

# Fundamentos de PHP

1.	Insertando Código PHP .....	2
1.1.	Caso 1 .....	2
1.2.	Caso 2 .....	2
1.3.	Caso 3 .....	3
1.4.	Caso 4 .....	3
2.	Instrucciones Básicas .....	4
2.1.	Separación de Instrucciones .....	4
2.2.	Comentarios .....	4
2.3.	Variables .....	5
3.	Impresión en el Navegador .....	6
3.1.	Echo .....	6
3.2.	Print() .....	6
3.3.	Printf .....	7
4.	Tipos de Datos .....	8
4.1.	Enteros .....	8
4.2.	Números de Punto Flotante .....	8
4.3.	Cadenas .....	9
4.4.	Averiguar el Tipo de una Variable .....	10
4.5.	Valor Lógico .....	10
5.	Constantes .....	11
6.	Expresiones y Operadores .....	13
6.1.	Operadores Aritméticos .....	13
6.2.	Operadores de Asignación .....	13
6.3.	Operadores de Comparación .....	14
6.4.	Operador de Ejecución .....	14
6.5.	Operadores de Incremento/Decremento .....	15
6.6.	Operadores Lógicos .....	15
6.7.	Operadores de Cadena .....	15

# 1. Insertando Código PHP

Los scripts basados en PHP están insertados en el código HTML, y para esto tenemos tres formas, que se describen a continuación.

## 1.1. Caso 1

### Sintaxis

```
<?
    // Aquí se inserta el script PHP
?>
```

### Ejemplo 1

```
<?
    Echo "<h1>Alianza Campeon</h1>";
    echo "<h1>Apertura - 2004</h1>";
?>
```

Disponible si se han habilitado las etiquetas cortas. Esto se puede hacer habilitando la opción de configuración `short_open_tag` en el archivo de configuración de PHP.

## 1.2. Caso 2

### Sintaxis

```
<?php
    // Aquí se inserta el script PHP
?>
```

### Ejemplo 2

```
<?php
    echo "Apache - PHP - MySQL<br>";
    echo "Una gran alternativa";
?>
```

## 1.3. Caso 3

### Sintaxis

```
<script language="php">  
    // Aquí se inserta el script PHP  
</script>
```

### Ejemplo 3

```
<script language="PHP">  
    echo "Maguma Studio<br>";  
    echo "Una gran herramienta";  
</script>
```

## 1.4. Caso 4

### Sintaxis

```
<% // Aquí se inserta el script PHP %>
```

### Ejemplo 4

```
<%  
    echo ("Podemos también usar etiquetas tipo ASP");  
%>
```

### Ejemplo 5

```
<% $cad = "Parece ASP"; %>  
<table border=1>  
    <tr>  
        <th>variable</th>  
        <th>valor</th>  
    </tr>  
    <tr>  
        <td>$cad</td>  
        <td><%= $cad; %></td>  
    </tr>  
</table>
```

Disponible sólo si se han habilitado las etiquetas tipo ASP usando la opción de configuración asp\_tags.

## 2. Instrucciones Básicas

### 2.1. Separación de Instrucciones

Las instrucciones se separan igual que en C o perl, terminando cada sentencia con un punto y coma.

La etiqueta de cierre (`</>`) también implica el fin de la sentencia, así el siguiente ejemplo 6.

#### Ejemplo 6

```
<?php
    echo "Esto es una prueba";
?>
```

Es equivalente al ejemplo 7.

#### Ejemplo 7

```
<?php
    echo "Esto es una prueba"
?>
```

### 2.2. Comentarios

PHP soporta comentarios tipo "C", "C++" y shell de Unix.

#### Ejemplo 8

```
<?php
    echo "ISL"; // Esto es un comentario tipo c++ para una línea
    /* Esto es un comentario multilínea
    otra línea más de comentario*/
    echo "<br>El mejor instituto";
    echo "<br>del PERU"; # Este es un comentario tipo shell
?>
```

## 2.3. Variables

Las variables en PHP no necesitan ser declaradas, podemos decir que en PHP las variables son débilmente tipadas.

