

Práctica 5: Funciones propias y nativas, include, require y operadores ternarios

Instrucciones:

Realiza los siguientes ejercicios en PHP y asegúrate de que cada función y script cumpla con los requerimientos especificados. La práctica está dividida en tres secciones: funciones con parámetros, manipulación de strings y arrays, y uso de operadores ternarios en PHP con HTML.

Crea una carpeta de nombre `pac05apellidoNombre`.

Parte 1: Funciones con parámetros (carpeta: `funcionesPropias`)

1. Generar saludo personalizado

Crea una función `generarSaludo($nombre)` que reciba un nombre como parámetro y devuelva un saludo personalizado en HTML. Por ejemplo, si se pasa "Juan", debería devolver `<h1>Hola, Juan!</h1>`.

2. Calcular el precio total de un producto

Crea una función `calcularTotal($precio, $cantidad, $impuesto)` que reciba como parámetros el precio de un producto, la cantidad y el porcentaje de impuesto. La función debe devolver el precio total con el impuesto aplicado.

3. Generar un resumen acortado

Crea una función `generarResumen($texto, $limite)` que reciba un texto y un límite de caracteres. La función debe devolver el texto acortado hasta el límite, agregando "..." al final si el texto ha sido recortado.

4. Conversión de temperaturas

Crea una función `convertirTemperatura($temperatura, $escala)` que reciba una temperatura y la escala a la que debe convertir ("C" para Celsius o "F" para Fahrenheit) y devuelva el resultado convertido.

5. Cálculo de edad a partir del año de nacimiento

Crea una función `calcularEdad($anioNacimiento)` que reciba un año de nacimiento y devuelva la edad actual.

6. **Determinar si un número es par o impar**

Crea una función `esPar($numero)` que reciba un número y devuelva `true` si es par, o `false` si es impar.

7. **Generar enlace de descarga**

Crea una función `generarEnlaceDescarga($archivo)` que reciba el nombre de un archivo y devuelva un enlace HTML con el texto "Descargar" apuntando a dicho archivo.

8. **Calcular descuento aplicado**

Crea una función `calcularDescuento($precioOriginal, $descuento)` que reciba un precio original y un porcentaje de descuento, y devuelva el precio final con el descuento aplicado.

9. **Conversión de horas a minutos**

Crea una función `convertirHorasMinutos($horas)` que reciba una cantidad de horas y devuelva el equivalente en minutos.

Parte 2: Manipulación de strings y arrays

Funciones de Strings (carpeta: funcionesString)

1. **Convertir texto a mayúsculas**

Crea una función `convertirMayusculas($texto)` que reciba una cadena de texto y devuelva la cadena en mayúsculas. Usa la función `strtoupper()`.

2. **Contar palabras en un texto**

Crea una función `contarPalabras($texto)` que reciba un texto y devuelva la cantidad de palabras que contiene. Usa la función `str_word_count()`.

3. **Obtener una subcadena**

Crea una función `obtenerSubcadena($texto, $inicio, $longitud)` que reciba un texto, una posición de inicio y una longitud, y devuelva la subcadena correspondiente. Usa la función `substr()`.

4. **Reemplazar palabras en una frase**

Crea una función `reemplazarPalabras($texto, $buscar, $reemplazar)` que reciba un texto, una palabra a buscar y otra palabra para reemplazarla. Usa la función `str_replace()`.

5. **Invertir una cadena de texto**

Crea una función `invertirTexto($texto)` que reciba una cadena de texto y devuelva el texto invertido. Usa la función `strrev()`.

6. **Comparar dos strings**

Crea una función `compararStrings($cadena1, $cadena2)` que reciba

dos cadenas de texto y devuelva true si son iguales o false si son diferentes. Usa la función `strcmp()`.

7. **Eliminar espacios en blanco**

Crea una función `eliminarEspacios($texto)` que reciba una cadena de texto y devuelva el texto sin espacios en blanco al principio o al final. Usa la función `trim()`.

