

# Submitir un formulario: método “post”

- Ayer vimos como eramos capaces de recuperar los datos introducidos en un formulario cuando el valor del atributo “method” era “**get**”.

```
<form action="nuevo.php" method="get">
```

- El atributo “method” puede tener otro valor: **post**.
- Veamos cuales son las diferencias entre uno y otro método, y cual de ellos elegir.

# Submitir un formulario: método “post”

- Método “**get**”:

- **Realiza una llamada a la URL** del atributo “action”, añadiendo los campos (atributo “name”) y valores del formulario en forma de nombre=valor&nombre=valor...
- Desventaja: la longitud máxima URL son 3.000 caracteres.
- **No se aconseja si la información a enviar es sensible**, ya que la URL quedaría almacenada en el historial del navegador.

## Método “**post**”:

- Los datos del formulario no aparecen en la URL.
- Se aconseja si **la información es sensible**, ya que los datos enviados no aparecen en el historial del navegador.

# Submitir un formulario: método “post”

- ¿Cómo se recuperan los datos en el archivo **nuevo.php** si usamos **method=”post”** en vez de **method=”get”**?
- Simplemento, usaremos la **variable “superglobal” \$\_POST**, el resto es igual.

```
$_POST['nombre'];  
$_POST['apellidos'];  
...
```

- Vamos a comprobarlo...

# Ejercicio

- Configurar el widget datepicker de jQueryUI para que el formato de la fecha sea:

**YYYY-mm-dd**

- Palabras clave:  
“jqueryui datepicker format date”

# Tipos de datos en MySQL: Texto

- **CHAR:**

- Se usa para almacenar **cadena de caracteres** de **longitud fija** (NIF, IBAN, etc.)
- Es **obligatorio** especificar la **longitud de la cadena** (máximo: 255 caracteres).
- Es un 50% más rápido que VARCHAR.

- **VARCHAR:**

- Se usa para almacenar **cadena de caracteres** de **longitud variable**.
- Es **obligatorio** especificar la **longitud de la cadena** (máximo 65.535 caracteres).

# Tipos de datos en MySQL: Números enteros (edad, etc.)

- **5 tipos:**

Type	Storage	Minimum Value	Maximum Value
	(Bytes)	(Signed/Unsigned)	(Signed/Unsigned)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

# Tipos de datos en MySQL: Números fraccionarios (precio, calificación, etc.)

- **DECIMAL:**

- Almacena **números fraccionarios con un formato fijo**. Se redondea para ajustar el número al formato.
- Se usa para almacenar **precios**. **Ejemplo: 12.34**

- **FLOAT:**

- Almacena **números fraccionarios aproximados**.
- Utiliza **4 bytes** para la precisión de la parte decimal.

- **DOUBLE:**

- Almacena **números fraccionarios aproximados**.
- Utiliza **8 bytes** para la precisión de la parte decimal.

# Tipos de datos en MySQL: Fechas y Horas

- **DATE:**

- Almacena fechas.
- Formato por defecto: **YYYY-mm-dd**
- Rango: 1000-01-01 al 9999-12-31

- **DATETIME:**

- Almacena fecha + hora.
- Formato por defecto: **YYYY-mm-dd H:i:s**
- Rango: 1000-01-01 00:00:00 a 9999-12-31 23:59:59.



# Tipos de datos en MySQL: Booleanos (si/no, activo/inactivo, etc.)

- **BOOLEAN:**

- Permite almacenar **datos de tipo booleano** (si/no, activo/inactivo, etc.).

# Ejercicio

- Creamos columnas en una tabla usando los tipos de datos vistos hasta ahora.

# El tipo de dato DECIMAL

- Si hemos intentado introducir un valor numérico fraccionario, en campo de tipo DECIMAL. Es probable que el valor no se haya almacenado como queríamos, es decir se haya almacenado solamente la **parte entera**.
- Si acudimos a la pestaña “Estructura” de la tabla , veremos que el tipo DECIMAL está establecido tal que así:

6 **precio**

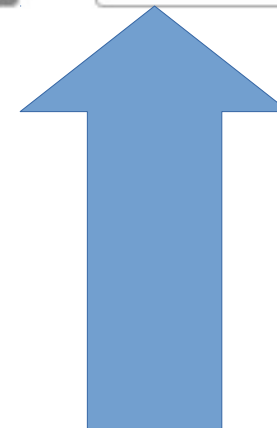
decimal(10,0)

# El tipo de dato DECIMAL

- La parte de la derecha de la imagen anterior es:  
decimal (10,0)
- El primer número indica el numero total de digitos (enteros y decimales), y el segundo el numero total de digito decimales.
- Por ejemplo para notas de exámenes del 1 al 10 con 2 decimales (ejem.: 7,86), deberíamos de escribir en la casilla “Longitud/valor” lo siguiente: (siguiente diapositiva).

# El tipo de dato DECIMAL

Name	Type ?	Length/Values ?
nota_media	DECIMAL	4,2



# El tipo de dato DECIMAL

- ¿Qué ocurre si introducimos un nota tal que esta?

7.4567

- Nuestro campo sólo puede almacenar esto:

X.XX

- Así que se redondeará las centesimas: al ser el **tercer decimal mayor o igual que 5**, al segundo decimal se le suma 1.

7.4567 → **7.46**

# Subir archivos desde formularios

- Pistas:

```
enctype="multipart/form-data">
```

```
var_dump($_FILES);  
move_uploaded_file($_FILES['foto']['tmp_name'], '/tmp/foto_alumno.jpg');
```

# Ejercicio

- ¿Qué palabras claves de busqueda usaríamos para saber el **tipo de dato** a usar para una hipotetica columna “**temperatura**”?



- Top
- Rgrep
- Apache directorios