cinco filas, mientras que el segundo se encarga de recorrer las dos columnas en cada fila. Dentro de cada celda de la tabla se imprime un espacio en blanco mediante ' '.

b) Este inciso muestra un código PHP dentro de un documento HTML que verifica si un formulario fue enviado correctamente mediante la comprobación de la variable \$_POST['submit']. Si el formulario no fue enviado, se genera un formulario en el cual el usuario puede ingresar su edad. El formulario utiliza el método POST y al enviarse, recarga la misma página a través de \$_SERVER['PHP_SELF'], que se refiere al archivo PHP actual. El formulario tiene un campo de entrada para la edad y un botón de envío. Una vez que el formulario ha sido enviado, el código pasa a la sección del else, donde se guarda el valor ingresado en el campo de edad usando \$_POST['age']. Luego, se verifica si la edad es mayor o igual a 21. Si es así, se imprime el mensaje "Mayor de edad". Si la edad es menor de 21, se imprime "Menor de edad". En conclusión, el código determina si una edad corresponde a un mayor de edad o a un menor de edad según lo que se ingrese en el formulario.

Ejercicio 4

En este ejercicio vemos dos códigos PHP, los cuáles se relacionan debido a que uno de ellos (datos.php) está dentro de otro a través del 'include'. El código principal comienza con una instrucción 'echo', donde se intenta imprimir las variables \$flor y \$color antes de incluir el archivo. En este punto, las variables aún no han sido definidas. Es por esto, que el 'echo' en cuestión sólo mostraría "El", ya que las variables no tienen valor pero la ejecución no se detendría. Después de esto, se ejecuta la instrucción para incluir al archivo datos.php, lo que provoca que las variables \$flor y \$color se definan con los valores 'clavel' y 'blanco', respectivamente. Finalmente, la segunda instrucción 'echo' imprime nuevamente "El \$flor \$color", pero esta vez las variables ya tienen valores asignados, por lo que el resultado será "El clavel blanco". De esta forma, se imprime el cartel "El" y debajo el cartel "El clavel blanco".

Ejercicio 5

PHP: arrays, funciones

Ejercicio 1

Los códigos proporcionados en este punto son equivalentes. En el primer caso, se utiliza la función `array()` para definir el array asociativo con sus respectivas claves y valores, incluyendo un valor sin clave asociada. En el segundo código, se usa la notación con corchetes para asignar los mismos valores a las mismas claves, y también se añade un valor sin clave especificada (`$a[] = 4`), por lo que se ubica en la siguiente posición numérica disponible. Ambos códigos generan el mismo array final.

Ejercicio 2

- a) En este apartado vemos un código PHP dónde se define `$matriz`, con dos claves "x" y "12" cuyos valores son "bar" y "true" respectivamente. Luego vemos las líneas que utilizan 'echo', mostrando como salida 'bar' y '1' (ya que en PHP el valor true se imprime como 1).
- b) En este código vemos una matriz que tiene dentro otro array (a través de la clave "unamatriz") con las claves "6", "13", y "a". Los valores correspondientes son 5, 9 y 42. De esta forma, la salida a través de los 'echo' será dichos valores.
- c) En este inciso se crean varios elementos en el array, pero algunos se eliminan con 'unset'. Después de las instrucciones de unset, el array `$matriz` queda vacío y todas las referencias a las claves se eliminan, por lo que cualquier intento de acceso a una clave del array en cuestión no producirá ninguna salida.

Ejercicio 3

Ejercicio 4