

Práctica 2. PHP+MySQL

Profesor: José M. Luna

Email: jmluna@uco.es

Enunciado de prácticas

- Haz una aplicación web similar a la de la práctica anterior, con la misma estructura de presentación de la información y de navegación, pero en la que los datos de los empleados estén almacenados en una base de datos en lugar de estar en el código HTML.
- **Emplea PHP y MySQL**

Introducción

- En la anterior práctica, **se necesitaba un nuevo fichero HTML para cada empleado**, y había que modificar los listados cada vez que un empleado se añadía, modificaba o eliminaba
- Una web sólo en HTML es muy estática y difícil de mantener
- **¿Cómo incluir contenido dinámico?**
 - Usando PHP
 - El contenido de la página puede ser diferente cada vez que el navegador solicita acceso

PHP

- PHP incluye un servidor propio
- El servidor de PHP se arranca con la siguiente instrucción por consola:

```
php -S localhost:8000 -t directorio
```

- El servidor busca automáticamente un fichero *index.php* o *index.html* en el directorio indicado
- Acceder desde el navegador como:

```
localhost:8000
```

```
localhost:8000/index.php
```

PHP

- Dentro del fichero *.php*, el código en PHP debe ir entre los símbolos
`<?php ... ?>`

- En PHP, el programa *Hola mundo* sería:

```
<?php
    echo "Hola mundo!"
?>
```

- **Todo el código que NO se encuentre** dentro de las etiquetas `<?php ... ?>` se envía al cliente como **código HTML**

PHP

- PHP permite incluir tanto comentarios de una línea como de varias líneas

```
//Comentario de una línea
```

```
/*  
 * Comentario  
 * de varias  
 * líneas  
 */
```

- Cada sentencia ha de terminar con punto y coma “;”

```
$x += 10;
```

- El símbolo \$ se coloca delante de **todas** las variables (NO hay que indicar el tipo de la variable)

```
$contador = 1;  
$real = 1.5;  
$cadena = "Hola";  
$array = array("Uno", "Dos", "Tres");
```

José María Luna

PHP

- Los arrays se crean con el constructor `array()`; incluyendo dentro los valores separados por comas

```
$equipo = array('Jose', 'Pablo', 'Lucia', 'Mario', 'Paula');
```

- Para acceder a un elemento del array, se hace con corchetes.

```
echo $equipo[2]; //Muestra el nombre Lucia
```

- Usando comillas dobles, PHP evaluará `$var` como una variable, incluyendo su valor

```
echo "Esta semana $count personas han visto tu perfil";
```

- Se pueden concatenar cadenas con el operador punto `.`

```
echo "Tiene " . $msgs . " mensajes.";
```

- Se puede concatenar una cadena con otra con el operador `.=`

```
$cadena1 .= $cadena2
```

José María Luna

PHP

- Algunas veces necesitamos mostrar mucho texto en PHP, que haciéndolo usando varias sentencias `echo` sería muy tedioso.
- Para ello, utilizar el operador `<<<`, que preserva los saltos de línea y otros espacios

```
echo <<<_END
```

```
    Esta es otra manera de imprimir  
    varias líneas conservando los espacios.
```

```
    El último _END debe aparecer justo al principio de línea  
    y sin contener nada más después, ni si quiera espacios ni comentarios.
```

```
_END;
```


PHP

- Bucle **for**

```
for ($count = 1 ; $count <= 12 ; ++$count){  
    echo "$count por 12 es " . $count * 12 . "<br>";  
}
```

- Bucle **foreach ... as**

```
$comprar = array("Huevos", "Leche", "Azucar", "Pan");  
  
foreach($comprar as $item)  
{  
    echo "$item<br>";  
}
```

PHP

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Biblioteca - Autores</title>
</head>
<body>
<?php

/* Incluye las funciones de consulta para la libreria */
require_once('queryFunctionsLibrary.php');

/* Crea un nuevo objeto para llamar a las consultas */
$q = new LibraryQueries();

/* Comprueba que se ha realizado bien la conexion a la base de datos */
if(empty($q->dbc)){
    echo "<h3 align='center'>¡Error!: No se pudo establecer la conexión con la base de datos.</h3><br/>";
    die();
}

/* Muestra la lista de libros */
echo <<<_END
    <h3 align="center">Listado completo de autores</h3>
    <table align="center" class="list" style="margin: 0 auto;">
        <tr align="center" class="list" >
            <th class="list" >Autor</th>
            <th class="list" >Nº de libros</th>
        </tr>
    _END;
```

PHP

```
$authors = $q->getAuthors();
foreach($authors as $author){
    echo <<<_END
        <tr align="center" class="list" >
            <td class="list" >$author[author]</td>
            <td class="list" >$author[nBooks]</td>
        </tr>
    _END;
}

//Cerramos la tabla
echo "</table>";

?>
</body>
</html>
```

MySQL

- Para acceder a `mysql` desde los ordenadores de la Universidad de Córdoba hay que seguir las instrucciones del siguiente enlace:

<http://oraclepr.uco.es/abd/>

- Acceder a *My Base de Datos* en el menú izquierdo y crear la base de datos.
 - El nombre de la base de datos creada será el nombre de usuario
 - El usuario y contraseña de `mysql` son los mismos que de la UCO
 - Servidor: `oraclepr.uco.es`

MySQL

- Para acceder por consola a `mysql` de manera local se hace de la siguiente manera