Toda variable debe tener un nombre al que se le debe anteponer el símbolo \$, además el nombre debe empezar con una letra.

El ámbito de una variable es global a nivel del archivo actual y los archivos incluidos; dentro de una función son locales a la función.

### Ejemplo 9

```
<?php
    $nombre = "Gustavo Coronel";
    echo "Mi nombre es: ", $nombre;
?>
```

### Ejemplo 10

```
<?php
    $x = 20; # $x es un entero
    $x = "Viva el Perú"; # $x es un cadena
?>
```

## 3. Impresión en el Navegador

### 3.1. Echo

#### Sintaxis

```
echo (string arg1, string [argn]...)
```

#### Ejemplo 11

```
<?php
    echo "Hola Mundo<br>";
    echo "Esto se extiende
    por varias líneas. <br>También puedes
    insertar código HTML";
?>
```

### 3.2. Print()

#### Sintaxis

```
print (string arg)
```

#### Ejemplo 12

```
<?php
    print("Hola Campeones")
?>
```

### 3.3. Printf

#### Sintaxis

```
int printf (string formato [, mixed args...])
```

#### Ejemplo 13

```
<?php
    $A = 10; $B = 15.5; $C = "MySQL la BD más rápida";
    printf("El valor de A es: %d", $A);
    printf("<br>Ahora relleno con ceros: %03d", $A);
    printf("<br>El valor de B es: %f", $B);
    printf("<br>Ahora relleno con ceros: %01.2f", $B);
    printf("<br>El valor de C es: %s", $C);
?>
```

## 4. Tipos de Datos

### 4.1. Enteros

Los enteros se pueden especificar usando una de las siguientes sintaxis:

```
$a = 1234; # número decimal  
$a = -123; # un número negativo  
$a = 0123; # número octal (equivalente a 83 decimal)  
$a = 0x12; # número hexadecimal (equivalente a 18 decimal)
```

#### Ejemplo 14

```
<?php  
    $a1 = 1234; # número decimal  
    $a2 = -123; # un número negativo  
    $a3 = 0123; # número octal (equivalente al 83 decimal)  
    $a4 = 0x12; # número hexadecimal (equivalente al 18 decimal)  
    echo "a1-> ", $a1, "<br>";  
    echo "a2-> ", $a2, "<br>";  
    echo "a3-> ", $a3, "<br>";  
    echo "a4-> ", $a4, "<br>";  
?>
```

### 4.2. Números de Punto Flotante

Los números en punto flotante ("double") se pueden especificar utilizando cualquiera de las siguientes sintaxis:

```
$a = 1.234;  
$a = 1.2e3;
```

#### Ejemplo 15

```
<?php  
    $a1 = 1.234;  
    $a2 = 1.2e3;  
    echo "a1-> ", $a1, "<br>";  
    echo "a2-> ", $a2;  
?>
```



## 4.3. Cadenas

Las cadenas se especifican usando como delimitadores la doble comilla (").

### Ejemplo 16

```
<?php
    $cad = "Esto es una cadena";
    echo $cad;
?>
```

Podemos insertar código HTML como parte de la cadena.

### Ejemplo 17

```
<?php
    $cad = "<H1>Software Libre</H1>";
    echo $cad;
?>
```

Para concatenar dos cadenas se utiliza el operador punto (.).

### Ejemplo 18

```
<?php
    $s1 = "Software Libre<br>";
    $s2 = "es una buena alternativa<br>";
    $s3 = "para Desarrollar Soluciones<br>";
    $s4 = "para Desarrollar Soluciones";
    echo $s1.$s2.$s3.$s4;
?>
```

También podemos incluir variables dentro de la cadena.

### Ejemplo 19

```
<?php
    $n = 15;
    $cad = "Numero = $n";
    echo $cad;
?>
```

## 4.4. Averiguar el Tipo de una Variable

Una forma de determinar el tipo de una variable es utilizando la función `gettype()`.

