

Relación 1. Introducción a los formularios

Sitio: [Centros - Granada](#)
Curso: Desarrollo web en entorno servidor
Libro: Relación 1. Introducción a los formularios

Imprimido por: Aguilera Aguilera, Javier
Día: miércoles, 22 de diciembre de 2021, 12:28

Tabla de contenidos

1. Ejercicio 1
2. Ejercicio 2
3. Ejercicio 3
4. Ejercicio 4
5. Ejercicio 5
6. Ejercicio 6
7. Ejercicio 7
8. Validación 1
9. Validación 2
10. Calculadora

1. Ejercicio 1

Escribe un formulario básico de login que pida el usuario y la contraseña.

El script sólo nos mostrará el usuario y la contraseña enviados por el método POST

2. Ejercicio 2

Vamos a modificar el formulario de login.

El nuevo formulario se encarga de comprobar los datos introducidos y, según sean correctos o no, da acceso al sistema al usuario o muestra un mensaje de error. Un fichero con un script php se encarga de procesar un formulario de login.

Como aún no usamos bases de datos simularemos un formulario de login donde hay que comprobar que el usuario y la contraseña son correctos. Si el usuario es "usuario" y la clave es "1234", se redirige a la página de bienvenida.

En caso contrario, lo hace a una página de error.

Para la redirección usa la función `header()`, que sirve para escribir en la cabecera de la respuesta HTTP.

Hay que enviar las cabeceras antes de empezar con el cuerpo de la respuesta. Esto implica que hay que utilizar la función `header()` antes de que se empiece a escribir la salida. Si se intenta llamar a `header()` después de haber realizado un `echo`, se producirá un error.

3. Ejercicio 3

Modifica el ejercicio anterior para que el formulario HTML y el bloque PHP que lo procesa se integren en un solo fichero.

Ahora hay que distinguir entre dos casos: cuando se accede al formulario para rellenarlo y cuando se envía para procesarlo.

Cuando se accede a la página usando el método GET, es decir, introduciendo la dirección en el navegador, al seguir un vínculo como resultado de una redirección con un header(Location:), se muestra el formulario. En cambio, si se accede mediante POST quiere decir que el cliente está enviando el formulario.

Se puede diferenciar entre los dos métodos de acceso consultando `$_SERVER["REQUEST_METHOD"]`.

Cuando un formulario llama al mismo fichero se recomienda usar:

```
action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>
```

en lugar del nombre del fichero.

4. Ejercicio 4

NOTA: Antes de comenzar el ejercicio investiga sobre `$_REQUEST`. En esta práctica el envío de datos debe hacerse por POST y la recuperación con **REQUEST**.

Se pide un formulario que solicite la base y la altura de un triángulo y calcule su área.

Solución en un único formulario (NO HAY VALIDACIONES)

5. Ejercicio 5

Se pide un formulario que:

- solicite los siguientes datos: nombre, teléfono y email .
- al pulsar el botón *enviar* muestre la siguiente página.

"¡Buenos días, **nombre**!

Te voy a enviar spam a **correo** y te llamaré de madrugada a **telefono**.

Enviado desde un iPhone"

6. Ejercicio 6

- Crea un array con los nombres de 4 alumnos y sus respectivas notas.

Marta: 7,8 Luis: 5 Lorena: 6,9 ...

- Muestra las notas de una forma ordenada.

Alumno Nota

Marta 7,8

Luis 5

- Da la posibilidad de añadir nuevos alumnos mediante un formulario
- Muestra la media en la parte inferior.

SOLUCIÓN para añadir nuevos alumnos. ¡¡¡OJO!!! si se añade un alumno que ya existe se sobrescribe el que ya teníamos. Debéis de tenerlo en cuenta y modificar el código (array_key_exists()).

Una solución más elegante sería con sesiones que veremos en los próximos días.

Otra modificación aconsejable es construir funciones para tareas como crear el array alumnos, añadir elementos al array, mostrar el array, etc. De esta forma el código será más limpio y legible. Dichas funciones deberían estar en un archivo independiente.

7. Ejercicio 7

Escribe un formulario que:

- Pide el año de nacimiento.

A partir de esa información:

- Calcula la edad.
- Si es mayor de edad, dile que puede pasar dentro.
- Si es menor, invítale a irse a casa a dormir.
- Si tiene más de 85 años, dile que es demasiado mayor para entrar en este local.
- Recuerda pedir además el día y el mes de nacimiento para saber si ha cumplido el año actual.

8. Validación 1

Crea un formulario para solicitar el nombre, el teléfono y la dirección de correo de tus amigos.

- La página solicita los datos y los valida (por ejemplo, que el nombre sólo tenga letras, que el número sean 9 números, que la dirección de correo contenga la @, un punto y letras inglesas, etc).
- Si los datos no son válidos, se solicitan de nuevo indicando los datos no válidos.
- Si los datos son válidos, los datos se muestran
- Además de asegurarnos de que los datos recibidos no están vacíos y de aplicar `filter_var()` donde corresponda se pide para los input tipo text se construya una función que se encargue de limpiar los caracteres especiales de estos datos. Dicha función devolverá el texto pero antes deberá aplicarle lo siguiente:
 - eliminar los espacios antes y después de los datos (`trim()`)
 - traducir caracteres especiales en entidades HTML (`htmlspecialchars()`)
 - eliminar las barras (`stripslashes()`)
- ¿Crees que podemos usar esta función para validar los datos tipo email antes de usar `filter_var()`? ¿ Y para el teléfono?

Nota: Constará de sólo una página (todo en uno) y la validación sólo se hará en el servidor.

Ejemplo de los datos a solicitar:

FORMULARIO Y RESULTADO EN UN ÚNICO ARCHIVO (FORMULARIO)

Escriba los datos siguientes:

Nombre:

Teléfono:

Correo:

Posible presentación de datos no válidos

VALIDACIÓN DE FORMULARIO (RESULTADO INVÁLIDO)

Por favor, corrija los datos incorrectos:

Nombre:

Teléfono: El teléfono no es correcto

Correo: El correo no es correcto

[Volver al principio](#)

En https://tryphp.w3schools.com/showphp.php?filename=demo_form_validation_complete

puedes consultar un ejemplo que te ayudará a crear tu propio código para este ejercicio.

9. Validación 2

Una empresa tiene un formulario para pedir los datos a sus futuros empleados.

En dicho formulario solicita:

- nombre
- apellidos
- email
- teléfono
- Empleo actual (puede tomar uno de los siguientes valores: sin empleo, media jornada, tiempo completo). Será tipo radio
- Lenguajes que domina (podrá tomar más de uno de los siguientes valores: Python, PHP, JavaScript, Java, C++). Será de tipo checkbox.
- URL donde comprobar alguno de tus últimos trabajos realizados.

Se pide la validación sólo en el lado del servidor.

Ten en cuenta que algunos campos del formulario como los checkbox son un array.

Ejemplo que puede servir como ayuda para realizar el ejercicio:

10. Calculadora

Hacer una calculadora en PHP . La interfaz de usuario será un formulario con dos input y cuatro botones: sumar, restar, dividir y multiplicar