

# Práctica 1: Introducción a PHP

Para realizar esta práctica, crea una nueva carpeta llamada `prac1XY`, siendo `X` la inicial de tu nombre, e `Y`, la inicial de tu apellido. Por ejemplo, si te llamas Luz García, la carpeta se llamará `prac1LG`. A lo largo de toda la práctica, `X` e `Y` significan estas iniciales.

Los ejercicios estarán en el fichero `prac1XY.php`. Antes de cada ejercicio, imprime un `<h2>` indicando el número de ejercicio.

## Ejercicio 1: Bucles anidados

Crea dos variables, `rows` y `columns`. Asigna valores de manera que:

- `rows` = (posición de la inicial de tu nombre en el alfabeto) módulo 8 + 4
- `columns` = (posición de la inicial de tu apellido en el alfabeto) módulo 6 + 5.

Ejemplo: Para “Sete Ruiz”, `S=20` y `R=19`. Por tanto, `rows` =  $20 \bmod 8 + 4 = 8$ ; y `columns` =  $19 \bmod 6 + 5 = 6$ . Calcula este número por código en el propio ejercicio.

Crea las siguientes figuras dependiendo de tus variables para las filas y columnas. Yo sigo con el ejemplo (filas=8, columnas=6).

Primera figura (rectángulo completo):

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Segunda figura (triángulo izquierdo):

```
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Tercera figura (marco):

```

* * * * *
*           *
*           *
*           *
*           *
*           *
*           *
*           *
* * * * *

```

Cuarta figura (tablero de ajedrez):

```

*   *   *
  *   *   *
*   *   *
  *   *   *
*   *   *
  *   *   *
*   *   *
  *   *   *

```

## Ejercicio 2: Arrays bidimensionales

Crea un array bidimensional de 6x6 que represente temperaturas de diferentes ciudades durante 6 días. Rellénalo con números aleatorios entre -10° y 45°.

Encuentra:

- La temperatura más baja y más alta
- El día con mayor variación térmica
- La temperatura media por ciudad

Crea una tabla HTML que muestre:

- Temperaturas en azul si  $< 0^\circ$
- Temperaturas en rojo si  $> 35^\circ$
- Temperatura mínima en **negrita subrayada**
- Temperatura máxima en *cursiva marrón*
- Fondo verde claro para los fines de semana (columna 6)
- Fondo amarillo claro para la ciudad con temperatura media más alta

Los estilos deben estar en `styles/styleXY.css`

Aquí se muestra un ejemplo de cómo podría salir. Pero no tiene que ser idéntico, hazlo como más te guste:

Temperaturas de ciudades por día (°C)								
Ciudad/Día	Día 1	Día 2	Día 3	Día 4	Día 5	Día 6	Día 7	Media
Ciudad 1	18°	12°	32°	1°	35°	-8°	29°	15°
Ciudad 2	33°	9°	20°	13°	-9°	5°	31°	11.8°
Ciudad 3	43°	1°	-3°	44°	36°	11°	32°	22°
Ciudad 4	31°	28°	-5°	3°	7°	14°	-8°	13°
Ciudad 5	17°	8°	41°	37°	25°	33°	37°	26.8°
Ciudad 6	27°	23°	17°	34°	-7°	36°	7°	21.7°

#### Estadísticas

**Temperatura mínima:** -9°C (Día 5, Ciudad 2)

**Temperatura máxima:** 45°C (Día 3, Ciudad 7)

**Día con mayor variación:** Día 7 (51°C de diferencia)

## Ejercicio 3: Funciones

En `functions/functionsXY.php`, crea:

- filterByType:** Recibe dos parámetros: un array y un tipo ("par", "impar", "primo", "positivo", "negativo"). Devuelve un array con los elementos que cumplen la condición.
- calculateStatistics:** Recibe un array de números. Devuelve un array asociativo con las claves `media`, `mediana` y `moda`, y los valores correspondientes a esos [cálculos](#).
- analizarPalabras:** Recibe un texto y devuelve un array asociativo con las claves `number_of_words`, `longest_word`, `shortest_word`.
- convertTemperature:** Recibe tres parámetros: una temperatura, la unidad de origen, la unidad de destino ("celsius", "fahrenheit", "kelvin"). Devuelve la temperatura convertida (o false si se escogió una unidad que no existe). En caso de no indicar unidades, se considerará por defecto que la de origen es celsius y la de destino es fahrenheit.
- Inventa una función que reciba dos parámetros, uno de tipo simple (int, boolean, float, string...) y otro que sea un array asociativo. La función no imprime nada, solo devuelve valores. Inventa el funcionamiento de dicha función para que tenga sentido. Explica lo que hace, qué significan los parámetros y qué valor devuelve, en un comentario justo antes de la declaración de la función.

## Ejercicio 4: Arrays asociativos

Crea `functions/shopXY.php` con:

