

¿Qué es el lenguaje PHP

PHP (proveniente del acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente en el desarrollo web y por la manera en la que se puede incrustar en el HTML por lo que permite desarrollar webs y aplicaciones web dinámicas, favoreciendo la conexión entre el servidor y la interfaz del usuario.

¿Para qué sirve el lenguaje PHP?

Este es un lenguaje versátil que se utiliza en una amplia gama de aplicaciones. Este lenguaje se incrusta en HTML para añadir nuevas funcionalidades. PHP sirve para conectar ciertos elementos de una página con determinados datos, permitiendo que el contenido cambie con las peticiones del usuario. Esta petición se manda a un servidor que, a su vez, contacta con una base de datos de donde extrae información, esta información vuelve a la página que varía (o no) en función de esta.

¿Cuál es la diferencia entre los archivos HTML y PHP?

Sabemos que la principal diferencia entre los archivos HTML y PHP es que, el primero es un lenguaje de marcado, mientras que el segundo es un lenguaje de programación. Lo que significa que los archivos HTML se usan para marcar y estructurar contenidos de un sitio web, mientras que el archivo PHP se usa para crear contenido dinámico y procesar datos.

¿Para qué sirve la etiqueta `<?php ?>`

Cuando PHP analiza un fichero, busca las etiquetas de apertura y cierre, que son `<?php` y `?>`, y que indican a PHP dónde empezar y finalizar la interpretación del código. Este mecanismo permite embeber a PHP en todo tipo de documentos, ya que todo lo que esté fuera de las etiquetas de apertura y cierre de PHP será ignorado por el analizador.

¿Cuáles son los pasos y requisitos para cargar un archivo PHP en un servidor?

los requisitos de PHP para Windows son:

- Múltiples sistemas operativos de Windows son compatibles con PHP 5.5+, pero los usuarios de 7.2.0+ no pueden utilizar Windows 2008 o Windows Vista.
- Visual C Runtime (CRT).
- Visual Studio 2012, 2015, 2017 o Microsoft Visual C++ Redistributable para Visual Studio 2019, dependiendo de tu versión de PHP.

Los pasos para cargar al servidor son los siguientes:

- Habilitar IIS y luego usar WebPI para instalar PHP. Después de lanzar WebPI para instalar PHP.
- Hacer clic en Agregar en la versión que desees, y luego haz clic en Instalar.

También se puede lograr instalando WampServer o XAMPP

Elabora un programa básico “Hola mundo” con PHP, describe el funcionamiento y elementos utilizados. Inserta el programa embebido con replit en tu página.

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

¿Para qué sirve la función echo en PHP?

Esta función se utiliza para imprimir o mostrar contenido en la página web. Puede mostrar texto, variables y combinaciones de ambos. Es una forma rápida y fácil de generar resultados visibles para los usuarios

¿Cuál es la sintaxis de la programación en PHP?

Un script de PHP se puede colocar en cualquier parte del documento, empezando con <?php y termina con ?>.

Por ejemplo:

```
<?php
// PHP code goes here
?>
```

La extensión que tienen que tener estos documentos es .php

MAYÚSCULAS Y MINÚSCULAS

En el lenguaje PHP, las palabras clave por ejemplo else, echo, if, las clases, funciones y funciones que definen a un usuario no son sensibles a las mayúsculas o minúsculas lo que significa que, en el siguiente ejemplo, el mensaje de imprimirá de la misma manera:

```

<!DOCTYPE html>
<html>
<body>

<?php
ECHO "Hola mundo<br>";
echo "Hola mundo <br>";
EcHo "Hola mundo <br>";
?>

</body>
</html>

```

Pero es importante saber que los nombres de las variables sí son sensibles a las mayúsculas y a las minúsculas, por ejemplo:

```

<!DOCTYPE html>
<html>
<body>

<?php
$color = "red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLoR . "<br>";
?>

</body>
</html>

```

En este caso solo se mostrará lo que corresponde a la manera de utilizar las mayúsculas y minúsculas de la variable específica

¿Cómo se declaran variables en PHP?

En PHP las variables se representan con un signo de dólar seguido por el nombre de la variable.

