

Relación de ejercicios

1. Escribe un programa que nos diga el horóscopo a partir del día y el mes de nacimiento.
2. Escribe un programa que ordene tres números enteros introducidos por teclado.
3. Crear una página que muestre un mensaje de bienvenida que dependa de la hora actual, de la siguiente manera (figura 4):
 - Si la hora se encuentra entre las 8 y las 13, mostrará 'Buenos días'
 - Si la hora se encuentra entre las 14 y las 20, mostrará 'Buenas tardes'
 - Si la hora se encuentra entre las 21 y las 7, mostrará 'Buenas noches'

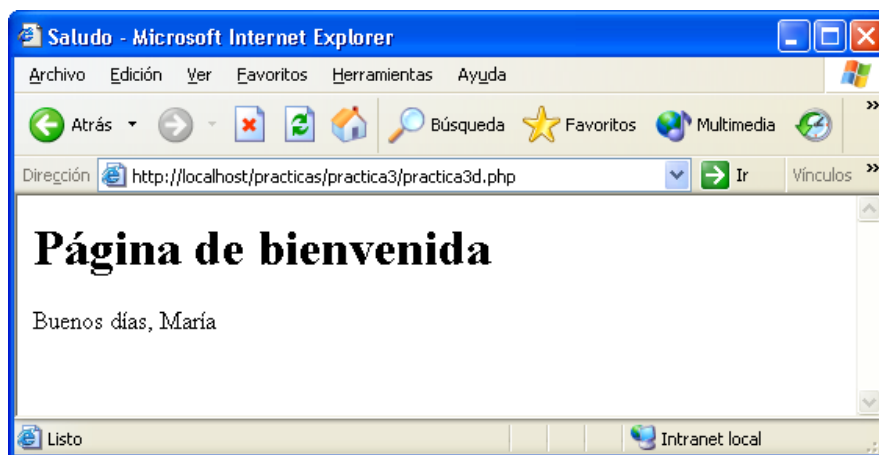


Figura 4: página con mensaje de bienvenida

4. Se pretende obtener información de un fichero a partir de su nombre. Para ello se van a definir las siguientes funciones:

a) string calcula_extension (string)

Recibe una cadena de caracteres que representa el nombre de un fichero y devuelve una cadena con la extensión del fichero.

Ejemplo: la llamada

```
$extension = calcula_extension ("curriculum.pdf");  
devuelve en la variable $extension el valor "PDF".
```

b) string tipo_fichero (string)

Recibe una cadena de caracteres que representa una extensión de fichero y devuelve una cadena con el tipo de fichero que corresponde a dicha extensión (véase Tabla I).

Ejemplo: \$tipo = tipo_fichero ("PDF");

devuelve en la variable \$tipo el valor "Documento Adobe PDF".

Tabla I: Tipos de ficheros

Extensión	Tipo
"PDF"	"Documento Adobe PDF"
"TXT"	"Documento de texto"
"HTML", "HTM"	"Documento HTML"
"PPT"	"Presentación Microsoft Powerpoint"
"DOC"	"Documento Microsoft Word"
"GIF"	"Imagen GIF"
"JPG"	"Imagen JPG"
"ZIP"	"Archivo comprimido ZIP"
Otra extensión	"Archivo" + extensión (por ejemplo, "Archivo PHP")

Se pide: Crear un fichero cadena.php con estas funciones y una página practica4a.php que, utilizando el código que se proporciona, obtenga y muestre el tipo de un fichero a partir de una variable con su nombre (figura 1).

```
<?PHP

$fichero = "curriculum.pdf";

$extension = calcula_extension ($fichero);

$tipo = tipo_fichero ($extension);

print ("El fichero '$fichero' es de tipo '$tipo'.\n");

?>
```

Nota

Para construir la primera de las funciones se pueden utilizar las siguientes funciones de manipulación de cadenas de PHP:

- stristr
- ltrim
- strtoupper

Consultar el manual de PHP para averiguar el objetivo y la sintaxis de cada una de ellas.

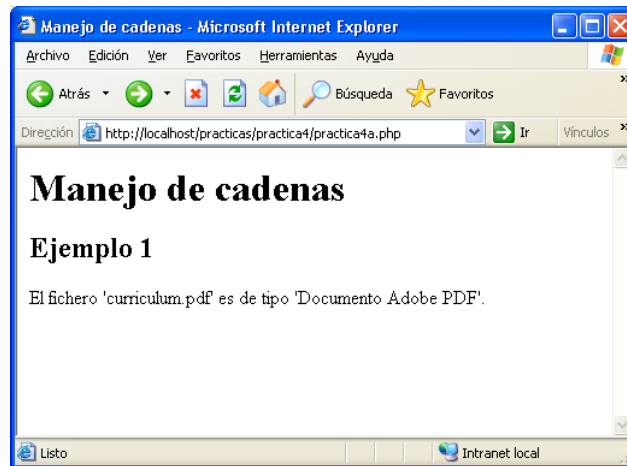


Figura 1: página que muestra el tipo de un fichero

5. Realiza un programa que simule el control de acceso a una caja fuerte. La combinación será un número de 4 cifras. Por lo tanto, a través de la URL se recibirán 4 valores. Si acertamos, se mostrará el mensaje “Caja abierta satisfactoriamente”. Si no acertamos se mostrará “Lo siento no es la combinación”.

