

UD 2. Prácticas

Índice de contenidos

Prácticas.....	2
Práctica 1: Introducción a PHP.....	2
Práctica 2: Formularios con PHP.....	4
Ejercicios de ampliación.....	8
Ampliación 1: Formulario con correo electrónico.....	8
Ampliación 2: URLs amigables.....	8
Ejercicios de refuerzo.....	10
Refuerzo 1: Etapas.....	10
Refuerzo 2: Validador de fechas.....	10
Refuerzo 3: Función potencia.....	10
Refuerzo 4: Número mayor en un array.....	10
Refuerzo 5: Suma de números pares.....	10
Refuerzo 6: Números primos en un rango.....	11
Refuerzo 7: Filtro de edades.....	11
Refuerzo 8: Descomposición de billetes.....	11
Refuerzo 9: CalcURLadora.....	11
Refuerzo 10: Clasificación de liga.....	12
Refuerzo 11: Catálogo de productos.....	12
Refuerzo 12: Tabla de multiplicar.....	13
Refuerzo 13: Mayor, menor y media.....	13
Refuerzo 14: Horóscopo.....	13
Refuerzo 15: Edades.....	14

Prácticas

Práctica 1: Introducción a PHP

Preparación

Crea una carpeta llamada "**practical**" en tu directorio de Apache. Dentro de esa carpeta, crea un archivo **index.html** donde podamos ver un listado con enlaces a cada uno de los scripts siguientes que hay que desarrollar.

En cada script, además del código, incluye al inicio un comentario PHP que contenga el enunciado. Llama a cada archivo de la forma **x.php** (x es el número de script)

Enunciado

1. Escribe el siguiente script y comprueba que aparece el mensaje "Hola Mundo" en el navegador:

```
<html>
<head>
  <title>Prueba de PHP</title>
</head>
<body>
  <?php echo '<p>Hola Mundo</p>'; ?>
</body>
</html>
```

2. Escribe un script embebido en una página html que muestre tu nombre completo por pantalla. Utiliza código PHP para imprimir el mensaje.
3. Modifica el script anterior para que muestre tu horóscopo y tu equipo de fútbol favorito. Cada dato se debe mostrar en una línea diferente. Utiliza PHP para imprimir tanto el mensaje como algo de HTML.
4. Escribe un script que utilice las variables \$x y \$y. Asígnale los valores 144 y 999 respectivamente. A continuación, muestra por pantalla el valor de cada variable y, en una lista, la suma, la resta, la división y la multiplicación de ambos números. Añade comentarios de una línea y de múltiples líneas.
5. Crea la variable \$nombre y la variable \$apellido y asígnale tu nombre y primer apellido respectivamente. Muestra su valor por pantalla de tal forma que el resultado sea el mismo que el del ejercicio 2.
6. Crea las variables \$nombre, \$horoscopo y \$equipo_futbol y asígnale los valores adecuados. Muestra los valores por pantalla de tal forma que el resultado sea el mismo que el del ejercicio 3.

7. Realiza un conversor de euros a dólares. La cantidad en euros que se quiere convertir deberá estar almacenada en una variable. La tasa de cambio es $1 \text{ €} = 0,97 \text{ \$}$. Imprime el resultado con dos decimales.
8. Escribe un script que pinte por pantalla un triángulo rectángulo relleno a base de asteriscos línea a línea. La base del triángulo debe estar formado por 9 asteriscos. Usa bucle.
9. Escribe un script que tenga recogidos en variables los valores de la velocidad inicial (v_0), el tiempo transcurrido (t) y la aceleración (a) de un objeto en movimiento y calcule la distancia recorrida según la siguiente fórmula:

$$d = v_0 * t + 0.5 * a * t^2$$

La fórmula debe ser implementada a través de una función que devuelva el resultado. El script debe mostrar el valor de cada variable y el resultado.

Práctica 2: Formularios con PHP

Descripción

Crear una pequeña aplicación web (en php) que gestione una sencilla lista de la compra, permitiéndonos añadir y borrar productos.

La página

La página mostrará:

- Un título en el que aparezca “LISTA DE LA COMPRA PARA EL 01/10/2022” (siendo esta fecha la fecha actual generada con PHP).
- El estado actual de la lista (1).
- Dos formularios:
 - Insertar artículo (2).
 - Eliminar artículo (3).

