

PHP

# Comentarios

```
< html>           # Comentario al estilo Unix
< head>
< title> COMENTARIOS< / title>
< / head>
< body>
< h1
align= "center"> COMENTARIOS< / h1> < hr>
<?
// Comentario de una linea
$saludo= "La casa de Juanito";
echo $saludo;
/*
Comentario de varias lineas
$saludo= "La casa de Juanito";
echo $saludo;
*/
?>
</body>
</ html>
```

# Arrays (Arreglos)

Son estructuras que permiten el almacenamiento de un conjunto de datos bajo un mismo nombre.

En php los elementos que componen el arreglo pueden ser de distintos tipos.

El primer elemento del arreglo tiene el índice 0.

# arreglos.php

```
< html>
< head>
< title> ARREGLOS< / title>
< / head>
< body>
< h1
align= "center"> ARREGLOS< / h1> < hr>
<?
    echo "El arreglo se llama
\ $producto< br> < br> ";
    /* Se definen los elementos del arreglo */
    $producto[0] = 1;
    $producto[1] = "chiles";
    $producto[2] = "herdez";
    $producto[3] = 15;
    $producto[4] = "";
    $producto[5] = 7.89;
echo "< table width= \ "80% \ "
    border= '1'> < tr> ";

    echo "< td> IDENTIFICADOR< / td> ";
    echo "< td> NOMBRE< / td> ";
    echo "< td> MARCA< / td> ";
    echo "< td> EN EXISTENCIA< / td> ";
    echo "< td> INFO.
ADICIONAL< / td> ";
    echo "< td> COSTO< / td> ";
    echo "< / tr> ";
    echo "< tr> ";
    echo "< td> $producto[0]< / td> ";
    echo "< td> $producto[1]< / td> ";
    echo "< td> $producto[2]< / td> ";
    echo "< td> $producto[3]< / td> ";
    echo "< td> $producto[4]< / td> ";
    echo "< td> $producto[5]< / td> ";
    echo "< / tr> ";
    echo "< / table> ";

?>
< / body>
< / html>
```

# Arrays Asociativos

Son arreglos especiales en los que el índice es un valor de tipo string, de modo que cada elemento del arreglo está definida por el par (clave, valor) ó también llamado par (key,value).

Son muy usados en PHP.

# arreglosasociativos.php

```
< html>
< head>
< title> ARREGLOS ASOCIATIVOS< / title>
< / head>
< body>
< h1 align= "center"> ARREGLOS
ASOCIATIVOS< / h1> < hr>
<?
    echo "El arreglo se llama
\ $producto< br> < br> ";
    / * Se definen los elementos del arreglo
asociativo */
    $producto['id'] = 1;
    $producto['nombre'] = "chiles";
    $producto['marca'] = "herdez";
    $producto['existencia'] = 15;
    $producto['informacion'] = "";
    $producto['costo'] = 7.89;
    echo "< table width= \ "80% "
        \
border= '1'> < tr> ";
    echo "< td> IDENTIFICADOR< / td> ";
    echo "< td> NOMBRE< / td> ";
```

```
        echo "< td> MARCA< / td> ";
        echo "< td> EN EXISTENCIA< / td> ";
        echo "< td> INFO.
ADICIONAL< / td> ";
        echo "< td> COSTO< / td> ";
        echo "< / tr> ";
        echo "< tr> ";
        echo "< td> " . $producto['id'] .
"< / td> ";
        echo "< td> " . $producto['nombre'] .
"< / td> ";
        echo "< td> " . $producto['marca'] .
"< / td> ";
        echo "< td> " . $producto
['ex istencia'] . "< / td> ";
        echo "< td> " . $producto
['informacion'] . "< / td> ";
        echo "< td> " . $producto['costo'] .
"< / td> ";
        echo "< / tr> ";
        echo "< / table> ";
    ?>
< / body>
< / html>
```

# Object (Objetos)

Es una estructura que define características propias (denominadas propiedades) y sus funcionalidades (denominadas métodos).

