



APRENDERAPROGRAMAR.COM

PROCESAR DATOS DE UN
FORMULARIO HTML EN
PHP CON \$_POST.
EJERCICIOS RESUELTOS Y
EJEMPLOS. (CU00834B)

Sección: Cursos

Categoría: Tutorial básico del programador web: PHP desde cero

Fecha revisión: 2029

Resumen: Entrega nº34 del Tutorial básico "PHP desde cero".

Autor: Enrique González Gutiérrez

RECUPERANDO DATOS DEL FORMULARIO. VARIABLES \$_POST

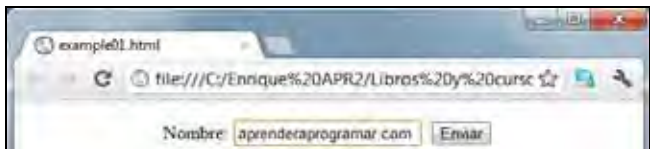
Cuando un usuario pulsa el botón enviar de un formulario, la información que contenían sus campos es enviada a una dirección URL desde donde tendremos que recuperarla para tratarla de alguna manera. Por ejemplo, si realiza una compra, tendremos que recuperar los datos para completar el proceso de pago. La información del formulario "viaja" almacenada en variables que podremos recuperar y utilizar mediante PHP. Una de las formas de recuperación consiste en usar \$_POST.



RECUPERAR VARIABLES CON POST

POST nos permite recuperar datos enviados desde formularios con el método POST. Escribe este código y guárdalo con un nombre de archivo como ejemplo1.html. A continuación, sube el fichero al servidor, introduce un nombre en el campo y visualiza el resultado.

```
<form name="formulario" method="post" action="ejemploPost.php">
  Nombre: <input type="text" name="nombre" value="">
  <input type="submit" />
</form>
```

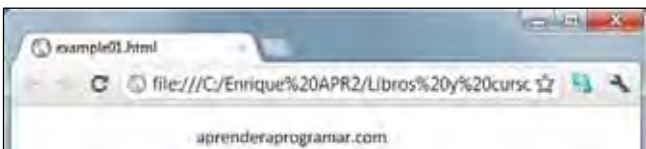


Antes de pulsar sobre el botón enviar, debemos crear el fichero ejemploPost.php que es el indicado en el formulario para el envío de los datos.

Escribe este código y guárdalo con un nombre de archivo como ejemploPost.php. A continuación, sube el fichero al servidor en la misma carpeta donde subiste el fichero html.

```
<?php
  $nombre = $_POST['nombre'];
  echo $nombre;
?>
```

Una vez completemos el paso anterior, pulsaremos sobre el botón enviar del ejemplo1.html y veremos una imagen similar a la siguiente.



Vamos a explicar el proceso que ha tenido lugar. El primer archivo es un documento HTML. Para ser más correctos, deberíamos haberlo escrito de esta manera:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo aprenderaprogramar.com</title>
    <meta charset="utf-8">
  </head>
  <body>
    <form name="formulario" method="post" action="ejemploPost.php">
      Nombre: <input type="text" name="nombre" value="">
      <input type="submit" />
    </form>
  </body>
</html>
```

Sin embargo, comprobamos que los navegadores tratan de interpretar y mostrar el resultado de un código HTML (o PHP) incluso cuando la sintaxis o la definición del documento no es del todo correcta. Esto debemos conocerlo, sin embargo recomendamos que siempre se trate de ser lo más correctos posibles a la hora de escribir código web.

Vemos que hemos definido un formulario en cuya cabecera hemos puesto que el método de envío de los datos va a ser el método post y que la URL de destino va a ser ejemploPost.php. A su vez, el formulario tiene un campo cuyo atributo name es "nombre". Ese atributo define el nombre de la variable que vamos a poder recuperar en la URL de destino.

En el archivo php hemos incluido la línea \$nombre = \$_POST['nombre']; que significa que creamos una variable php denominada \$nombre donde almacenamos la información del campo 'nombre' a través del método post proveniente del formulario. Si tuviéramos otros campos que hubiéramos definido como apellidos, teléfono, edad, el tratamiento sería similar. Por ejemplo:

```
$apellidos = $_POST['apellidos'];
$telefono = $_POST['celular'];
$edadPersona = $_POST['edad'];
```

Fijate en que una cosa es la variable en la que almacenamos la información recuperada, y otra cosa es el nombre del campo del formulario de donde proviene. Por ejemplo en \$apellidos = \$_POST['apellidos']; coinciden el nombre de la variable que utilizamos con el nombre del campo del formulario. Sin embargo, en \$telefono = \$_POST['celular']; no coinciden. En este caso, el campo que proviene del formulario se llama 'celular' mientras que la información que venga en ese campo la almacenamos en una variable a la que hemos llamado \$telefono. Finalmente, en \$edadPersona = \$_POST['edad']; estamos almacenando en una variable a la que hemos llamado \$edadPersona la información proveniente de un campo del formulario denominado 'edad'.

Con frecuencia los nombres de las variables y de los campos del formulario se hacen coincidir, pero en otras ocasiones no. Esto queda a elección del programador.

Como podemos observar, recuperar datos enviados por un formulario con el método POST es bastante simple usando PHP.

EJERCICIO RESUELTO N° 1

Diseñar un formulario web que pida la altura y el diámetro de un cilindro en metros. Una vez el usuario introduzca los datos y pulse el botón calcular, deberá calcularse el volumen del cilindro y mostrarse el resultado en el navegador.

SOLUCIÓN

La solución esquematizada en pseudocódigo es la siguiente:

1. Inicio

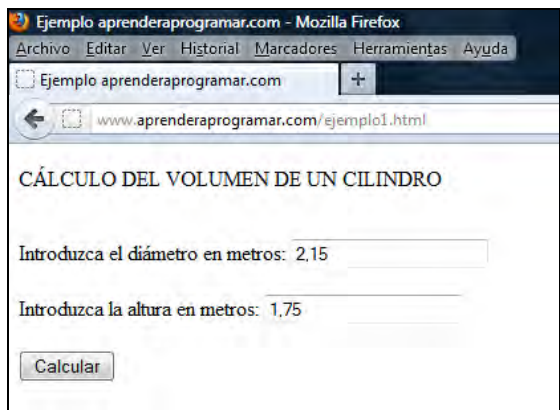
2. Mostrar "Introduzca el diámetro, en metros" : Pedir D
3. Mostrar "Introduzca la altura, en metros" : Pedir H
4. $R = D/2$: $Pi = 3,141593$
5. $V = Pi * (R ^ 2) * H$
6. Mostrar "El volumen del cilindro es de", V, "metros cúbicos"

7. Fin

Esquematizar la solución en pseudocódigo es una buena idea antes de realizar la programación, pues nos permite definir conceptualmente cómo va a ser nuestro código antes de escribirlo. Es sobre todo adecuado para personas que se están iniciando en la programación.

En primer lugar, crearemos el archivo html con el formulario.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Ejemplo aprenderaprogramar.com</title>
    <meta charset="utf-8">
  </head>
  <body>
    <form name="formularioDatos" method="post" action="ejemploPost1.php">
      <p> CÁLCULO DEL VOLUMEN DE UN CILINDRO </p>
      <br/>
      Introduzca el diámetro en metros: <input type="text" name="diam" value="">
      <br/> <br/>
      Introduzca la altura en metros: <input type="text" name="altu" value="">
      <br/> <br/>
      <input value="Calcular" type="submit" />
    </form>
  </body>
</html>
```



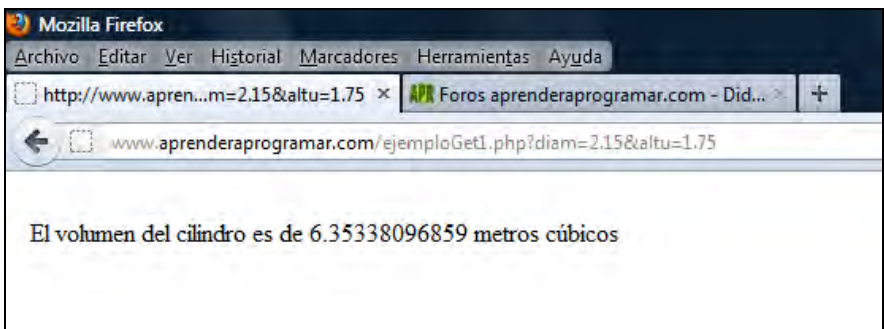
Por otro lado, crearemos el archivo php con el tratamiento de datos:

```
<?php //Ejemplo aprenderaprogramar.com
$diametro = $_POST['diam'];
$altura = $_POST['altu'];
$radio = $diametro/2;
$Pi = 3.141593;
$volumen = $Pi*$radio*$radio*$altura;

echo "<br/> &nbsp; El volumen del cilindro es de". $volumen. "metros cúbicos";
?>
```