(1) La lista

Toda la información referente a la lista de la compra sera almacenada en un array de productos en el que para cada producto recogeremos en un array asociativo los campos "nombre", "cantidad", "precio" y "total".

A la hora de mostrar la lista de la compra se debe mostrar, en una tabla, una fila por cada elemento de la lista de la compra. Por cada fila deben aparecer las 4 columnas: Nombre, Cantidad, Precio y Total.

Al final del listado se debe mostrar otro campo calculado que sume todos los precios totales, es decir, todas las columnas Total indicando el Precio Total de la Compra. Si no hay nada que mostrar se debe avisar al usuario.

Los formularios (2) y (3)

El destino del formulario será la misma página.

Si las operaciones se han llevado a cabo correctamente, se debe mostrar un mensaje de confirmación. Si ha habido algún error, se debe mostrar un mensaje de error explicando el problema y respetar los datos introducidos por el usuario.

(2) Insertar artículos

Para la inserción de elementos en la lista de la compra, crea un formulario en el que se solicite la siguiente información:

- Nombre
- Cantidad
- Precio

Ten en cuenta que:

- El Nombre hace referencia al producto que se quiere añadir a la lista, la Cantidad indicará el número de unidades que se quiere comprar de ese producto y el Precio será el valor unitario de cada producto.
- El Nombre será un campo obligatorio y no se deberá añadir el registro al array si está vacío o si el producto ya existe.
- El total se calculará a partir de los datos introducidos mediante una función.
- Si se añade el artículo correctamente, se debe mostrar un mensaje de confirmación.

(3) Eliminar artículos

Al eliminar un registro de la lista este debe ser eliminado del array.

Otros aspectos

- En el código debe haber comentarios allí donde sea necesario.
- Deben utilizarse funciones, pueden crearse tantas como se precisen. Al menos existirán dos funciones:
 - `calcularPrecioTotalProducto()`: devolverá la cantidad calculada (se hará paso por referencia).
 - `calcularPrecioTotalCompra()`: será la función la que devuelva la cantidad calculada.
- Todas las funciones deben ser implementadas en un fichero independiente llamado `funciones.php`

Esqueleto

Para facilitar el desarrollo, se adjunta un esqueleto de la práctica que, a través de campos hidden, permite la propagación del array de productos.

lista.php

```

<?php
// TODO: Incluir el archivo de funciones
$productos = array();
if (/* TODO: Si el botón de insertar ha sido pulsado... */) {
    $productos = array_merge($productos, recuperarProductos());
}
if (/* TODO: Si el botón de borrar ha sido pulsado... */) {
    $productos = array_merge($productos, recuperarProductos());
}
?>
<html>
    <head>
        <title>Lista de la compra</title>
    </head>
    <body>
        <h2>Añadir producto</h2>
        <form>
            <?php echo hiddenProductos($productos); ?>
        </form>
        <h2>Borrar producto</h2>
        <form>
            <?php echo hiddenProductos($productos); ?>
        </form>
    </body>
</html>

```

funciones.php

```

<?php
// Recibe un array con información de un producto POR REFERENCIA y le asigna
el valor de total
function calcularPrecioTotalProducto(/* producto */)
{
}

// Recibe lista de productos
// Devuelve la suma de todos los totales
function calcularPrecioTotalCompra(/* productos */)
{
}

function hiddenProductos($productos)
{
    $html = "";
    foreach ($productos as $producto) {
        $html .= '<input type="hidden" name="hidNombres[]" value="' .
        $producto['nombre'] . '" />';
        $html .= '<input type="hidden" name="hidCantidades[]" value="' .
        $producto['cantidad'] . '" />';
        $html .= '<input type="hidden" name="hidPrecios[]" value="' .
        $producto['precio'] . '" />';
        $html .= '<input type="hidden" name="hidTotales[]" value="' .
        $producto['total'] . '" />';
    }
    return $html;
}

function recuperarProductos()
{
    $productos = array();
    if (isset($_POST['hidNombres'])) {
        for ($i=0; $i<count($_POST['hidNombres']); $i++) {
            $producto = array();

```

```
        $producto['nombre'] = $_POST['hidNombres'][$i];
        $producto['cantidad'] = $_POST['hidCantidades'][$i];
        $producto['precio'] = $_POST['hidPrecios'][$i];
        $producto['total'] = $_POST['hidTotales'][$i];
        $productos[] = $producto;
    }
    return $productos;
}
```

Ejercicios de ampliación

Ampliación 1: Formulario con correo electrónico

Crea un formulario de contacto para un sitio web. El formulario constará de los siguientes campos:

- Nombre
- Correo electrónico
- Tipo de mensaje: queja, sugerencia o consulta
- Mensaje

También debe recoger la IP del visitante de forma oculta.