# CONVERSIÓN AUTOMÁTICA DE TIPOS

Cuando operamos con variables de distinto tipo, el intérprete de PHP tiende a homogeneizar sus diferentes tipos en función de la operación que se pretende realizar.



# conv\_auto.php

```
< html>
< head>
< title> CONVERSIÓN AUTOMÁTICA DE TIPOS< / title>
< / head>
< body>
< h1 align= "center"> CONVERSIÓN AUTOMÁTICA DE TIPOS< / h1> < hr>
<?
$num= 15;
$cadena1= "23";
$cadena2= "250aux";
$cadena3= "1.245";
$cadena4= "hola";
$suma= $num + $cadena1;
echo 'Sumando $num + $cadena1 = ' . $suma . "< br> ";
$suma= $num + $cadena2;
echo 'Sumando $num + $cadena2 = ' . $suma . "< br> ";
$suma= $num + $cadena3;
echo 'Sumando $num + $cadena3 = ' . $suma . "< br> ";
$suma= $num + $cadena4;
echo 'Sumando $num + $cadena4 = ' . $suma . "< br> ";
?>
< / body>
< / html>
```

# CONVERSIÓN DE TIPOS

Es posible convertir las variables a un tipo específico si se desea, es el llamado casting en C o conversión de tipo.

Se escribe entre paréntesis el tipo deseado antes de la variable, que puede ser integer, float, string, boolean, array, object).

Ejemplo:

```
(float)$micadena;
```

# casting.php

```
< html>
< head>
< title> CONVERSIÓN DE TIPOS (CASTING)< / title>
< / head>
< body>
< h1 align= "center"> CONVERSIÓN DE TIPOS (CASTING)< / h1> < hr>
<?
$cadena2= "250.789 es un buen número";
$cast= (integer)$cadena2;
echo 'Contenido de $cast : ' . $cast . "< br> ";
$cast= (float)$cadena2;
echo 'Contenido de $cast : ' . $cast . "< br> ";
$cast= (boolean)$cadena2;
echo 'Contenido de $cast : ' . $cast . "< br> ";
$cast= (array)$cadena2;
echo 'Contenido de $cast : ' . $cast . "< br> ";
echo "$cast[0]< br> ";
$cast= (object)$cadena2;
echo 'Contenido de $cast : ' . $cast . "< br> ";
?>
< / body>
< / html>
```

# CONSTANTES

Una constante es una variable que mantiene el mismo valor durante toda la ejecución del programa.

Constantes predefinidas de PHP:

PHP\_VERSION

PHP\_OS

TRUE

FALSE

Sintaxis de definición de constantes:

```
define("constante",valor)
```

# constantes.php

```
< html>
< head>
< title> CONSTANTES< / title>
< / head>
< body>
< h1 align= "center"> CONSTANTES< / h1> < hr>
<?
define("DOLAR",11.47);
define("euro",15.02);
echo "Valor de dolar " . DOLAR . "< br> ";
echo "Valor de euro " . euro . "< br> ";
echo "Estamos trabajando con : < br> ";
echo "Version de PHP : " . PHP_VERSION . "< br> ";
echo "Sistema Operativo : " . PHP_OS . "< br> ";
echo TRUE . "< br> ";
echo FALSE . "< br> ";
?>
< / body>
< / html>
```

# OPERADORES

## ARITMÉTICOS:

+ \$a + \$b

- \$a - \$b

\* \$a \* \$b

/ \$a / \$b

% \$a % \$b

# OPERADORES

## DE ASIGNACIÓN:

= \$a = \$b

+ = \$a + = \$b < = > \$a = \$a + \$b

- = \$a - = \$b

\* = \$a \* = \$b

/ = \$a / = \$b

% = \$a % \$b =

.= \$a .= \$b

# OPERADORES

## DE CADENAS:

- . Operador de concatenación de 2 cadenas.
- .= Concatenación y asignación.



