

Ejercicios de PHP

PHP

El código PHP se ejecuta en el servidor. Para que el código funcione correctamente vamos a usar XAMPP. Una vez hecho el ejercicio lo copiaremos en la carpeta htdocs de XAMPP. Para comprobar su funcionamiento abriremos el navegador web e introduciremos la dirección de nuestro servidor web (localhost en mi caso).

Para estos ejercicios hemos utilizado los apuntes que están en <http://www.php.net/manual/es/getting-started.php>.

1. Crea un documento PHP de nombre **ejemploPHP.php** que muestra el mensaje “Bienvenido a PHP” usando código PHP.
2. Crea un documento PHP de nombre **phpinfo.php** que muestra la información acerca del PHP que hay instalado en nuestro sistema.
3. Crea un documento PHP de nombre **navegador.php** que muestra la información acerca del navegador web que estoy usando.
4. Crea un documento PHP de nombre **comprobarnavegador.php** que saque un mensaje con el nombre del navegador web que estoy usando (Internet Explorer, Mozilla Firefox, Google Chrome, Otro).
5. Crea un documento HTML de nombre **FormsHolaPHP.html** que muestra un formulario con los datos de Nombre y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **hola.php** que coge los datos, y los saca por pantalla.
6. Crea un documento HTML de nombre **FormsHolaPHPcss.html** que muestra un formulario con los datos de Nombre y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **holacss.php** que coge los datos, y los saca por pantalla utilizando la hoja de estilos **estilos.css**.
7. Crea un documento HTML de nombre **FormsHolaPHPcssmenu.html** que muestra un formulario con los datos de Nombre y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **holacssmenu.php** que coge los datos, y los saca por pantalla utilizando el mismo formato que la página de inicio del proyecto de HTML **index.html**.
8. Crea un documento HTML de nombre **FormsAlumnosPHP.html** que muestra un formulario con los datos de Nombre, Apellidos, Grupo, Nota y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al

servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnos.php** que coge los datos, y los saca por pantalla.

9. Crea un documento HTML de nombre **FormsAlumnosPHPmenu.html** que muestra un formulario con los datos de Nombre, Apellidos, Grupo, Nota y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosmenu.php** que coge los datos, y los saca por pantalla utilizando el mismo formato que la página de inicio del proyecto de HTML **index.html**.

Manipulación de XML mediante PHP

Para realizar los ejercicios de manipulación de XML mediante PHP vamos a usar los apuntes del enlace <http://www.php.net/manual/es/refs.xml.php>

10. Crea un documento PHP de nombre **crearxml.php** que crea un nuevo documento XML de nombre **datos.xml** que contendrá datos de alumnos.
11. Crea un documento PHP de nombre **usarxmlxsl.php** que muestra por pantalla el documento XML de nombre **datos.xml** usando como archivo de transformación **alumnos.xsl**.
12. Crea un documento PHP de nombre **usarxmlxslmenu.php** que muestra por pantalla el documento XML de nombre **datos.xml** usando como archivo de transformación **alumnos.xsl** utilizando el mismo formato que la página de inicio del proyecto de HTML **index.html**.

Nota: Si se produce un error del tipo “***Class 'XSLTProcessor' not found***” deberemos de editar el archivo **php.ini**, que se encuentra dentro de la carpeta php de XAMPP, añadir la siguiente línea y reiniciar el servidor web. Los datos de los ficheros van entre comillas simples (**'datos.xml'**).

extension=php_xsl.dll

13. Crea un documento HTML de nombre **FormsAlumnosPHPXML.html** que muestra un formulario con los datos de Nombre, Apellidos, Grupo, Nota y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosxml.php** que coge los datos, crea un nuevo documento xml de nombre **datos.xml** y muestra por pantalla los bytes grabados.
14. Crea un documento HTML de nombre **FormsAlumnosPHPXMLXSL.html** que muestra un formulario con los datos de Nombre, Apellidos, Grupo, Nota y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosxmlxsl.php** que coge los datos, crea un

nuevo documento xml de nombre **datos.xml** y los saca por pantalla usando como archivo de transformación **alumnos.xsl**.

