



Variables PHP

En PHP las variables se representan con un signo de dólar seguido por el nombre de la variable. El nombre de la variable es sensible a minúsculas y mayúsculas.

Los nombres de variables siguen las mismas reglas que otras etiquetas en PHP. 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]*'
```

Nota: Para los propósitos de este manual, una letra es a-z, A-Z, y los bytes del 127 al 255 (0x7f-0xff).

Para más información sobre funciones relacionadas con variables, vea la [Referencia de Funciones de Variables](#).

```
<?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
?>
```

De forma predeterminada, las variables siempre se asignan por valor. Esto significa que cuando se asigna una expresión a una variable, el valor completo de la expresión original se copia en la variable de destino. Esto quiere decir que, por ejemplo, después de asignar el valor de una variable a otra, los cambios que se efectúen a una de esas variables no afectará a la otra. Para más información sobre este tipo de asignación, vea [Expresiones](#).

PHP también ofrece otra forma de asignar valores a las variables: [asignar por referencia](#). Esto significa que la nueva variable simplemente referencia (en otras palabras, "se convierte en un alias de" ó "apunta a") la variable original. Los cambios a la nueva variable afectan a la original, y viceversa.

Para asignar por referencia, simplemente se antepone un signo ampersand (&) al comienzo de la variable cuyo valor se está asignando (la variable fuente). Por ejemplo, el siguiente segmento de código produce la salida '**Mi nombre es Bob**' dos veces:



```
<?php
$foo = 'Bob';           // Asigna el valor 'Bob' a $foo
$bar = &$foo;           // Referenciar $foo vía $bar.
$bar = "Mi nombre es $bar"; // Modifica $bar...
echo $bar;
echo $foo;              // $foo también se modifica.
?>
```

Algo importante a tener en cuenta es que sólo las variables con nombre pueden ser asignadas por referencia.

```
<?php
$foo = 25;
$bar = &$foo;           // Esta es una asignación válida.
$bar = &(24 * 7);       // Inválida; referencia una expresión sin nombre.

function test()
{
    return 25;
}

$bar = &test();         // Inválido.
?>
```

No es necesario inicializar variables en PHP, sin embargo, es una muy buena práctica. Las variables no inicializadas tienen un valor predeterminado de acuerdo a su tipo dependiendo del contexto en el que son usadas - las booleanas se asumen como **FALSE**, los enteros y flotantes como cero, las cadenas (p.ej. usadas en [echo](#)) se establecen como una cadena vacía y las matrices se convierten en un array vacío.

Ejemplo #1 Valores predeterminados en variables sin inicializar

```
<?php
// Una variable no definida Y no referenciada (sin contexto de uso); imprime NUL
L
var_dump($variable_indefinida);

// Uso booleano; imprime 'false' (Vea operadores ternarios para más información
sobre esta sintaxis)
echo($booleano_indefinido ? "true\n" : "false\n");

// Uso de una cadena; imprime 'string(3) "abc"'
$cadena_indefinida .= 'abc';
var_dump($cadena_indefinida);

// Uso de un entero; imprime 'int(25)'
$int_indefinido += 25; // 0 + 25 => 25
var_dump($int_indefinido);
```

AULA MATRIZ

```
// Uso de flotante/doble; imprime 'float(1.25)'
$flotante_indefinido += 1.25;
var_dump($flotante_indefinido);

// Uso de array; imprime array(1) { [3]=> string(3) "def" }
$array_indefinida[3] = "def"; // array() + array(3 => "def") => array(3 => "def"
)
var_dump($array_indefinida);

// Uso de objetos; crea un nuevo objeto stdClass (vea http://www.php.net/manual/
en/reserved.classes.php)
// Imprime: object(stdClass)#1 (1) { ["foo"]=> string(3) "bar" }
$objeto_indefinido->foo = 'bar';
var_dump($objeto_indefinido);
?>
```

Depender del valor predeterminado de una variable sin inicializar es problemático al incluir un archivo en otro que use el mismo nombre de variable. También es un importante riesgo de seguridad cuando la opción register_globals se encuentra habilitada. Un error de nivel E_NOTICE es emitido cuando se trabaja con variables sin inicializar, con la excepción del caso en el que se anexan elementos a un array no inicializado. La construcción del lenguaje isset() puede ser usada para detectar si una variable ya ha sido inicializada.