

Profesor: Javier Puche

Primeros pasos con PHP

P1-UD2

Desarrollo Web en Entorno Servidor

APLICACIÓN ADIVINAR NÚMERO

Objetivo

El objetivo era adivinar un número fijado previamente por el ordenador, dando retroalimentación sobre si es mayor o menor en cada intento (envío del formulario) y mostrando al adivinarlo el número de intentos consumidos. El número aleatorio debe calcularse una sola vez, en el primer acceso (acceso sin ningún parámetro enviado)

Pasos Realizados

1. Creación formulario HTML

Primero se hace un diseño inicial con la creación de una página HTML básica con un formulario, que definimos con form, con un campo de entrada numérico para que el usuario ingrese el número a adivinar y un botón de envío, tipo submit.

En el form indicamos el atributo action con `$_SERVER['PHP_SELF']`, lo que significa que cuando se envía el formulario, los datos se enviarán de vuelta al mismo script PHP en el que se encuentra actualmente.

```
html>
<body>
  <h1>Adivina un número del 1 al 100</h1>
  <!-- indicamos en action el propio php (PHP_SELF) para que el formulario se dirija
  <form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="post">
    <label for="nombre">Numero: </label>
    <!-- Se crea un campo de entrada de tipo número y obligatorio que permite
    al usuario ingresar un número del 1 al 100.
    Y un botón de tipo submit para enviar el formulario -->
    <input type="number" name="num" min="1" max="100" required >
    <input type="submit" name="adivinar" value="Adivinar" />
```

2. Número Aleatorio, Contador de intentos y campos ocultos.

Después de los inputs del formulario, sin cerrar este, abrimos etiqueta PHP.

Antes de generar un número aleatorio, se verifica si existe la variable con **isset** `$_POST["numMaquina"]`. Si no existe (lo que significa que es la primera vez que se carga la página o el juego se reinicia), se genera un número aleatorio entre 1 y 100 y se almacena en la variable `$numMaquina`.

Y con el contador de intentos hacemos lo mismo: Se verifica si existe la variable con **isset** `$_POST["contador"]`. Si no existe (en la primera carga de la página o al iniciar el juego), se inicializa `$contador` en 0.

```
<?php
// Comprueba si ya hay un número aleatorio para adivinar, si no,
//se genera uno del 1 al 100.
if (isset($_POST["numMaquina"])) {
    $numMaquina = $_POST["numMaquina"];
} else {
    $numMaquina = rand(1, 100);
}

// Comprueba si ya hay contador, si no, se inicializa en 0.
if (isset($_POST["contador"])) {
    $contador = $_POST["contador"];
} else {
    $contador = 0;
}
```

Muy importante, estas dos variables las tenemos que almacenar dentro del formulario HTML pero a la vez que estén ocultas para el usuario, por lo que para ello creamos input de tipo hidden, para que almacenen estos valores. Permiten mantener el mismo número aleatorio y el registro de intentos en el contador.

Estos campos los ponemos al final del documento, después de cerrar la parte de php y antes de cerrar el formulario y html (luego explico la razón).

```
3     }
4     ?>
5     <!-- Se incluyen campos ocultos para almacenar el número aleatorio
6     y el contador de intentos. -->
7     <input type="hidden" name="numMaquina" value="<?php echo $numMaquina; ?>">
8     <input type="hidden" name="contador" value="<?php echo $contador; ?>">
9     </form>
10    </body>
11    <html>
```

4. Procesamiento de Datos del Formulario

Cuando el usuario envía el formulario haciendo clic en el botón "Adivinar", se verifica si existe la variable con `isset $_REQUEST["adivinar"]` (lo que indica que se ha enviado el formulario).

Si el formulario se ha enviado, se obtiene el número ingresado por el usuario a través de `$_REQUEST["num"]` y se lo asignamos a la variable `$num`.

Teniendo ya estos datos, vamos a la lógica de la aplicación, que es ir comparando el número de usuario `$num` con el número aleatorio `$numMaquina` para determinar si es igual, mayor o menor. Se hace una retroalimentación

En los tres casos, el contador se aumenta en 1 para ir sumando intentos y lo imprimimos por pantalla.