Un nombre de variable válido tiene que empezar con una letra o un carácter de subrayado (underscore), seguido de cualquier número de letras, números y caracteres de subrayado. Como expresión regular se podría expresar como: '[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*'

Por ejemplo:

```
<?php
$var = 'Roberto';
$Var = 'Juan';
echo "$var, $Var"; // imprime "Roberto, Juan"

$4site = 'aun no'; // inválido; comienza con un número
$_4site = 'aun no'; // válido; comienza con un carácter de subrayado
$täyte = 'mansikka'; // válido; 'ä' es ASCII (Extendido) 228
?>
```

¿Cómo se declaran funciones en PHP?

Para crear funciones en PHP se debe de escribir la palabra reservada function seguida del nombre de la función y los paréntesis, se declara la función, y mediante el uso de llaves se declara que líneas de código pertenecen a la función.

El nombre de una función deberá empezar siempre con una letra y no puede contener ni guiones, ni caracteres extraños, tan solo letras (mayúsculas o minúsculas) y números, acompañados de guion bajo.

Por ejemplo:

```
#php

function nombre_de_mi_funcion() {
    //líneas de código que pertenecen a la función
}
```

¿Cómo se llaman funciones en PHP?

Para llamar funciones es necesario indicar los valores para los parámetros de entrada. Si nos dejamos algún parámetro de entrada sin valor estaremos incurriendo en una falta leve, pero que en el peor de los casos impedirá que la función se ejecute correctamente. En el caso de tener habilitada la visualización de errores nos aparecerá un Warning.

Por ejemplo:

```

<?php

function con_parametros ( $var1, $var2, $var3 ) {

$suma = $var1 + $var2 + $var3;

echo 'La suma de los tres números es : ' . $suma . '<br/>';

}

con_parametros( 1, 2, 3 );

con_parametros( 24, 33, 10 );

?>

```

¿Cuáles ciclos de programación están disponibles en PHP?

- While, el cual pasa por el bloque del código mientras una condición especificada sea “verdadera”
 - Sintaxis:

```
while (condición) sentencia;
```

- Do...While, este tipo de ciclo pasa por el bloque de código una vez, y luego se repetirá siempre que una condición especial sea verdadera. A diferencia del anterior, este bucle siempre ejecuta al menos la primera repetición.
 - Sintaxis:

```
do {

    sentencia;

}

while (condición);
```

- For, que recorre el bloque de código un número específico de veces y se usa cuando sabe exactamente cuántas veces se quiere ejecutar el enunciado o bloque de enunciados, por lo que a diferencia de las anteriores debe incluir una variable de control, la cual se incrementa o disminuye de forma automática.
 - Sintaxis:

```
for (inicialización; condición; incremento) {

    sentencia;

}
```

- Foreach: Recorre un bloque de código para cada elemento de un arreglo o matriz. En cada ciclo, el valor del elemento de matriz actual se asigna internamente a \$valor y el puntero interno de matriz se mueve en uno de tal forma que en el siguiente ciclo se procesará el siguiente elemento.
 - Sintaxis:

```
foreach (expresión_arreglo as $valor) {  
    sentencia;  
}
```

¿Cuáles funciones de condicional están disponibles en PHP?

Las funciones condicionales disponibles en php son:

- If: permite la ejecución condicional de fragmentos de código al cumplirse una condicion.
 - Ejemplo:

```
<?php  
if ($x > $y) {  
    echo "$x es mayor que $y";  
}
```

- Else: Sirve para ejecutar una sentencia cuando otra no se cumple. else extiende una sentencia if, y se ejecuta cuando ésta es false.
 - Ejemplo:

```
<?php  
if ($x > $y) {  
    echo "$x es mayor que $y";  
} else {  
    echo "$y es mayor que $x";  
}
```

- Elseif/else if: Es una combinación entre if y else. Se ejecuta cuando if es false, pero sólo si la expresión condicional del elseif es true.

Puede haber varios elseif dentro de una sentencia if (aunque en ese caso suele ser más recomendable usar switch). No hay diferencias de comportamiento entre elseif y else if.

elseif sólo se ejecuta si el if precedente o cualquier elseif anterior son evaluadas como false.

- Ejemplo:

```
<?php  
if ($x > $y) {  
    echo "$x es mayor que $y";  
} elseif ($x == $y) {  
    echo "$x es igual que $y";  
} else {  
    echo "$y es mayor que $x";  
}
```

- While: Es el tipo más sencillo de loop en PHP. Se ejecutan las sentencias dentro del while siempre y cuando se evalúen como true. El valor de la expresión se comprueba cada vez al inicio del loop, y la ejecución no se detendrá hasta que finalice la iteración (cada vez que PHP ejecuta las sentencias en un loop es una iteración). Si la expresión while se evalúa como

false, las sentencias no se ejecutarán ni siquiera una vez. También es posible agrupar varias instrucciones while dentro de una.

