PHP

Estructuras de control

Estructuras de selección

if

Las estructura if es una estructura condicional que permite ejecutar una conjunto de instrucciones con base en la condición evaluada. Su sintaxis es:

```
if (condición_verdadera) {
    // código a ejecutar
}
```

if-else

Las estructura if-else es el complemento a la estructura if, se evalúa la condición y, si ésta es correcta ejecuta las instrucciones del bloque if, de lo contrario (si la condición es falsa) ejecuta las instrucciones del bloque else. Su sintaxis es:

```
if (condición_verdadera) {
    // código a ejecutar
    // si la condición es verdadera
} else{
    // código a ejecutar
```

Operadores lógicos

Nombre	Ejemplo	Resultado
AND (y)	$a and $b	TRUE si tanto $a como $b son TRUE.
OR (o inclusivo)	$a or $b	TRUE si cualquiera de $a o $b es TRUE.
XOR (o exclusivo)	$a xor $b	TRUE si $a o $b es TRUE, pero no ambos.
NOT (no)	!$a	TRUE si $a no es TRUE.
AND (y)	$a && $b	TRUE si tanto $a como $b son TRUE.
OR (o inclusivo)	$a ‖ $b	TRUE si cualquiera de $a o $b es TRUE.

Operadores de relación

Nombre	Ejemplo	Resultado
Igualdad	$a == $b	TRUE si $a es igual a $b.
Idéntico	$a === $b	TRUE si $a es igual a $b, y son del mismo tipo. (a partir de PHP 4).
Diferente	$a != $b	TRUE si $a no es igual a $b.
No idénticos	$a !== $b	TRUE si $a no es igual a $b, o si no son del mismo tipo. (a partir de PHP 4).
Menor que	$a < $b	TRUE si $a es estrictamente menor que $b.
Mayor que	$a > $b	TRUE si $a es estrictamente mayor que $b.
Menor o igual que	$a <= $b	TRUE si $a es menor o igual que $b.
Mayor o igual que	$a >= $b	TRUE si $a es mayor o igual que $b.

ej1.php

```
<?php
	$num=5;
	if (($num%2)==0){
		print("El número es par<br>");
	}else{
		print("El número es impar<br>");
	}
?>
```

switch-case

Las estructura de selección switch-case permite ejecutar varios segmentos de código dependiendo de la variable a evaluar. Su sintaxis es:

```
switch($var) {
    case ”valor1":   // Bloque 1
        // código a ejecutar
        break;
    case ”valor2":   // Bloque 2
        // código a ejecutar
```

ej2.php

```
<?php
	$caracter="be";
	switch($caracter){
		case "a":
			print("Se eligiO A");
			break;
		case "be":
			print("Se eligiO B");
			break;
		case "ce":
```

Operador ternario

El operador ternario es una estructura similar a if-else, pero en una sóla línea. Se evalúa la condición, si es verdadera ejecuta la primera parte del código, de lo contrario ejecuta la segunda parte. Su sintaxis es:

condición ? Verdadera : falsa;

ej3.php

```php
<?php
	$a=-6;
	$b = $a>0 ? $a : $a*-1;
	printf("El valor absoluto de %d es: %d", $a, $b);
?>
```

Estructuras de repetición

while

El ciclo while es un bucle de control que se utiliza para ejecutar un conjunto de instrucciones varias veces hasta que la condición lógica se cumpla. Su sintaxis es la siguiente:

```
while (condicion_logica) {
    // bloque de código a ejecutar
    // hasta que se cumpla la condición
    // lógica
}
```

ej4.php

```
<?PHP
	$tam=1;
	while ($tam<=7) {
		echo "<font size=$tam>Tamaño $tam</font><br>";
		$tam++;
	}
?>
```

each y list

Es posible recorrer un arreglo utilizando las funciones each() y list(). La función list(), almacena en los parámetros ($i, $valor), que son el índice y el valor devuelto por la función each(), que tiene como parámetro el arreglo a recorrer.

each y list

Además, la función list(), avanza automáticamente el apuntador al siguiente elemento del arreglo y, cuando se haya llegado al final del arreglo, la función devuelve false y sale del ciclo.

ej5.php

```
<?PHP
    $materia[100] = "Economía";
    $materia[2080] = "Administración";
    $materia[1450] = "Derecho";
    $materia[350] = "Ciencias Sociales";
    $materia[1220] = "Teología";

    echo "<H2>"."Lista de materias";
    echo "<H3>"."<HR>";
    while (list($i,$valor)=each($materia)){
```

ej6.php

```
<?PHP
    $ciudad = array("Par" => "Paris",
                    "Lon" => "Londres",
                    "Ate" => "Atenas",
                    "Ber" => "Berlin",
                    "Lim" => "Lima");

    echo "<H2>Ciudades</H2>";
    while (list($i,$valor)=each($ciudad)) {
        echo "Identificador: " . $i . "<br>";
```

do-while

El ciclo do-while es un bucle de control que se utiliza para ejecutar un conjunto de instrucciones varias veces hasta que la condición lógica se cumpla. Su sintaxis es la siguiente:

```
do{
    // bloque de código a ejecutar
} while (condicion_logica)
```

ej7.php

```
<?PHP
	$n=100;
	  echo "Múltiplos de 7 entre [100, 400]<br>
<hr>";
	do {
		if ($n%7 == 0) {
			echo "$n - ";
		}
		$n++;
	} while ($n<400);
```

Funciones next y prev

En PHP un arreglo puede tener elementos en cualquier posición. Otra forma para recorrer los elementos de un arreglos es mediante las funciones next() y prev().

ej8.php

```
<?PHP
    $lista[10] = "Marcelo";
    $lista[20] = "Alicia";
    $lista[15] = "Alejandra";
    $lista[35] = "Mario";
    $lista[12] = "Alberto";

    reset($lista);
    echo ("<H2>Lista de Nombres</H2>");
    echo ("<Hr>");
```

Como se puede observar, los nombres son cargados en posiciones aleatorias. La función reset() lleva el apuntador al principio del arreglo y para poder encontrar la posición de un elemento se utiliza la función key(), la cual acepta como parámetro el vector y devuelve la posición.

Para poder obtener el contenido en esa posición se utiliza la función current(), que recibe el arreglo y devuelve el valor almacenado. La

Si se desean imprimir los valores del arreglo en forma invertida, solamente se debe cambiar la instrucción reset($Nombre) por end($Nombre) y la instrucción while(next($Nombre)) por while(prev($Nombre)).

for

El ciclo for es un bucle repetitivo que permite realizar de manera cíclica un conjunto de instrucciones hasta que se cumpla la condición establecida. Su sintaxis es la siguiente:

```
for ( inicializar ; condicion_logica ; paso ){
   // Código a ejecutarse hasta que la condición
   // lógica se cumpla
}
```

ej9.php

```
<center><H2>Tabla de datos</H2></center>
    <Table align=Center Border=1 width=60%>
    <TR>
    <TH>Número</TH>
    <TH>Cuadrado</TH>
    <TH>Cubo</TH>
    </TR>
<?PHP
    for ($i=1; $i<=15; $i++) {
        $cuadrado = $i * $i;
```

Función range

Esta función permite asignar a los elementos de un arreglo los valores comprendidos en un determinado rango. Acepta como parámetros los valores de inicio y fin de los datos a almacenar.

ej10.php

```
<?PHP
	$arregloAleatorio = range(100,150);
	echo "<H2>Arreglo generado de manera aleatoria.</H2>";
	echo "<H3><Hr>";
	for (;list($i,$num)=each($arregloAleatorio);) {
		echo $num . "<br>";
	}
?>
```

Ejercicios