Si el número es igual, si acierta, se saca por pantalla dicho número y un mensaje informativo, y como se acaba el juego, inicializamos variables contador, a 0, y numMaquina, generando nuevo número aleatorio, de esta manera puede seguir jugando desde la misma sesión.



Si el número es mayor o menor, se dice por pantalla con mensaje de que lo vuelva intentar.

```
// Verifica si se ha enviado el formulario con el botón "Adivinar".
if ( isset($_REQUEST["adivinar"])){
    $num = $_REQUEST["num"];

    // Se comprueba si el número ingresado en casilla es igual al número aleatorio
    if ($num == $numMaquina){
        $contador++; // aumentamos el contador en 1 para sumar un intento.
        echo "<br> Tu número " . $num . " es correcto<br>HAS GANADO!<br>";
        echo "<br> Intentos: " . $contador . "<br>";
        echo "<br> Introduce otro número para volver a jugar.";
        // y como se ha acabado el juego reiniciamos contador de intentos y
        // generamos nuevo núm aleatorio
        $contador = 0;
        $numMaquina = rand(1, 100);

    } elseif ($num > $numMaquina){
        $contador++;
        echo "<br> Tu número " . $num . " es mayor.<br>Vuelve a intentarlo.";
        echo "<br> Intentos: " . $contador . "<br>";
    } else {
        $contador++;
        echo "<br> Tu número " . $num . " es menor.<br>Vuelve a intentarlo.";
        echo "<br> Intentos: " . $contador . "<br>";
    }
}
```

Capturas de ejecución:

- Al abrir sesión:

Numero:

- Introduzco número, es mayor y se muestra contador en 1.

Numero:
Tu número 50 es mayor.
Vuelve a intentarlo.
Intentos: 1

- Introduzco otro número, es menor y se muestra contador como 2 intentos:



← → ↻ localhost/php/adivinarNumero.php

Importar marcadores... FRANCISCO DE GOYA Archivos - Cloud Educ...

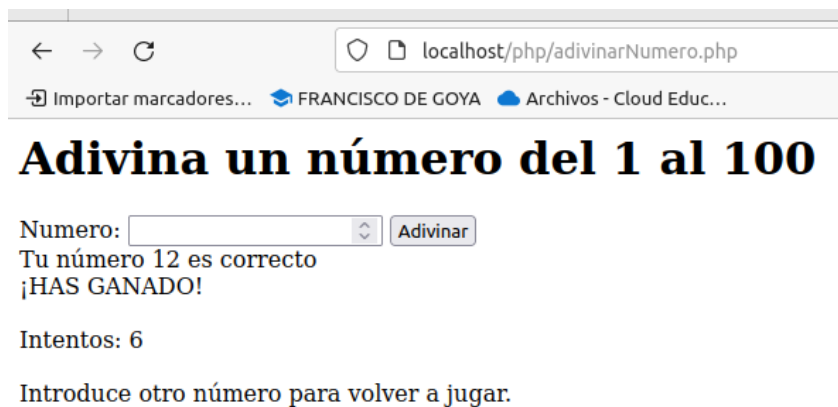
Adivina un número del 1 al 100

Numero: Adivinar

Tu número 10 es menor.
Vuelve a intentarlo.

Intentos: 2

- Y después de varios intentos, introduzco número y es correcto, se muestra contador en 6 intentos y se muestra mensaje de introducir otro número para volver a empezar, ya que en el código hemos puesto que cada vez que acierte el número, se inicialice el contador y se cree un nuevo número aleatorio.



← → ↻ localhost/php/adivinarNumero.php

Importar marcadores... FRANCISCO DE GOYA Archivos - Cloud Educ...

Adivina un número del 1 al 100

Numero: Adivinar

Tu número 12 es correcto
¡HAS GANADO!

Intentos: 6

Introduce otro número para volver a jugar.

Dificultades.

Al principio puse los campos ocultos del formulario junto con el resto de los campos del formulario, pero claro salía error ya que detectaba que los valores que tenía que almacenar no se actualizaban pues como estaba el php después, cada vez que se enviaba el formulario. Por lo que al colocarlos al final, aseguramos que se actualicen con los valores correctos antes de que el formulario se vuelva a mostrar después de darle a enviar.