# operadores1.php

```
< html>
< head>
< title> OPERADORES 1< / title>
< / head>
< body>
< h1 align= "center"> OPERADORES
1< / h1> < hr>
<?
$var1= 2;
$var2= 4;
$var3= 5;
$suma= $var1+ $var2;
$concatena= 'La suma de $var1 + $var 2 =
' . $suma . "< br> ";
$suma+= $var3;
$concatena.= 'El resultado acumulado de
$suma = ' . $suma . "< br> ";
$resta= $var1- $var2;
$concatena.= 'La resta de $var1 - $var 2 =
' . $resta . "< br> ";
$resta- = $var3;
$concatena.= 'El resultado acumulado de
$resta = ' . $resta . "< br> ";
```

```
$mult= $var1*$var2;
$concatena.= 'La multiplicación de $var1 *
$var 2 = ' . $mult . "< br> ";
$mult*= $var3;
$concatena.= 'El resultado acumulado de
$mult = ' . $mult . "< br> ";
$div= $var1/ $var2;
$concatena.= 'La división de $var1 / $var 2
= ' . $div . "< br> ";
$div/= $var3;
$concatena.= 'El resultado acumulado de
$div = ' . $div . "< br> ";
$mod= $var1% $var2;
$concatena.= 'El módulo de $var1 % $var 2
= ' . $mod . "< br> ";
$mod% $var3;
=
$concatena.= 'El resultado acumulado de
$mod = ' . $mod . "< br> ";
echo $concatena;
?>
< / body>
< / html>
```

# OPERADORES

## DE INCREMENTO Y DECREMENTO.

++	++	\$a incrementa \$a en uno y después devuelve \$a
\$a++		devuelve \$a y después lo incrementa
--	--	\$a predecremento
\$a--		postdecremento

# pre\_post.php

```
< html>
< head>
< title> Pre- incremento y Post- incremento
</title>
< /head>
< body>
< h1 align= center> Incrementos< / h1> <
  hr>
<?
$variable= 5;
echo "Postincremento< br> ";
echo "$variable< br> ";
echo $variable+ + ."< br> ";
echo "$variable< br> < br>
```

```
$variable= 5;
echo "Preincremento< br> ";
echo "$variable< br> ";
echo + + $variable."< br> ";
echo "$variable< br> ";
?>
< / body>
< / html>
```

# OPERADORES

DE COMPARACIÓN : Se utilizan para comparar expresiones, el resultado es true o false, se utilizan en la toma de decisiones.

`== $a == $b`

`!= $a != $b`

`=== $a === $b` Idénticos(iguales y del mismo tipo)

`!== $a !== $b`

`< $a < $b`

`> $a > $b`

`<= $a <= $b`

`>= $a >= $b`

# OPERADORES

A NIVEL DE BIT : Realizar operaciones a nivel de bits, es decir con ceros y unos.

&    \$a & \$b    AND

|    \$a | \$b    OR

^    \$a ^ \$b    XOR

~    ~ \$b    CA1

<<    \$a < < \$b    DESP. IZQ.

>>    \$a > > \$b    DESP. DER.

# OPERADORES

**LÓGICOS** : Se utilizan con expresiones que devuelven valores lógicos, con estos operandos se pueden combinar varias expresiones y evaluarlas en una sola expresión.

&& ó and    \$a && \$b TRUE si a y b son TRUE

|| ó or      \$a || \$b TRUE si a ó b son TRUE

!            !a      Niega el valor lógico

x X-or      \$a x or \$b TRUE si alguno es TRUE pero no los 2 a la vez

# PRECEDENCIA DE OPERADORES

De mayor a menor precedencia:

! ~ + + - - (int) (double) (string) (array) (object) ( )

\* / %

+ - .

<< >>

< <= > >=

= = != == ==

&

^

|

&&

||

?:

= + = - = \*= / = . = % &= |= ^ = ~= < < = > > =

=

and

x or

or





# ESTRUCTURAS DE CONTROL

Las estructuras de control o sentencias de control nos permiten modificar el flujo de ejecución de un programa,

permitiéndonos que la ejecución no tenga que ser secuencial, sino que nos permite bifurcar el flujo del programa (estructuras condicionales) o que determinado código se ejecute un determinado número de veces (estructuras cíclicas).

# ESTRUCTURAS CONDICIONALES

Son estructuras que nos permiten elegir diferentes caminos de ejecución, cuando se cumple una determinada condición.

