



CURSO DE PROGRAMACIÓN WEB

EJERCICIOS PHP

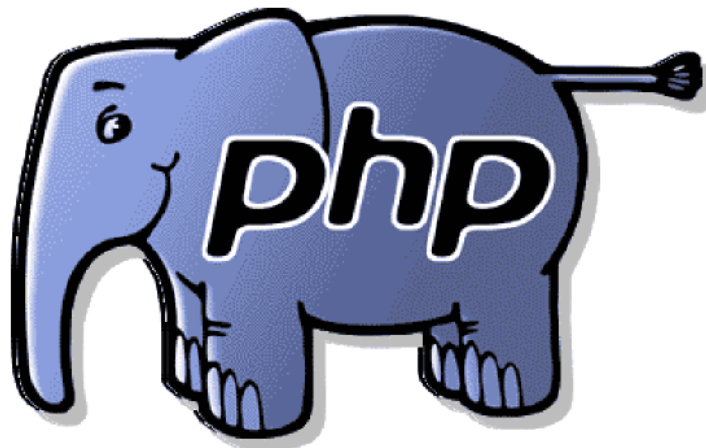


Tabla de contenido

Ejercicio 1	3
Ejercicio 2	4
Ejercicio 3	4
Ejercicio 4	5
Ejercicio 5	5
Ejercicio 6	6
Ejercicio 7	7
Ejercicio 8	8
Ejercicio 9	9
Ejercicio 10	10
Ejercicio 11	11
Ejercicio 12	11

Para ver el funcionamiento de estos ejemplos puedes visitar la web

<http://www.carpehosting.com/ejemplosPHP/>

Ejercicio 1

- 1) Abre un nuevo documento PHP y crea las etiquetas de inicio y fin de código PHP.
- 2) Define una constante con el valor “El valor que contiene es: ” y de nombre “principio”, y otra que se llame “esp” y contenga el valor “
”.
- 3) Declara e inicializa las siguientes variables:

Nombre	Valor
num1	1
num2	2
num10	10
nom1	“Cipsa”
nom2	“Barcelona”
txt1	“23”
txt2	“ en ”

- 4) Crea una variable \$multiplicacion que como valor contenga el multiplicar las 3 variables numéricas entre sí y muéstrala en el navegador.
- 5) Crea una variable \$textos, que concatene todas las variables de texto existentes por orden de aparición y muéstrala en el navegador.
- 6) Utilizando estas variables y las constantes la salida en el navegador deberá ser la siguiente:

VALORES DE LAS VARIABLES

El valor que contiene es: 1
El valor que contiene es: 2
El valor que contiene es: 10
El valor que contiene es: Cipsa
El valor que contiene es: 2
El valor que contiene es: 23
El valor que contiene es: en

OPERACIONES CON LAS VARIABLES

La suma de las variables numéricas es: 13
Cipsa se encuentra en Barcelona
La multiplicación de las variables numéricas entre si es: 20 Los
texto concatenados son: CipsaBarcelona23en

Nota: cada vez que se visualice un número en pantalla deberás utilizar las variables definidas, en el caso de los textos podrás utilizar más textos, aparte de los existentes en las variables.

Todos los ejercicios resueltos los encontrarás en la carpeta Archivos dossier PHP/
Ejercicios resueltos

Ejercicio 2

En este ejercicio generarás un calendario de un mes de 31 días, teniendo en cuenta que el día 1 es domingo, y que todos los domingos tienen que estar marcados en rojo. La manera de mostrar este calendario en la página web será utilizando una tabla.

- 1) Abre el documento Ejercicio2Incompleto.php con el editor de código.
- 2) Declara las variables \$uno, \$fin y \$semana con los valores 1, 31 y 1 respectivamente.
- 3) Delante de la llave coloca una instrucción while que se ejecute siempre que \$ sea menor o igual a \$fin.
- 4) Delante de la llave coloca una sentencia for que declare la variable \$contador y contenga el valor 1, que se ejecute siempre que \$contador menor o igual a semana * 7 y que incremente \$contador.
- 5) Delante de la llave escribe una sentencia if que se ejecute siempre que \$uno sea menor o igual que \$fin.
- 6) Delante de la llave escribe otra sentencia if que se ejecute siempre y cuando \$contador sea igual a 1.

Si todo ha funcionado correctamente serás capaz de ver el calendario en el navegador.

