# Etiquetas

## <?php

Etiqueta de apertura

## <?=

Etiqueta echo abreviada.

## ?>

Etiqueta de cierre

**NOTA:**

Se recomienda que si el archivo solo tiene código PHP se omita la etiqueta de cierre.

## Bloques de HTML condicionados

### <?php if ($expresión == true): ?>

### <?php else: ?>

### <?php endif; ?>

# Comentarios

## //

Una línea

## #

Una línea

## /\* \*/

Múltiple línea

# Variables

Para declarar variables debemos iniciar el nombre con el signo de pesos ($).

Ejemplo:

$intVar = 12;

Para imprimir valor de variable en body utilizamos **echo.**

Ejemplo:

echo $intVar;

Se pueden especificar explícitamente el final del nombre de una variable.

Ejemplo:

echo “${strVar}s”

# Tipos de datos

En PHP no se tiene que especificar el tipo de dato de una variable, pero si podemos saber qué tipo de dato contiene una variable con la función **var\_dump($variable)** o **gettype($variable).**

Ejemplos:

var\_dump($intVar); **//Salida int(12)**

gettype($strVar); **//Salida string**

Si queremos cambiar de tipo de dato alguna variable utilizamos settype($var, ‘tipo\_de\_dato’);

Ejemplo:

settype($boolVarTrue, ‘string’); **//Cambia la variable a string con valor 1**

Para validar un tipo de dato utilizamos la función is\_int, is\_string, is\_bool, is\_float, is\_array, is\_null, etc.

Ejemplo:

if (is\_int($intVar)) {

//Todo

}

Podemos hacer cast de valor en una variable a otro tipo.

Ejemplo:

if ((bool) 2) {

//Todo

}

## Strings

Si utilizamos comillas simples se considerará lo que contenga la variable como simple texto, si utilizamos comillas dobles podremos utilizar variables dentro del string y PHP contemplará su valor en el texto.

Ejemplo:

$intVar = 5;

$strVar = ‘Hola $intVar’;

echo $strVar; **//Salida “Hola $intVar”**

$strVar = “Hola $intVar”;

echo $strVar; **//Salida “Hola 5”**

$strVar = <<<”STR”

Hola $intVar

“STR”

**//Salida “Hola 5”**

## Arreglos

En PHP todos los arreglos son en realidad mapas, con una llave y un valor, si no especificamos la llave, se asigna por default un número a partir del 0.

NOTA: Un arreglo puede contener llaves explícitas de diferente tipo e/o implícitas.

NOTA: Puede ponerse una coma al final del último elemento.

Ejemplo 1:

$arrayVar = [

‘red’,

‘yellow’,

‘blue’

]

$arrayVar[1] **//Salida ‘yellow’**

Ejemplo 2:

$arrayVar = array(

1 => ‘red’,

3 => ‘yellow’,

‘cuatro’ => ‘blue’,

‘green’, **//Se asignará a la llave numérica 4**

);

$arrayVar[4] **//Salida ‘green’**

Ejemplo 3:

$arrayVar = array();

$arrayVar[1] = ‘red’;

$arrayVar[3] = ‘yellow’;

$arrayVar[3.5] = ‘amarillo’; **//Reemplazará la llave 3**

$arrayVar[‘cuatro’] = ‘blue’;

$arrayVar[] = ‘green’; **//Se asignará a la llave numérica 4**

Puede haber arreglos multidimensionales:

Ejemplo:

$array = array(  
 "foo" => "bar",  
 42 => 24,  
 "multi" => array(  
 "dimensional" => array(  
 "array" => "foo"

)

)  
);

$array["multi"]["dimensional"]["array"] **//Salida ‘foo’**

Para recorrer un arreglo se puede utilizar la función foreach:

Ejemplo:

foreach ($arrayVar as $color) {

echo “El color es $color\n”;

}

Para recorrerlo y pasar por referencia el elemento del arreglo para modificarlo:

Ejemplo:

foreach ($arrayVar as &$color) {

$color = strtoupper($color);

}

unset($color);

print\_r($colors);

## Funciones

<http://php.net/manual/es/funcref.php>

# Formularios

## $\_POST[‘fieldName’]

Para obtener el valor de un campo en el formulario vía POST.

## $\_GET[‘fieldName’]

Para obtener el valor de un campo en el formulario vía GET.

## $\_REQUEST[‘fieldName’]

Para obtener el valor de un campo en el formulario vía POST, GET o COOKIE.

## htmlspecialchars($\_POST[‘fieldName’])

Para asegurarse que cualquier carácter que sea especial en HTML se codifique adecuadamente, de manera que nadie pueda inyectar etiquetas HTML o Javascript en la página.