BLOG FAKE

Objetivo

Un formulario con tres controles: uno de texto (etiqueta: Nuevo mensaje) un botón de enviar y un área de texto de solo lectura. Cada pulsación a Enviar provocará que el área de texto acumule el texto enviado del campo Nuevo mensaje al final de los mensajes anteriores.

Pasos realizados

1 Creación formulario HTML

Se utiliza un formulario HTML con el atributo action indicando que se envíe a la misma página con (`$_SERVER['PHP_SELF']`) cuando se presiona el botón "Enviar".

Hacemos un campo de entrada de texto `<input type="text">` con el nombre "texto" para que el usuario ingrese un mensaje. Ponemos el atributo required para ponerlo como campo obligatorio y no se deje vacío.

E incluimos un botón de envío `<input type="submit">` con el nombre "enviar".

Y por último un campo textarea de solo lectura, con readonly, para que no se pueda escribir ahí y solo sirva para ir asignando los textos a través del código php al enviar el formulario. MUY IMPORTANTE no cerrar la etiqueta textarea, pues dentro se va hacer todo el código php, para que lo que se ejecute/muestre con el código, quede dentro del campo.

2 Procesamiento de Datos del Formulario

La lógica del código es ir encadenando mensajes, es decir, introducir un mensaje al que llamaremos mensajeNuevo en el campo textarea llamado area, pero si ya hay un mensaje en el area, pasando a llamarse mensajeAnterior, pero si ya ha un mensaje en el textarea, entonces se encadenan ambos mensajes, el mensajeNuevo se encadena con el mensajeAnterior, y ambos pasan a ser mensajeAnterior, y ya no existe mensajeNuevo hasta que se vuelva a introducir mensajeNuevo.

Ahora iniciamos el código PHP, dentro del campo textarea, primero comprobamos si se ha enviado el formulario y si se ha ingresado un nuevo mensaje utilizando `if (isset($_REQUEST["enviar"]) && isset($_REQUEST["texto"]))`., ambas condiciones en el mismo if.

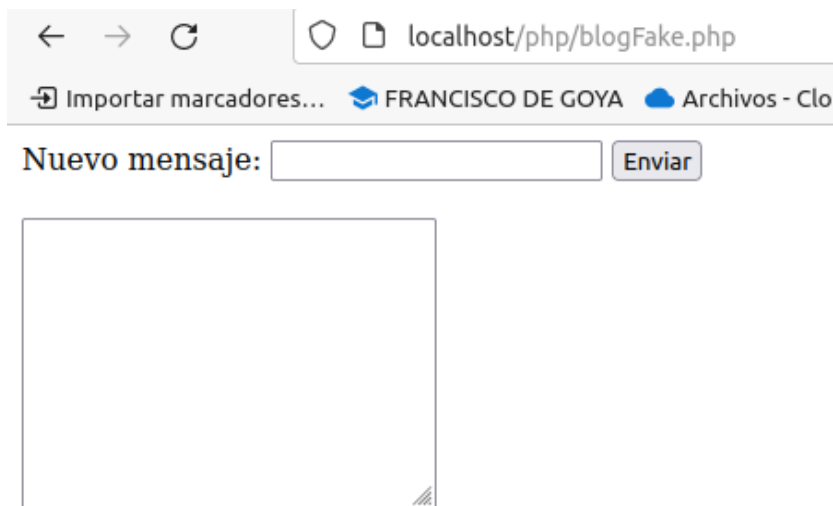
Y creamos variable mensajeAnterior de tipo string a la que asignamos un string vacío.

Si es así entonces con un if comprobamos si existe mensajeAnterior en el area con `$_REQUEST["area"]` y si es así hacemos lo siguiente:

- A la variable `mensajeAnterior` le asignamos el valor del campo `area` que obtenemos con `$_REQUEST["area"]`.
- Creamos variable `mensajeNuevo` y le asignamos el valor ingresado por el usuario con `$_REQUEST["texto"]`.
- Y por último, con un `echo` concatenamos `mensajeAnterior` con `mensajeNuevo` y con un salto de línea, todo se concatena con un punto. Y de esta manera los mostrarlos en el área de texto.

