

Conceptos Basicos De Php



Elaborado Por:

Daniel Pardo Cuenca.

Universidad Nacional Abierta y a Distancia

2017

¿Qué es PHP?

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Bien, pero ¿qué significa realmente? Un ejemplo nos aclarará las cosas:

Ejemplo #1 Un ejemplo introductorio

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>

    <?php
      echo "¡Hola, soy un script de PHP!";
    ?>

  </body>
</html>
```

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!"). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php` y `?>` que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts.

¿Qué Puede Hacer PHP?

Cualquier cosa. PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Aunque PHP puede hacer mucho más.

Existen principalmente tres campos principales donde se usan scripts de PHP.

- Scripts del lado del servidor. Este es el campo más tradicional y el foco principal. Son necesarias tres cosas para que esto funcione. El analizador de PHP (módulo CGI o servidor), un servidor web y un navegador web. Es necesario ejecutar el servidor con una instalación de PHP conectada. Se puede acceder al resultado del programa de PHP con un navegador, viendo la página de PHP a través del servidor. Todo esto se puede ejecutar en su máquina si está experimentado con la programación de PHP. Véase la sección sobre las instrucciones de instalación para más información.
- Scripts desde la línea de comandos. Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo de esta manera. Este tipo de uso es ideal para scripts que se ejecuten con regularidad empleando cron (en *nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden usarse para tareas simples de procesamiento de texto. Véase la sección Uso de PHP en la línea de comandos para más información.
- Escribir aplicaciones de escritorio. Probablemente PHP no sea el lenguaje más apropiado para crear aplicaciones de escritorio con una interfaz gráfica de usuario, pero si se conoce bien PHP, y se quisiera utilizar algunas características avanzadas de PHP en aplicaciones del lado del cliente, se puede utilizar PHP-GTK para escribir dichos programas. También es posible de esta manera escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal. Si está interesado en PHP-GTK, puede visitar su propio.

PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Esto incluye cualquier servidor web que pueda utilizar el binario de PHP FastCGI, como lighttpd y nginx. PHP funciona tanto como módulo como procesador de CGI.

De modo que con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas.

Con PHP no se está limitado a generar HTML. Entre las capacidades de PHP se incluyen la creación de imágenes, ficheros PDF e incluso películas Flash (usando libswf y Ming) generadas sobre la marcha. También se puede generar fácilmente cualquier tipo de texto, como XHTML y cualquier otro tipo de fichero XML. PHP puede autogenerar estos ficheros y guardarlos en el

sistema de ficheros en vez de imprimirlos en pantalla, creando una caché en el lado del servidor para contenido dinámico.


Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos. Escribir una página web con acceso a una base de datos es increíblemente simple utilizando una de las extensiones específicas de bases de datos (p.ej., para mysql), o utilizar una capa de abstracción como PDO, o conectarse a cualquier base de datos que admita el estándar de Conexión Abierta a Bases de Datos por medio de la extensión ODBC. Otras bases de datos podrían utilizar cURL o sockets, como lo hace CouchDB.

PHP también cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. También se pueden crear sockets de red puros e interactuar usando cualquier otro protocolo. PHP tiene soporte para el intercambio de datos complejos de WDDX entre virtualmente todos los lenguajes de programación web. Y hablando de interconexión, PHP tiene soporte para la instalación de objetos de Java y emplearlos de forma transparente como objetos de PHP.

¿Qué necesito?

En este manual se asume que se cuenta con un servidor que tiene soporte activado para PHP y que todos los ficheros con la extensión .php son tratados por PHP. En la mayoría de servidores, esta es la extensión predeterminada para los ficheros de PHP, aunque puede preguntar al administrador de su servidor para estar seguro. Si el servidor tiene soporte para PHP, entonces no es necesario hacer nada. Simplemente cree sus ficheros .php, guárdelos en su directorio web y el servidor los analizará por usted. No hay necesidad de compilar nada o instalar otras herramientas. Piense en estos ficheros habilitados para PHP como simples ficheros HTML con el añadido de una nueva familia de etiquetas mágicas que permiten todo tipo de cosas.

Digamos que quiere ahorrar el preciado ancho de banda y trabajar localmente. En este caso, querrá instalar un servidor web, como [» Apache](#), y por supuesto [» PHP](#). Lo más seguro es que también quiera instalar una base de datos como [» MySQL](#).



Su primera página con PHP

Comience por crear un fichero llamado hola.php y póngalo en el directorio raíz de su servidor web (`DOCUMENT_ROOT`) con el siguiente contenido:

Ejemplo #1 Nuestro primer script de PHP: hola.php

