

Seminario 1 - PHP 8.4 Básico

Desarrollo de Aplicaciones Multiplataforma

Programación Web

Ejercicio 1. Número máximo de un array

Crea una función que obtenga el número máximo de un array de números.

Ejercicio 2. Sumatoria de un array

Crea una función que obtenga la sumatoria de un array de números.

Ejercicio 3. Conversión de millas a kilómetros

Crea una función que dada una distancia en millas calcule su correspondiente en kilómetros.

Nota: 1 milla = 1.60934 kilómetros

Ejercicio 4. Palíndromo

Crea una función que determine si una cadena de texto es un palíndromo.

Nota: Un palíndromo es una palabra o frase que se lee igual de izquierda a derecha que de derecha a izquierda, por ejemplo: "ana", "reconocer", "anilina".

Ejercicio 5. Contar ocurrencias de una letra

Crea una función que cuente cuántas veces aparece una letra en un texto.

Ejercicio 6. Contar ocurrencias de una subcadena

Crea una función que cuente cuántas veces aparece una subcadena en un texto.

Ejercicio 7. Capitalizar palabras

Crea una función que ponga en mayúscula la primera letra de cada palabra de un texto.

Ejemplo: "hola mundo" → "Hola Mundo"

Ejercicio 8. Suma de dígitos

Crea una función que sume los dígitos de un número.

Ejemplo: $\text{sumaDigitos}(245) = 2 + 4 + 5 = 11$

Ejercicio 9. Máximo común divisor (MCD)

Crea una función que calcule el máximo común divisor de dos números naturales.

Nota: Se dice que dos números son relativamente primos si su factor común más grande (MCD) es 1. Ejemplo 1: Los factores de 20 son 1, 2, 4, 5, 10 y 20. Los factores de 33 son 1, 3, 11, y 33.

Ejercicio 10. Fibonacci

Crea una función que calcule el término n-ésimo de la sucesión de Fibonacci.

Nota: En matemática, la sucesión de Fibonacci se trata de una serie infinita de números naturales que empieza con un 0 y un 1 y continúa añadiendo números que son la suma de los dos anteriores: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597...

Ejercicio 11. Números primos relativos

Crea una función que determine si dos números son primos relativos.

Nota: Se dice que dos números son relativamente primos si su factor común más grande (MCD) es 1.

Ejercicio 12. Número capicúa

Crea una función que determine si un número dado es capicúa.

Nota: Un número capicúa es aquel que se lee igual de izquierda a derecha que de derecha a izquierda, por ejemplo: 121, 1331, 45654.

Ejercicio 13. Generador de tabla HTML

Crea una función que dada una cadena de texto con formato Emmet devuelva su correspondiente etiqueta HTML, teniendo en cuenta sólo los atributos de clase e id.

Ejemplos:

```
in: a -> out: <a></a>
```

```
in: div.oferta -> out: <div class="oferta"></div>
```

```
in: div.coche#VWPolo -> out: <div class="coche" id="VWPolo"></div>
```

Ejercicio 14. Mosaico numérico

Crea una función que dado un número n imprima el siguiente 'mosaico' (para n = 6):

```
1
```

```
22  
333  
4444  
55555
```

```
666666
```

Ejercicio 15. Comparar arrays elemento a elemento

Crear una función que reciba dos arrays de enteros y devuelva un array de booleanos que determine si los elementos, uno a uno, de ambos arrays son iguales.

Ejemplo: `comparar([1, 2, 3], [1, 2, 4]) → [true, true, false]`

Ejercicio 16. Producto de elementos de un array

Crea una función que calcule el producto de todos los elementos en un array de números.

Ejemplo: `producto([2, 3, 4])` → 24

Ejercicio 17. Filtrar números pares

Crea una función que dada un array de números, devuelva un nuevo array con solo los números pares.

Ejemplo: `filtrarPares([1, 2, 3, 4, 5, 6])` → [2, 4, 6]

Ejercicio 18. Número primo

Crea una función que determine si un número es primo.

Nota: Un número primo es un número natural mayor que 1 que solo es divisible por 1 y por sí mismo.

Ejercicio 19. Eliminar vocales

Crea una función que, dada una cadena de texto, elimine todas las vocales de la cadena.

Ejemplo: `eliminarVocales("Hola Mundo")` → "Hl Mnd"

Ejercicio 20. Factorial

Crea una función que calcule el factorial de un número.