Todos los ejercicios resueltos los encontrarás en la carpeta Archivos dossier PHP/
Ejercicios resueltos

Ejercicio 3

- 1) Abre un nuevo documento php y elimina todo el contenido que genera el programa, resultando un archivo vacío, coloca las etiquetas de inicio y fin de PHP, y guárdalo con el nombre include.php en la carpeta de ejecución.
- 2) Escribe una función include para el archivo cabecera.html que encontrarás tú carpeta.
- 3) Has un “echo” de la etiqueta title de html, que contenga el título que quieras.
- 4) Incluye el archivo estilo.html.
- 5) Incluye el archivo cuerpo.html.
- 6) Si visualizas el archivo php en el navegador, observarás una página web con estilos incluidos, ahora puedes mirar los archivos que el include ha incrustado y verás que está dividido en secciones para su mejor mantenimiento.

Todos los ejercicios resueltos los encontrarás en la carpeta Archivos dossier PHP/ Ejercicios resueltos

Ejercicio 4

En este ejercicio utilizarás dos arrays, uno de usuarios y otro de contraseñas, y deberás validar por medio de estructuras de control, si el usuario y la contraseña existen y también si el usuario y su contraseña se corresponden. Para ello y de momento el usuario y la contraseña que quieres averiguar si existe lo debes almacenar en variables, lo cual luego podrás realizar en otros ejercicios más avanzados por medio de formularios HTML.

- 1) Declara tres variables una \$usuario, otra \$password, asígnale los valores que quieras y la tercera se llamará \$existe y tendrá el valor false.
- 2) Declara dos arrays uno de nombre \$usuariosRegistrados y otro \$passRegistrados y asígnale 5 valores a cada uno de ellos. Recuerda que el usuario que coloques en el índice 0 se corresponderá con la contraseña que este situada en el índice 0 de las contraseñas.
- 3) Utiliza una sentencia for para recorrer el array \$usuariosRegistrados y anida en su interior un if que compare el elemento que actualmente analiza el for con el \$usuario. Si este if es verdadero que guarde el índice del elemento en una variable llamada \$indice.
- 4) Dentro del anterior if declara otro if que compare \$password con \$indice, si esta comparación es verdadera cambia el valor de la variable \$existe a true.
- 5) Ahora viene el último paso, después del cierre de la sentencia for y todos sus anidamientos deberás generar otra estructura if, que compare el valor de existe con true, y si esta comparación es verdadera, debes hacer un echo de bienvenida al usuario, si es falsa informar que el usuario no existe.

Todos los ejercicios resueltos los encontrarás en la carpeta Archivos dossier PHP/ Ejercicios resueltos

Ejercicio 5

En este ejercicio deberás realizar cambios sobre un texto, utilizando las funciones que existen en PHP.

- 1) Crea un nuevo documento php y las etiquetas de inicio y fin del código.

2) Crea una variable llamada texto que contenga la frase:

La palabra estudiante es un sustantivo masculino que se refiere al educando o alumno dentro del ámbito académico, que estudia como su ocupación principal. 3)

Muestra en el navegador el texto con las siguientes variaciones:

- a. En mayúsculas
- b. En minúsculas
- c. Cambia todas las letras 'a' por 'á'
- d. La cantidad de veces que aparece la letra 'a'
- e. La posición en la cual comienza la palabra 'refiere'
- f. La frase con cada palabra iniciada en mayúscula
- g. Cambia la palabra 'masculino', por 'femenino'

Si todo ha funcionado correctamente cuando veas la página en el navegador la salida será la siguiente:

LA PALABRA ESTUDIANTE ES UN SUSTANTIVO MASCULINO QUE SE REFIERE AL EDUCANDO O ALUMNO DENTRO DEL ÁMBITO ACADÉMICO, QUE ESTUDIA COMO SU OCUPACIÓN PRINCIPAL.

la palabra estudiante es un sustantivo masculino que se refiere al educando o alumno dentro del ámbito académico, que estudia como su ocupación principal.

Lá pálabrá estudiánte es un sustantivo másculino que se refiere ál educándo o álumno dentro del ámbito ácadémico, que estudiá como su ocupación principál.

La letra 'a' aparece 15 veces en la frase

La palabra 'refiere' comienza en la posición 56

La Palabra Estudiante Es Un Sustantivo Masculino Que Se Refiere Al Educando O Alumno Dentro Del Ámbito Académico, Que Estudia Como Su Ocupación Principal.