15. Crea un documento HTML de nombre **FormsAlumnosPHPXMLXSLmenu.html** que muestra un formulario con los datos de Nombre, Apellidos, Grupo, Nota y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosxmlxslmenu.php** que coge los datos, crea un nuevo documento xml de nombre **datos.xml** y los saca por pantalla usando como archivo de transformación **alumnos.xsl** utilizando el mismo formato que la página de inicio del proyecto de HTML **index.html**.
16. Crea un documento HTML de nombre **FormsAlumnosPHPXMLXSLAdd.html** que muestra un formulario con los datos de Nombre, Apellidos, Grupo, Nota y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosxmlxsladd.php** que coge los datos, los añade al documento xml de nombre **datos.xml** (si no existe lo crea) y los saca por pantalla usando como archivo de transformación **alumnos.xsl**.

PHP y XPATH

Para poder usar XPath en PHP debemos incluir el siguiente código:

```
#creo una nueva variable para usar con xpath
$xml = new DOMXPath($nuevoxml);

#register the php: namespace (required)
$xml->registerNamespace("php", "http://php.net/xpath");

#Register PHP functions (no restrictions)
$xml->registerPHPFunctions();
```

Una vez habilitado el uso de XPath construimos la consulta que queremos ejecutar y la ejecutamos. En el siguiente ejemplo se busca un **nombre** que se envía mediante POST dentro de un documento XML de nombre **datos.xml**.

```
#consulta xpath
$nombre = $_POST['nombre'];

$consulta = "alumno[nombre='$nombre']";

#Busco los nodos que cumplen la consulta xpath
$alumnos = $xml->query($consulta);
```

Una vez ejecutada la consulta compruebo si ha devuelto algún registro.

```
#compruebo cuantos nodos ha producido la consulta
```

```
$numNodos = $alumnos->length;
```

```
if($numNodos > 0) {
```

```
#si encuentro datos con ese nombre
```

```
}
```

```
else{
```

```
#si no encuentro datos con ese nombre
```

```
}
```

Para recorrer los registros devueltos por la consulta utilizo la sentencia foreach.

```
foreach ($alumnos as $alumno) {
```

```
#por cada registro encontrado
```

```
}
```

Para obtener el valor de un campo de un registro utilizo:

```
$nombre = $alumno->getElementsByTagName("nombre")->item(0)->nodeValue;
```

Para borrar un registro utilizo:

```
$alumno->parentNode->removeChild($alumno);
```

Para actualizar los datos de un campo de un registro utilizo:

```
$nombre = $_POST['nombre'];
```

```
$alumno->getElementsByTagName("nombre")->item(0)->nodeValue = $nombre;
```

Para ejecutar una consulta compuesta por ejemplo para buscar un alumno que tiene un nombre y un grupo determinado escribimos

```
$nombre = $_POST['nombre'];
```

```
$grupo = $_POST['grupo'];
```

```
$consulta = "alumno[nombre='$nombre' and grupo='$grupo']";
```

Si queremos crear un documento xml con los datos obtenidos en la consulta y procesar esos datos usando un documento xsl escribimos

```
#carga el fichero xsl
```

```
$xsl = new DOMDocument(1.0);  
$xsl->load('alumnos.xml');  
#lo transformo  
$proc = new XSLTProcessor();  
$proc->importStyleSheet($xsl);  
#Nuevo documento  
$nuevoxml = new DOMDocument('1.0','utf-8');  
#elemento raiz  
$raiz = $nuevoxml->createElement('alumnos');  
$raiz = $nuevoxml->appendChild($raiz);  
foreach ($alumnos as $alumno) {  
    #por cada registro encontrado  
    $nuevoNodo = $nuevoxml->importNode($alumno,true);  
    $raiz->appendChild($nuevoNodo);  
}  
$xslresultado = $proc->transformToXML($nuevoxml);  
echo $xslresultado;
```

17. Crea un documento HTML de nombre **FormsAlumnosPHPBuscarNombre.html** que muestra un formulario con un campo de datos para el Nombre y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor ([localhost](#) en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosbuscarnombre.php** que busca los datos de los alumnos que tienen ese nombre en el documento xml de nombre **datos.xml** y los saca por pantalla usando como archivo de transformación **alumnos.xsl**. Si no encuentra ningún alumno mostrará el mensaje “*No se ha encontrado ningún Alumno con esos datos.*”.

18. Crea un documento HTML de nombre **FormsAlumnosPHPBuscarNombreMenu.html** que muestra un formulario con un campo de datos para el Nombre y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor ([localhost](#) en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosbuscarnombremenu.php** que busca los datos de los alumnos que tienen ese nombre en el documento xml de nombre **datos.xml** y los saca por

pantalla usando como archivo de transformación **alumnos.xsl**. Si no encuentra ningún alumno mostrará el mensaje “*No se ha encontrado ningún Alumno con esos datos.*”. El código PHP irá dentro de la sección de contenido de una página web con cabecera, menú, pie, y contenido.

