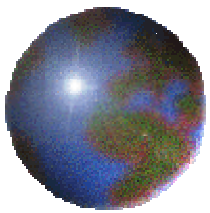


Desarrollo de Portales WEB Lenguaje PHP

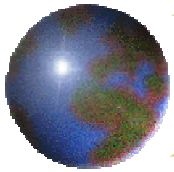
Gonzalo Anchante Hurtado

Docente

INSTITUTO SUPERIOR TECNOLÓGICO

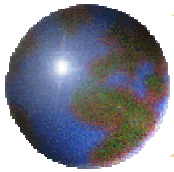
IDAT

R.M. N° 420 - 94 - ED



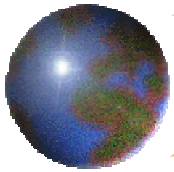
Lenguaje PHP básico

1. Expresiones y operadores
2. Estructuras de control
3. Funciones
4. Tablas
5. Bibliotecas de funciones



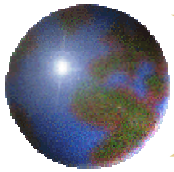
Estructuras de control

- Estructuras selectivas:
  - if-else
  - switch
- Estructuras repetitivas:
  - while
  - for
  - foreach



Introducción

- Todo archivo de comandos PHP se compone de una serie de sentencias. Una sentencia puede ser una asignación, una llamada a función, un bucle, una sentencia condicional e incluso una sentencia que no haga nada (una sentencia vacía).



Introducción

- Las sentencias normalmente acaban con punto y coma. Además, las sentencias se pueden agrupar en grupos de sentencias encapsulando un grupo de sentencias con llaves. Un grupo de sentencias es también una sentencia o Estructura de Control.

Estructuras de control -if

- Estructura selectiva **if-else**

```
if (condición)
   sentencia
```

```
if (condición)
   sentencia 1
else
   sentencia 2
```

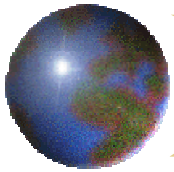
```
if (condición1)
   sentencia 1
else if (condición2)
   sentencia 2
...
else if (condición n)
   sentencia n
else
   sentencia n+1
```

- Las sentencias compuestas se encierran entre llaves
- elseif puede ir todo junto

Estructuras de control – if else

- Ejemplo de estructura selectiva if-else:

```
<?PHP
   $sexo=“M”;
   $nombre=“Andrea”;
   if ($sexo == ‘M’)
         $saludo = "Bienvenida, ";
   else
         $saludo = "Bienvenido, ";
   $saludo = $saludo . $nombre;
   print ($saludo);
?>
```



Estructuras de control - if

- A menudo, se desea tener más de una sentencia ejecutada de forma condicional.
- Por supuesto, no hay necesidad de encerrar cada sentencia con una cláusula if.
- En vez de eso, se pueden agrupar varias sentencias en un grupo de sentencias. Por ejemplo, este código mostraría a es mayor que b si *$a* fuera mayor que *$b*, y entonces asignaría el valor de *$a* a *$b*:

```
if ($a > $b) {
      print "a es mayor que b";
      $b = $a;
  }
```

