

# Desarrollo web en entorno servidor - DWES

## DWES03 - Introducción PHP.

Profesor: Pepe Lluyot  
pepe.lluyot@iescristobaldemonroy.es

### Introducción a PHP - Relación de ejercicios 2

Para poder realizar los siguientes ejercicios, vamos a elaborar una web con html5 con una estructura básica que nos servirá de base para incluir el código PHP.

#### Contenido:

<b>Introducción a PHP - Relación de ejercicios 2</b>	<b>1</b>
Sentencias de control: if, else, elseif	1
operador ternario	2
switch	2
Sentencias de control repetitivas	2
foreach	3
funciones	3
include, include_one, require, require_one	4
Funciones recursivas	5
Ejercicio repaso	5
Ejercicio para investigar algunas funciones:	6

#### Ejercicios (bloque II)

##### Sentencias de control: if, else, elseif

1. Crea un script que almacene la temperatura actual en una variable. Hacer uso de la función **rand()** para generar un número aleatorio entre -5 y 45.  
Si la temperatura es mayor o igual a 30 grados, muestra el mensaje "*Hace calor*". Si es menor de 15 grados, muestra "*Hace frío*". En cualquier otro caso, muestra "*El clima es templado*". Indicando en todo caso la temperatura obtenida.

- 
2. Crea un script que almacene el día de la semana (1-7) en una variable. Si se recibe el valor del día por URL mediante el parámetro *dia*, utilízalo; de lo contrario, genera un día aleatorio utilizando la función **rand()**. Usa **if**, **else**, y **elseif** para determinar si el día es laborable (de lunes a viernes) o es fin de semana (sábado o domingo). Si el día no está comprendido entre 1 y 7 mostrar un mensaje indicándolo.

### **operador ternario**

3. Utiliza el **operador ternario** para determinar si una persona es mayor de edad o no. Crea una variable con la edad de una persona y muestra el mensaje "Eres mayor de edad" o "Eres menor de edad" según sea el caso.
4. Crea un script que reciba el puntaje de un examen (de 0 a 10) por URL mediante el parámetro *nota*. Si no se recibe, genera una nota aleatoria entre 0 y 10. Usa el operador ternario para mostrar:
  - "Suspenso" si la nota es menor que 5.
  - "Aprobado" si la nota está entre 5 y 7.
  - "Notable" si la nota está entre 7 y 9.
  - "Sobresaliente" si la nota es 9 o más.

Además, indica si la nota ha sido redondeada (es entera) o si tiene decimales.

### **switch**

5. Crea una variable que contenga el nombre de una fruta y usa la sentencia **switch** para mostrar el color de la fruta. Por ejemplo, si la fruta es "manzana", muestra "El color es rojo", si es "plátano", muestra "El color es amarillo", etc.
6. Crea un script que almacene el número del día de la semana (1-7) en una variable. Utiliza una sentencia **switch** para mostrar el nombre del día en español. Luego, calcula cuántos días quedan hasta el próximo domingo y muestra este número.

### **Sentencias de control repetitivas**

7. Crea un script que use un bucle **while** para contar desde 1 hasta 10 y mostrar cada número en una nueva línea.
8. Crea un script que defina un array que contenga los nombres de varios estudiantes. Usa un bucle **while** para iterar sobre el array y mostrar cada nombre de estudiante en la pantalla. Al mismo tiempo, cuenta cuántos estudiantes tienen más de 5 letras en su nombre con la función **strlen()**. Cuando el bucle termine, muestra el número total de estudiantes y cuántos de ellos tienen más de 5 letras en su nombre.

- 
9. Utiliza un bucle **do..while** para contar hacia atrás desde 5 hasta 1 y mostrar cada número en una nueva línea.
  10. Crea un script donde definas un array que contenga las edades de varios estudiantes. Usa un bucle **do...while** para iterar sobre el array y calcular la suma total de las edades. Mientras iteras, cuenta cuántos estudiantes son mayores de edad (18 años o más). Al final del bucle, calcula la edad media dividiendo la suma total de las edades por el número total de estudiantes y muestra este valor junto con cuántos estudiantes son mayores de edad.
  11. Usa un bucle **for** para mostrar la tabla de multiplicar del número 7 (del 7x1 al 7x10). Si el resultado de una multiplicación es par se mostrará de un color y si es impar de otro. (Usa estilos css)

### **foreach**

12. Crea un script que defina un array simple con los nombres de varios colores. Usa un bucle **foreach** para mostrar cada color en la pantalla.
13. Crea un script que defina un **array asociativo** que contenga información sobre algunos estudiantes, donde las claves sean los nombres y los valores sean sus calificaciones (del 0 al 10, con decimales). Usa un bucle foreach para:
  - Mostrar el nombre de cada estudiante junto con su calificación.
  - Contar cuántos estudiantes tienen una calificación mayor o igual a 5.
  - Mostrar la calificación más alta y la más baja.