19. Crea un documento HTML de nombre **FormsAlumnosPHPBorrarNombre.html** que muestra un formulario con un campo de datos para el Nombre y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosborrarnombre.php** que busca los datos de los alumnos que tienen ese nombre en el documento xml de nombre **datos.xml** y los borra. Después muestra el contenido de **datos.xml** por pantalla usando como archivo de transformación **alumnos.xsl**. Si no encuentra ningún alumno mostrará el mensaje “*No se ha encontrado ningún Alumno con esos datos.*”.
20. Crea un documento HTML de nombre **FormsAlumnosPHPBorrarNombreMenu.html** que muestra un formulario con un campo de datos para el Nombre y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosborrarnombremenu.php** que busca los datos de los alumnos que tienen ese nombre en el documento xml de nombre **datos.xml** y los borra. Después muestra el contenido de **datos.xml** por pantalla usando como archivo de transformación **alumnos.xsl**. Si no encuentra ningún alumno mostrará el mensaje “*No se ha encontrado ningún Alumno con esos datos.*”. El código PHP irá dentro de la sección de contenido de una página web con cabecera, menú, pie, y contenido.
21. Crea un documento HTML de nombre **FormsAlumnosPHPModificarNombre.html** que muestra un formulario con los datos de Nombre, Apellidos, Grupo, Nota y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosmodificarnombre.php** que comprueba si el nombre existe en el documento xml de nombre **datos.xml**. Si existe modifica los datos y muestra los datos del documento xml de nombre **datos.xml** por pantalla. Si no existe muestra un mensaje de error.
22. Crea un documento HTML de nombre **FormsAlumnosPHPModificarNombreMenu.html** que muestra un formulario con los datos de Nombre, Apellidos, Grupo, Nota y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo **alumnosmodificarnombremenu.php** que comprueba si el nombre existe en el documento xml de nombre **datos.xml**. Si existe modifica los datos y muestra los datos del documento xml de nombre **datos.xml** por pantalla. Si

no existe muestra un mensaje de error. El código PHP irá dentro de la sección de contenido de una página web con cabecera, menú, pie, y contenido.

23. Crea un documento HTML de nombre **LoginUsuarios.html** que muestra un formulario con los datos de Username, Password y un botón para Enviar los datos. Al hacer clic en el botón se mandan los datos al servidor (localhost en mi caso) usando POST. En el servidor se procesan usando el archivo loginusuarios.php que comprueba si en el documento xml de nombre usuarios.xml existe un usuario con ese username y esa password. Si existe muestra un mensaje de bienvenida. Si no existe muestra un mensaje de error. El código PHP irá dentro de la sección de contenido de una página web con cabecera, menú, pie, y contenido.

PHP - Variables de Sesión

Para realizar los ejercicios de Variables de Sesión vamos a usar los apuntes del enlace <http://www.php.net/manual/es/session.examples.basic.php>.

24. Crea un documento PHP de nombre **sesiones.php** que muestra el número de veces que se ha accedido a la página usando una variable de sesión de nombre **posicion**.

25. Crea un documento PHP de nombre **sesionesnavegar.php** que muestra el valor de un contador, que inicialmente vale 0, y cuyo valor se va guardando usando una variable de sesión. Además, mostrará dos enlaces de nombre aumentar y disminuir que incrementan o decrementan en uno el valor del contador y muestran el nuevo valor en la página. Para ello se usan los documentos PHP de nombre **aumentar.php** y **disminuir.php**.

Nota: Se usan enlaces en vez de botones porque el evento onclick se debe tratar con JavaScript (en el cliente) y no con PHP (en el servidor).

26. Crea un documento PHP de nombre **sesionesnavegarxml.php** que nos permite navegar por las posiciones del documento **datos.xml**. Para ello usa el valor de una variable de sesión de nombre **posición**, que inicialmente vale 1, y cuyo valor se va actualizando, y el de otra variable de sesión de nombre **numeroregistros** que contiene el número de nodos (datos de tipo alumno) que hay en datos.xml. El documento, mostrará un texto con la posición actual y el número de registros que hay así como unos enlaces de nombre anterior, posterior, primero y último que se encargarán de ir actualizando el valor de la posición. Para ello se usan los documentos PHP de nombre **anterior.php**, **posterior.php**, **primero.php** y **ultimo.php**.