En PHP existen 2:

if

switch

# IF

Sintaxis:

```
if(expresión){  
    sentencias;  
}
```

NOTA : expresión debe de ser una expresión lógica, es decir que devuelva verdadero o falso.

# if.php

```
< html>
< head>
< title> IF< / title>
< / head>
< body>
< h1
align= "center"> IF< / h1> < hr
>
<?
if(4> 2){
echo "4 es mayor que
2< br> ";
}
$a= 3;
$b= 9;
$c= 6;
```

```
echo "Los tres numeros para comparar
son : \ $a = $a,\ $b= $b y \ $c = $c< br> ";
if(($a> $b) && ($a> $c)){
echo "El mayor es $a";
}
if(($b> $a) && ($b> $c)){
echo "El mayor es $b";
}
if(($c> $a) && ($c> $b)){
echo "El mayor es $c";
}
?>
< / body>
< / html>
```

# ifanidados.php

```
< html>
< head>
< title> IF ANIDADOS< / title>
< / head>
< body>
< h1 align= "center"> IF
ANIDADOS< / h1> < hr>
<?
$a= 3;
$b= 9;
$c= 6;
echo "Los tres numeros para comparar son : \ $a = $a,
  \ $b= $b y \ $c = $c< br> ";
if($a> $b){      if($a> $c){
echo "El mayor es $a";
}
}
```

```
if($b> $a){
    if($b> $c){
        echo "El mayor es $b";
    }
}
if($c> $a){
    if($c> $b){
        echo "El mayor es $c";
    }
}
?>
< / body>
< / html>
```

# IF ... ELSE

Sintaxis:

```
if(expresión){  
    sentencias;  
}  
else{  
    sentencias;  
}
```

# ifelse.php

```
< html>
< head>
< title> IF ELSE< / title>
< / head>
< body>
< h1 align= "center"> IF ELSE< / h1> < hr>
<?
$a= 3;
$b= 9;
$c= 6;
echo "Los tres numeros para comparar son : \ $a = $a, \ $b= $b y \ $c = $c< br> ";
if($a> $b)
    $mayor= $a;
else
    $mayor= $b;
if($mayor< $c)
    $mayor= $c;
echo "El mayor es $mayor";
?>
< / body>
< / html>
```

# IF ... ELSEIF

Sintaxis:

```
if(expresión){  
    sentencias;  
}  
elseif(expresión){  
    sentencias;  
}  
else{  
    sentencias;  
}
```



# elseif.php

```
< html>
< head>
< title> ELSE IF< / title>
< / head>
< body>
< h1 align= "center"> ELSE IF< / h1> < hr>
<?
$calificacion= 6;
if($calificacion> = 9){
echo "Felicidades tienes MB";
}
elseif($calificacion< 9 && $calificacion> = 8){
echo "Bien hecho, tienes B";
}
elseif($calificacion< 8 && $calificacion> = 6){
echo "Apenas, tienes S";
}
else{
echo "Estas reprobado";
}
?>
< / body>
< / html>
```

# IF COMPACTO

Sintaxis:

< expresión1 > ? < expresión2 > :  
expresión3 >

expresión1 es la condición lógica (true or false)

expresión2 se devuelve si es true

expresión3 se devuelve si es false

# ifcompacto.php

```
< html>
< head>
< title> IF COMPACTO< / title>
< / head>
< body>
< h1 align= "center"> IF COMPACTO< / h1> < hr>
<?
$a= 3;
$b= 9;
$c= 6;
echo "Los tres numeros para comparar son : \ $a = $a, \ $b= $b y \ $c = $c< br> ";
($a> $b)?$mayor= $a:$mayor= $b;
$mayor= ($mayor< $c)?$c:$mayor;
echo "El mayor es $mayor< br> ";
$dia= date(w);
echo "$dia< br> ";
echo($dia= 0 || $dia= 6)"Es fin de semana":"Es un dia entre semana";
?>
< / body>
< / html>
```

# SWITCH

Se utiliza para comparar un dato con un conjunto de posibles valores.

