ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

“INTRODUCCIÓN AL PHP PARTE I”

AUTORÍA
EZEQUIEL JEREZ CALERO

TEMÁTICA
LENGUAJES DE PROGRAMACIÓN

ETAPA
BACHILLERATO, FORMACIÓN PROFESIONAL

Resumen

Este documento trata de ser un pequeño manual de php. Como este lenguaje es muy extenso, aquí nos ceñiremos a una introducción sobre éste y a sintaxis básica.

Palabras clave

Lenguaje, php, introducción al php, programación, web, aplicaciones web.

1. INTRODUCCIÓN

1.1 Tipos de páginas Web.

Una sencilla clasificaron de los tipos de páginas Web ser en:

- Páginas estáticas
- Páginas dinámicas

Las páginas Web dinámicas son aquellas cuya información que presentan se genera a partir de alguna acción o petición del el usuario en la página. Contrariamente a las páginas estáticas, en las que su contenido se encuentra predeterminado, en las dinámicas la información aparece inmediatamente después de una solicitud echa por el usuario. Una página dinámica permite visualizar la información contenida en una base de datos, así como almacenar y hacer actualizaciones de cierta información a través de un formulario. Además se pueden manejar foros y el usuario tiene la posibilidad de cambiar a su gusto el diseño y el contenido de la página, entre otras cosas.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

Las páginas Web estática cuando sus contenidos no pueden ser modificados mediante ninguna intervención del usuario ni tampoco a través de ningún programa.

1.2 Script.

Un script es un conjunto de instrucciones escritas en lenguaje de programación determinado que van dentro del mismo documento de las páginas Web y pueden ser ejecutadas en un momento determinado por un intérprete.

La forma de diferenciar los script de otro lenguaje o código es con las cabeceras que

<script> y </script> que delimitan el inicio y el fin del script respectivamente.

La posibilidad de insertar en un mismo documento scripts desarrollados en distintos lenguajes obliga a especificar cuál se ha utilizado en cada caso, para que en el momento en el que vayan a ser ejecutados se invoque el intérprete adecuado.

Por eso en la etiqueta de apertura del script se debe poner el lenguaje en el que se va a escribir dicho script.

Pero aquí el lenguaje que estudiaremos será el PHP y este tiene una sintaxis específica más fácil de recordar para hacer los script que es la siguiente:

```
<?
......instrucciones..
?>
```

1.3 Lenguajes.

Hay muchos lenguajes de script pero se pueden clasificar en dos tipos:

- Del lado del cliente.
- Del lado del servidor.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

1.3.1 Lenguajes del lado del cliente.

Un lenguaje es del lado del cliente cuando el intérprete que ejecuta el script es accesible desde el lado del cliente sin que tengamos que hacer ninguna petición al servidor.

Cuando los scripts contenidos en un documento son de este tipo, el servidor lo entrega al cliente si efectuar ningún tipo de modificación.

Ejemplos de lenguajes del lado del cliente:

- DHTML
- JavaScript
- VBScript

DHTML se trata del html dinámico.

JavaScript es un lenguaje de script basado en objetos, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador Web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas Web dinámicas.

VBScript es un lenguaje de *script* derivado de *VisualBasic* y diseñado específicamente para los navegadores de *Microsoft*.

1.3.2 Lenguajes del lado del cliente servidor

Un lenguaje es del lado del servidor cuando la ejecución de sus scripts se efectúa, por el servidor y antes de dar respuesta a la petición, de manera que el cliente no recibe el documento original sino el resultante de esa interpretación previa.

Cuando se este lenguaje el cliente recibe un documento en que se han ejecutado los script y se sustituye por resultados de las ejecuciones.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

Ejemplos de lenguajes del lado del servidor:

- PHP
- ASP
- Perl
- JSP

1.4 Requisitos para el uso del lenguaje PHP.

Para usar PHP requiere tener *instalado* y *configurado*:

- Un software de servidor que soporte el protocolo HTTP.
- El intérprete de PHP.
- Un software de servidor de bases de datos capaz de ser gestionado mediante funciones propias de PHP.

2. PÁGINAS PHP.

Suelen ser páginas html o htm que se le cambia la extensión por php. En ella se pueden añadir instrucciones en lenguaje php que suelen ir entre etiquetas y hay varias formas de ponerlo:

- <? .... ?> Sólo si se activa la función short_tags() o la bandera de configuración short_open_tag.
- <?php .... ?>
- <script languaje="php"> .... </script>
- <% .... %> Sólo si se activan los tags para ficheros 'asp' con la bandera de configuración asp_tags.