La palabra estudiante es un sustantivo femenino que se refiere al educando o alumno dentro del ámbito académico, que estudia como su ocupación principal.

Todos los ejercicios resueltos los encontrarás en la carpeta Archivos dossier PHP/ Ejercicios resueltos

Ejercicio 6

En este ejercicio crearás funciones que te ayuden a gestionar un listado (matriz) de números, algunas de las funciones que crearás ya existen en el paquete php. Para realizar el ejercicio sigue los siguientes pasos:

- 1) Crea una matriz llamada num y asígnale 10 números.
- 2) Crea una función llamada contar que requiera un parámetro llamado arr, dentro de dicha función sigue estos pasos:
 - a. Declara una variable llamada contador y asígnale el valor 0
 - b. Crea una sentencia foreach que recorra los elementos contenidos en el parámetro arr.

- c. Dentro de la estructura foreach incrementa en 1 el valor de contador
- d. Al final de la función devuelve el valor de la variable contador.

Con esta función contarás cuantos elementos existen en la matriz pasada como parámetro.

- 3) Crea una función llamada suma que reciba un parámetro llamado arr y que se encargue de sumar todos los valores de la matriz, para ello realiza los siguientes pasos:
 - a. Crea una variable llamada x e inicialízala con el valor 0.
 - b. Recorre los elementos contenidos en la matriz con una sentencia foreach y ve incrementando el valor de "x" sumando cada número de la matriz.
 - c. Al final de la función y fuera de la sentencia foreach devuelve el valor de "x".
- 4) Crea una función llamada menor que requiera un parámetro llamado arr, que servirá para seleccionar el número más bajo de la matriz. Para ello realiza lo siguiente:
 - a. Declara una variable llamada x y asígnale el valor del primer elemento de la matriz.
 - b. Crea una sentencia foreach para recorrer los elementos de la matriz.
 - c. Dentro del foreach coloca una sentencia if que compare el valor actual de x con el elemento actual de la matriz, si el valor actual es menor que el anterior asigna el este valor a la variable "x".
 - d. Al final de la función devuelve el valor de "x".
- 5) Crea una función llamada mayor, que busque el número más grande contenido en la matriz.
- 6) Utilizando dichas funciones muestra en el navegador los siguientes valores:
 - a. El menor número
 - b. El mayor número
 - c. La suma de todos los valores
 - d. El promedio de los valores
 - e. Y utilizando la función date de php muestra la fecha con el formato dd/mm/aaaa y la hora con el formato hh:mm:ss.

Ejercicio 7

En este ejercicio crearás una estructura para guardar datos personales de personas(nombre, apellido, y fecha de nacimiento) en objetos, y luego mostrarás estos datos en una tabla HTML.

Para ello sigue los siguientes pasos:

- 1) Crea una interfaz llamada datos, que contenga la declaración de dos funciones

- llamadas “mostrarNombre” y “mostrarApellido”.
- 2) Crea una clase llamada agenda que implemente la interfaz datos.
 - 3) Dentro de dicha clase:
 - a. Declara dos variables protegidas llamadas “nombre y apellido”.
 - b. Crea la función __construct que requiera dos parámetros llamados “unNombre y unApellido” y de estos valores a las variables protegidas de la clase.
 - c. En la implementación de la función mostrarNombre devuelve por medio de la sentencia return el valor de la variable nombre del objeto actual.
 - d. Y en la implementación de la función mostrarApellido devuelve por medio de la función return el valor de la variable apellido del objeto actual.
 - 4) Crea otra clase llamada nuevaAgenda que extienda la clase agenda.
 - 5) Dentro de dicha clase:
 - a. Crea una variable llamada “fechaNacimiento” que sea privada.
 - b. Crea la función __construct que requiera tres parámetros llamados “unNombre, unApellido y unNacimiento”. Dentro de esta función ejecuta la función __construct de la clase base y envíale los valores que necesita, y asigna el valor del parámetro “unNacimiento” a la variable privada de la clase.
 - c. Crea la función “muestraNacimiento” que devuelva por medio de la sentencia return el valor de la variable de la clase llamada “fechaNacimiento”.
 - d. Crea otra función llamada “mostrarEdad” que calcule mediante la función date y la fecha de nacimiento del objeto la edad de la persona, basándose solamente en el año; y devuelva dicho valor mediante la sentencia return.
 - 6) Crea una matriz de cinco objetos de la clase “nuevaAgenda” enviándole los datos de cinco personas inventadas.
 - 7) Mediante una sentencia foreach muestra los datos de dichas personas en el navegador mediante una tabla, lógicamente utilizando los objetos creados anteriormente.