Capturas de ejecución

-Formulario al iniciar sesión:

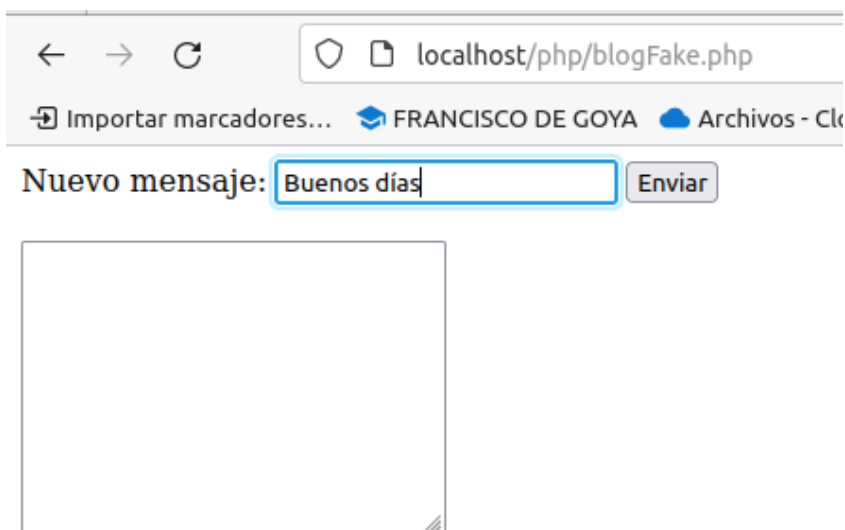


← → ↻ localhost/php/blogFake.php

Importar marcadores... FRANCISCO DE GOYA Archivos - Clo

Nuevo mensaje:

-Introducimos texto y damos a enviar:



← → ↻ localhost/php/blogFake.php

Importar marcadores... FRANCISCO DE GOYA Archivos - Clo

Nuevo mensaje:



-Vemos que el texto ingresado se ha colocado en el textarea:

A screenshot of a web browser window. The address bar shows 'localhost/php/blogFake.php'. Below the address bar, there are links for 'Importar marcadores...', 'FRANCISCO DE GOYA', and 'Archivos'. The main content area has a label 'Nuevo mensaje:' followed by an empty text input field and an 'Enviar' button. Below this is a large text area containing the text 'Buenos días'.

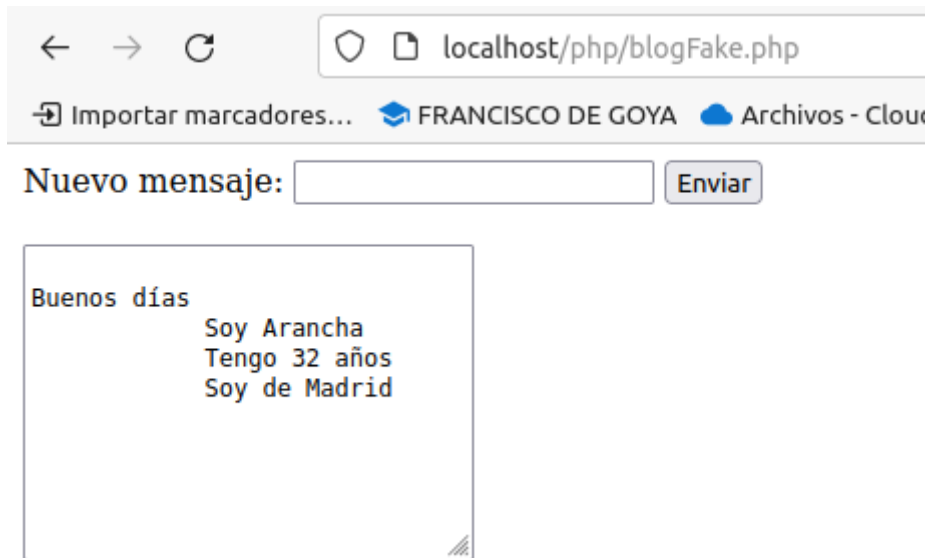
-Introducimos un segundo texto y damos a enviar:

A screenshot of a web browser window. The address bar shows 'localhost/php/blogFake.php'. Below the address bar, there are links for 'Importar marcadores...', 'FRANCISCO DE GOYA', and 'Archivos - Cloud'. The main content area has a label 'Nuevo mensaje:' followed by a text input field containing 'Soy Arancha' and an 'Enviar' button. Below this is a large text area containing the text 'Buenos días'.

