



EJERCICIOS PHP

RELACIÓN I - EJERCICIOS INICIALES

1- Haz un programa en PHP que muestre el mensaje "Hello world" de diferentes formas:

- como texto plano html
- como un encabezado de nivel 2 html
- como un párrafo con estilo: color, tipografía, alineación, etc.
- con un salto de línea entre hello y world
- añádele la información sobre la instalación php (phpversion() y phpinfo())
- investiga en la siguiente dirección como mostrar la fecha y la hora del sistema en el momento de la ejecución:

<https://www.php.net/manual/es/function.date.php>

PUNTOS DE INTERÉS: sintaxis de bloque php dentro de html, salida básica con echo, manipulación de las etiquetas html dentro del código php, comentarios en una y varias líneas, información de la configuración

2- Haz un programa PHP que muestre un valor de ejemplo de cada **tipo de dato escalar** en php con **echo** utilizando la función **var_dump()**, y también con **printf** formateado.

Prueba diversas posibilidades de formateo de salida tal y como vienen descritas en: https://www.w3schools.com/php/func_string_printf.asp para:

- bool
- int
- float
- string

PUNTOS DE INTERÉS: tipos de datos escalares en PHP, declaración de variables, variables variables (\$\$), regla de formación de nombres de variables, salida formateada con printf

3- Investiga qué y cuales son las **superglobals** en php (<https://www.php.net/manual/es/language.variables.superglobals.php>), y haz un programa que muestre, en forma de lista no numerada, para la superglobal \$_SERVER los valores de:

```
'DOCUMENT-ROOT'  
'PHP-SELF'  
'SERVER-NAME'  
'SERVER_SOFTWARE'  
'SERVER_PROTOCOL'  
'HTTP_HOST'
```

```
'HTTP_USER_AGENT'  
'REMOTE_ADDR'  
'REMOTE_PORT'  
'SCRIPT_FILENAME'  
'REQUEST_URI'
```

Prueba un volcado de `$_SERVER` con `var_dump($_SERVER)` y también con `print_r($_SERVER)`. ¿Cuál es la diferencia?

PUNTOS DE INTERÉS: concepto de superglobals, arrays asociativos, información de `var_dump()` y `print_r()` para la fase de debug (no usar en producción)

4- En un programa PHP, declara un array constante en el que se almacenarán los días de la semana. Muestra por pantalla:

- el primer día de la semana
- todos los días secuencialmente
- lo mismo que el anterior, pero en formato de lista numerada

PUNTOS DE INTERÉS: array clásico, recorrido secuencial usando `for`, función que calcula la longitud de una array, creación de estructuras de listas en html desde php

5- Crea un array asociativo constante, en el que utilices como clave el día de la semana, y como valor, la temperatura máxima de ese día en formato real. A continuación, muestra:

- la temperatura del primer día de la semana
- la temperatura de todos los días, secuencialmente
- lo mismo que el anterior, pero en formato de lista numerada
- idem, en forma de tabla

PUNTOS DE INTERÉS: array asociativo, recorrido secuencial usando `foreach`, acumulación de resultados con `+=`, creación de estructuras de listas y tablas html, límites del estilo css

6- Declara en un programa PHP una clase fruta, con dos atributos: nombre y color, y dos funciones, `set_name()` y `get_name()`. Declara e inicializa dos instancias: apple y banana, inicializa los nombres y muéstralos por pantalla

PUNTOS DE INTERÉS: declaración y manipulación básica de clases y objetos en php, visibilidad, getters, setters y constructores

7- Calcula la nota final de una persona a partir de la media de dos notas numéricas iniciales, y descontando 0.25 por cada falta sin justificar. Muestra el resultado por pantalla, indicando si la persona aprueba o suspende.

PUNTOS DE INTERÉS: estructura de control de bifurcación `if` (simple), operadores matemáticos

8- Crea en un script PHP dos arrays asociativos paralelos, uno con la rúbrica de 4 calificaciones (inicial, primera, segunda y tercera) y otro con las notas particulares de una persona. A continuación, computará la nota final de esa persona, y muéstrala por pantalla.