Estructuras de control -if

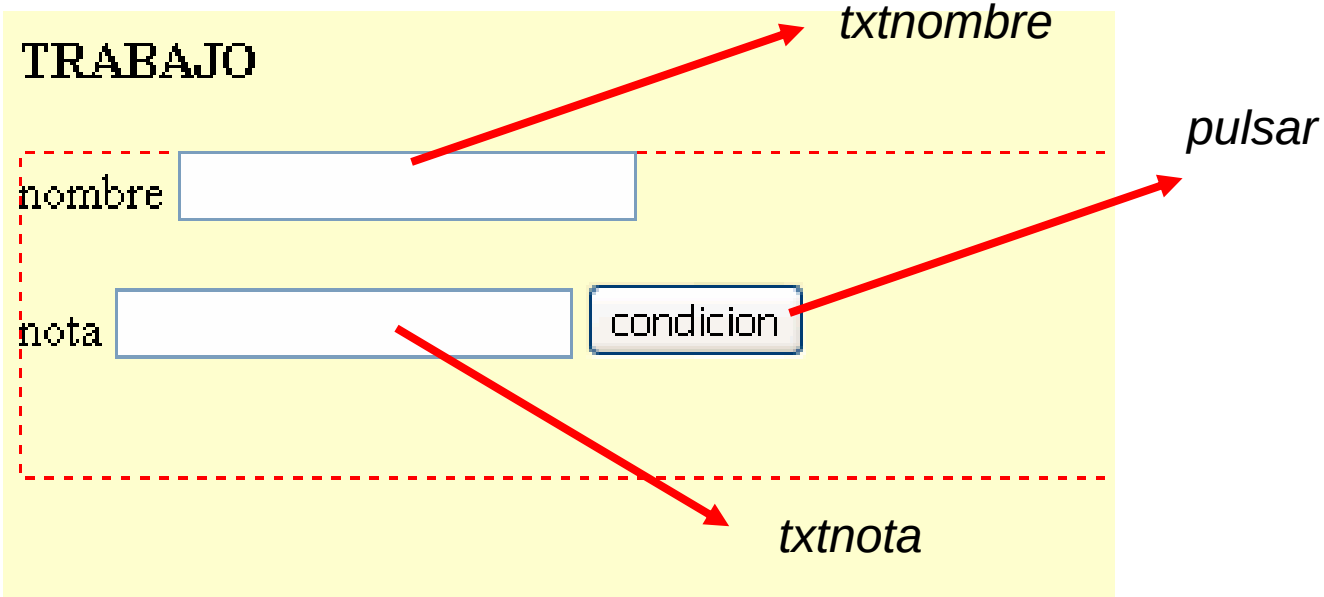
- A menudo queremos ejecutar una sentencia si se cumple una cierta condicion, y una sentencia distinta si la condición no se cumple. Esto es para lo que sirve else. else extiende una sentencia if para ejecutar una sentencia en caso de que la expresión en la sentencia if se evalúe como **FALSE**

```
if ($a > $b) {
      print "a es mayor que b";
 } else {
      print "a NO es mayor que b";
 }
```

Estructuras de control –if-elseif

Ejercicio1 :

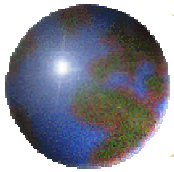
Ingresar el nombre y nota de un alumno, luego mostrar si el alumno esta desaprobado o aprobado, en función a la nota (0 – 10 desaprobado; 11-20 aprobado). Además colocar cuanto le falto para aprobar con el mínimo.



Estructuras de control –if-elseif

- Elseif, como su nombre sugiere, es una combinación de if y else. Como else, extiende una sentencia if para ejecutar una sentencia diferente en caso de que la expresión if original se evalúa como **FALSE**.
- No obstante, a diferencia de else, ejecutará esa expresión alternativa solamente si la expresión condicional elseif se evalúa como **TRUE**.
- Por ejemplo, el siguiente código mostraría a es mayor que b, a es igual a b o a es menor que b

```php
<?PHP
if ($a > $b) {
     print "a es mayor que b";
 } elseif ($a == $b) {
     print "a es igual que b";
 } else {
     print "a es menor que b";
 }

?>
```



Estructuras de control – switch

- La sentencia switch es similar a una serie de sentencias IF en la misma expresión.
- En muchas ocasiones, se quiere comparar la misma variable (o expresión) con muchos valores diferentes, y ejecutar una parte de código distinta dependiendo de a qué valor es igual. Para ello sirve la sentencia switch.

Estructuras de control - switch

- Estructura selectiva **switch**

```
switch (expresión)
{
   case valor_1:
         sentencia 1
         break;
   case valor_2:
         sentencia 2
         break;
   …
   case valor_n:
         sentencia n
          break;
   default
         sentencia n+1
}
```

Estructuras de control – switch

- Es importante entender cómo se ejecuta la sentencia switch para evitar errores. La sentencia switch ejecuta línea por línea (realmente, sentencia a sentencia).
- Al comienzo, no se ejecuta código. Sólo cuando se encuentra una sentencia case con un valor que coincide con el valor de la expresión switch PHP comienza a ejecutar las sentencias.
- PHP continúa ejecutando las sentencias hasta el final del bloque switch, o la primera vez que vea una sentencia break. Si no se escribe una sentencia break al final de una lista de sentencias case, PHP seguirá ejecutando las sentencias del siguiente case.