-Vemos que se ha puesto en el textarea, debaja del texto anterior (aunque con un formato extraño):

A screenshot of a web browser window. The address bar shows 'localhost/php/blogFake.php'. Below the address bar, there are links for 'Importar marcadores...', 'FRANCISCO DE GOYA', and 'Archivos - Cloud'. The main content area has a label 'Nuevo mensaje:' followed by a text input field and an 'Enviar' button. Below this is a large text area containing the text 'Buenos días' and 'Soy Arancha' on separate lines.

-Y probamos introduciendo más textos, y vemos que se han colocado todos una debajo de otro:



Dificultades

Este me ha costado un poco más pero la principal razón era que no me acordaba como rellenar un text area y por lo tanto estaba intentando introducir el resultado del código php como valúe en el textarea.

Para concatenar los mensajes, he mirado en el pdf del aula y enseguida ví que se concatenaban los strings con puntos.



CONVERSION EUROS A PESETAS

Objetivo

El objetivo de este código PHP es crear una página web que permita a los usuarios convertir una cantidad en euros a pesetas y mostrar el resultado. Además, se proporciona conversiones adicionales a dólares, libras, yenes y francos suizos.

Pasos Realizados:

1 Creación HTML e Inicio PHP:

Se crea una estructura HTML básica y dentro incluimos el código PHP, pues como en este ejercicio al enviar el formulario, el propio deja de aparecer, debe ir dentro del código y al final, pues su funcionalidad es que primero se verifica si el formulario ha sido enviado mediante el uso de `if (isset($_POST["convertir"]))`. Si el formulario ha sido enviado, se realizarán los cálculos y se mostrará el resultado. Si no, se mostrará el formulario para ingresar la cantidad en euros.

Este formulario lo creamos con un input de tipo number para que el usuario introduzca la cantidad, y botón de tipo submit para Convertir.

2 Procesamiento de Datos del Formulario:

Dentro de la condición `if (isset($_POST["convertir"]))`, se obtiene la cantidad ingresada por el usuario utilizando `$_POST["euros"]` y se la asignamos a variable euros.

Se comprueba si se ingresó una cantidad con `if (! $euros == "")`. Si no se ingresó nada, se muestra un mensaje de error.

3 Cálculos y Conversiones:

Si se ingresó una cantidad, se realizan los siguientes cálculos:

La cantidad en pesetas se calcula multiplicando la cantidad en euros por el factor de conversión 166.386, y asignamos esta cantidad a la variable pesetas.

Además de la conversión a pesetas, se calculan conversiones adicionales a dólares, libras, yenes y francos.

4 Mostrar el Resultado:

Dentro del mismo if, se muestra el resultado de la conversión a pesetas con un echo junto con un mensaje que indica también su equivalencia en el resto de las monedas, mostrando también la propia cantidad en euros ingresada.

5 Manejo de Errores:

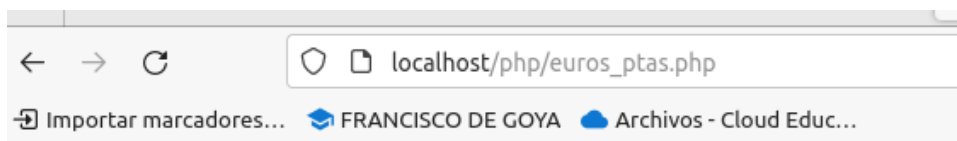
Si el usuario no ingresó una cantidad, se muestra un mensaje de error indicando que debe introducir una cantidad.

6 Link de Vuelta:

Generamos un enlace [Volver] que permite al usuario regresar al formulario para realizar una nueva conversión si lo desea, por lo que como link, ponemos el propio nombre del archivo.