Sintaxis:

```
switch($variable){  
  case valor1:  
    sentencias;  
    break;  
  case valor2:  
    sentencias;  
    break;  
  case valorN:  
    sentecnia;  
    break;  
  default:  
    sentencias;  
}
```

# switch.php

```
< html>
< head>
< t itle> SWITCH< / title>
< / head>
< body>
< h1 align= "center"> SWITCH< / h1> < hr>
<?
$idia= date(n);
$smes= "";
swit ch($idia){
case 1:
$smes = "Enero";
break;
case 2:
$smes = "Febrero";
break;
case 3:
$smes = "Marzo";
break;
case 4:
$smes = "Abril";
break;
case 5:
$smes = "Mayo";
break;
case 6:
$smes = "Junio";
break;
```

```
case 7:
$smes = "Julio";
break;
case 8:
$smes = "Agosto";
break;
case 9:
$smes = "Septiembre";
break;
case 10:
$smes = "Octubre";
break;
case 11:
$smes = "Noviembre";
break;
case 12:
$smes = "Diciembre";
break;
default:
$smes= "No es una entrada válida";
}
echo "Estamos en el mes de $smes";
?>
< / body>
< / html>
```

# Practica SWITCH

Crear un formulario que permita al usuario seleccionar la imagen de fondo de una página, guardar el script con el nombre de fondo.php

# ESTRUCTURAS CÍCLICAS

Se utilizan para ejecutar una o mas instrucciones un determinado numero de veces, generalmente se utilizan para contar o para recorrer los elementos de un arreglo.

En PHP existen 4 tipos:

for

foreach

while

do while

# FOR

Nos permite realizar un conjunto de instrucciones un determinado número de veces.

Sintaxis:

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



# for1.php

```
< html>
< head>
< title> FOR< / title>
< / head>
< body>
< h1 align= "center"> FOR< / h1> < hr>
<?
echo "< center> Despliega un listado del 1 al 10< / center> ";
for($i= 1;$i< = 10;$i+ + ){
    echo "$i< br> ";
}
echo "< center> Despliega la suma del 1 al 10< / center> ";
$acumula= 0;
for($i= 0;$i< = 10;$i+ + )
    $acumula+ = $i;
echo "La suma del 1 al 10 es : $acumula";
?>
< / body>
< / html>
```

# for2.php

```
< html>
< head>
< title> FOR 2< / title>
< / head>
< body>
< h1 align= "center"> FOR 2< / h1> < hr>
<?
echo "< center> Tabla del 7< / center> ";
echo "< center> ";
for($i= 1,$y= 7;$i< = 10;$i+ + ){
    echo "$i X $y = " . $i * $y . "< br> ";
}
echo "< / center> ";
?>
< / body>
< / html>
```

# for3.php

```
< html>
< head>
< t itle> FOR PARA RECORRER UN ARREGLO< / title>
< / head>
< body>
< h1 align= "center"> FOR PARA RECORRER UN ARREGLO< / h1> < hr>
<?
$producto[0] = 1;
$producto[1] = "chiles";
$producto[2] = "herdez";
$producto[3] = 15;
$producto[4] = "";
$producto[5] = 7.89;
?>
< t able border= "1" align= "center"> < tr> < t d> ID< / td> < td> NOMBRE< / td> < td> MARCA< / td>
< t d> EXISTENCIA< / td> < td> INFO. ADICIONAL< / td> < td> COSTO< / td> < / tr> < tr>
<?
for($icontador= 0;$icontador< = 5;$icontador+ + )
echo "< td> ".$producto[$icontador]."< / t d> ";
?>
< / tr> < / table>
< / body>
< / html>
```

# Practicas FOR

Generar el ejercicio de etiquetas HTML usando ciclo for para presentar las características dentro de una tabla. (Utilizar un arreglo asociativo y recorrerlo.

Generar una tabla con las características de css utilizando un formulario utilizando un combobox.

# FOREACH

Se utiliza para recorrer las estructuras de tipo arreglo, obteniendo en cada iteración uno de sus elementos componentes.

