

Desarrollo de Portales WEB *Lenguaje PHP*

Gonzalo Anchante Hurtado
Docente

INSTITUTO SUPERIOR TECNOLÓGICO



R.M. N° 420 - 94 - ED



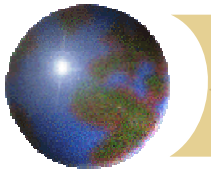
Presentación:

Profesor: Gonzalo Anchante Hurtado

Correo: gonzalo.anchante@inei.gob.pe
gonzalo.anchante@gmail.com

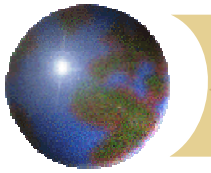
Calificaciones:

- Participación durante la clase
- Examen Parcial
- Notas de prácticas de clase o laboratorio
- Puntualidad en Clase y entrega de trabajos



Tema 1: Introducción

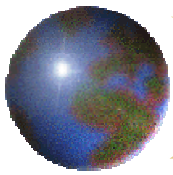
- 1.Introducción a PHP
- 2.Instalación y configuración de Apache
- 3.Instalación y configuración de PHP
- 4.Recursos de PHP



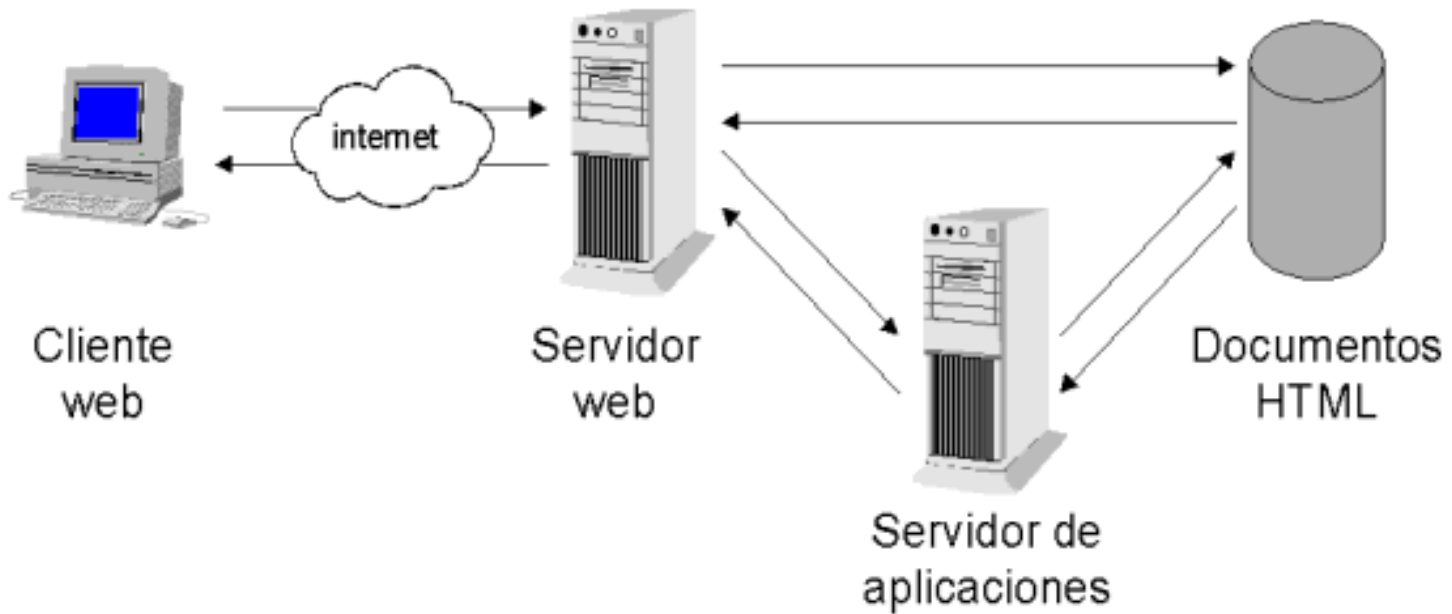
Introducción a PHP

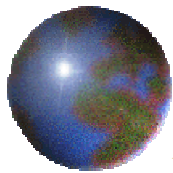
❖ **Lenguajes de *script***

- ❑ PHP es un lenguaje de *script* del lado del servidor. Otros lenguajes similares son ASP, JSP o ColdFusion
- ❑ Los scripts PHP están incrustados en los documentos HTML y el servidor los interpreta y ejecuta antes de servir las páginas al cliente
- ❑ El cliente no ve el código PHP sino los resultados que produce