PUNTOS DE INTERÉS: recorrido secuencial de array asociativo con foreach y acceso directo por clave a un segundo array, operadores matemáticos, bifurcación simple

9- En un programa PHP, valora a partir de los 3 lados de un triángulo si es equilátero, isósceles y escaleno, y muestra esa valoración por pantalla

PUNTOS DE INTERÉS: bifurcaciones if anidadas, utilización de operadores booleanos and, or y not, testing con diferentes juegos de datos de las diferentes ramas de un algoritmo

10- Haz un programa PHP que resuelva una ecuación de segundo grado siempre que los resultados sean reales

PUNTOS DE INTERÉS: bifurcaciones if anidadas, operadores y funciones para potencia y raíz cuadrada, testing intensivo con juegos de datos que buscan deliberadamente “romper” la ejecución del programa

11- Mejora el intento anterior para que si alguno de los coeficientes **a**, **b** o **c** fuera 0, el programa gestione el cálculo de resultados de manera más adecuada:

- Si $a=0$, la ecuación no es de segundo grado, solo hay una raíz:
$$x = -c/b$$
- Si $b=0$, las raíces se calculan de manera más sencilla:
$$x1 = -\sqrt{-c/a} \text{ y } x2 = \sqrt{-c/a}$$
- Si $c=0$, las raíces son, sacando factor común: $x(ax+b)=0$:
$$x1=0 \text{ y } x2=-b/a$$

Aparte de esta casuística, hay que evitar dividir por cero.... Resuelve toda estas posibilidades y **refactoriza** el código para que sea limpio y óptimo

PUNTOS DE INTERÉS: bifurcaciones if anidadas a varios niveles, validación de datos y refactorización

12 - Realiza un programa php que, a partir de una nota numérica entera entre 1 y 10 devuelva:

- Sobresaliente si es 9 ó 10
- Notable si es 7 u 8
- Bien si es un 6
- Suficiente si es un 5
- Suspenso, si es 1,2,3 ó 4

Utiliza la bifurcación múltiple o switch y comprueba que la nota esté en el rango adecuado de valores permitidos (sea entera y entre 1 y 10)

PUNTOS DE INTERÉS: bifurcaciones múltiples, sentencia break y validación de datos

13- Haz un script PHP que calcule el factorial de un número natural (entero y positivo). Haz que se muestren los cálculos que se van haciendo

PUNTOS DE INTERÉS: bucle for, contador, acumulador de productos, recorrido hacia atrás, inicialización de acumulador para productos, validación de datos, overflow

14- Haz un programa PHP que calcule la suma de los n primeros números naturales (siendo n entero y positivo)

PUNTOS DE INTERÉS: bucle for, contador, acumulador de sumas, recorrido hacia adelante, inicialización de acumulador para sumas, validación de datos

15- Haz un programa php que te diga si un número entero y positivo es primo o no

PUNTOS DE INTERÉS: bucle for vs bucle while, contador, salida abrupta de un bucle (break, break n)

16- Haz un programa que muestre todos los divisores de un número entero y positivo. Irá mostrando cada número que se prueba y si resulta ser divisor, aparecerá marcado visiblemente, por ejemplo con otro color. Por ejemplo:

Divisores de 10:

1 2 3 4 5 6 7 8 9 10

PUNTOS DE INTERÉS: bucle while, contador, acumulador de resultados, formateo de la salida con estilo, optimización, validación de datos

17- Haz un script en PHP que calcule la división de dos números naturales utilizando el algoritmo de Euclides para la división

PUNTOS DE INTERÉS: bucle while, contador, acumulador de restas, validación de datos

18- Haz un programa en PHP que calcule el máximo común divisor de dos números naturales utilizando el algoritmo de Euclides

PUNTOS DE INTERÉS: bucle while, acumulador de restas, validación de datos

19- Haz un script PHP en el que conviertas en binario un número natural decimal

PUNTOS DE INTERÉS: bucle while, acumulador divisiones, acumulación de cadenas, array_push(), validación de datos, testing

20- Mejora el ejercicio anterior para que se pueda convertir a binario, octal o hexadecimal

PUNTOS DE INTERÉS: bucle do-while, bucle while, acumulador divisiones, acumulación de cadenas, array_push(), switch, validación de datos, testing