### Ejemplo 20

```
<?php
    $nombre = "Claudia";
    $edad   = 20;
    echo "El tipo de \$nombre es: ", gettype($nombre), "<br>";
    echo "El tipo de \$edad es: ", gettype($edad);
?>
```

## 4.5. Valor Lógico

El valor lógico como tal no existe, el equivalente a *Falso* será un valor nulo, cero, arreglo ó cadena vacía; caso contrario se asume verdadero.

### Ejemplo 21

```
<?php
    $cad = "Alianza Campeon";
    if ($cad){
        echo "Es verdadero";
    }else{
        echo "Es falso";
    }
?>
```

## 5. Constantes

PHP define varias constantes y proporciona un mecanismo para definir más en tiempo de ejecución. Las constantes son como las variables, salvo por las dos circunstancias de que las constantes deben ser definidas usando la función *define()*, y que no pueden ser redefinidas con otro valor.

### Ejemplo 22

```
<?php
define("PI",3.141516);
$radio = 5;
$area = PI*$radio*$radio;
echo "PI: ",PI,"<br>";
echo "Radio: $radio<br>";
echo "Area: $area";
?>
```

Las constantes predefinidas siempre disponibles son:

<b>__FILE__</b>	El nombre del archivo de comandos que está siendo interpretado actualmente. Si se usa dentro de un archivo que ha sido incluido o requerido, entonces se da el nombre del archivo incluido, y no el nombre del archivo padre.
<b>__LINE__</b>	El número de línea dentro del archivo que está siendo interpretado. Si se usa dentro de un archivo incluido o requerido, entonces se da la posición dentro del archivo incluido.
<b>PHP_VERSION</b>	La cadena que representa la versión del analizador de PHP en uso.
<b>PHP_OS</b>	El nombre del sistema operativo en el cuál se ejecuta el analizador PHP.
<b>TRUE</b>	Valor verdadero.
<b>FALSE</b>	Valor falso.
<b>E_ERROR</b>	Denota un error distinto de un error de interpretación del cual no es posible recuperarse.
<b>E_WARNING</b>	Denota una condición donde PHP reconoce que hay algo erróneo, pero continuará de todas formas; pueden ser capturados por el propio archivo de comandos.
<b>E_PARSE</b>	El intérprete encontró sintaxis inválida en el archivo de comandos. La recuperación no es posible.

**E\_NOTICE**

Ocurrió algo que pudo ser o no un error. La ejecución continúa. Los ejemplos incluyen usar una cadena sin comillas como un índice "hash", o acceder a una variable que no ha sido inicializada.

**Ejemplo 23**

```

<?php
function report_error($file, $line, $message) {
    echo "<b>ERROR</b><br>";
    echo "<b>Archivo:</b> $file<br>";
    echo "<b>Linea:</b> $line<br>";
    echo "<b>Mensaje:</b> $message";
}
report_error(__FILE__, __LINE__, "Algo esta mal!!!");
?>

```

**Ejemplo 24**

```

<body>
<table border=1 width=400>
  <tr>
    <th>Constante</th>
    <th>Valor</th>
  </tr>
  <tr>
    <td>PHP_VERSION</td>
    <td><? Echo PHP_VERSION ?></td>
  </tr>
  <tr>
    <td>PHP_OS</td>
    <td><? Echo PHP_OS ?></td>
  </tr>
</table>
</body>

```

## 6. Expresiones y Operadores

Sin duda alguna que las expresiones constituyen la base de todo lenguaje, estas se construyen en base a operadores.

### 6.1. Operadores Aritméticos

Recordemos la aritmética básica de colegio.

Ejemplo	Nombre	Resultado
$\$a + \$b$	Adición	Suma de $\$a$ y $\$b$ .
$\$a - \$b$	Substracción	Diferencia entre $\$a$ y $\$b$ .
$\$a * \$b$	Multiplicación	Producto de $\$a$ and $\$b$ .
$\$a / \$b$	División	Cociente de $\$a$ entre $\$b$ .
$\$a \% \$b$	Módulo	Resto de $\$a$ dividido entre $\$b$ .