Capturas de ejecución:

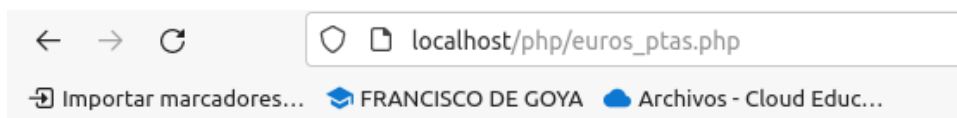
-Al abrir la sesión:



Conversor de Euros a Pesetas

Cantidad en euros:

-Introduzco una cantidad y damos al botón convertir:



Conversor de Euros a Pesetas

Cantidad en euros:



-Aparece el resultado, la cantidad introducida y la conversión a pesetas, además de las conversiones al resto de monedas.



Dificultades:

Una vez hecho el ejercicio anterior, al hacer este ha sido más sencillo. Pero una dificultad surgida fue que estaba realizando el formulario fuera del código php por lo que siempre estaba presente y no se borraba cuando se enviaba el formulario y tampoco estaba planteando bien la lógica inicial de “si se envía el formulario”.

CONVERSOR MONEDAS

Objetivo

Este ejercicio es muy parecido al anterior, también se trata de convertir una cantidad ingresada por el usuario en otras monedas, con la diferencia de que en este se usa un campo de select/option para seleccionar el tipo de moneda, y de esta manera al enviar el formulario, solo se muestra la equivalencia a la moneda seleccionada.

Como es muy parecido al ejercicio anterior no voy a explicar todos los pasos, solo lo que cambia.

En este caso, una vez que entramos en el if (! \$euros == ""), cogemos con \$_POST["monedas"] el contenido del campo option correspondiente, donde “monedas” corresponde al name del select de los options.

Y dentro, con cuatro ifs, uno por cada tipo de moneda, comparamos la variable moneda con un string del mismo nombre que el contenido de los campos options (dolares, libras, yenes y



francos) y vamos haciendo los cálculos, asignado a la variable del tipo de moneda la multiplicación de los euros ingresados por el usuario por su tipo de cambio.

El resto es todo igual al ejercicio anterior.

Capturas de ejecución:

-Formulario al iniciar sesión:

Conversor de Monedas

-Introducimos cantidad y seleccionamos libras esterlinas, y damos al botón convertir:

Conversor de Monedas

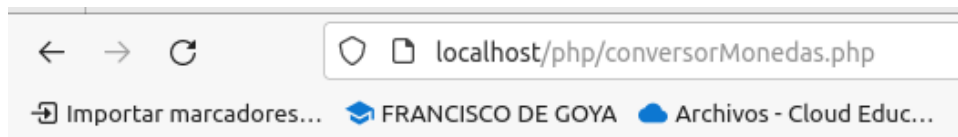
-Vemos que aparece el resultado, la cantidad ingresada y su equivalencia en libras, la moneda seleccionada, y vemos también que el formulario ya no está y aparece un link con la palabra Volver.

Conversor de Monedas

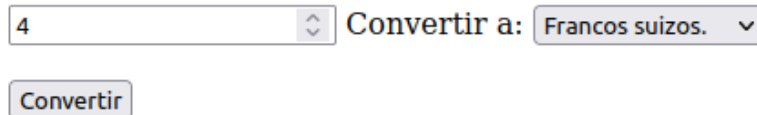
5 euro(s) equivale(n) a: 4.635 libras esterlinas.

[\[Volver\]](#)

-Damos a Volver y volvemos al formulario. Probamos con otra cantidad y esta vez seleccionamos la moneda de Francos suizos:



Conversor de Monedas



Conversor de Monedas

4 euro(s) equivale(n) a: 6.06 francos suizos.

[\[Volver\]](#)

Dificultades

Después de hacer el ejercicio anterior que era parecido, la única dificultad fue como coger/captar el valor de cada campo option, ya que es diferente a un campo input. Pero ha sido fácil de resolver al buscarlo en internet.

BUSCAR CANCIONES

Solo envió la parte de formulario html, pues no he tenido tiempo de terminarlo. Pero quería enviarlo igualmente.



CONCLUSIONES

Esta práctica ha sido muy útil para refrescar los conocimientos de creación de formularios HTML y sobretodo comprender mucho mejor su funcionalidad y cómo funcionan “por dentro”, es decir, como se procesan los datos del usuario. También he aprendido bastante PHP, sobre todo la sintaxis que es diferente al resto de lenguajes, y me ha hecho practicar la lógica y la soltura.