

UNDERCODE

TALLER DE PHP #1

INTRODUCCIÓN A PHP



TEMAS

Introducción a PHP
Backend y Frontend
Instalación
Editores de Texto
Variables
Y más!...

TUTOR

2Fac3R

ERES LIBRE DE COPIAR, DISTRIBUIR Y COMPARTIR ESTE MATERIAL

ÍNDICE

| | |
|--|-----------|
| ÍNDICE | 2 |
| INTRODUCCIÓN A PHP | 3 |
| Requisitos necesarios | 3 |
| FUNCIONAMIENTO DEL BACKEND Y FRONTEND | 4 |
| ¿Qué es el Frontend? | 4 |
| ¿Qué es el Backend? | 4 |
| INSTALACIÓN | 4 |
| EDITORES DE TEXTO | 5 |
| PROGRAMACIÓN EN PHP | 6 |
| ¡Hola Mundo! | 6 |
| COMENTARIOS | 7 |
| Ejemplo: ¿Cuántos días tengo de vida? | 7 |
| VARIABLES | 8 |
| Ejemplo: Reglas de variables | 9 |
| CONSTANTES | 9 |
| Ejemplo: Definición de constantes | 9 |
| TIPOS DE DATOS | 10 |
| Tabla de tipos de datos. | 10 |
| ¿Y... AHORA QUÉ? | 11 |
| REFERENCIAS | 12 |

INTRODUCCIÓN A PHP

PHP (*Hypertext Preprocessor*) es un lenguaje de programación multiplataforma de código abierto orientado al desarrollo web y que corre del lado del servidor (*backend*).

A diferencia de los lenguajes de programación que se ejecutan del lado del cliente, como Javascript (*frontend*) el intérprete¹ de PHP se aloja del lado del servidor y es éste quien se encarga de que todo funcione correctamente.

PHP ha sido desarrollado para manejar contenido de forma *dinámica*, también puede trabajar de forma amigable con otros lenguajes en un mismo archivo, podemos incluirlo junto con javascript, html, entre otros.

Aunque el desarrollo de PHP está centrado en la programación de scripts del lado del servidor, se puede utilizar para muchas otras cosas.

Con PHP podrás generar páginas con contenido dinámico, manejar sesiones de usuarios, generar sistemas de información junto con bases de datos, entre muchas otras cosas más, depende de tu imaginación.

“La efectividad de un lenguaje depende de la imaginación del programador”

Requisitos necesarios

- ☐ Tener conocimientos básicos de HTML.
- ☐ Ganas de aprender ;)

¹ Intérprete:

Es un programa informático capaz de analizar y ejecutar otros programas, se diferencian de los compiladores o de los ensambladores en que mientras estos traducen un programa a código máquina, los intérpretes sólo realizan la traducción a medida que sea necesaria.

FUNCIONAMIENTO DEL BACKEND Y FRONTEND

Antes de continuar con el lenguaje, es necesario dejar claro algunos conceptos que son importantes en el desarrollo web.

¿Qué es el Frontend?

Cuando ingresas a un sitio web como facebook o twitter con tu navegador (cliente), lo que estás viendo en tu navegador (chrome, firefox, etc.) es decir todo el contenido multimedia (imágenes, video, texto, audio, etc.) se le conoce como lado del cliente o frontend.

¿Qué es el Backend?

Cuando ingresamos a un sitio web, por ejemplo *http://www.google.com*, estamos realizando una petición al servidor de Google, que tiene almacenado todos los datos necesarios para responder a nuestra consulta, a sí mismo, cuando nosotros usamos un servicio estamos haciendo que un proceso interno se ejecute del lado del servidor, invisible para nosotros pero se está ejecutando, cuando termina de ejecutarse nos devuelve el resultado por medio del cliente, generalmente el navegador web.

INSTALACIÓN

Debido a que PHP es un lenguaje de programación web podemos utilizarlo en cualquier sistema operativo, sólo necesitamos el servidor (Apache, por ejemplo) y el intérprete de PHP, para una instalación más sencilla recomiendo los siguientes enlaces:

- ❖ [WampServer](#) (Windows)
- ❖ [XAMPP](#) (Windows, GNU/Linux, Os X)

En ambos casos la instalación es sencilla, el típico “*siguiente* → *siguiente*” así que no veo necesario explicarlo.

EDITORES DE TEXTO

Para el desarrollo de nuestros *scripts*² vamos a utilizar algunos editores de texto, mi recomendación es la siguiente:

- Bloc de Notas (Sí, el de Windows)
- Nano, GEdit (GNU/Linux)
- [Notepad++](#)
- [Átom](#)
- [Sublime Text](#)
- [Brackets](#) (**recomendado**)

¿Cuál usar? Pruébalos y decide por ti mismo, generalmente todos te ofrecen lo mismo, pero diferente, simplemente usa el que más te guste.

Mi recomendación sería, inicia con los editores base de tu sistema operativo, por ejemplo el bloc de notas, y después cambia a otro y notarás la diferencia.