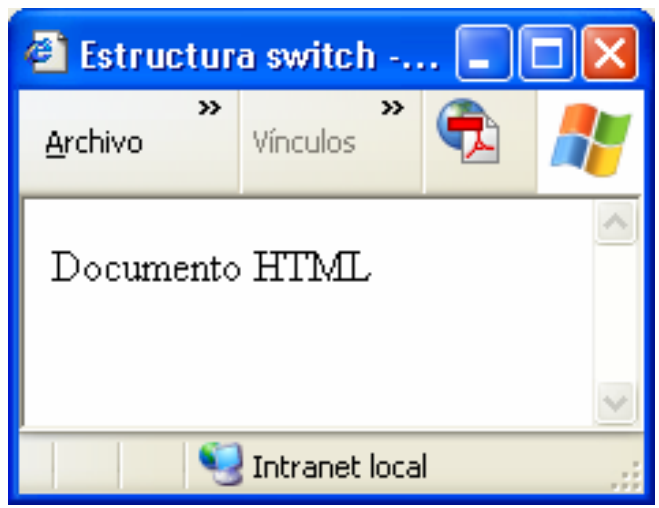
Estructuras de control – switch

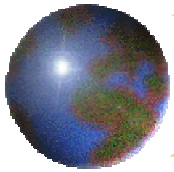
- Aquí, si $i es igual a 0, ¡PHP ejecutaría todas las sentencias print! Si $i es igual a 1, PHP ejecutaría las últimas dos sentencias print y sólo si $i es igual a 2, se obtendría la conducta 'esperada' y solamente se mostraría 'i es igual a 2'. Así, es importante no olvidar las sentencias break

```
<?PHP
 $i=0;
switch ($i) {
      case 0:
          print "i es igual a 0";
      case 1:
          print "i es igual a 1";
      case 2:
          print "i es igual a 2";
 }
?>
```

Estructuras de control –switch,case

```php
<?php
$extension="HTM";
  switch ($extension)
{
   case ("PDF"):
      $tipo = "Documento Adobe PDF";
      break;
   case ("TXT"):
      $tipo = "Documento de texto";
      break;
   case ("HTML"):
   case ("HTM"):
      $tipo = "Documento HTML";
      break;
    default:
      $tipo = "Archivo " . $extension;
}
print ($tipo);

  ?>
```





Estructuras de control - while

- El significado de una sentencia while es simple. Le dice a PHP que ejecute la(s) sentencia(s) anidada(s) repetidamente, mientras la expresión while se evalúe como **TRUE**.

- El valor de la expresión es comprobado cada vez al principio del bucle, así que incluso si este valor cambia durante la ejecución de la(s) sentencia(s) anidada(s), la ejecución no parará hasta el fin de la iteración

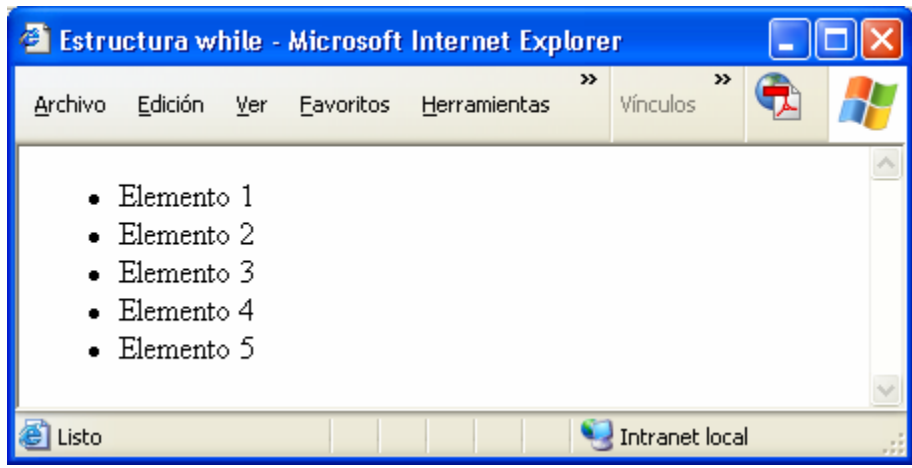
Estructuras de control - while

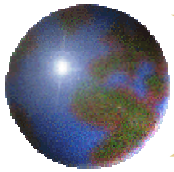
- Estructura repetitiva **while**

```
while (condición)
  sentencia
```

Estructuras de control - While

```
<body>

<p><strong>TRABAJO DE ESTRUCTURAS DE VI&Ntilde;ETAS </strong></p>
<p>
  <?PHP

    print ("<UL>\n");
    $i=1;
    while ($i <= 10)
    {
        print ("<LI>Elemento $i</LI>\n");
        $i++;
    }
    print ("</UL>\n");
?>
  <?PHP

    print ("<OL type ='I'>\n");
    $i=1;
    while ($i <= 10)
    {
        print ("<LI>Elemento $i</LI>\n");
        $i++;
    }
    print ("</OL>\n");
?>

</p>
</body>
```



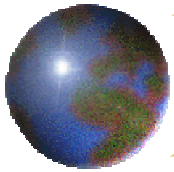


Estructuras de control - DO

- Los bucles do..while son muy similares a los bucles while, excepto que las condiciones se comprueban al final de cada iteración en vez de al principio.
- La principal diferencia frente a los bucles regulares while es que se garantiza la ejecución de la primera iteración de un bucle do..while (la condición se comprueba sólo al final de la iteración), mientras que puede no ser necesariamente ejecutada con un bucle while regular.