Ejercicio 8

En este ejercicio enviarás los datos de un formulario a un archivo php, que mostrará dichos archivos en el navegador y mediante un enlace te permitirá volver al formulario, pero manteniendo los datos introducidos.

Deberás crear dos archivos php, uno llamado Ejercicio8.php y otro llamado Ejercicio8datos.php. En el primero de los dos archivos deberás crear un formulario que permita al usuario introducir los siguientes datos: Nombre, primer apellido, segundo apellido, contraseña y una casilla del tipo checkbox que verifique si el usuario quiere recibir más información, todos estos datos se enviarán por el método post.

En el archivo Ejercicio8datos.php deberás recibir los datos y mostrarlos con el siguiente formato:

El nombre del usuario es: xxxx

El primer apellido del usuario es: xxxx

El segundo apellido del usuario es: xxxx

La contraseña del usuario es: xxx

El usuario desea recibir información

[Volver al formulario completado con los datos](#)

Recuerda que el último elemento es un enlace que debe permitir al usuario volver a la página anterior, y debe enviar los datos.

Ahora deberás retocar el archivo “Ejercicio8.php”, para que reciba los datos enviados por la otra página, pero debes tener en cuenta que estos datos no siempre estarán incluidos, ya que cuando el usuario entre a la página si no rellenado el formulario anteriormente, estos campo deben estar vacíos.

Ejercicio 9

En este ejercicio utilizarás una sesión y una cookie para asegurar el acceso a una página, de alta importancia. Para ello debes crear tres archivos PHP llamados: “Ejercicio9.php”, “Ejercicio9Comprueba.php” y “paginaAsegurada.php”.

En el archivo “Ejercicio9.php”, debes crear un formulario HTML con dos campos llamados “nombre y contraseña”, uno del tipo text y el otro de tipo password, el formulario enviará los datos al archivo “Ejercicio9Comprueba.php” por medio del método POST.

El archivo “Ejercicio9Comprueba.php” deberás seguir varios pasos:

- 1) Crea dos variables para guardar los datos que te envíe el formulario y guarda los valores en ellas.
- 2) Ahora deberás crear una sentencia if que compare los valores de las variables con los que tú consideres como usuario y contraseña, es decir crea un usuario y contraseña (más adelante estos datos se podrán guardar en bases de datos).
 - a. En el caso que la comprobación sea verdadera, deberás inicializar una sesión, crear una variable de sesión llamada “miVariable” que contenga el valor “Segura” y crear una cookie llamada “miCookie” que guarde el valor “El valor cifrado”.
 - b. En el caso que la comprobación sea falsa deberás crear una cookie llamada “miCookie”, sin valor. (Este paso se realiza para borrar cualquier cookie creada anteriormente).
- 3) Y por último después de realizar la comprobación enviarás al navegador a la página “paginaAsegurada.php”, por medio de una función header de PHP.

Falta por realizar el último archivo, que realizará los siguientes pasos:

- 1) Creará una sesión, o la continuará si ya existe.
- 2) Comprobará por medio de una sentencia if si los datos guardados en la variable de sesión como en la cookie coinciden con los “buenos”:
 - a. Si es verdadero, te dará la bienvenida a la página
 - b. Si es falso te informará que no puedes ver la página y destruirá la sesión (por seguridad).

Si todo ha funcionado correctamente al entrar en la primera página, el usuario se encontrará con un formulario, si los datos que introduce son correctos, le dejará entrar a la página, pero si los datos son incorrectos, le bloqueará la página.

Ejercicio 10

En este ejercicio deberás contralrar la subida de archivos a tu sitio web, y también la cantidad de visitas que has recibido. Para realizar este ejercicio deberás crear dos archivos llamados “Ejercicio10.php y Ejercicio10Receptor.php”. En el primer archivo solamente deberá existir un formulario que envíe los datos al segundo archivo, por medio de un input del tipo file, recuerda que al enviar archivos al servidor deberás colocar un enctype en la etiqueta form.