6. Sea la cadena de caracteres

\$texto = "uno-dos-tres-cuatro-cinco";

Crear una página que, a partir de esta cadena, muestre una lista con los elementos de la misma que resultan de partirla usando como carácter separador el guión '-' (figura 2).

Utilizar las siguientes funciones de PHP:

- explode
- count

Consultar el manual de PHP para averiguar el objetivo y la sintaxis de cada una de ellas.

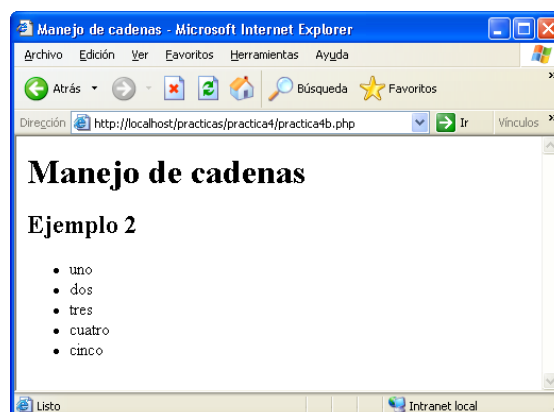


Figura 2: página que extrae las palabras de una cadena

7. Rellena un array bidimensional de 6 filas por 9 columnas con número enteros positivos comprendidos entre 100 y 999 (ambos incluidos). Todos los números deben de ser distintos, es decir, no se puede repetir ninguno. Muestra a continuación por pantalla el contenido del array de tal forma que se cumplan los siguientes requisitos:
 - a. Los números de las dos diagonales donde está el mínimo deben aparecer en color verde
 - b. El mínimo debe aparecer en color azul
 - c. El resto de los números deben de aparecer en color negro.
8. Realiza un programa que escoja al azar 10 cartas de la baraja española y que diga cuántos puntos suman según el juego de la brisca. Emplea un array asociativo para obtener los puntos a partir del nombre de la figura de la carta. Asegúrate de que no se repite ninguna carta, igual que si las hubieras cogido de una baraja de verdad.
Los valores de las cartas son:
'As' => 11, 'Dos' => 0, 'Tres' => 10, 'Cuatro' => 0, 'Cinco' => 0,
'Seis' => 0, 'Siete' => 0, 'Sota' => 2, 'Caballo' => 3, 'Rey' => 4
9. Crea una función para resolver la ecuación de segundo grado. Esta función recibe los coeficientes de la ecuación y devuelve un array con las soluciones. Si no hay soluciones reales, devuelve FALSE.
10. Almacena la función anterior en el fichero matematicas.php. Crea un fichero que la incluya y la utilice
11. Escribe una función que reciba una cadena y compruebe si es un palíndromo
12. Escribe una función que reciba un array de números, y un número, el límite. La función tiene que devolver un nuevo array que contenga solo los elementos del array original menores que el límite.
13. Dada la frase “Esta cadena tiene muchas letras” muestra:
 - La posición de la primera ocurrencia de la letra ‘a’.
 - La posición de la primera ocurrencia de la letra ‘m’.
 - La posición de la primera ocurrencia de la palabra “tiene”.
 - La primera ocurrencia desde el final de la letra ‘a’.
 - La frase desde la aparición de la palabra “cadena” hasta el final.
 - La cadena desde el carácter 12 hasta el final.
 - La cadena devolviendo 6 caracteres desde el carácter 18.
 - La cadena devolviendo los 6 últimos caracteres.
 - La cadena desde la posición 26, contando desde atrás, 6 caracteres.
 - La cadena empezando en el carácter 4 y terminando en el 7 desde atrás.

Se mostrará una página parecida a la siguiente:

La primera ocurrencia de 'a' es 3
La primera ocurrencia de 'm' es 18
La primera ocurrencia de 'tiene' es 12
La primera ocurrencia desde el final de 'a' es 29
La frase desde la aparición de la palabra "cadena" hasta el final es : cadena tiene muchas letras

Partes de la cadena

tiene muchas letras muchas
letras
cadena
cadena tiene muchas

14. Dada la frase "Bienvenidos a la aventura de aprender PHP en 30 horas" utilizar funciones de cadena para:

- Mostrar la parte central de la frase
- Averiguar la posición de la palabra PHP
- Reemplazar la palabra "aventura" por la cadena "misión".

15. Crea un array de dos dimensiones, de manera que en una dimensión contenga el tipo de colores (fuerte o suave) y en la otra 3 colores de cada tipo. Muestra una tabla como la siguiente recorriendo el array:

Colores fuertes:	Rojo:FF0000	Verde:00FF00	Azul: 0000FF
Colores suaves:	Rosa:FE9ABC	Amarillo:FFD700	Malva:DDA0DD

16. Crea un array llamado pila como éste: \$pila = array("cinco" => 5, "uno"=>1, "cuatro"=>4, "dos"=>2, "tres"=>3); Muestra el array ordenado por asort, arsort, ksort, sort, rsort.

17. Crea un array con los países de la Unión Europea y sus capitales. Muestra por cada uno de ellos la frase: "La capital de <> es <>". Luego ordena la lista por el nombre del país y luego por el nombre de la capital.