27. Crea un documento PHP de nombre **sesionesnavegarxmldie.php** que modifica **sesionesnavegarxml.php** para que, si se no se pueden cargar los datos que hay en datos.xml muestre un mensaje de error y finalice la ejecución. Para ello se usan los documentos PHP de nombre

anteriordie.php, posteriordie.php, primerodie.php y ultimodie.php. Para ello usará la función **die** que tiene la siguiente sintaxis:

\$nuevoxml->load('datos.xml') or die("Error. No se han podido cargar los datos.");

28. Tomando como base **sesionesnavegarxmldie.php**, crea un documento PHP de nombre **sesionesnavegarxmlforms.php** que nos permite navegar por las posiciones del documento **datos.xml**. Para ello usa el valor de una variable de sesión de nombre **posición**, que inicialmente vale 1, y cuyo valor se va actualizando y el de otra variable de sesión de nombre **numeroregistros** que contiene el número de nodos (datos de tipo alumno) que hay en **datos.xml**. El documento, mostrará un texto con la posición actual y el número de registros que hay, un **formulario con los datos** del alumno de esa posición, y unos enlaces de nombre anterior, posterior, primero y último que se encargarán de ir actualizando el valor de la posición. Para ello se usan los documentos PHP de nombre **anteriorforms.php, posteriorforms.php, primeroforms.php y ultimoforms.php**.

29. Crea un documento PHP de nombre **sesionesnavegarxmlformscompleto.php** que nos permite navegar por las posiciones del documento **datos.xml**. Para ello el valor de una variable de sesión de nombre **posición**, que inicialmente vale 1, y cuyo valor se va actualizando y el de otra variable de sesión de nombre **numeroregistros** que contiene el número de nodos (datos de tipo alumno) que hay en **datos.xml**. El documento, mostrará un texto con la posición actual y el número de registros que hay, un formulario con los datos del alumno de esa posición, y unos enlaces de nombre anterior, posterior, primero y último que se encargarán de ir actualizando el valor de la posición. También se incluirán los botones de nombre añadir, borrar, y modificar para tratar con los registros. Para ello se usan los documentos PHP de nombre **anteriorformscompleto.php, posteriorformscompleto.php, primeroformscompleto.php, ultimoformscompleto.php, y actualizardatos.php**.

Nota:

Se usan **botones** en vez de enlaces para las opciones **añadir, borrar, y modificar** porque necesitamos enviar los datos que han sido modificados (mediante POST). Para controlar que botón se ha pulsado hacemos que el formulario ejecute el documento PHP **actualizardatos.php** y, en las propiedades de los botones hacemos que tengan un valor en la propiedad **name**. Ese valor de la propiedad **name** es el que usaremos en **actualizardatos.php** para comprobar que botón se ha pulsado.

Cuando **no hay registros** en **datos.xml** muestra el mensaje "No hay registros" antes de los campos de texto del formulario y pone en los campos de texto unos valores predefinidos (nombre "", apellidos "", grupo "1DW3", y nota "5.0").

Si en las cajas de texto deben de aparecer datos compuestos (por ejemplo un nombre compuesto que contiene espacios en blanco) debemos usar el componente de formulario **textarea** en vez del text.

```
echo "<p>Nombre:";  
  
echo "<textarea name=\"nombre\" >";  
  
echo "$nombre";  
  
echo "</textarea></p>";
```

Para simular el comportamiento de un text con un textarea podemos usar el siguiente código:

```
echo "<p>Nombre:";  
  
echo "<textarea name=\"nombre\" cols=\"30\" rows=\"1\" style=\"overflow:  
hidden;resize: none;\" >";  
  
echo "$nombre";  
  
echo "</textarea></p>";
```

30. Copia el contenido del Proyecto de XML en una carpeta de nombre **ProyectoXMLDatos**. Modifica la página **xml.html** que se encuentra en la carpeta xml para que las opciones del menú ejecuten código **jquery** (javascript) para cargar la hoja de transformación (XSL) que corresponda a dicha opción del proyecto, se la apliquen a los datos que se encuentran en el documento **datos.xml**, y muestren el resultado utilizando el mismo formato que la página de inicio del proyecto de HTML **index.html**.