El formulario debe comprobar que no haya campos en blanco y que el email "sea un email" y, si todo está correcto, enviarte el contenido del formulario a tu correo electrónico, por ejemplo usando la función `mail()` de PHP.

Nota: probablemente necesitarás instalar un servicio de email en tu equipo y una cuenta de algún proveedor de correo electrónico.

Ampliación 2: URLs amigables

Crea un script PHP llamado `tienda.php` que reciba a través de la URL dos variables: `categoria` e `id`. Ejemplo:

```
tienda.php?categoria=televisores&id=42392
```

El script lo único que debe hacer es imprimir el mensaje: El visitante quiere consultar el producto {ID} de la categoría {CATEGORIA}.

A continuación crea la siguiente URL amigable:

```
/CATEGORIA/producto/ID
```

De forma que la url `/televisores/producto/42392` equivalga a la URL `tienda.php?categoria=televisores&id=42392`

Notas:

- Puedes consultar el siguiente enlace:
<https://www.arsys.es/blog/soluciones/configuracion-de-programas/url-amigables-htaccess/>.
- Es posible que necesites instalar el módulo `mod_rewrite` de Apache.

Ejercicios de refuerzo

Refuerzo 1: Etapas

A partir de una edad, muestra por pantalla:

- “bebé” si tiene menos de 3 años
- “niño” si tiene entre 3 y 12 años
- “adolescente” entre 13 y 17 años
- “adulto” entre 18 y 66
- “jubilado” a partir de 67

Refuerzo 2: Validador de fechas

Crea una función que reciba un día, mes y año y devuelva *true* si la fecha es válida o *false* en caso contrario. Puedes utilizar el código de la actividad 2.3 para comprobar si el año es bisiesto.

Refuerzo 3: Función potencia

Escribe una función PHP que reciba dos parámetros (A y B) y devuelva el valor de la potencia de A elevado a B (AB).

Escribe también un programa PHP que haga uso de esa función para calcular potencias.

Refuerzo 4: Número mayor en un array

Crea un array numérico con al menos 10 valores y escribe un programa que recorra el array usando un bucle `foreach`, encontrando y mostrando el número mayor.

Refuerzo 5: Suma de números pares

Escribe un programa que cree un array numérico con 20 números enteros aleatorios entre 1 y 100. Puedes usar la función `rand(int $min, int $max) : int` para generar los números. Luego, recorre el array y suma solo los números pares, mostrando el resultado final.

Refuerzo 6: Números primos en un rango

Escribe un programa que imprima todos los números primos entre 1 y 100. Utiliza un bucle `for` para iterar por el rango. Deberás crear previamente la función `es_primo($num)`.

Refuerzo 7: Filtro de edades

Inicializa un array numérico llamado `$personas`.

Luego crea un array asociativo llamado `$persona1` y rellénalo con las siguientes claves: “nombre”, “apellido1”, “apellido2” y “edad”.

Añade el array asociativo al array numérico anterior.

Vuelve a crear otros 4 arrays asociativos de personas (`$persona2`, `$persona3`, `$persona4` y `$persona5`) y añádelos al array numérico.

En total, deberás tener un array numérico con 5 elementos. Cada elemento será un array asociativo.

Por último, usa un bucle para recorrerlo y mostrar solo las personas que son mayores de edad (≥ 18 años).

Refuerzo 8: Descomposición de billetes

A partir de una cantidad de dinero, mostrar su descomposición en billetes (500, 200, 100, 50, 20, 10, 5) y monedas (2, 1), para que el número de elementos sea mínimo. Por ejemplo, al introducir 139 debe mostrar:

```
1 billete de 100
0 billete de 50
1 billete de 20
1 billete de 10
1 billete de 5
2 moneda de 2
```

Refuerzo 9: CalcURLadora

Se pide crear una calculadora utilizando los parámetros GET de la URL. No se pide un formulario, sino introducir manualmente los parámetros en la URL.