Los editores de texto nos ayudan a hacer un coloreado (highlight) de nuestro código para poder diferenciar entre funciones, variables, etc, de otra forma sería más difícil la lectura.

Otras funciones importantes que nos proveen los editores es el autocompletado que nos ayuda a programar más rápido, sin posibilidad de errores de dedo, la función de detección de errores, entre otras.

² Script:

Es un programa de texto plano que contiene una serie de instrucciones paso a paso para resolver o ejecutar un sistema de programación.

PROGRAMACIÓN EN PHP

¡Hola Mundo!

Abrimos nuestro editor de texto y guardamos el archivo como “hola.php”, por defecto sería en `/var/www` en Linux y `/wampp/www` o `/wampp/htdocs` en Windows, dependiendo de la configuración que hayas asignado en la instalación.

Ejemplo: *Hola Mundo!*

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

Asumiendo que has iniciado el servidor ya sea con wampp o xampp, vamos a el navegador e ingresamos a nuestro archivo.

127.0.0.1/hola.php

localhost/hola.php³

Explicación:

`<?php` → Iniciamos el código php

`echo` → Mostramos algo en pantalla

`?>` → Terminamos la ejecución de php

³ Localhost & 127.0.0.1

Localhost es un nombre reservado que tienen todas las computadoras es traducido como la dirección IP de loopback 127.0.0.1 en IPv4 y como la dirección ::1 en IPv6.

En pocas palabras es la dirección que apunta “a ti mismo” dentro de la red.

COMENTARIOS

Los comentarios en programación se utilizan generalmente para agregar al código texto sin que sea ejecutado por el intérprete de PHP, el programador aprovecha esto para dejar **documentación importante acerca del propio código**, recordatorios o cualquier cosa importante para el programador.

No es necesario que entiendas el código del ejemplo al 100% ahora, lo que importa es que veas el uso de los comentarios.

```
<?php
// Comentario de 1 línea.

# Otra forma de comentar 1 línea.

/*
    Comentario con varias líneas
    usualmente para mencionar
    el autor del código.
*/
```

Ejemplo: *¿Cuántos días tengo de vida?*

```
<?php
/*
    “¿Cuántos días tengo de vida?”
    Taller de PHP v2
    By 2Fac3R
    2018
*/

# Reemplaza por tus datos
$year = ;
$month = ;
$day = ;

$born = mktime(0, 0, 0, $month, $day, $year); //Obtiene la marca de tiempo
$today = time(); // Obtiene el día actual
$total = round( $today - $born / 86400 ); // resultado

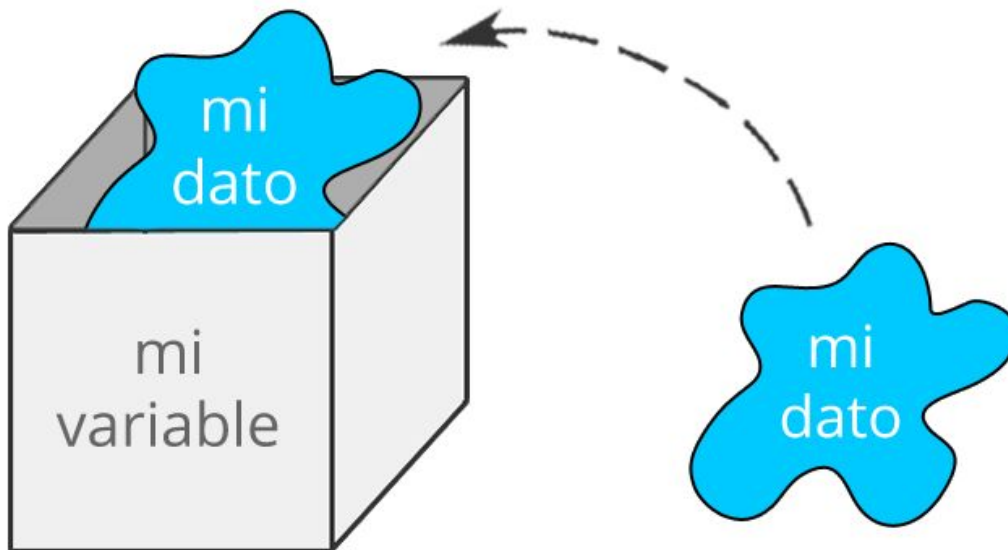
echo "Tienes $total días de vida";4
```

⁴ A partir de PHP 5 es opcional utilizar `?>` para terminar los códigos en PHP.

VARIABLES

Las [variables](#) son espacios reservados en la memoria que pueden cambiar de contenido a lo largo de la ejecución de un programa. Una variable corresponde a un área reservada en la memoria principal del ordenador.

Imagina una caja que almacena algún contenido, en la programación serían *datos*, por ejemplo:



En PHP las variables comienzan con el símbolo de dólar (\$) seguido del nombre de la variable, éste nombre es *case-sensitive*⁵.