Al código que pongamos entre esas etiquetas lo determinaremos script y podemos poner todos los script que queramos dentro de la página.

3. INTRODUCCIÓN A LA SINTAXIS PHP Y COMANDOS.

3.1 eho y print.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

La instrucción **echo** y **print** se utilizan para mostrar por pantalla una cadena de texto introducida a mano, aunque ahora los utilicemos para lo mismo tienen peculiaridades distinta que más adelante iremos mostrando y explicando.

Ejemplo:

```
<html>
      <head>
            <title>Prueba</title>
      <head>
      <body>
            <?
            print “Escribe un texto ...<br>”;

            print (“Escribe un texto ...<br>”);

            echo “Escribe un texto ...”;
            ?>
      </body>
</html>
```

Como podéis observar print tiene dos formas utilizarse pero que el resultado es el mismo. Una característica del php es que cada instrucción debe finalizarse con un punto y coma (**;**) y no suelen utilizarse dos instrucciones en la misma línea para que sea más fácil la depuración de los programas.

3.2 Comillas concatenadas.

Si queremos **utilizar unas comillas dentro de otras comillas** hay que utilizar distintos tipos de comilla (“ ”) o (´´) pero siempre las que abren y cierran tienen que ser las mismas.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

También otra solución seria para las internas es poner / “ /” para que php las interprete. Pero en ningún caso está permitido sustituir las comillas exteriores (las que encierran la cadena) por \".

Ejemplo de la utilización de las comillas.

```
<html>
  <head>
  </head>
  <body>
          <?
          echo "Este texto solo lleva las comillas de la instrucción <br>";
          echo "La palabra 'comillas' aparecerá entrecomillada <br>";
          echo 'La palabra "comillas" aparecerá entrecomillada <br>';
          echo "La palabra \"comillas" usando la barra invertida <br>";
          ?>
  </body>
</html>
```

4. COMENTARIOS.

Los **comentarios** suelen usarse para que en futuras revisiones a tu código por tu parte y sobretodo por parte de otros programadores, es un buen costumbre comentar lo que se esta haciendo en PHP. De esta manera el código será mucho más sencillo de comprender y a su vez de modificar, corregir.

Los forma de usar los comentarios son:

- ***//* ó #** para comentarios de una solo línea.
- ***/* */*** para comentarios de varias línea.

Ejemplo de la utilización de los tipos de comentarios:

```
<html>
  <head>
          <title>Comentarios</title>
  </head>
  <body>
          <?
          //Ejemplo de comentarios de una sola línea

          echo "Los comentarios no se muestra en el php <br>";

          /* Este comentario es
```

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

```
            el que ocupa
            varias líneas */

          #Otra forma de poner comentarios
          ?>
  </body>
</html>
```

**Nota los comentarios no pueden anidarse.

5. CONSTANTES.

Una **constante** es como una variable pero con la diferencia que una vez toma un valor este no puede variar durante la ejecución del script, otra particularidad de las constantes es que son globales, por lo que se pueden leer desde el interior de una función sin tener que pasarlas como parámetro.

Las constantes en PHP tienen que ser definidas por la función define() y además no pueden ser redefinidas con otro valor.

- define("Nombre","Valor");

No es necesario escribir entre comillas los valores de las constantes cuando se trata de constantes numéricas y punto entre caracteres numéricos es considerado como separador de parte decimal.

Además, existen una serie de variables predefinidas denominadas:

- _FILE_: Fichero que se está procesando.
- _LINE_: Línea del fichero que se está procesando
- _PHP_VERSION: Versión de PHP.
- PHP_OS: Sistema operativo del cliente.
- TRUE: Verdadero.
- FALSE: Falso.

5.1 Ampliación de echo y print.

Con **echo** se pueden mostrar de forma simultánea varias cadenas de caracteres y/o constantes y variables. Basta con ponerlas una a continuación de otra utilizando una coma como separador entre

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

cada una de ellas. También en vez de utilizar la coma se puede usar un punto, que lo interpretaría lo mismo.

Con **print** solo es posible usar el punto como elemento de unión y no la coma como en el echo porque esto conllevaría a un error de PHP.

Cuando enlacemos elementos distintos (cadenas, constantes o números) hay que tener dos cosas en cuenta:

- Las cadenas tienen que ir cerradas por sus comillas.
- Los nombres de constantes o variables nunca van entre comillas.