Fijate que hemos escrito la potencia del radio como \$radio * \$radio. En otros lenguajes existe el operador de exponenciación, pero en php esta operación se tiene que realizar recurriendo a una función matemática. Esta función la estudiaremos en otro momento.

Finalmente obtenemos un resultado. Haz pruebas introduciendo como valores de diámetro y altura 2,15 y 1,75 en vez de 2.15 y 1.75. Posiblemente no obtengas un resultado adecuado si usas las comas, ya que en PHP el separador de la parte decimal de un número es el punto.



EJERCICIO RESUELTO N° 2

Diseñar un desarrollo web simple con php que pida al usuario el precio de tres productos en tres establecimientos distintos denominados "Tienda 1", "Tienda 2" y "Tienda 3". Una vez se introduzca esta información se debe calcular y mostrar el precio medio del producto.

SOLUCIÓN

La solución esquematizada en pseudocódigo es la siguiente:

1. Inicio
 2. Mostrar "Introduzca el precio del producto en el establecimiento número 1, en euros" : Pedir Precio1
 3. Mostrar "Introduzca el precio del producto en el establecimiento número 2, en euros" : Pedir Precio2
 4. Mostrar "Introduzca el precio del producto en el establecimiento número 3, en euros" : Pedir Precio3
 5. $Media = (Precio1 + Precio2 + Precio3) / 3$
 6. Mostrar "El precio medio del producto es", Media, "euros"
7. Fin

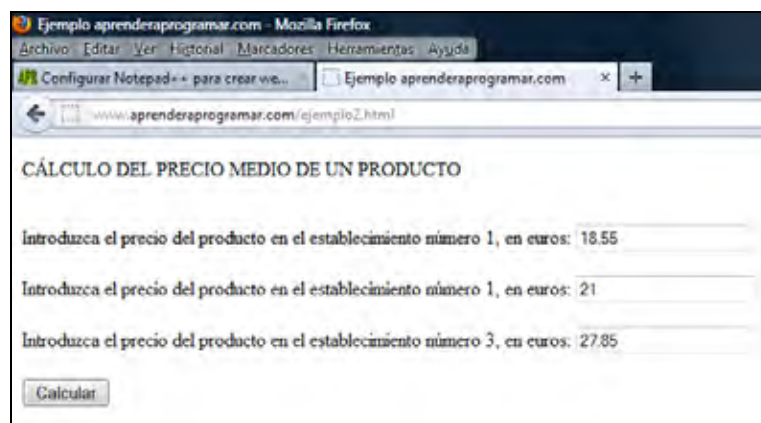
```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title>Ejemplo aprenderaprogramar.com</title>
    <meta charset="utf-8">
</head>
<body>
    <form name="formularioDatos" method="post" action="ejemploPost2.php">
    <p> CÁLCULO DEL PRECIO MEDIO DE UN PRODUCTO </p>
    <br/>
    Introduzca el precio del producto en el establecimiento número 1, en euros:
    <input type="text" name="precio1" value="">
    <br/> <br/>
    Introduzca el precio del producto en el establecimiento número 1, en euros:
    <input type="text" name="precio2" value="">
    <br/> <br/>
    Introduzca el precio del producto en el establecimiento número 3, en euros:
    <input type="text" name="precio3" value="">
    <br/> <br/>
    <input value="Calcular" type="submit" />
    </form>
</body>
</html>
    
```

Es posible que durante el desarrollo del curso te encuentres visualizaciones de este tipo, donde podrás comprobar que los acentos o tildes no se ven bien.