### 6.2. Operadores de Asignación

Operador	Ejemplo	Equivalente
=	$\$a = 7;$	
+=	$\$a += 3;$	$\$a = \$a + 3;$
-=	$\$a -= 3;$	$\$a = \$a - 3;$
*=	$\$a *= 3;$	$\$a = \$a * 3;$
/=	$\$a /= 3;$	$\$a = \$a / 3;$
%=	$\$a \% = 3;$	$\$a = \$a \% 3;$

## 6.3. Operadores de Comparación

Ejemplo	Nombre	Resultado
<code>\$a == \$b</code>	Igualdad	Cierto si \$a es igual a \$b.
<code>\$a === \$b</code>	Identidad	Cierto si \$a es igual a \$b y si son del mismo tipo (sólo PHP4 ó superior)
<code>\$a != \$b</code>	Desigualdad	Cierto si \$a no es igual a \$b.
<code>\$a &lt; \$b</code>	Menor que	Cierto si \$a es estrictamente menor que \$b.
<code>\$a &gt; \$b</code>	Mayor que	Cierto si \$a es estrictamente mayor que \$b.
<code>\$a &lt;= \$b</code>	Menor o igual que	Cierto si \$a es menor o igual que \$b.
<code>\$a &gt;= \$b</code>	Mayor o igual que	Cierto si \$a es mayor o igual que \$b.

Otro operador condicional es el operador "?:"(o ternario), que funciona como en C y otros muchos lenguajes.

```
(expr1) ? (expr2) : (expr3);
```

La expresión toma el valor *expr2* si *expr1* se evalúa a cierto, y *expr3* si *expr1* se evalúa a falso.

## 6.4. Operador de Ejecución

PHP soporta un operador de ejecución: el apóstrofe invertido (`). *¡No son apostrofes normales!* PHP intentará ejecutar la instrucción contenida dentro de los apóstrofes invertidos como si fuera un comando del shell; y su salida devuelta como el valor de esta expresión.

### Ejemplo 25

```
<?php
    $output = `dir`;
    echo "<pre>$output</pre>"
?>
```

## 6.5. Operadores de Incremento/Decremento

PHP soporta los operadores de predecremento y post incremento al estilo de C.

Ejemplo	Nombre	Efecto
++\$a	Preincremento	Incrementa \$a en uno y después devuelve \$a.
\$a++	Postincremento	Devuelve \$a y después incrementa \$a en uno.
--\$a	Predecremento	Decrementa \$a en uno y después devuelve \$a.
\$a--	Postdecremento	Devuelve \$a y después decrementa \$a en uno.

## 6.6. Operadores Lógicos

Ejemplo	Nombre	Resultado
\$a and \$b	Y	Cierto si tanto \$a como \$b son ciertos.
\$a or \$b	O	Cierto si \$a o \$b son ciertos.
\$a xor \$b	O exclusiva	Cierto si \$a es cierto o \$b es cierto, pero no ambos a la vez.
! \$a	Negación	Cierto si \$a no es cierto.
\$a && \$b	Y	Cierto si tanto \$a como \$b son ciertos.
\$a    \$b	O	Cierto si \$a o \$b son ciertos.

## 6.7. Operadores de Cadena

### 6.7.1. Operadores de Concatenación

Para concatenar dos cadenas utilizamos el operador *Punto* (.).

#### Ejemplo 26

```
<?php
    $cad1 = "PHP is ";
    $cad2 = "PowerFull.";
    echo $cad1.$cad2;
?>
```

### 6.7.2. Operador de Concatenación y Asignación

Este es el operador punto e igual (.=), y agrega a una cadena, otra cadena.

#### Ejemplo 27

```
<?php
    $cad = "Este es el equipo: ";
    $cad .= "Gustavo, Sergio, Hugo y Ricardo";
    echo $cad;
?>
```