Ejemplo:

```
<html>
  <head>
    <title>Comentarios</title>
  </head>
  <body>
    <?
    //Definimos la constante dolares
    define (“dolares”, 224.62);

    //Definimos la constante cadena
    define (“cadena”, “Esta constante es una cadena”);

    echo “Valor de la constante dolares: ”, dolares, “<br>”;
    echo “Valor de la constante dolares: ”. dolares. “<br>”;
    print “Valor de la constante cadena: ”.cadena. “<br>”;
    echo “Con echo los números no necesitan ir entre comillas: ” ,3, “<br>”;
    print “En el caso de print si son necesarias: ” .”3”. “<br>”;

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

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

6. VARIABLES

Una **variable** es un nombre que contiene unos determinados datos, ya sean de texto o numéricos, y en php tienen la peculiaridad de ir precedidas por el signo del dólar (**$**).

Los nombres de las variables deben de llevar una letra después del signo dólar o no serán validas para php. Además php distingue entre mayúscula y minúsculas.

Para PHP las letras mayúsculas y las minúsculas son distintas.

Otra peculiaridad de php es que las variables no hay que declararla para utilizarlas, tampoco es necesario definir el tipo de variable que es y a lo largo del programa puede ir cambiando.

Ejemplo:

```
<html>
<head>
        <title>Variables</title>
</head>
<body>
        <?
        //Añadiendole un valor a las variable
        $Pepe = “Lo ves”;
        $pepe = “No son iguales”;