- Ejemplo:

```
<?php
$i = 1;
while($i <= 10){
    echo $i;
    $i++;
}
```

- Do-while: Muy similares a los loops while, simplemente aquí la expresión para el loop se verifica al final en lugar de al principio, esto garantiza que el código se ejecute por lo menos la primera vez. Este loop se ejecutaría sólo una vez, ya que después no cumple la condición.

- Ejemplo:

```
<?php
$i = 0;
do {
    echo $i;
} while ($i > 0);
```

- For: Los loops for son los más complejos en PHP.

- Ejemplo:

```
<?php
for ($i = 1; $i <= 10; $i++) {
    echo $i;
} // Devuelve 123456789
```

- Las expresiones o conjunto de expresiones van separadas por punto y coma ; y sólo hay 3.
- La primera expresión, \$i = 1, se ejecuta una vez incondicionalmente al comienzo del bucle.
- La segunda expresión, \$i <= 10, es una condición, si es true, se ejecutará la tercera expresión.
- La tercera expresión, \$i++, es la acción a realizar si se cumple la segunda expresión.

Cada una de las expresiones pueden estar vacías o contener múltiples expresiones, lo que resulta útil en ciertas ocasiones. Si la expresión 2 está vacía, el loop será definido como true.

```

<?php
// Todos los siguientes ejemplos son válidos y devuelven lo mismo, 123456789
// EJEMPLO 1
for($i = 1; $i <= 10; $i++) {
    echo $i;
}

// EJEMPLO 2
for ($i = 1; ; $i++){
    if($i > 10) {
        break;
    }
    echo $i;
}

// EJEMPLO 3
$i = 1;
for( ; ; ){
    if($i > 10){
        break;
    }
    echo $i;
    $i++;
}

// EJEMPLO 4
for ($i = 1, $j = 0; $i <= 10; $j += $i, print $i, $i++);

```

Dependiendo de donde se coloque el operador incremento o decremento, dará un resultado distinto:

```

for ($i = 1; $i <= 10; print $i++); // 12345678910
for ($i = 1; $i <= 10; print ++$i); // 234567891011

```

Se pueden hacer iteraciones por medio de arrays:

```

$gente = array(
    array('nombre' => 'Carlos', 'salt' => 123),
    array('nombre' => 'Maria', 'salt' => 124));
// Numero de gente
$size = count($gente);
// Loop que da un salt de 4 dígitos aleatorio
for ($i=0; $i < $size; ++$i) {
    $gente[$i]['salt'] = mt_rand(0000, 9999);
}
var_dump($gente);

```

- Foreach: foreach permite una forma fácil de iterar sobre arrays u objetos. Cuando foreach inicia su ejecución, el puntero apunta directamente al primer elemento del array, por lo que no es necesario llamar a la función reset() antes que un loop foreach. Es recomendable no cambiar el puntero dentro del loop.
 - Ejemplo:

```

<?php
// Devuelve directamente el value de cada key, comenzando desde el primero
foreach ($array as $value) {
    // Lo que se desee mostrar
}
// Devuelve cada key con cada value, para poder trabajar con cualquiera de los dos
foreach ($array as $key => $value){
    // Lo que se desee mostrar
}

```



```
<?php
$array = array(1, 2, 3, 4);
foreach ($array as &$amp;value){
    $value = $value * 2;
}
// cada valor del array vale ahora : 2, 4, 6, 8
unset($value);
```

- Break: se usa para terminar la ejecución de un ciclo prematuramente y pasar a la instrucción inmediatamente posterior.
 - Ejemplo:

```
// Ejemplo sin valor numérico
$array = array('uno', 'dos', 'parar', 'tres');
while(list(, $valor) = each($array)){
    if($valor == 'parar'){
        break;
    }
    echo "$valor<br>";
}
// Ejemplo con valor numérico
$i = 0;
while (++$i){
    switch($i){
        case 5:
            echo "He llegado a 5 <br>";
            break 1; // Aquí sólo saldría del switch
        case 10:
            echo "He llegado a 10 <br>";
            break 2; // Sale del switch y del while
        default:
            break;
    }
}
```