```
$productos = [
    'prod1' => [
        'nombre' => 'portátil gaming',
        'precio' => 899.99,
        'stock' => 15,
        'categoria' => 'electrónica'
    ],
    'prod2' => [
        'nombre' => 'mesa escritorio',
        'precio' => 120.50,
        'stock' => 8,
        'categoria' => 'hogar'
    ],
    'prod3' => [
        'nombre' => 'ratón inalámbrico',
        'precio' => 25.99,
        'stock' => 0,
        'categoria' => 'electrónica'
    ]
];
```

Funciones:

- **formatPrice:** Recibe precio y devuelve "899,99 €"
- **calculateIVA:** Recibe precio e IVA (por defecto 21%), devuelve precio con IVA
- **getStock:** Devuelve array con productos con stock > 0

En el fichero principal `prac1XY.php`, muestra una tabla con:

- Nombre formateado (primera letra mayúscula)
- Precio con IVA
- Stock en verde si > 10, amarillo si > 0, rojo si = 0

### Ejercicio 4.1

Añade `$productosConDescuento` que sea como el anterior array asociativo, pero que incluya descuentos solo a algunos productos (introduciendo la clave `descuento` solo en algunos elementos).

Vuelve a mostrar por pantalla el contenido en la misma tabla que antes, pero en caso de que tenga descuento, con el precio original tachado y el nuevo precio con descuento.

## Ejercicio 5: Cumplir requisitos

Inventa un ejercicio que tenga sentido en sí mismo. Deberás explicar en un párrafo html (<p>) el funcionamiento general. Debe cumplir los siguientes requisitos:

- Debe tener al menos un par de funciones en un fichero separado. Cada función debe tener en un comentario justo antes de su definición, lo que hace, los parámetros que recibe, y el valor que retorna. Todo debidamente justificado.
- Al menos una de las funciones tiene que trabajar con algún array.
- Debe tener al menos una hoja de estilos css en otro fichero separado. Los resultados que se muestran en tu pantalla deberán hacer uso de los estilos definidos aquí, y deben ser cambiantes entre ellos (utiliza id y/o clases en html para diferenciarlos). Al menos deberá haber 3 estilos distintos que se apliquen por id y/o por clases.
- Debe hacer uso de al menos 3 funciones de PHP, sin contar los echo. Puedes ver la documentación de PHP o los apuntes para inspirarte. Pueden ser de manipulación de cadenas, de matemáticas, de control de errores (isset, isint...).
- El ejercicio tiene que tener una funcionalidad concreta y la debes contar. No vale con hacer código aleatorio sin cohesión.

## Observaciones generales

A tener en cuenta en toda la práctica:

- Consulta la rúbrica de calificación en la práctica en el Aula Virtual.
- Demuestra que sabes utilizar bien PHP. Crea un código con sentido, respetando los estándares de nomenclatura, con nombres que tengan sentido. Pon comentarios en aquellas partes que son más complejas.
- Si el programa entregado tiene errores o *warnings* al interpretarse, se restará 3 puntos a la nota final.
- Habrá defensa individual de la práctica. Si se detecta que no se comprende su propio código o no se sabe hacer modificaciones sencillas, se restarán 3 puntos de la nota, además de anular los ítems correspondientes que no se conocen.
- Si se detectan copias, se calificará con 0 a todas las partes implicadas.
- No se aceptan entregas atrasadas.
- Para la entrega crea un .zip con todos los ficheros.