        echo “Pepe contiene: ”, $Pepe, “<br>";
        echo “pepe contiene: ”, $pepe, “<br>";
        ?>
</body>
</html>
```

6.1 Ámbito de las variables.

Una función definida dentro de un script puede ser utilizada desde cualquier parte del script con excepciones de las funciones y si están contenidas en ficheros externos.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

Las variables que se definen dentro de una función se limitan a su uso local dentro de la función y no se pueden utilizar fuera de ella.

Si dentro de una función llamamos una variable declarada fuera, PHP considerará esa llamada como si la variable tuviera valor cero si es numérico o cadena vacía si es una cadena de texto.

Igual ocurriría si desde fuera de una función hiciéramos alusión a una variable definida en ella.

Pero hay excepciones como son las variables globales y superglobales que se pueden utilizar en cualquier parte del script.

6.2 Variables globales.

Variables globales son aquellas que pueden ser usadas, tanto por la parte principal del programa como por cualquier procedimiento o función que se encuentre en el.

Se definen como *global nombre de la variable* y además de forma simultánea, pueden definirse varias variables.

Por ejemplo:

- global $a1;
- global $a1, $a2, $a3;

6.3 Variables superglobales.

A partir de PHP 4.1.0., se dispone de un conjunto de variables de tipo array que mantienen información del sistema, llamadas superglobales porque se definen automáticamente en un ámbito global.

Estas variables vienen predefinidas y el usuario no puede modificar su contenido.

La lista de estas variables es la siguiente: $GLOBALS, $_SERVER, $_GET, $_POST, $_COOKIE, $_FILES, $_ENV, $_REQUEST, $_SESSION.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

7. VARIABLES PREDEFINIDAS.

Hay dos variables distintas para recoger el mismo valor y esto se justifica por lo siguiente. Las variables superglobales se introdujeron en PHP a partir de la versión 4.1.0 y no existían con anterioridad, porque se usaban array asociativo y para que los programadores que cambiaran a versiones más modernas no tuvieran que reescribir el código.

Es por esta razón, por la que se mantienen dos variables distintas para recoger el mismo valor. Pero aunque los valores de ambas variables van a ser siempre idénticos, no ocurre lo mismo con su ámbito.

Las primeras son de ámbito superglobal y no necesitan ser declaradas expresamente con globales cuando se trata de utilizar sus valores dentro de cualquier función.

En el segundo caso va a ser imprescindible que sean declaradas como globales antes de que sus valores puedan ser utilizados dentro de una función.

7.1 Tipos de variables predefinidas.

- **Variables de sesión:** Las sesiones son un método seguro y eficaz de guardar y mantener datos del usuario durante toda su visita. Podemos guardar por ejemplo una variable que diga si está identificado en nuestro sistema o no y si lo está, también podemos guardar sus datos.

- **Variables del método POST:** Estas variables se usan para recoger la información que envía el cliente en el servidor.

- **Variables del método GET:** Son muy similares a las anteriores. La existencia de los dos tipos se justifica porque también existen dos tipos de métodos o maneras de enviar datos desde el cliente hasta el servidor.

- **Variables de transferencia de ficheros:** Se utilizan para la transferencia de ficheros del cliente al servidor.

- **El tipo GLOBALS:** Su finalidad es recoger en una tabla con los nombres de todas las variables establecidas como globales así como sus valores.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

7.2 Variables estáticas.

Las variables **estáticas** se usan para guardar el último valor de una variable definida dentro de una función.

La instrucción que permite establecer una variable como **estática** es la siguiente:

static nombre = valor;

7.3 Variables de variables.

PHP permite un mecanismo para mantener variables con un nombre no fijo. La forma de hacerlo sería esta:

- $$nombre_variable_previa;

Veamos un ejemplo.

$color="verde"; (Variable normal)

$$color="es horrible"; (variable de variables)

Habría tres formas de escribir la instrucción:

- print $$color;
- print ${$color};
- print $verde;

Cualquiera de las instrucciones anteriores nos produciría la misma salida: es horrible.

Cuando la variable utilizada para definir una variable de variable cambia de valor no se modifica ni el nombre de esta última ni tampoco su valor.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

8. TIPOS DE VARIABLES.

En php no hace falta definir el tipo de la variable que vas a usar y soporta ocho tipos primitivos.

Cuatro tipos escalares:

- Boolean
- Integer
- Float (número de punto-flotante, también conocido como 'double')
- String

Dos tipos compuestos:

- Array
- Object

Y finalmente dos tipos especiales:

- Resource
- NULL

8.1 Determinación de tipos de variables.

Para saber de que tipo es una variable php tiene una función que devuelve el tipo que es:

- gettype(variable)

8.2 Forzado de tipos.

En php se puede hacer que una variable cambien su tipo con el forzado de tipos ya sea para operar con ella o lo que quieras hacer con dicha variable.

Siempre que los valores que contenga estén dentro del rango del nuevo tipo de variable.

Los tipos se pueden forzar tanto en el momento de definir la variable como después de haber sido definida.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

8.3 Forzado y asignación simultanea de valores.

Para forzar una variable solo vasta con poner el tipo que quieres forzar dentro de la variable entre paréntesis.

Por ejemplo:

- $var= ((tipo)valor);
- $b=((string)85);

8.4 Forzado de tipos en variables ya definidas.

Lo mejor para el forzado de tipos de variables que ya están definidas es utilizar la función settype.

Por ejemplo:

- settype(var,tipo)
- settype($c,'integer')

Convertiría a tipo entero la variable $c.

La ejecución de la instrucción settype devuelve (da como resultado) un valor que puede ser: true o false (1 ó 0) según la conversión se haya realizado con éxito o no haya podido realizarse.

8.6 Operaciones con distinto tipo de variables.

