

# Envío de datos de un formulario con PHP



- Cuando un usuario rellena un formulario en una página web los datos hay que enviarlos de alguna manera. Vamos a considerar las dos formas de envío de datos posibles: usando el método POST o usando el método GET.
- La diferencia entre los métodos get y post radica en la forma de enviar los datos a la página cuando se pulsa el botón “Enviar”. Mientras que el método GET envía los datos usando la URL, el método POST los envía de forma que no podemos verlos.

# Práctica 1



- Crea un archivo HTML que contenga un formulario que contenga un checkbox (¿Desea recibir factura?), dos input radio (Varón/Mujer), tres input tipo texto (Nombre, Dirección, Email) y un combobox para forma de envío (normal, por avión o urgente), junto con un botón de envío. Establece como url de envío la dirección recibe.php y método de envío get y comprueba que se recuperan los datos enviados correctamente a través de la url.

# \$ \_GET PHP: recuperar datos de formularios HTML



GET nos permite recuperar parámetros desde la URL o desde formularios enviados con el método GET.

# **\$ \_POST PHP. Procesar datos de un formulario HTML**



POST nos permite  
recuperar datos  
enviados desde  
formularios con el  
método POST.

# Request PHP. Diferencias, ventajas y problemas de GET, POST, REQUEST



MÉTODO	CONCEPTO	OBSERVACIONES
GET	GET lleva los datos de forma "visible" al cliente (navegador web). El medio de envío es la URL. Para recoger los datos que llegan en la url se usa \$_GET.	Los datos son visibles por la URL, por ejemplo: www.aprenderaprogramar.com/ action.php?nombre=pedro&apellidos1= gomez
POST	POST consiste en datos "ocultos" (porque el cliente no los ve) enviados por un formulario cuyo método de envío es post. Es ideal para formularios. Para recoger los datos que llegan por este método se usa \$_POST.	La ventaja de usar POST es que estos datos no son visibles al usuario de la web. En el caso de usar get, el propio usuario podría modificar la URL escribiendo diferentes parámetros a los reales en su navegador, dando lugar a que la información tratada no sea la prevista.
REQUEST	No es un método de envío propiamente dicho, es decir, no podemos poner en un formulario HTML que el method sea REQUEST. Con la variable \$_REQUEST recuperaremos los datos de los formularios enviados tanto por GET como por POST.	La ventaja principal de esta forma de recuperar los datos de un formulario es que no tenemos que saber con cual método fue enviado. La desventaja principal, como veremos en el siguiente ejemplo, es que no podremos diferenciar una variable enviada por GET o por POST.

# Práctica 2



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.

## 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

# Práctica 3



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.

## **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**



# Práctica 3



- 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 el ejercicio



## 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

# Prueba con estos datos



- 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.

# Base de Datos MySQL



- Crear una base de datos MySQL En nuestro servidor. Crear una tabla.

```
DROP DATABASE if exists BD_Colegio;
CREATE DATABASE BD_Colegio;
    use BD_Colegio;

CREATE TABLE alumno (
    al_id int NOT NULL AUTO_INCREMENT,
    al_nombres varchar(100),
    al_apellidos varchar(100),
    al_telefono varchar(50),
    al_email varchar(200),
    al_procedencia varchar(200)
);
```

# INSERCIÓN DE DATOS

```
INSERT INTO alumno (al_nombres, al_apellidos, al_telefono, al_email, al_procedencia)
values ("Jorge Luis", "Bocanegra Diaz", "999555111", "Trujillo"),
      ("Marco", "Perez Ulloa", "994554114", "La esperanza"),
      ("Cristina", "Vilcapoma Chavez", "997557117", 'La rinconada');
```

# Práctica 4



- Crea una tabla denominada <<ciudades>> que conste de los siguientes 6 campos:
  - id
  - ciudad
  - pais
  - habitantes
  - superficie
  - tieneMetro
- id INT con índice PRIMARY y auto-incrementable, ciudad VARCHAR con largo máximo 80, pais VARCHAR con largo máximo 50, habitantes BIGINT (entero largo), superficie DOUBLE (decimal largo), tieneMetro BOOLEAN (valor booleano, verdadero o falso).