Estructuras de control - DO

```php
<?PHP
$i = 0;
 do {
      print " El numero es: $i <br>";
      $i++;
 } while ($i<50);

?>
```



Estructuras de control For

- **Los bucles for son los bucles más complejos en PHP**.
- Estructura repetitiva **for**

```
for (inicialización; condición; incremento)
  sentencia
```

Estructuras de control -for

- Al comienzo de cada iteración, se evalúa *expr2* . Si se evalúa como **TRUE**, el bucle continúa y las sentencias anidadas se ejecutan. Si se evalúa como **FALSE**, la ejecución del bucle finaliza.

```
<?PHP

for ($i = 1; $i <= 10; $i++) {
      print $i;
  }

?>
```

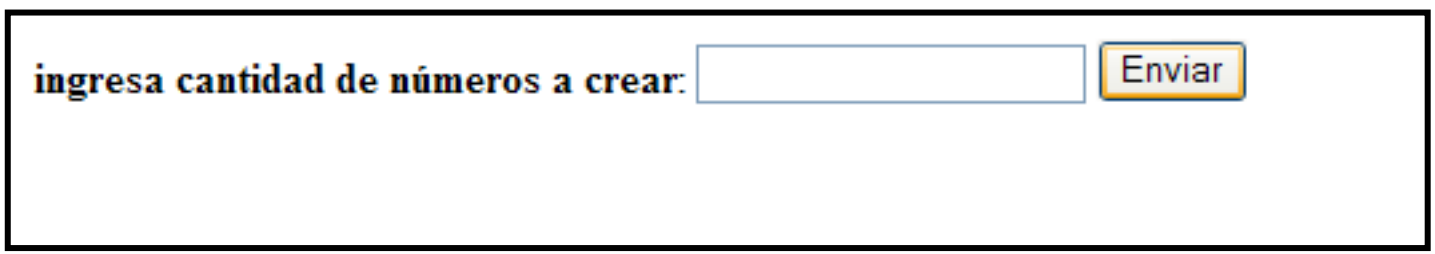
Estructuras de control - for

```php
<?php

    print ("<UL>\n");
    for ($i=1; $i<=25; $i++)
        print ("<LI>Elemento $i</LI>\n");
    print ("</UL>\n");
?>
```

Estructuras de control –for

ejercicio 2

- Diseñar un algoritmo en PHP que permita mostrar un listado de los primeros “N” números naturales positivos impares. El valor debe ser ingresado por teclado.



Colocar en el cuaderno y dar solución

Ejercicio 3. Archivo Ejercicio: Usando la estructura de control FOR genera el código que permita imprimir una tabla de N filas y N celdas.

Ejercicios Prácticos

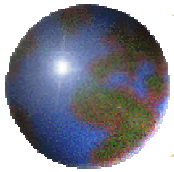
Ejercicio 04

REPORTE DE PRODUCTOS

1 - 1	1 - 2	1 - 3	1 - 4
2 - 1	2 - 2	2 - 3	2 - 4
3 - 1	3 - 2	3 - 3	3 - 4
4 - 1	4 - 2	4 - 3	4 - 4
5 - 1	5 - 2	5 - 3	5 - 4
6 - 1	6 - 2	6 - 3	6 - 4
7 - 1	7 - 2	7 - 3	7 - 4
8 - 1	8 - 2	8 - 3	8 - 4
9 - 1	9 - 2	9 - 3	9 - 4
10 - 1	10 - 2	10 - 3	10 - 4

Colocar en el cuaderno y dar solución

Luego .. Pintar el fondo de la columna 2 y 4 de color amarillo u otro color



ejercicio 4

- Realizar un algoritmo en php que permita ingresar un valor por teclado (entre 1 y 4 dígitos) y muestre lo siguiente:
  - Nro de millares
  - Nro de centenas:
  - Nro de decenas.
  - Nro de unidades.

PLANTEADO Ejercicio 5

- Diseñe un algoritmo que lea tres longitudes y determine si forman o no un triángulo. Si es un triángulo determine de que tipo de triángulo se trata entre: equilátero (si tiene tres lados iguales), isósceles (si tiene dos lados iguales) o escaleno (si tiene tres lados desiguales). Considere que para formar un triángulo se requiere que: "el lado mayor sea menor que la suma de los otros dos lados".