

Front-End y Back-End

Front-End

Se le llama front-end a la parte de una pagina web donde los usuarios interactúan con ella.

Esta hecha por una combinación de HTML,CSS y JavaScript.

Back-End

El Back-End esta enfocado en hacer que todo lo que está detrás de un sitio web funcione correctamente.

Toma los datos, los procesa y los envía al usuario, además de encargarse de las consultas o peticiones a la Base de Datos, la conexión con el servidor, entre otras tareas que debe realizar.

Normalmente hechas con lenguajes como:

PHP, Ruby, ASP, Python, JavaScript, SQL, MongoDB, MySQL, etc.

PHP

PHP

Sus siglas significan **Hypertext Preprocessor** es un lenguaje de script para servidores, diseñado específicamente para el desarrollo web.

Se puede embeber fácilmente en un archivo HTML y en archivos de PHP se puede escribir código HTML.

PHP

El código PHP se ejecuta del lado del servidor y el resultado lo regresa al navegador.

La única información que el cliente recibe del resultado al ejecutar un código PHP, es el resultado en si y no el código PHP escrito; En otras palabras, el output de tu programa.

Php puede hacer todo lo relacionado con programación del lado del servidor o mejor conocido como el Back-end.

PHP | Estándares de Programación

PHP tiene algunas reglas para estandarizar la programación y sea mas entendible en general las cuales son las siguientes.

- **PHP tags:** Uno debe usar la version estandar de los tags envez de la version corta
- **Comentarios:** Se usa el estándar de C y C++, (`//`) (`/* */`)
- **Estructuras de control y ciclos:** Deben tener un espacio entre la palabra clave y donde se abren los paréntesis para diferenciarlos de las funciones.
- **Funciones:** No debe de haber un espacio entre el nombre de la función y los paréntesis
- **Nombres Variables:**
 - Usar minúsculas para nombrar las variables.
 - Usar `'_'` para separar palabras en las variables.
 - Variables estáticas deben de comenzar con al letra `'s'`.
 - Variables Globales deben de comenzar con la letra `'g'`.
- **Bloques de código:** Deben de estar debidamente alineados.

PHP | Sintaxis Básica

La forma de introducir código PHP en un archivo se llama **Escapando** y tiene varias formas de hacerlo.

- **Canonical PHP Tags:** Empieza con `<?php` y termina con `?>`.
- **Versión Corta:** Empieza con `<?` y termina con `?>`
- **Tipo ASP:** Se ocupa habilitar la opcion en el archivo php.ini. Empieza con `<%` Termina con `%>`.

PHP | Sintaxis Básica

- PHP Ignora todos los espacios en una línea o en múltiples líneas hasta que encuentre un punto y coma ‘;’, PHP trata múltiples líneas como un solo comando.
- Todas las funciones, palabras claves y nombres de clases del lenguaje de PHP **no son case-sensitive** excepto las variables.
- Es un lenguaje a base de llaves, para definir bloques de códigos ({}).

PHP | Variables

Tiene su forma unica de declarar y almacenar variables.

- Toda variable declara debe comenzar con el signo de dolares '\$' seguido por el nombre de la variable.
- El nombre de la variable solo puede contener caracteres alfanuméricos y guiones ('a-z', 'A-Z', '0-9', '_').
- El nombre de la variable debe comenzar con letras o guionbajo y nunca con números
- No se ocupa declarar el tipo de dato que va contener la variable.
- Variables son case-sensitive.

PHP | Tipo de datos

- Integer
- Double
- String
- NULL
- Boolean
- Arrays
- Objects
- Sources: Se utilizan para guardar referencias a funciones o algun recurso externo de php.

PHP | Operadores

Aritméticos

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power

PHP | Operadores

Asignación

Assignment	Same as...	Description
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right
<code>x += y</code>	<code>x = x + y</code>	Addition
<code>x -= y</code>	<code>x = x - y</code>	Subtraction
<code>x *= y</code>	<code>x = x * y</code>	Multiplication
<code>x /= y</code>	<code>x = x / y</code>	Division
<code>x %= y</code>	<code>x = x % y</code>	Modulus

PHP | Operadores

Comparación

Operator	Name	Example	Result
==	Equal	<code>\$x == \$y</code>	Returns true if \$x is equal to \$y
===	Identical	<code>\$x === \$y</code>	Returns true if \$x is equal to \$y, and they are of the same type
!=	Not equal	<code>\$x != \$y</code>	Returns true if \$x is not equal to \$y
<>	Not equal	<code>\$x <> \$y</code>	Returns true if \$x is not equal to \$y
!==	Not identical	<code>\$x !== \$y</code>	Returns true if \$x is not equal to \$y, or they are not of the same type
>	Greater than	<code>\$x > \$y</code>	Returns true if \$x is greater than \$y
<	Less than	<code>\$x < \$y</code>	Returns true if \$x is less than \$y
>=	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if \$x is greater than or equal to \$y
<=	Less than or equal to	<code>\$x <= \$y</code>	Returns true if \$x is less than or equal to \$y
<=>	Spaceship	<code>\$x <=> \$y</code>	Returns an integer less than, equal to, or greater than zero, depending on if \$x is less than, equal to, or greater than \$y. Introduced in PHP 7.