```
<html>
<head>
  <title>Prueba de PHP</title>
</head>
<body>
  <?php echo '<p>Hola Mundo</p>'; ?>
</body>
</html>
```

Utilice su navegador web para acceder al fichero con el URL de su servidor, finalizado con la referencia al fichero /hola.php. Si está programando localmente, este URL será algo parecido a `http://localhost/hola.php` o `http://127.0.0.1/hola.php`, pero esto depende de la configuración de su servidor web. Si todo está configurado correctamente, el fichero será analizado por PHP y se enviará el siguiente contenido a su navegador:

```
<html>
<head>
  <title>Prueba de PHP</title>
</head>
<body>
  <p>Hola mundo</p>
</body>
</html>
```

Este programa es extremadamente simple y realmente no es necesario utilizar PHP para crear una página como esta. Lo único que muestra es: Hola mundo empleando la sentencia `echo` de PHP. Observe que el fichero no necesita ser ejecutable o especial de ninguna forma. El servidor reconoce que este fichero necesita ser interpretado por PHP debido al empleo de la extensión ".php", ya que el servidor está configurado para enviarlo a PHP. Piense como si fuera un fichero HTML normal que tiene una serie de etiquetas especiales disponibles con las que puede hacer muchas cosas interesantes.

Si intentó usar este ejemplo y no produjo ningún resultado, se le preguntó si deseaba descargar el fichero, o se mostró todo el fichero como texto, lo más seguro es que PHP no se encuentre habilitado en su servidor o no esté configurado adecuadamente. Pídale a su

administrador que lo habilite utilizando el capítulo [Instalación](#) del manual. Si está trabajando localmente, lea también el capítulo dedicado a la instalación para asegurarse de que todo esté configurado adecuadamente. Asegúrese de que está accediendo al fichero mediante http y que el servidor muestre el resultado. Si está abriendo el fichero desde el sistema de ficheros, probablemente no será analizado por PHP. Si el problema persiste, no dude en usar alguna de las múltiples opciones del [» Soporte para PHP](#).

El objetivo de este ejemplo es el formato de las etiquetas especiales de PHP. En este ejemplo utilizamos `<?php` para indicar el inicio de una etiqueta de PHP. Después ponemos la sentencia y abandonamos el modo PHP añadiendo la etiqueta de cierre `?>`. De esta manera, se puede entrar y salir del modo PHP en un fichero HTML cada vez que se quiera. Para más información, lea la sección del manual titulada [Sintaxis básica de PHP](#).

(Toda la información anterior imágenes y contenido fue tomado de la página principal de PHP.)

Link: <http://php.net/manual/es/getting-started.php>

Ejercicios básicos de ejemplo

Ejemplo 1: Concatena dos cadenas con el operador punto (.) e imprimir su resultado.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <title>ejercicio 4.1</title>
</head>
<body>

  <?php
    $ini = "Hola ";
    $fin = " a todos";
    $todo = $ini.$fin;
    echo $todo;
  ?>

</body>
</html>
```

Ejemplo 2: hacer un programa que muestre en pantalla información de PHP con la función `phpinfo()`. Muestre la información centrada horizontalmente en la pantalla.

```
<html>
<head>
  <title>ejercicio 4.3</title>
</head>

<body>
<center>
<?php

  echo phpinfo();

</center>
</body>
</html>
```

Ejemplo 3: Mostrar en pantalla una tabla de 10 por 10 con los números del 1 al 100

```
<html>
<head>
  <title>ejercicio 4.4</title>
</head>

<body>

<?php

echo "<table border=1>";
$n=1;
for ($n1=1; $n1<=10; $n1++)
{
  echo "<tr>";
  for ($n2=1; $n2<=10; $n2++)
  {
    echo "<td>", $n, "</td>";
    $n=$n+1;
  }

  echo "</tr>";
}
echo "</table>";
?>

</body>
</html>
```

(Los ejemplos anteriores fueron obtenidos del documento, programación en php a través de ejemplos, Manuel Palomo Duarte)

Link: http://servicio.uca.es/softwarelibre/publicaciones/apuntes_php