Tiene 2 sintaxis:

Generalmente para arreglos:

```
foreach(nombre_arreglo as $valor){  
    sentencias;  
}
```

Generalmente para arreglos asociativos:

```
foreach(nombre_arreglo as $clave => $valor){  
    sentencias;  
}
```

# foreach.php

```
< html>
< head>
< t itle> FOR EACH PARAARREGLOS< / title>
< / head>
< body>
< h1 align= "center"> FOR EACH PARAARREGLOS< / h1> < hr>
<?
$product o[0] = 1;
$product o[1] = "chiles";
$product o[2] = "herdez";
$product o[3] = 15;
$product o[4] = "";
$product o[5] = 7.89;
?>
< t able border= "1" align= "center"> < tr> < t d> ID< / td> < td> NOMBRE< / td> < td> MARCA< / td>
< t d> EXISTENCIA< / td> < td> INFO. ADICIONAL< / td> < td> COSTO< / td> < / tr> < tr>
<?
foreach($producto as $value)
echo "< td> $value< / t d> ";
?>
< / tr> < / table>
< / body>
< / html>
```

# foreach2.php

```
< html>
< head>
< title> FOR EACH PARA ARREGLOS ASOCIATIVOS< / title>
< / head>
< body>
< h1 align= "center"> FOR EACH PARA ARREGLOS ASOCIATIVOS< / h1> < hr>
<?
$producto['ID'] = 1;
$producto['NOMBRE'] = "chiles";
$producto['MARCA'] = "herdez";
$producto['EXISTENCIA'] = 15;
$producto['INFORMACION ADICIONAL'] = "Producto Agotado";
$producto['COSTO'] = 7.89;
?>
< table border= "1" align= "center"> < tr>
<?
foreach($producto as $key => $value)
echo "< td> $key = $value< / td> < / tr> ";
?>
< / table>
< / body>
< / html>
```

# WHILE

Se ejecuta un número indeterminado de veces, siempre y cuando el resultado de comprobar la condición sea verdadero.

Sintaxis:

```
while(condición){  
    sentencias;  
}
```



# while.php

```
< html>
< head>
< title> WHILE< / title>
< / head>
< body>
< h1 align= "center"> WHILE< / h1> < hr>
<?
$contador= 1;
$iacumulador= 0;
while($contador< = 10){
$iacumulador+ = $contador;
$contador+ + ;
}
echo "El resultado de la suma del 1 al 10 es : $iacumulador";
?>
< / body>
< / html>
```

# Practica WHILE

Realizar un script que genere las tablas con los ejercicios de HTML y CSS con arreglos asociativos, se debe de proporcionar una interfaz al usuario para permitirle escoger que tabla se generará.

# do while

Es lo mismo que un ciclo while, la única diferencia es que por lo menos se ejecuta una vez, ya que la condición se evalúa al final del ciclo.

Sintaxis:

```
do {  
    sentencias;  
} while(condición);
```

# break y continue

**break** : Se utiliza para forzar la terminación de un ciclo, o en el caso del switch para que no se sigan evaluando los case.

**Continue** : Se utiliza dentro de los ciclos, cuando queremos que no se efectuen una serie de instrucciones del ciclo y queremos pasar a la siguiente iteración

# break.php

```
< html>
< head>
< title> BREAK< / title>
< / head>
< body>
< h1 align= "center"> BREAK< / h1> < hr>
<?
for($icontador= 0;$icontador< = 15;$icontador+ + ){
if($icontador= = 7)
break;
else
echo "$icontador< br> ";
}
?>
< / body>
< / html>
```

# continue.php

```
< html>
< head>
< title> CONTINUE< / title>
< / head>
< body>
< h1 align= "center"> CONTINUE< / h1> < hr>
<?
for($icontador= 0;$icontador< = 15;$icontador+ + ){
if($icontador= = 7 || $icontador= = 9 || $icontador= = 13)
continue;
else
echo "$icontador< br> ";
}
?>
< / body>
< / html>
```

# practica BREAK y CONTINUE

Crear un Script que permita seleccionar las etiquetas HTML por criterios de sus atributos y las despliegue en una tabla.