- A continuación inserta los siguientes datos en la tabla.



id	ciudad	pais	habitantes	superficie	tieneMetro
1	México D.F.	México	555666	23434.34	1
2	Barcelona	España	444333	1111.11	0
3	Buenos Aires	Argentina	888111	333.33	1
4	Medellín	Colombia	999222	888.88	0
5	Lima	Perú	999111	222.22	0
6	Caracas	Venezuela	111222	111.11	1
7	Santiago	Chile	777666	222.22	1
8	Antigua	Guatemala	444222	877.33	0
9	Quito	Ecuador	333111	999.11	1
10	La Habana	Cuba	111222	333.11	0

# Funciones MySql con PHP



## FUNCIÓN mysqli\_connect (mysql\_connect)

```
$nombreConexion = mysqli_connect($hostname , $nombreUsuario , $contraseña);
```

Para servidores no actualizados usar mysql\_connect en lugar de mysqli\_connect

## FUNCIÓN mysqli\_close (mysql\_close)

```
mysqli_close($nombreConexión);
```

Para servidores no actualizados usar mysql\_close en lugar de mysqli\_close



# Funciones MySql con PHP



## FUNCIÓN `mysqli_select_db` (`mysqli_select_db`)

```
mysqli_select_db ($nombreConexión, $nombreBaseDatos);
```

Para servidores no actualizados usar `mysql_select_db ($nombreBaseDatos, $nombreConexión);`

## FUNCIÓN `mysqli_query` (`mysqli_query`)

```
$result = mysqli_query($nombreConexion, "Consulta aqui"); ó mysqli_query($nombreConexion, $query);
```

Para servidores no actualizados usar `$result = mysql_query("Consulta aqui", $nombreConexion);`

```
$tildes = $link->query("SET NAMES 'utf8'"); //Para que se muestren las tildes
```

# USO DE WHILE PARA EXTRAER TODOS LOS RESULTADOS DE UNA CONSULTA

```
while ($fila = mysqli_fetch_array($result)){  
    // Operaciones con los resultados que tenemos en $fila  
}
```

# Práctica 5



- Utilizando la tabla <<ciudades>> que hemos creado para el ejercicio anterior del curso.
- Crea un archivo php donde se realice una consulta a la base de datos para extraer todos los datos de la tabla y a continuación muestra por pantalla los valores de país y ciudad de cada fila.

# Práctica 6



- Utilizando la tabla <<ciudades>> que hemos utilizado para ejercicios anteriores del curso.
- Crea un archivo php donde se realice una consulta a la base de datos para extraer todos los datos de la tabla y a continuación calcula la suma de habitantes de todos los países y cuántas de estas ciudades tienen metro. Muestra el resultado por pantalla.
- Realiza una consulta a la base de datos para extraer todas las ciudades determinando cuántas ciudades en la base de datos empiezan por B.

# Inserción de datos



```
mysqli_query($link, "INSERT INTO nombreTabla VALUES (datoCol1, datoCol2, ... datoColN)");
```

- Para insertar nuevos datos en la base de datos se debe especificar la conexión y luego las palabras reservadas INSERT INTO (insertar dentro) luego el nombre de la tabla y por ultimo la pabra VALUES e indicar cada valor en el orden que tiene la base de datos.
- Recordar que si se desea solo insertar algunos datos se debe especificar el nombre de las columnas.

## Práctica 7

- Usar la tabla anterior e insertar una fila más. Luego visualizar desde el servidor en phpmyadmin y luego mostrar el nuevo listado por pantalla.

## Práctica final

- Realizar un CRUD (Create, Read, Update, Delete) usando la tabla <Ciudades>.