- Los parámetros serán los siguientes:
 - `n1`
 - `n2`
 - `operacion`

- En la página se mostrará el resultado de la operación con los dos números introducidos (n1 y n2):
 - Si “operacion” es “suma”, los sumará.
 - Si “operacion” es “resta”, los restará.
 - Si “operacion” es “multiplicacion”, los multiplicará.

Refuerzo 10: Clasificación de liga

Crea un script que contenga un array con los nombres de los equipos de futbol de la liga, ordenados según su clasificación actual. Dicho script mostrará una tabla con la clasificación de los equipos, a no ser que reciba alguna de las siguientes variables por GET:

- Si se recibe por GET la variable “posicion”, se debe imprimir un mensaje que indique qué equipo ocupa esa posición, y qué equipos están por encima y por debajo.
- Si se recibe por GET la variable “nombre”, se debe imprimir un mensaje que indique la posición del equipo que tenga ese nombre.

Refuerzo 11: Catálogo de productos

Vas a crear una página PHP que simule un catálogo de productos y permita filtrar los productos por dos parámetros GET: `categoria` y `precio_max`. Dependiendo de los valores que se pasen en la URL, la página mostrará los productos que coincidan con los filtros.

El catálogo de productos se implementa a través de un array lleno de arrays asociativos. Cada array asociativo se corresponde con un producto. Puedes copiar el siguiente código:

```
// Lista de productos (nombre, categoría, precio)
$productos = [
    ["nombre" => "Teléfono móvil", "categoria" => "Electronica", "precio" => 800],
    ["nombre" => "Televisor", "categoria" => "Electronica", "precio" => 1200],
    ["nombre" => "Camiseta", "categoria" => "Ropa", "precio" => 20],
    ["nombre" => "Pantalón", "categoria" => "Ropa", "precio" => 40],
    ["nombre" => "Silla", "categoria" => "Hogar", "precio" => 150],
    ["nombre" => "Mesa", "categoria" => "Hogar", "precio" => 300],
];
```

La página debe recibir dos parámetros a través de la URL:

- `categoria`: La categoría del producto (puede ser "Electrónica", "Ropa", "Hogar", etc.).
- `precio_max`: Un precio máximo para los productos mostrados.

Filtra la lista de productos para mostrar solo los que coincidan con la categoría especificada y que tengan un precio igual o inferior al valor de `precio_max`.

Si no se pasan los parámetros, se debe mostrar la lista completa de productos.

Cuando termines, pruébalo:

- **Sin filtros** (muestra todos los productos): <http://localhost/refuerzo11.php>
- **Filtrando por categoría** (debería mostrar "Camiseta" y "Pantalón"): <http://localhost/refuerzo11.php?categoria=Ropa>
- **Filtrando por categoría y precio máximo** (debería mostrar "Teléfono móvil" porque es el único producto en la categoría "Electronica" cuyo precio es menor o igual a 1000): http://localhost/refuerzo11.php?categoria=Electronica&precio_max=1000

Refuerzo 12: Tabla de multiplicar

Escribe un programa que pida un número al usuario mediante un formulario y muestre su tabla de multiplicar hasta el 25 en una tabla HTML.

Refuerzo 13: Mayor, menor y media

Escribe un programa que pida cinco números al usuario mediante un formulario y los guarde en un array.

Luego debe llamar a una función pasándole el array como parámetro, y la función calculará cuál de los cinco números es el mayor, cuál el menor y cuánto vale la media, devolviendo esos tres valores en otro array.

Por último, se mostrarán en la pantalla el mayor, el menor y la media.

Refuerzo 14: Horóscopo

Crea un formulario que reciba el nombre del usuario y su día, mes y año de nacimiento. Una vez enviado el formulario se mostrará el signo del zodiaco correspondiente y un breve texto relacionado. Se debe controlar que la fecha sea válida y que los campos no estén en blanco.

Refuerzo 15: Edades

Tras leer la edad de una persona, mostrar la edad que tendrá dentro de 10 años y hace 10 años. Además, muestra qué año será en cada uno de los casos. Finalmente, muestra el año de jubilación suponiendo que trabajarás hasta los 67 años. En este caso, no hace falta que previamente crees un formulario, puedes probar el ejercicio via URL: `edades.php?edad=33`