- Continue: se utiliza para detener la iteración actual de un ciclo pero no termina el ciclo, sino que comienza a evaluar la condición que abre la estructura de control
 - Ejemplo:

```
<?php
for ($i=0; $i < 10; $i++) {
    if($i % 2 == 0)
        continue;
    print "$i ";
} // Devuelve 1 3 5 7 9
```

- Switch: switch es como una serie de sentencias if. Es útil para comparar una misma variable o expresión con valores diferentes y ejecutar un código diferente a otro dependiendo de esos valores.
 - Ejemplo:

```
<?php
switch ($i) {
    case "perro":
        echo "\$i es un perro";
        break;
    case "gato":
        echo "\$i es un gato";
        break;
    case "avestruz":
        echo "\$i es un avestruz";
        break;
}
```

- Declare: declare sirve para fijar directivas de ejecución para un bloque de código.
 - Ejemplo:

```
<?php
declare(ticks=1){
    // Código afectado
}

declare(ticks=1);
// Código afectado
```

- Return: return devuelve el control del programa al módulo que lo invoca. La ejecución vuelve a la siguiente declaración después del módulo que lo invoca.
- Include/include_once: incluye y ejecuta un archivo.
 - Ejemplo:

```
<?php
// archivo1.php
$color = 'azul';
// archivo2.php
echo $color; // Devuelve un Notice: Undefined variable: color
include 'archivo1.php';
echo $color; // Devuelve azul
```

- Require/require_once: require hace lo mismo que include pero en caso de fallar devuelve un error fatal de nivel E_COMPILE_ERROR, por lo que no puede continuar el script. include sólo emite un E_WARNING que permite continuar el script.
- Goto: se utiliza para saltar a otra sección del script. La etiqueta de destino debe estar dentro del mismo fichero y ámbito.
 - Ejemplo:

```
<?php
goto x;
echo 'Hola!';

x:
echo 'Adios!'; // sólo se imprimirá Adios!
```

Identifica los diferentes métodos reservados para el tratamiento de archivos.

- r: Apertura para lectura. Puntero al principio del archivo
- r+: Apertura para lectura y escritura. Puntero al principio del archivo
- w: Apertura para escritura. Puntero al principio del archivo y lo sobrescribe. Si no existe se intenta crear.
- w+: Apertura para lectura y escritura. Puntero al principio del archivo y lo sobrescribe. Si no existe se intenta crear.
- a: Apertura para escritura. Puntero al final del archivo. Si no existe se intenta crear.
- a+: Apertura para lectura y escritura. Puntero al final del archivo. Si no existe se intenta crear.
- x: Creación y apertura para sólo escritura. Puntero al principio del archivo. Si el archivo ya existe dará error E_WARNING. Si no existe se intenta crear.
- x+: Creación y apertura para lectura y escritura. Mismo comportamiento que x.

- c: Apertura para escritura. Si no existe se crea. Si existe no se sobrescribe ni da ningún error. Puntero al principio del archivo.
- c+: Apertura para lectura y escritura. Mismo comportamiento que C.

Elabora un programa básico con PHP que permita crear un archivo de texto en el servidor, describe el funcionamiento y elementos utilizados. Inserta el programa embebido con replit en tu página.

¿Cómo se utiliza el método Open/Read en el manejo de archivos con PHP?

ABRIR UN ARCHIVO

Los archivos en PHP se abren con la función `fopen()`, que requiere dos parámetros: el archivo que se quiere abrir y el modo en el que abrir el archivo. La función devuelve un puntero en el archivo si es satisfactoria o cero si no lo es. Los archivos se abren para realizar operaciones de lectura o escritura.

```
$fp = fopen("miarchivo.txt", "r");
```

Se puede abrir un archivo, pero también una URL externa, ya que `fopen()` realmente lo que hace es crear una conexión, por eso hay que cerrarla posteriormente.

```
$fp = fopen("http://localhost:8000", "r");
```

LEER UN ARCHIVO

Una vez abierto el archivo, vamos a leerlo y guardar sus contenidos en una variable con `fread()`:

```
$file = "miarchivo.txt";
$fp = fopen($file, "r");
$contents = fread($fp, filesize($file));
```

La variable `$contents` guardará el contenido que obtengamos con la función `fread()`. Esta función requiere dos parámetros, el archivo abierto y la longitud que queremos leer de dicho archivo (en bytes). En este caso hemos empleado la función `filesize()` para obtener el tamaño del archivo y así devolver todo su contenido.

