



FUNCIONES

EJERCICIO 1

Escribir una función HTML – PHP “corregir_primera_letra()” que reciba una cadena y convierta la primera letra en mayúscula.

EJERCICIO 2

Escribir una función HTML – PHP “corregir_mayusculas()” que reciba una cadena y ponga el primer carácter en mayúsculas y el resto en minúsculas

EJERCICIO 3

Escribir una función HTML – PHP “contar_letra_a()” que cuente el número de ‘a’ que hay en una frase.

EJERCICIO 4

Escribir una función HTML – PHP “contar_mayusculas()” que cuente el número de letras mayúsculas que hay en una frase.

EJERCICIO 5

Escribir una función HTML – PHP “contar_letra()” que reciba una frase y una letra, la función devolverá el número de veces que aparece la letra en la cadena.

Agregar a esta función un tercer parámetro “casesensitive” con el que el usuario pueda elegir si quiere que la función distinga entre mayúsculas y minúsculas. Si el usuario no dice nada, la función sí distinguirá entre mayúsculas y minúsculas.

EJERCICIO 6

Modifica el ejercicio 5 de la relación 1 del tema 7 para que el calendario se cree mediante funciones.





EJERCICIO 7

Programar una página en HTML – PHP que pida al usuario su nombre completo y sus notas de matemáticas, lengua, historia y dibujo. El usuario introducirá las notas de forma numérica y la página web deberá mostrar, utilizando funciones, su boletín de notas:

Alumno	Juan Ramírez
Matemáticas	Sobresaliente
Lengua	Notable
Historia	Notable
Dibujo	Insuficiente

EJERCICIO 8

Crear una función PHP que reciba una fecha y diga los días que han pasado desde esa fecha hasta hoy. En caso de que la fecha recibida sea una fecha futura, devolverá los días en negativo.

Programar después un sitio en HTML – PHP que pida al usuario el nombre de 3 personas así como el día que cada uno de ellos debía hacer el pago de su cuota. El sitio Web mostrará un cuadro como el siguiente:

Nombre	Fecha de pago	Recargo
Nombre1	Fecha1	X
Nombre2	Fecha2	X

Nota: tener en cuenta que por cada día de retraso al pagar la cuota, el recargo será de 5€





EJERCICIO 9

Crear una función PHP que reciba el código de un color y un número. La función escribirá una celda de una tabla (sólo la celda) con el color de fondo pasado como parámetro y el número escrito.

Crear después una aplicación Web que pida al usuario un mes concreto y (por medio de listas desplegables) le permita elegir el color para los días festivos, para los fines de semana y para los laborables. La aplicación Web mostrará el mes elegido por el usuario, marcando cada día con el color correspondiente.

EJERCICIO 10

Crear una función que reciba un array asociativo con las posiciones: nombre, apellidos, nota1, nota2, nota3. La función mostrará el boletín oficial de las notas de dicho alumno, mostrando la nota final (calculada como la media de las 3 notas).

Escribir una aplicación Web que pida al usuario el nombre y apellidos de 3 alumnos, además pedirá de cada uno de ellos la nota del primer examen, la nota del segundo examen y la nota del tercer examen.

La aplicación deberá mostrar para cada alumno su boletín de notas.

NOTA: el boletín debe mostrarse con un formato adecuado, usando tablas, cabeceras etc (tal como serían unas notas reales)

EJERCICIO 11

Crea un archivo PHP que incluya las siguientes funciones; una función para comprobar que el nombre de usuario no tiene caracteres especiales(quitando espacios) una función para sustituir espacios por _. Una función para comprobar el tamaño de un archivo, una función para obtener el tipo del archivo, una función para comprobar si una carpeta dada existe y, en caso contrario, crearla.





Una función la cual, si se le pasa un archivo, lo guarda en una carpeta de nombre, el proporcionado por el usuario, utilizando el nombre que ya tiene junto con la fecha de guardado en formato hhmmssDDMMYYYY; en caso de proporcionar dos archivos, se comprobarán cuales son sus tipos y se mostrará cuál de los dos es de mayor tamaño.

Crea otro archivo PHP que incluya las funciones antes mencionadas junto con un formulario para pedir los datos (el nombre y 2 archivos). El programa deberá comprobar que el nombre proporcionado es correcto, en caso contrario se recargará el formulario. Una vez comprobado el nombre, se realizará las funciones descritas en el párrafo anterior, en función de si el usuario ha proporcionado uno o dos archivos.

EJERCICIO 12

Crea un archivo PHP con un formulario en el que se le pida al usuario una cadena de texto, un carácter y un desplegable con las opciones “Izquierda”, “Derecha” y “Ambos”. La parte de PHP tendrá una función que reciba los elementos proporcionados por el formulario. Dicha función deberá eliminar todas las ocurrencias del carácter elegido, en la parte del string seleccionada y devolverlo.

La función admitirá que no se le pase el parámetro de “lado”, en cuyo caso seleccionará por defecto la izquierda.

OPCIONAL: Haz la función recursiva.





EJERCICIO 13

Escriba un programa que muestre el resultado de buscar una palabra (o varias) en Wikipedia.

Notas:

- La wikipedia ofrece un servicio web que permite solicitar el resultado de búsquedas en la propia Wikipedia.
- La solicitudes pueden tener la forma:
`https://es.wikipedia.org/w/api.php?action=opensearch&prop=extracts&format=json&formatversion=2&srwhat=title&search=XXX` donde **XXX** es la palabra o cadena a buscar.
- La respuesta es una cadena **JSON** con el siguiente formato:
 - palabra buscada
 - títulos de las páginas encontradas (hasta 10) ordenadas por relevancia
 - comienzo del contenido de las páginas
 - direcciones de las páginas
- Para obtener los datos de la API vamos a usar las funciones de CURL. Primero, hay que crear **una variable** donde guardamos los datos que nos devuelve **`curl_init()`**. Después habrá que invocar a **`curl_setopt()`** varias veces para establecer las opciones de CURL. La función admite 3 parámetros: 1º la variable donde hemos guardado **`curl_init()`**, 2º la opción a establecer (las que vamos a utilizar nosotros son **`CURLOPT_URL`**, **`CURLOPT_RETURNTRANSFER`** y **`CURLOPT_USERAGENT`**) y 3º el valor que le vamos a dar a la opción.
- En concreto los valores que le vamos a dar a las opciones son los siguientes: **`CURLOPT_URL`** -> La URL de la que vamos a obtener los





datos; `CURLOPT_RETURNTRANSFER` -> TRUE; `CURLOPT_USERAGENT`

-> La URL de nuestra página (puede ser localhost).

- Una vez hemos establecido las opciones, invocamos **`curl_exec()`** y guardamos el resultado en una variable.





- Este es, por ejemplo, el resultado (recortado) de [buscar la palabra perro](#):

```
[ "perro",  
  ["Perro", "Perro lobo", "Perrito caliente", ... ],  
  ["", "Un perro lobo ... ", "\"perro caliente\"  
redirige aquí; ... ", ... ],  
  ["https://es.wikipedia.org/wiki/Perro", "https://es.wik  
ipedia.org/wiki/Perro_lobo", ... ]]
```
- Por último, cerramos CURL utilizando **curl_close()**.
- El programa mostrará la respuesta de Wikipedia en forma de lista con enlaces a las páginas obtenidas.

NOTA EXTRA: Apóyate de la función del ejercicio anterior para limpiar el texto del JSON.