Los nombres de las variables tienen reglas que debemos tener en cuenta. Un nombre de variable válido comienza con una *letra o guión bajo*, seguido de cualquier cantidad de letras, números o guiones bajos, cualquier otro inicio de variable es inválido y php nos dará error.

⁵ **Case-Sensitive** es una expresión usada en informática que se aplica a los textos en los que tiene relevancia escribir un carácter en mayúsculas o minúsculas, es decir, no es lo mismo “ab” que “aB”.

Ejemplo: Reglas de variables

```
<?php
$var = 'Bob';
$Var = 'Joe';
echo "$var, $Var";    // Muestra "Bob, Joe"

$4site = 'ejemplo';  // inválido; empieza con un número
$_4site = 'ejemplo 2'; // válido; empieza con un guión bajo
?>
```

[Más información](#) sobre variables.

CONSTANTES

En programación, una constante es un valor que no puede ser alterado/modificado durante la ejecución de un programa, únicamente puede ser leído.

Las constantes no llevan el signo de dólar (\$), pueden ser definidas y accedidas desde cualquier sitio sin importar las reglas de acceso de variables, no pueden ser redefinidas o eliminadas una vez se han definido.

Ejemplo: Definición de constantes

```
<?php
define("CONSTANTE", "Hola mundo.");
echo CONSTANTE; // muestra "Hola mundo."
echo Constante; // muestra "Constant" y se emite un aviso.
?>
```

Más información sobre [constantes](#) y [constantes predefinidas](#).⁶

⁶ ¿Cómo debo programar?

Por convención entre programadores, las constantes siempre son declaradas en MAYÚSCULAS para diferenciarlas a las variables dentro de el código.

TIPOS DE DATOS

| Tipo | Descripción. | // código |
|-------------------------|---|--|
| String | Secuencia de caracteres entre comillas simples o dobles. | <code>echo "Cadena de texto";</code> |
| Integer | Es un número no-decimal entre -2,147,483,648 y 2,147,483,647. | <code>echo 1337;</code> |
| Float | Es un número con punto decimal. | <code>echo 3.1416;</code> |
| Boolean | Valor lógico: Verdadero o Falso. | <code>echo true;</code> <code>echo false;</code> |
| Array | Almacena múltiples valores en una sola variable. | <code>\$fruits =</code> <code>array("Manzana",</code> <code>"Pera", "Melón");</code> |
| Object | Almacena datos e información y cómo procesarla. | <code>class myClass{</code> <code> // atributos</code> <code> myMethod(){</code> <code> // acción</code> <code> }</code> <code>}</code> |
| NULL | Es un tipo de variable que no tiene un valor asignado. | <code>\$var = null;</code> |

Tabla de tipos de datos.⁷

⁷ Pro-Tip

Siempre programa en inglés. Sí, tienes que aprender inglés my friend. ;)

¿Y... AHORA QUÉ?

PHP es un lenguaje complejo que ha sufrido años de giros, modificaciones, y actualizaciones. Es muy inconsistente y a veces con errores. Cada versión tiene sus propias características únicas y caprichos, y es difícil hacer un seguimiento de qué versión tiene qué problemas. Es fácil ver por qué se pone tanto odio a veces entre programadores.

A pesar de eso, es el lenguaje más popular en la web hoy en día. Debido a su larga historia, encontrarás muchos tutoriales sobre cómo hacer cosas básicas, el problema es que de cinco tutoriales, tienes buenas posibilidades de encontrar cinco formas totalmente diferentes de hacer algo. ¿Cuál es la manera "correcta"? Es realmente difícil de descubrir.

Esa es también una de las razones por las que los nuevos programadores de PHP son tan frecuentemente culpados de códigos poco actualizados o desagradables. ¡No pueden evitarlo si el primer resultado de Google fue un artículo de hace cuatro años que enseñaba un método de cinco años!

Por esto y mucho más, te dejo a tu disposición [referencias](#) importantes para tu aprendizaje, no te quedes con los conocimientos de un sólo curso o libro, siempre intenta seguir las normas que a lo largo de los años, los programadores han intentado asignar al lenguaje para su mejor uso y eficiencia.

Ten en cuenta que es sólo una introducción al lenguaje de programación PHP, existen muchas otras cosas interesantes que puedes hacer y un mundo inmenso por explorar, el límite es tu imaginación!.

Éste documento fue escrito con la finalidad de actualizar el taller de PHP escrito originalmente para la comunidad Underc0de, por lo tanto, cualquier duda o comentario, ya sabes dónde encontrarme.

By 2Fac3R.

REFERENCIAS

- [Sitio Oficial de PHP](#)
- [Underc0de: Comunidad de habla hispana sobre Seguridad Informática y Programación.](#)
- [Proyecto: Escuela de Hackers & Programadores.](#)
- [PHP the right way](#) - ([Proyecto en español](#)).
- [PHP the wrong way?](#)
- [Best Practices](#)
- [Github: La mayor red de programadores del mundo.](#)
- [W3Schools: El sitio más grande de desarrollo web.](#)
- [Sobre mí.](#)