La información se mostrará en una **tabla html**.

### **funciones**

14. Crea una **función** que genere un número aleatorio entre 1 y 100. La función debe devolver un mensaje que indique si el número generado es par o impar. Por ejemplo "El número 32 es par". Realiza la llamada varias veces, mostrando por pantalla el texto que devuelve.
15. Crea una **función** llamada generarTabla que reciba un número entero como **parámetro**. Este número representará tanto el número de filas como el número de columnas de la tabla. La función debe generar y devolver una cadena de texto que contenga una tabla HTML, donde cada celda esté llena con números correlativos, comenzando desde 1. Luego, el script debe llamar a la función varias veces con diferentes valores y mostrar las tablas generadas.

- 
16. Crear un script que permita calcular la media de las notas de un grupo de estudiantes y clasificar los resultados en función de la media obtenida. Para ello:
- **paso de parámetros:** La aplicación debe recibir un array de notas a través de la URL. Las notas se pasarán en la siguiente estructura:  
ej16.php?notas=6.5,7.0,8.5,9.0,4.5
  - **Validación de Notas:** Cada nota debe ser un número entre 0 y 10. Si hay alguna nota fuera de este rango, el programa debe mostrar un mensaje indicando que las notas deben estar entre 0 y 10 y terminar su ejecución.
  - **Función:** Definir una función llamada `calcularMedia($notas)` que reciba el array de notas como parámetro y retorne la media de las notas. Hacer uso de la función en el script para calcular la media de las notas obtenidas.
  - **Mensaje:** Se mostrará un mensaje por pantalla indicando la media del grupo y si el grupo ha aprobado o suspendido (aprobado si su media es  $\geq 5$ ). Además debe contar cuántas notas son aprobadas ( $\geq 5$ ) y cuántas son suspendidas ( $< 5$ ).

*Nota: puedes usar la función **explode**, **array\_sum**.*

17. Genera una función de forma que al llamarla imprima el contenido de un array de forma formateada usando las etiquetas `<pre>`. Hacer uso de **print\_r** con el segundo argumento a `true`.

### **include, include\_one, require, require\_one**

18. Crea un archivo PHP llamado `config.php` que contenga las variables de configuración necesarias para conectarte a una base de datos, incluyendo el host, el usuario, la contraseña y el nombre de la base de datos. Luego, crea un archivo PHP llamado `index.php` que incluya `config.php` usando **include**, y que muestre por pantalla la información de conexión.
19. Define un archivo llamado `ej18_funciones.php` que contenga una función llamada `saludar($nombre)` que reciba un nombre como parámetro y retorne un saludo. Luego, en otro archivo llamado `ej18_index.php`, utiliza **include once** para incluir `ej18_funciones.php` y llama a la función `saludar` con diferentes nombres. Asegúrate de que el archivo no se incluya más de una vez.
20. Crea un archivo PHP llamado `config.php` que contenga las variables necesarias para la conexión a una base de datos, similar al primer ejercicio. Luego, crea un archivo PHP llamado `index.php` que use **require** para cargar `config.php` y mostrar la

---

información de conexión. Asegúrate de que el script se detenga si config.php no se encuentra.

21. Define un archivo PHP llamado funciones.php que contenga la misma función saludar(\$nombre) del ejercicio anterior. Luego, en un archivo llamado index.php, utiliza **require\_once** para cargar funciones.php. Llama a la función saludar con diferentes nombres. Asegúrate de que el archivo no se requiera más de una vez y verifica que el script funcione correctamente.