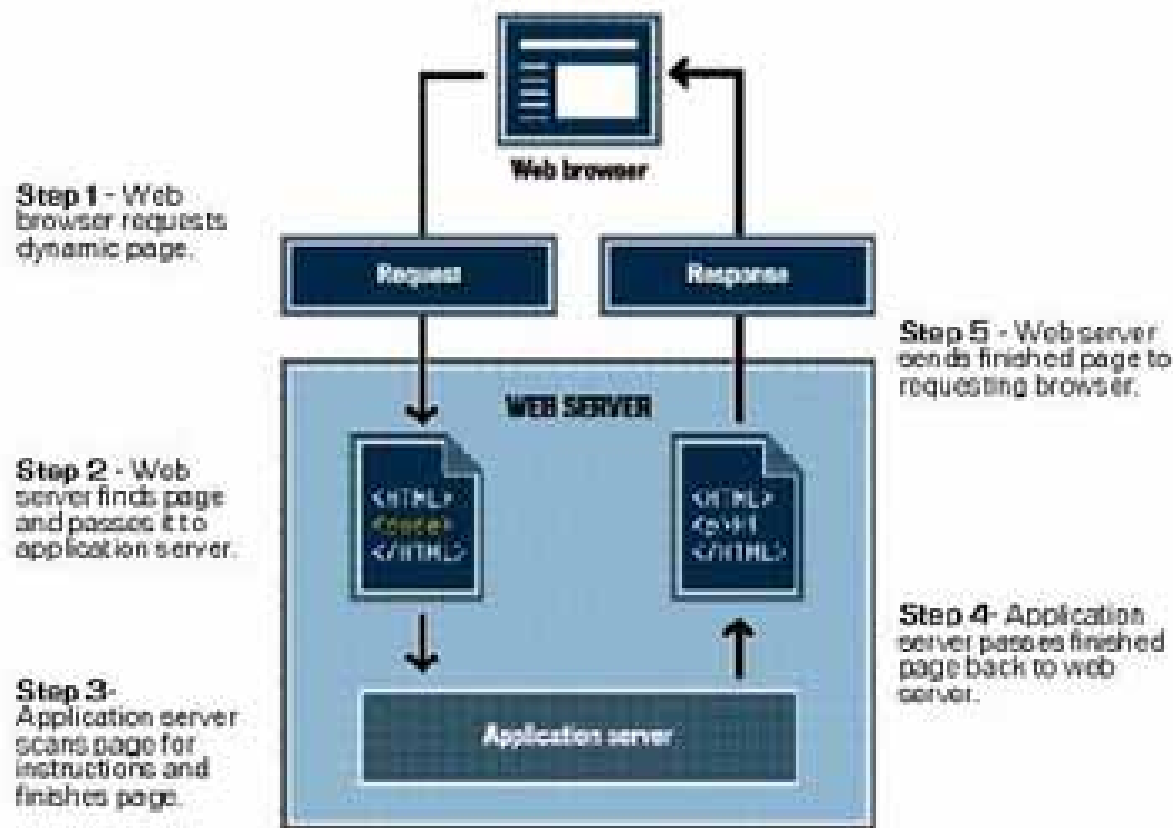


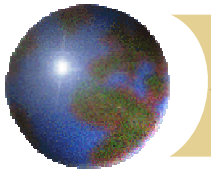
Introducción a PHP





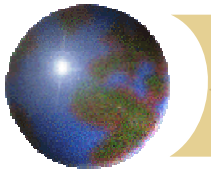
Introducción a PHP





Servidor WEB

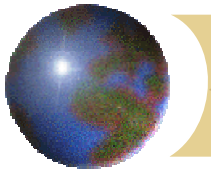
- ❁ Un servidor web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos



Servidor WEB

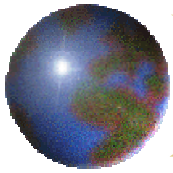
- ❖ Sistema de transferencia de información basado en arquitectura cliente/servidor, empleando protocolo **TCP/IP** permite intercambio de archivos y páginas web entre puntos de la red.
- ❖ El servidor de páginas es el sitio primordial de un sitio web, ya que genera información a los usuarios finales.