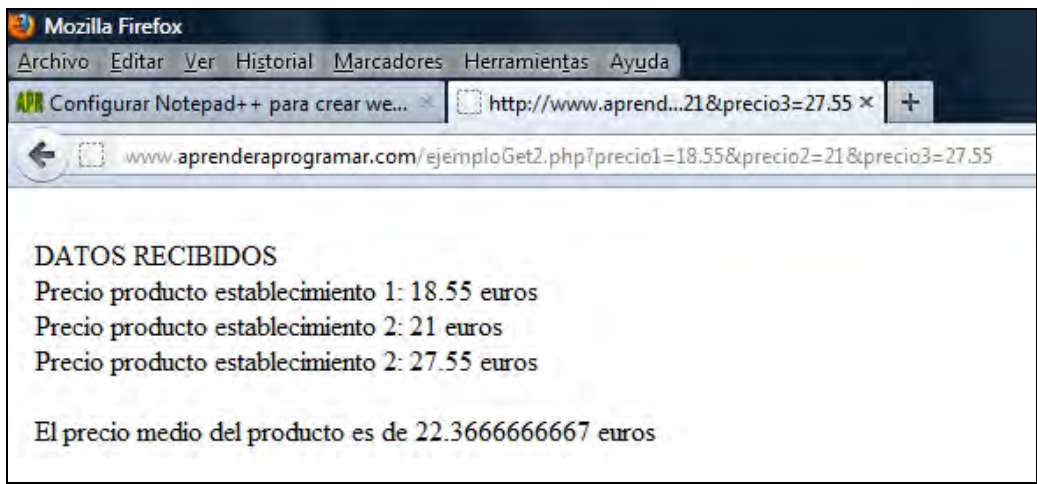


Recordarte que la solución a esto es, cuando estamos trabajando con Notepad++, elegir en el menú Formato la opción "Codificar en UTF-8 sin BOM". En caso de que por error el archivo esté en otro formato, elige la opción "Convertir en UTF-8 sin BOM" para dejar correctamente la codificación del archivo. También puede ser necesario introducir la etiqueta: `<meta charset="utf-8">` entre las etiquetas `<head> ... </head>` para lograr que la visualización sea correcta.



Por otro lado necesitamos el archivo php.

```
<?php //Ejemplo aprenderaprogramar.com
    $precio1 = $_POST['precio1'];
    $precio2 = $_POST['precio2'];
    $precio3 = $_POST['precio3'];
    $media = ($precio1+$precio2+$precio3)/3;
    echo "<br/> &nbsp; DATOS RECIBIDOS";
    echo "<br/> &nbsp; Precio producto establecimiento 1: ". $precio1. " euros";
    echo "<br/> &nbsp; Precio producto establecimiento 2: ". $precio2. " euros";
    echo "<br/> &nbsp; Precio producto establecimiento 3: ". $precio3. " euros <br/>";
    echo "<br/> &nbsp; El precio medio del producto es de ". $media. " euros";
?>
```



EJERCICIO PROPUESTO

Diseñar un desarrollo web simple con PHP que dé respuesta a la necesidad que se plantea a continuación.

Un operario de una fábrica recibe cada cierto tiempo un depósito cilíndrico de dimensiones variables, que debe llenar de aceite a través de una toma con cierto caudal disponible. Se desea crear una aplicación web que le indique cuánto tiempo transcurrirá hasta el llenado del depósito. El caudal disponible se considera estable para los tiempos que tardan los llenados de depósitos y lo facilita el propio operario, aportando el dato en litros por minuto.

ORIENTACIÓN PARA LA SOLUCIÓN

La solución esquematizada en pseudocódigo es la siguiente:

1. Inicio

2. Mostrar "Introduzca el caudal disponible en litros / minuto": Pedir Q
3. Mostrar "Introduzca el diámetro del depósito, en metros" : Pedir D
4. Mostrar "Introduzca la altura del depósito, en metros" : Pedir H
5. $\pi = 3,141593$
6. $R = D / 2$
7. $V = \pi * (R ^ 2) * H$: Vlitros = $V * 1000$
8. $t\text{minutos} = V\text{litros} / Q$
9. Mostrar "El tiempo que transcurrirá hasta el llenado del depósito es de", tminutos, "minutos"

10. Fin

Realiza el ejercicio y comprueba los resultados. Para que tengas una referencia, si el caudal disponible es de 125 litros por minuto y los valores de diámetro y altura del cilindro son 2.15 y 1.75, el resultado que debes obtener es que el depósito tarda en llenarse será de aproximadamente 50 minutos.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00835B

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=70&Itemid=193