### Funciones recursivas

22. Crea un archivo PHP llamado factorial.php que contenga una **función recursiva** llamada factorial(\$n) para calcular el factorial de un número. La función debe retornar el producto de todos los enteros positivos hasta un número dado n.
23. Crea una función recursiva para que calcule la suma de los n primeros números naturales
24. Crea una función recursiva para que sume los dígitos de un número entero

### Ejercicio repaso

25.- En este ejercicio, implementarás un sistema para registrar animales en un zoológico y llevar un seguimiento de las interacciones que los visitantes tienen con ellos. Cada animal tiene un comportamiento particular y su número de interacciones depende de cuántos visitantes se acercan a ellos.

El script debe permitir registrar interacciones, mostrar el animal más interactivo, y calcular el promedio de interacciones por especie. Asegúrate de que todos los cálculos funcionen correctamente, incluso si hay varias especies y animales con el mismo número de interacciones. Para ello:

- Crea un array **\$animales** que contenga información sobre diferentes animales. Cada animal será representado como un array asociativo con las claves: nombre (nombre del animal), especie (especie del animal), e interacciones (número de interacciones que ha tenido con los visitantes):

```
$animales = [  
    ['nombre' => 'Simba', 'especie' => 'León', 'interacciones' => 5],
```

---

```
['nombre' => 'Dumbo', 'especie' => 'Elefante', 'interacciones' => 10],  
['nombre' => 'George', 'especie' => 'Mono', 'interacciones' => 8], etc....];
```

- Crea una función **registrar\_interaccion**(&\$animales, \$nombre\_animal) que reciba el array de animales por referencia y el nombre de un animal y aumente en 1 el contador de interacciones de ese animal.
  - Si el animal no existe en el array, la función debe mostrar un mensaje de error indicando que no se encontró el animal.
  - Si no hay animales registrados en el array, la función debe mostrar un mensaje indicando que el zoológico está vacío.
- Crea una función **animal\_mas\_interactivo**( \$animales) que devuelva el nombre del animal que ha tenido más interacciones. Si hay más de un animal con la misma cantidad máxima de interacciones, la función debe devolver todos los nombres de esos animales. (Se puede hacer uso de las funciones **array\_column** y **max**)
- Crea una función **promedio\_interacciones\_por\_especie**( \$animales) que calcule el promedio de interacciones de los animales para cada especie.
  - La función debe devolver un array asociativo donde las claves sean los nombres de las especies y los valores sean los promedios de interacciones.

*NOTA: echar un vistazo a las funciones **array\_column** y **in\_array**, para poder realizar búsquedas más simples en un array multidimensional*

### Ejercicio para investigar algunas funciones:

26.- Tenemos un archivo de texto llamado productos.txt que contiene información sobre varios productos en un formato estructurado. Cada línea del archivo representa un producto y tiene el siguiente formato:

**ID|Nombre|Precio|Cantidad** ([enlace](#))

Debes generar un script que realice las siguientes acciones:

- Leer el archivo productos.txt y cargar la información de los productos en un array bidimensional asociativo.
- Después imprime una tabla con los productos, mostrando su ID, Nombre, Precio, y Cantidad.

---

Puedes generar varias funciones:

- `cargarProductosDesdeArchivo($archivo)`: Lee el archivo línea por línea, divide cada línea en las partes correspondientes (ID, Nombre, Precio y Cantidad), y carga cada producto en un array asociativo.

Usa las funciones: **fopen**, **fread** para leer cada fila del fichero. Usa **explode** por cada línea para obtener los valores de cada campo. Cierra el fichero con **fclose**

- `imprimirProductos($productos)`: Imprime los productos en formato de tabla con las columnas ID, Nombre, Precio y Cantidad, ajustando los espacios para alinearlos correctamente.

Puedes generar una función genérica que reciba un array bidimensional asociativo, y vuelque su contenido en una tabla HTML y así tener una función genérica que obtiene una tabla html con sus campos.

-----

(snippets)

```
"php inicio vacio": {  
    "prefix": "pi",  
    "body": [  
        "<?php",  
        "\t$0",  
        "?>"  
    ],  
    "description": "apertura y cierre php"  
}
```

---

operador ??

---

Este operador comprueba si la primera variable es nula en cuyo caso devuelve la segunda variable y en caso contrario devuelve la primera, esto es

**`$value ?? $other`**

Es lo mismo que

**`$value != null ? $value : $other`**

Que es lo mismo que

```
if ($value != null) {  
    return $value;  
} else {  
    return $other;  
}
```