8. **Contar ocurrencias de una palabra en un texto**

Crea una función `contarOcurrencias($texto, $palabra)` que reciba un texto y una palabra, y devuelva cuántas veces aparece la palabra en el texto. Usa la función `substr_count()`.

9. **Dividir una cadena en palabras**

Crea una función `dividirPalabras($texto)` que reciba un texto y devuelva un array con todas las palabras. Usa la función `explode()`.

Funciones de Arrays (carpeta: funcionesArrays)

1. **Sumar valores de un array**

Crea una función `sumarArray($numeros)` que reciba un array de números y devuelva la suma total de sus valores. Usa la función `array_sum()`.

2. **Ordenar un array alfabéticamente**

Crea una función `ordenarArrayAlfabetico($nombres)` que reciba un array de nombres y lo ordene alfabéticamente. Usa la función `sort()`.

3. **Filtrar elementos mayores a un valor**

Crea una función `filtrarMayores($numeros, $valor)` que reciba un array de números y un valor límite, y devuelva un nuevo array con los números mayores a ese valor. Usa la función `array_filter()`.

4. **Buscar un valor en un array**

Crea una función `buscarEnArray($array, $valor)` que reciba un array y un valor, y devuelva true si el valor se encuentra en el array, o false si no se encuentra. Usa la función `in_array()`.

5. **Contar elementos de un array**

Crea una función `contarElementos($array)` que reciba un array y devuelva la cantidad de elementos que contiene. Usa la función `count()`.

6. **Obtener el valor máximo de un array**

Crea una función `obtenerMaximo($numeros)` que reciba un array de números y devuelva el valor máximo. Usa la función `max()`.

7. **Obtener el valor mínimo de un array**

Crea una función `obtenerMinimo($numeros)` que reciba un array de números y devuelva el valor mínimo. Usa la función `min()`.

8. **Eliminar duplicados de un array**

Crea una función `eliminarDuplicados($array)` que reciba un array y devuelva un nuevo array sin elementos duplicados. Usa la función `array_unique()`.

9. **Combinar dos arrays**

Crea una función `combinarArrays($array1, $array2)` que reciba dos arrays y los combine en uno solo. Usa la función `array_merge()`.

10. **Dividir un array en fragmentos**

Crea una función `dividirArray($array, $tamaño)` que reciba un array y un tamaño, y divida el array en fragmentos del tamaño indicado. Usa la función `array_chunk()`.

Parte 3: Uso de operadores ternarios en PHP y HTML (carpeta: pt-3ternarios)

1. **Mostrar mensaje de bienvenida (ex1.php)**

Crea un script PHP que determine si un usuario está autenticado. Si la variable `$autenticado` es `true`, debe mostrar un mensaje de bienvenida dentro de un `<h2>`. Si no lo está, debe mostrar un mensaje de "Por favor, inicie sesión".

2. **Mostrar disponibilidad de un producto (ex2.php)**

Crea una página que muestre si un producto está disponible o agotado. Usa una variable `$stock` para definir si hay unidades disponibles (más de 0). Si el stock es mayor que 0, debe mostrar "Producto disponible" en verde dentro de un `<p>`, y si no, "Producto agotado" en rojo.

3. **Campo de formulario con valor por defecto (ex3.php)**

En un formulario de contacto, crea un campo `<input>` que muestre un valor por defecto en el campo de "nombre" si la variable `$nombre` no está definida o está vacía. Usa un operador ternario para definir si muestra el valor o un texto por defecto como "Ingrese su nombre".

4. **Cambio de idioma en una página web (ex4.php)**

Supón que tienes un sitio web multilingüe. Crea una página que muestre un saludo en español si la variable `$idioma` es "es", y en inglés si la variable es "en". Usa un operador ternario para cambiar el saludo entre "Hola" y "Hello".

5. **Mostrar ícono de usuario logueado o invitado (ex5.php)**

Crea una página que muestre un ícono de usuario dependiendo si el

usuariu està logueado o no. Si la variable \$logueado es true, se mostrarà un ícono de usuariu dentro de un , si no lo está, se mostrarà un ícono de invitado.