¿Cómo se utiliza el método Create/Write en el manejo de archivos con PHP?

CREAR ARCHIVO

Para crear un archivo podemos utilizar nuevamente la instrucción `fopen`, la cual buscará el archivo y si no se encuentra la creará en la dirección especificada.

Por ejemplo:

```
1 <?php
2 $file_handle = fopen('/home/tutsplus/files/tmp.txt', 'w');
3 ?>
```

ESCRIBIR EN UN ARCHIVO

Para escribir en un archivo podemos utilizar la instrucción `fwrite`, que permite escribir el contenido de una cadena en el flujo de archivo al que hace referencia el identificador del archivo.

Por ejemplo:

```
1 <?php
2 $file_handle = fopen('/home/tutsplus/files/tmp.txt', 'a+');
3 fwrite($file_handle, 'Tuts+ is a great online resource to learn skills you want!');
4 fwrite($file_handle, "\n");
5 fwrite($file_handle, 'Visit tutstplus.com to know more!');
6 fclose($file_handle);
7 ?>
```

En este ejemplo se abre el archivo con el modo “a+” que lo abre para lectura y escritura, con el puntero del archivo colocado al final de este último. Por lo tanto, nuestro contenido será anexado al final del archivo, después de cualquier otro contenido.

Finamente, concluye cuales serían las ventajas de integrar PHP en una página web.

Las ventajas que pudimos detectar a la hora de integrar PHP en una página web es, primeramente, lo fácil que es trabajar con este lenguaje ya que es bastante amigable al usuario y se parece a otros lenguajes populares de programación.

Otra ventaja que se encuentra es que tiene frameworksexcelentes para el desarrollo web y soporta una gran cantidad de bases de datos y peticiones.

De la misma manera es muy rápido y permite que cualquiera pueda aplicar modificaciones para mejorarlo

Referencias:

https://www.w3schools.com/php/php_intro.asp

PHP: ¿Qué es PHP? - Manual. (s. f.). <https://www.php.net/manual/es/intro-what-is.php>

Juan de Assembler Institute. (2023, 4 octubre). ¿Qué es PHP y para qué sirve? - Assembler Institute. Assembler Institute. <https://assemblerinstitute.com/blog/que-es-php/>

Vargas, B. (2023, 20 febrero). ¡Descubre las notables Diferencias entre HTML y PHP! Byron Vargas ®. https://www.byronvargas.com/diferencias/descubre-las-notables-diferencias-entre-html-y-php/#google_vignette

Ravoof, S. (2023, 21 agosto). Cómo Instalar PHP en Cualquier Servidor (Linux, macOS, Windows). Kinsta®. <https://kinsta.com/es/blog/instalar-php/>

Developer, M. (2023, 6 julio). PHP echo: Cómo imprimir texto y variables en tu web. Mauricio Developer. https://www.mauriciodeveloper.com/post/php-echo-como-imprimir-texto-y-variables-en-tu-web-948#google_vignette

Rootstack. (s. f.). PHP: cuál es la sintaxis básica de este lenguaje de programación. Rootstack. <https://rootstack.com/es/blog/php-cual-es-la-sintaxis-basica-de-este-lenguaje-de-programacion>

PHP: Conceptos básicos - Manual. (s. f.). <https://www.php.net/manual/es/language.variables.basics.php>

Gómez, J. (s. f.). Cómo se crean funciones en PHP | Ejemplos sencillos de funciones PHP.
<https://www.srcodigofuente.es/curso-php/aprender-como-crear-funciones-y-como-se-usan>

Redacción. (2023, 18 julio). Bucles en programación PHP: conoce los 4 tipos de bucles. Saber Digital.
<https://saber.digital/los-4-tipos-de-bucles-en-programacion-php/>

Estructuras de control en PHP. (s. f.). <https://diego.com.es/estructuras-de-control-en-php>

Manejo de archivos en PHP. (s. f.). <https://diego.com.es/manejo-de-archivos-en-php>

Soni, S. (2020, 26 junio). Cómo crear, escribir, leer y eliminar archivos en PHP. Code Envato Tuts+.
<https://code.tutsplus.com/es/create-write-read-and-delete-files-in-php--cms-34950t>

Surra, B. (2023, 20 julio). PHP: características, beneficios y frameworks. MyTaskPanel Consulting.
<https://www.mytaskpanel.com/php-caracteristicas-beneficios/>