Nota: El factorial de un número n (representado como $n!$) es el producto de todos los números enteros positivos desde 1 hasta n . Por ejemplo, $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

Ejercicio 21. Invertir cadena

Crea una función que invierta una cadena de texto. Por ejemplo, "hola" debería convertirse en "aloh".

Ejercicio 22. Número perfecto

Crea una función que, dado un número, devuelva true si es un número perfecto (la suma de sus divisores propios positivos es igual al número), o false en caso contrario.

Ejemplo: 6 es un número perfecto porque sus divisores propios son 1, 2 y 3, y $1 + 2 + 3 = 6$.

Ejercicio 23. Número Armstrong

Crea una función que, dado un número entero, devuelva true si es un número Armstrong (un número que es igual a la suma de sus propios dígitos elevados a una potencia). Por ejemplo, 153 es un número Armstrong porque $1^3 + 5^3 + 3^3 = 153$.

Ejercicio 24. Calculadora de descuentos con constantes

Crea un programa que utilice constantes para definir diferentes tipos de descuentos (DESCUENTO_ESTUDIANTE, DESCUENTO_JUBILADO, DESCUENTO_VIP) y una función que calcule el precio final de un producto aplicando el descuento correspondiente según el tipo de cliente.

Valores de las constantes:

- DESCUENTO_ESTUDIANTE: 15%
- DESCUENTO_JUBILADO: 20%
- DESCUENTO_VIP: 25%

Ejemplo: calcularPrecioFinal(100, "estudiante") → 85

Ejercicio 25. Clasificador de notas con match

Crea una función que utilice la expresión match de PHP 8 para clasificar una nota numérica (0-10) en su correspondiente calificación textual.

Clasificación:

- 9-10: Sobresaliente
- 7-8: Notable
- 5-6: Aprobado
- 0-4: Suspenso

Ejemplo: clasificarNota(8) → "Notable"

Ejercicio 26. Validador de datos con operador null coalescing

Crea una función que reciba un array asociativo con datos de usuario (nombre, email, edad, ciudad) y utilice el operador null coalescing (??) para asignar valores por defecto cuando algún campo esté ausente o sea null.

Valores por defecto:

- nombre: "Anónimo"
- email: "sin-email@example.com"
- edad: 18
- ciudad: "Desconocida"

Ejemplo:

```
validarDatos(['nombre' => 'Juan', 'edad' => 25])
```

```
→ ['nombre' => 'Juan', 'email' => 'sin-email@example.com', 'edad' => 25,  
  'ciudad' => 'Desconocida']
```

Ejercicio 27. Acceso seguro a propiedades con nullsafe operator

Crea una función que reciba un array asociativo que representa un usuario con datos anidados (dirección, teléfono, etc.) y utilice el operador nullsafe (`?->`) para acceder de forma segura a propiedades que podrían no existir, devolviendo un mensaje apropiado.

Nota: Simula objetos usando arrays asociativos o stdClass.

Ejemplo de estructura:

```
$usuario = [  
  
    'nombre' => 'Ana',  
    'direccion' => [  
        'calle' => 'Gran Vía',  
        'ciudad' => 'Madrid'  
    ]  
  
];
```

La función debe intentar acceder a `$usuario['direccion']['codigoPostal']` de forma segura.

Ejercicio 28. Calculadora interactiva

Crea un programa que simule una calculadora interactiva. El programa debe solicitar al usuario dos números y una operación (+, -, *, /) usando `readline()` o simulando entrada de datos, y mostrar el resultado. Debe validar que los números sean válidos y manejar la división por cero.

Ejemplo de interacción:

```
Introduce el primer número: 10
```

```
Introduce el segundo número: 5  
Introduce la operación (+, -, *, /): *
```

```
Resultado: 10 * 5 = 50
```

Ejercicio 29. Conversor de temperaturas con constantes mágicas

Crea un programa que convierta temperaturas entre Celsius, Fahrenheit y Kelvin. Utiliza constantes para las fórmulas de conversión y constantes mágicas de PHP (__FUNCTION__, __LINE__) para mostrar información de depuración.

Fórmulas:

- Celsius a Fahrenheit: $(C \times 9/5) + 32$
- Celsius a Kelvin: $C + 273.15$
- Fahrenheit a Celsius: $(F - 32) \times 5/9$
- Kelvin a Celsius: $K - 273.15$

Ejemplo: `convertirTemperatura(25, 'celsius', 'fahrenheit')` → 77

¡A trabajar!

Recuerda:

- Escribe código limpio y bien comentado
- Prueba tus funciones con diferentes casos de prueba
- Utiliza las características modernas de PHP 8.4
- Maneja correctamente los tipos de datos y errores