```
mysql -u user -p
```

```
mysql -u root
```

- Para acceder por consola a `mysql` proporcionado por la UCO

```
mysql -h oraclepr.uco.es -u loginUCO -p
```

```
sudo apt-get mysql-server  
sudo apt-get mysql-client
```

MySQL

- Mostrar bases de datos

```
SHOW DATABASES;
```

- Crear base de datos

```
CREATE DATABASE empresa;
```

- Usar base de datos

```
USE empresa;
```

- Se crean usuarios con CREATE USER

```
CREATE USER 'username'@'hostname' IDENTIFIED BY 'password';
```

```
CREATE USER 'jose'@'localhost' IDENTIFIED BY 'josePWD';
```

Conectar con MySQL desde PHP

- Creamos una clase que conectará con la base de datos e incluirá las consultas a la misma
- Primero incluimos el usuario y contraseña de MySQL y el objeto para conectarse a la base de datos

```
class ConsultasEmpresa {  
    //Usuario  
    public $usuario = 'jose';  
  
    //Contraseña  
    public $pass = 'josePW';  
  
    //Objeto para conectar con la base de datos  
    public $dbc;
```

Conectar con MySQL desde PHP

- El constructor de la clase conecta con la base de datos y la tabla correspondiente, almacenando la conexión en el objeto *dbc*

```
/* Constructor. Llama a la función que conecta con la base de datos */
public function __construct(){
    //Conectar con la base de datos
    $this->dbc = $this->dbconnect();
}

/* Conectar con la base de datos y la tabla correspondiente*/
public function dbconnect() {
    $dbc = null;

    try {
        $dbc = new PDO('mysql:host=localhost; dbname=empresa', $this->usuario, $this->pass, array(PDO::ATTR_PERSISTENT => true));
    } catch (PDOException $e) {
        return null;
    }

    return $dbc;
}
```

OJO, si se usa mysql de la uco, la conexión debe ser:

```
$dbc = new PDO('mysql:host=oraclepr.uco.es; dbname=usuarioUCO', ...
```


Conectar con MySQL desde PHP

- Dentro de la misma clase que hemos creado, podemos incluir funciones que accedan a la base de datos

```
public function getEmpleados(){  
    $emps = array();  
  
    $sentence = $this->dbc->prepare("SELECT * FROM empleados");  
  
    if ($sentence->execute()) {  
        while ($row = $sentence->fetch()) {  
            $emps[] = $row;  
        }  
    }  
  
    return $emps;  
}
```

Conectar con MySQL desde PHP

- Ejemplo **biblioteca**

```
<?php

class LibraryQueries {

    //Username
    public $user = 'jose';

    //Password
    public $password = 'josePWD';

    //Database connection PDO object
    public $dbc;

    /* Constructor. It connects to the database */
    public function __construct(){
        //Connect to the database
        $this->dbc = $this->dbconnect();
    }
}
```

Conectar con MySQL desde PHP

```
/* Connect to the database and table library */
public function dbconnect() {
    $dbc = null;

    try {
        $dbc = new PDO('mysql:host=localhost; dbname=library', $this->user, $this->password,
array(PDO::ATTR_PERSISTENT => true));
    } catch (PDOException $e) {
        return null;
    }
    return $dbc;
}

/* Get an array with all books */
public function getAllBooks(){
    $books = array();
    $i = 0;
    $sentence = $this->dbc->prepare("SELECT * FROM books");
    if ($sentence->execute()) {
        while ($row = $sentence->fetch()) {
            $books[$i] = $row;

            if ($row['available'])
                $books[$i]['available'] = "Si";
            else
                $books[$i]['available'] = "No";
            $i++;
        }
    }
    return $books;
}
```

Conectar con MySQL desde PHP

```
/* Get an array with the different authors in the database and their number of books */
public function getAuthors(){
    $authors = array();
    $i = 0;
    $sentence = $this->dbc->prepare("SELECT author, count(title) as nBooks FROM books GROUP BY author");

    if ($sentence->execute()) {
        while ($row = $sentence->fetch()) {
            $authors[$i] = $row;
            $i++;
        }
    }
    return $authors;
}

/* Get an array with the different categories in the database and their number of books */
public function getCategories(){
    $categories = array();
    $i = 0;
    $sentence = $this->dbc->prepare("SELECT category, count(title) as nBooks FROM books GROUP BY category");

    if ($sentence->execute()) {
        while ($row = $sentence->fetch()) {
            $categories[$i] = $row;
            $i++;
        }
    }
    return $categories;
}
```

Conectar con MySQL desde PHP

```
/* Get an array with the books for a specific author */
public function getBooksForAuthor($author){
    $books = array();
    $i = 0;

    $sentence = $this->dbc->prepare("SELECT * FROM books WHERE author = ?");

    if ($sentence->execute(array($author))) {
        while ($row = $sentence->fetch()) {
            $books[$i] = $row;

            if ($row['available'])
                $books[$i]['available'] = "Si";
            else
                $books[$i]['available'] = "No";

            $i++;
        }
    }

    return $books;
}
```

.....
.....
.....