PHP permite la realización de operaciones aritméticas con cualquiera de los tres tipos de variables y adecua el resultado al tipo más apropiado. Lo que haría PHP seria lo siguiente:

- Al operar con dos enteros, si el resultado está dentro del rango de los enteros, devuelve un entero.
- Si al operar con dos enteros el resultado desborda el rango entero, convierte su valor, de forma automática, al tipo coma flotante.
- Al operar un entero con una variable tipo coma flotante el resultado es de coma flotante.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

- Al operar con una cadena lo hace como si se tratara de un entero. Si hay caracteres numéricos al comienzo, los extrae (hasta que aparezca un punto o un carácter no numérico) y los opera como un número entero.
- Si una cadena no comienza por un carácter numérico PHP la operará tomando su valor numérico como cero.

9. UTILIZANDO FORMULARIOS.

Lo que hemos visto es hasta ahora es como escribir un script y ver los resultados y veremos la forma de que interactúen el cliente y el servidor.

9.1 Envió a través del navegador.

La forma más simple de que un cliente pueda enviar valores a un servidor es incluir esos valores en la propia petición, insertándolos directamente en la barra de direcciones del navegador.

Por ejemplo:

- pagina.php?n1=v1&n2=v2

Desglosemos ese ejemplo.

- pagina.php es la dirección de la página que contiene el script que ha de procesar los valores transferidos.
- ? es un carácter obligatorio que indica que detrás de él van a ser insertados nombres de variables y sus valores.
- n1, n2 son los nombres de las variables. Que nunca llevan el $.
- = es el separador de los nombres de las variables y sus valores respectivos.
- v1, v2 es el valor asignado a cada una de las variables. Los valores de las variables nunca se ponen entre comillas
- & es el símbolo que separa los distintos bloques variable = valor.

9.2 Recepción de datos.

Cuando el servidor recibe una petición con extensión .php en la que tiene un signo ? e incluye nombre de variables y valores se incluyen, de forma automática, en variables predefinidas del tipo:

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

- $HTTP_GET_VARS['v1']
- $HTTP_GET_VARS['v2']

Si la versión es superior a la 4.1.0, se guardan en este tipo de variables:

- $_GET['n1']
- $_GET['n2']

Estas variables tienen el valor que se le asigna en la barra de direcciones.

Si el **php.ini** está configurado con la opción **ON**, los valores transferidos desde el navegador son del tipo:

- $n1
- $n2

La opción ultima es más fácil de usar y cómoda para la recepción de datos enviados. Aunque cada uno puede usar la opción que más fácil le resulte usar.

9.3 Interpretación de los datos recibidos a través de formularios.

Hay dos métodos (**method**) de envío: '**GET**' y '**POST**'.

- Método GET
  Utiliza las mismas variables predefinidas. Los nombres de las variables son en este caso, los incluidos como name en cada una de las etiquetas del formulario.

  Respecto a los valores de cada variable, éstos serían los recogidos del formulario:

  - En los campos tipo: *text*, *password* y *textarea* serían los valores introducidos por el usuario en cada uno de esos campos.
  - En los campos tipo radio, recogería el valor indicado en la casilla marcada.
  - En el tipo checkbox se transferirían únicamente las variables y los valores que corresponden a las casillas marcadas.
  - En el campo tipo hidden se transferiría el valor contenido en su etiqueta.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

- En los campo del select sería transferido como valor de la variable la parte del formulario contenida entre las etiquetas <option></option> de la opción seleccionada.

- Método POST

  En el caso de que el método de envío sea POST hay una diferencia a tener en cuenta en cuanto a las variables que recogen la información. Ahora será:

  $HTTP_POST_VARS['n1']

  quien haga la función atribuida en el método anterior a:

  $HTTP_GET_VARS['n1']

  y ocurrirá algo similar con las superglobales, que pasarían a ser del tipo:

  $_POST['n1']

  en sustitución del $_GET['n1'] usado en el caso del método GET

  Si register_globals está en On el comportamiento de las variables directas es idéntico con ambos métodos.

9.4 Identificación del método de envió.

PHP recoge en una variable el método utilizado para enviar los datos desde un formulario. Se trata de la variable REQUEST_METHOD.

Puede ser invocada como una *variable directa* (en caso de que register globals esté en on) o a través de una de las variables de servidor.

En el primer caso la variable se llamaría:

$REQUEST_METHOD

y en el segundo:

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

$HTTP_SERVER_VARS[REQUEST_METHOD]

Cuando PHP permita el uso de variables superglobales se puede utilizar:

$_SERVER[REQUEST_METHOD]

9.5 Diferencias entre el método GET y POST.

Las diferencias que tienen son las siguientes:

- Método GET

Las particularidades de este método son las siguientes:

  - Al ser enviado el formulario se carga en el navegador la dirección especificada como action, se le añade un ? y a continuación se incluyen los datos del formulario.
  - Únicamente son aceptados los caracteres ASCII.
  - Tiene una limitación en el tamaño máximo de la cadena que contiene los datos a transferir.

- Método POST

No tiene las limitaciones indicadas para el caso de GET en lo relativo a visibilidad ni en cuanto a aceptación de caracteres no ASCII.

10. BIBLIOGRAFIA.

- *Manual oficial de PHP,* extraído el día 24 de febrero del 2010 desde http://www.hackingballz.com/herramientas/manual-oficial-de-php/index.html.

- *Guía teórica al PHP,* extraído el día 24 de febrero del 2010 desde http://www.htmlpoint.com/php/guida/index.html.

ISSN 1988-6047 DEP. LEGAL: GR 2922/2007 Nº 30 – MAYO DE 2010

- *Tutorial básico PHP,* extraído el día 24 de febrero del 2010 desde http://geneura.ugr.es/~maribel/php/.

- *Curso PHP,* extraído el día 24 de febrero del 2010 desde http://www.programacionweb.net/cursos/curso.php?num=10.

- *Luke Welling, Laura Thonsom. 2003. Desarrollo web con PHP y MySQL. Madrid. Anaya*

Autoría

- Nombre y Apellidos: Ezequiel Jerez Calero
- Centro, localidad, provincia: IES Abdera, Adra, Almería
- E-mail: ejcalero@gmail.com