PHP | Operadores

Incremento/Decremento

Operator	Name	Description
<code>++\$x</code>	Pre-increment	Increments \$x by one, then returns \$x
<code>\$x++</code>	Post-increment	Returns \$x, then increments \$x by one
<code>--\$x</code>	Pre-decrement	Decrements \$x by one, then returns \$x
<code>\$x--</code>	Post-decrement	Returns \$x, then decrements \$x by one

PHP | Operadores

Lógicos

Operator	Name	Example	Result
and	And	\$x and \$y	True if both \$x and \$y are true
or	Or	\$x or \$y	True if either \$x or \$y is true
xor	Xor	\$x xor \$y	True if either \$x or \$y is true, but not both
&&	And	\$x && \$y	True if both \$x and \$y are true
	Or	\$x \$y	True if either \$x or \$y is true
!	Not	!\$x	True if \$x is not true

PHP | Operadores

Operadores de Cadenas.

Operator	Name	Example	Result
.	Concatenation	\$txt1 . \$txt2	Concatenation of \$txt1 and \$txt2
.=	Concatenation assignment	\$txt1 .= \$txt2	Appends \$txt2 to \$txt1

PHP | Superglobals

Son arreglos especiales definidos en PHP, que hacen fácil obtener información acerca del contexto de la variable.

Se pueden acceder a ellos desde cualquier funcion, clase o archivo con tan solo declararlos. Se utilizan para transmitir información de una pagina a otra.

- **`$GLOBALS`** Almacena todas las variables globales en un arreglo, donde el index del arreglo es el nombre de la variable.
- **`$_SERVER`** Guarda informacion acerca de los headers, paths y lugares de los scripts.
- **`$_REQUEST`** Recolecta información enviada por el tag de HTML `<form>`, no es usada mucho porque `$_GET` y `$_POST` hacen los mismo.
- **`$_GET`** Recolecta información enviada por el tag de HTML `<form>`, la información es visible en la barra de direcciones, la información no esta oculta.

PHP | Superglobals

- **\$_POST** Recolecta información enviada por el tag de HTML `<form>`, la información no se muestra en la barra de direcciones ya que esta oculta.
- **\$_SESSION** Contiene información sobre variables de una sesión, tal como el usuario, correo, etc.
- **\$_COOKIE** Es un archivo que se crea en la computadora del usuario, y permite crear ese archivo o adquirir información de ese archivo
- **\$_FILES** Contiene todos los items que se subirán al servidor mediante el método HTTP POST.
- **\$_ENV** Contiene variables de entorno, las variables de entorno son adquiridas a través del SHELL en el cual está corriendo PHP

PHP | Constantes

Son identificadores o Nombres, los cuales podemos asignarles valores fijos.

```
define(name, value, case_insensitive)
```

- name: nombre de la constante.
- value: valor fijo.
- case_insensitive: Define si va a ser case sensitive o no.

Las constantes son variables globales por defecto

PHP | Estructuras de Selección

IF

```
if (condition){  
    // if TRUE then execute this code  
}
```

PHP | Estructuras de Selección

IF-ELSE

```
if (condition) {  
    // if TRUE then execute this code  
}  
else{  
    // if FALSE then execute this code  
}
```

PHP | Estructuras de Selección

IF-ELSEIF-ELSE

```
if (condition) {  
    // if TRUE then execute this code  
}  
elseif {  
    // if TRUE then execute this code  
}  
elseif {  
    // if TRUE then execute this code  
}  
else {  
    // if FALSE then execute this code  
}
```

PHP | Estructuras de Selección

SWITCH

```
switch(n) {  
    case statement1:  
        code to be executed if n==statement1;  
        break;  
    case statement2:  
        code to be executed if n==statement2;  
        break;  
    case statement3:  
        code to be executed if n==statement3;  
        break;  
    case statement4:  
        code to be executed if n==statement4;  
        break;  
    .....  
    default:  
        code to be executed if n != any case;
```


PHP | Estructuras de Selección

Ternario

```
(condition) ? if TRUE execute this : otherwise execute this;
```

PHP | Estructuras de Repetición

FOR

```
for (initialization expression; test condition; update expression) {  
    // code to be executed  
}
```

PHP | Estructuras de Repetición

WHILE

```
while (if the condition is true) {  
    // code is executed  
}
```

PHP | Estructuras de Repetición

DO WHILE

```
do {  
  
    //code is executed  
  
} while (if condition is true);
```

PHP | Estructuras de Repetición

FOREACH

```
foreach (array_element as value) {  
    //code to be executed  
}
```

PHP | Funciones

Sin parámetros

```
function function_name(){  
    executable code;  
}
```

Con Parámetros

```
function function_name($first_parameter, $second_parameter) {  
    executable code;  
}
```

PHP | Output

Ambas se usan para desplegar cadenas

- Echo puede desplegar múltiples cadenas.

```
echo "Hello,This is a display string example!";  
echo "Multiple ","argument ","string!";
```

- Print solamente puede desplegar 1 cadena.

```
print "Hello, world!";
```

PHP | Output

Tabla comparativa

	"echo"	"print"
type	This is a type of output string.	This is a type of output string.
parenthesis	Not written with parenthesis.	It can or can not be written with parenthesis.
strings	It can output one or more strings.	It can output only one string at a time, and returns 1.
Functionality	echo is faster than print.	print is slower than echo.
argument	echo(\$arg1[, \$arg2....])	print(\$arg)