En el segundo archivo deberás seguir los siguientes pasos:

- 1) Comprobar la existencia del archivo llamado “contador.txt”, que en principio no existirá en el servidor y será el encargado de registrar el número de visitas recibidas en esta página:
 - a. Si existe deberás abrir el archivo, leer su contenido y guardarlo en una variable llamada datos, incrementar en una unidad su contenido y cerrar el archivo, después abrirás nuevamente el archivo pero en modo escritura, escribirás el contenido de la variable datos y cerrarás el archivo. A continuación crearás un echo que muestre el valor de la variable datos, para mostrar el contador en la página. Y por último ejecutarás una función llamada archivo (que crearás después), y le enviarás como parámetro la variable datos.
 - b. Si el archivo no existe, deberás crearlo y escribir en su interior el número uno, ya que eres el primer visitante, e informar al visitante que es el número uno. A continuación ejecutarás la función llamada archivo y le enviarás como parámetro el número 1.
- 2) Ahora deberás crear la función archivo que recibirá un parámetro, esta función deberá:
 - a. Crear un directorio con la fecha del día hoy en el formato DD_MM_AAAA, para que guarde los archivos en carpetas con la fecha de subida.
 - b. Mover el archivo de temporal a dicha carpeta
 - c. Renombrar el archivo, que debe llamarse “archivo1.jpg”, siendo el número 1 el número de visita actual.
 - d. Por último mostrarás en el navegador la foto subida en un tamaño de 200. 3)Fuera del código PHP crearás un enlace que te permita volver a la página

principal.

Si todo ha funcionado correctamente, al visitar la página podrás subir archivos que se guardarán en carpetas, y con nombres que has elegido tú. Y también se contarán las visitas que se efectúen de tu página.

Ejercicio 11

En este ejercicio guardarás datos en una base, y los mostrarás en una tabla, permitiendo agregar más datos por medio de un formulario HTML. Para ello, sigue los siguientes pasos:

- 1) Crea un nuevo archivo PHP, y dentro del body del documento abre las etiquetas php necesarias.
- 2) Conecta con el servidor MySQL, por medio de los datos de conexión que se te han entregado.
- 3) Crea dos funciones llamadas “crearBBDD y crearTabla”; en la primera función crearás una nueva base de datos con el nombre de tu usuario Cipsa , y en la segunda crearás una tabla llamada “propia”, con las siguientes características:
 - a. Se necesita guardar 3 datos, nombres, apellidos y contraseñas, y por supuesto deberán estar ligados a un identificador auto incrementable.
 - b. La clave primaria debe ser el identificador
 - c. Y debe ser una tabla del tipo ‘innoDB’
- 4) Ahora deberás crear el bloque principal de código de la página, empezando por comprobar con un if si es posible seleccionar la base de datos:
 - a. Si no es posible, deberás realizar las llamadas a las funciones creadas en el paso anterior y recargar la página utilizando un header.
 - b. Si es posible deberás realizar otra comprobación, saber si hay datos en el POST:
 - i. Si hay datos, deberás crear un nuevo registro en la tabla, mostrar un mensaje de aviso “Los datos han sido añadidos a la base de datos”, y recargar la página utilizando la siguiente función “header(“Refresh:5; URL=Ejercicio11.php”);” .
 - ii. Si no hay datos en el POST, significa que el usuario debe rellenar el formulario, por lo tanto lo mostrarás.
 - c. Después de comprobar lo anterior deberás crear una sentencia de consulta, y mostrar los datos en una tabla HTML

Si todo ha funcionado correctamente, serás capaz de agregar datos a la base de datos y de visualizarlos por medio de una tabla.

Ejercicio 12

En este ejercicio deberás crear y leer un archivo XML, que puede ser una alternativa rápida y eficaz de guardar datos que no necesiten la seguridad de una base de datos.

Para ello, deberás crear 2 archivos llamados “Ejercicio12CreadorXML.php y Ejercicio12.php”.

En el primero de los archivos, deberás crear el archivo XML utilizando el DOM, el archivo XML debe tener la siguiente estructura:

```
<noticias>
    <noticia>
        <titulo> Titulo de la noticia </titulo>
        <textoNoticia> Un texto sobre una noticia </textoNoticia>
    </noticia>
    ...
</noticias>
```

Y guardarlo bajo el nombre “archivo.xml”.

En el segundo archivo deberás mostrar la información recogida del archivo XML, utilizando la tecnología SimpleXML y mostrar cada título dentro de una etiqueta h1 y cada texto en una etiqueta p.

Ahora deberás ejecutar el primer archivo, para crear el XML y luego ejecutar el segundo para mostrar su información.