Recurso: www.saifernet.com/thesau/textos/th.s.htm



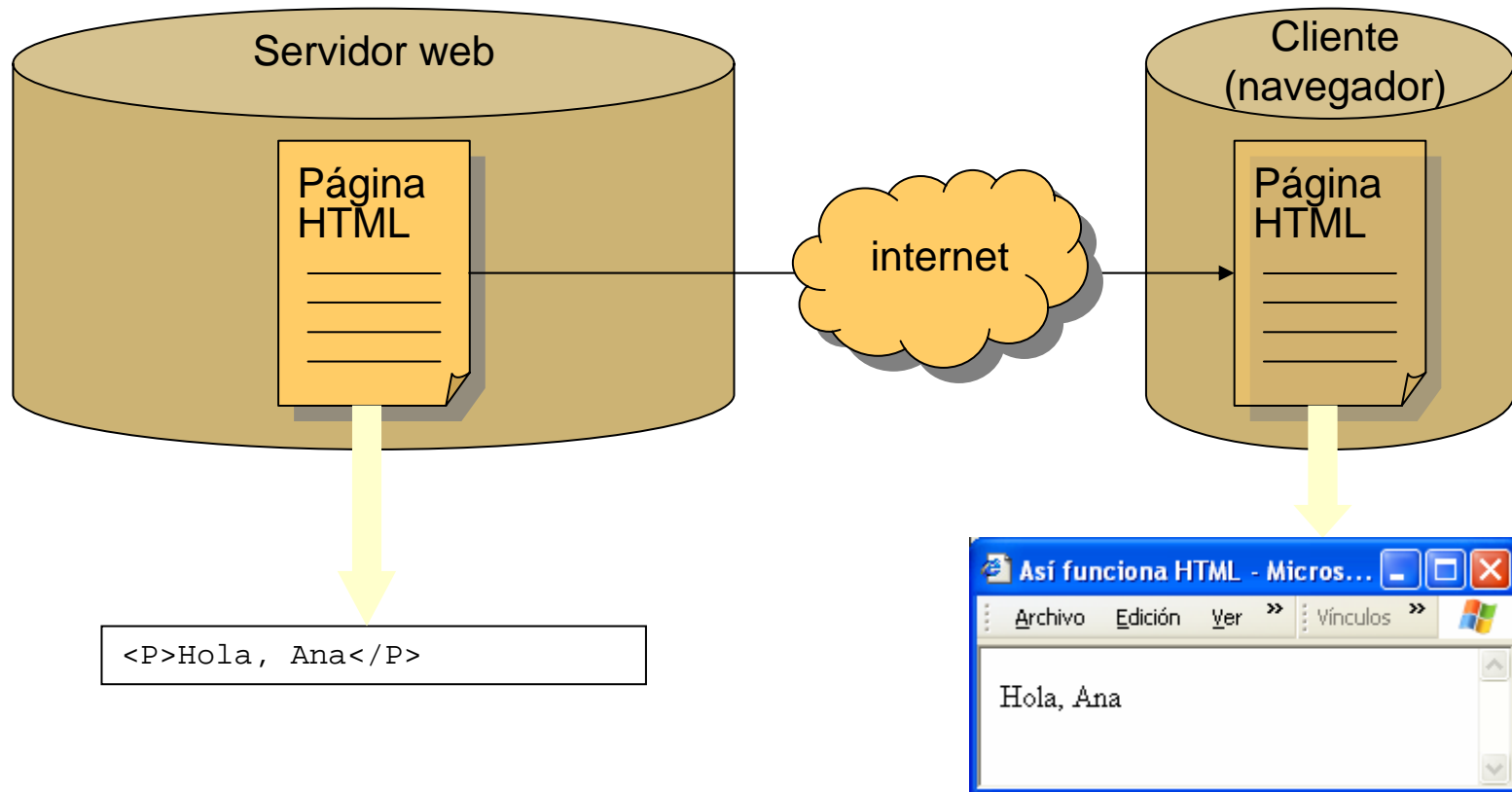
Servidor de Aplicaciones

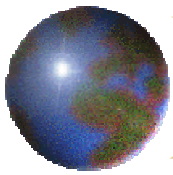
- ✿ En informática se denomina servidor de aplicaciones a un servidor en una red de computadores que ejecuta ciertas aplicaciones de software.
- ✿ Usualmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Un servidor de aplicaciones generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación.



Introducción a PHP

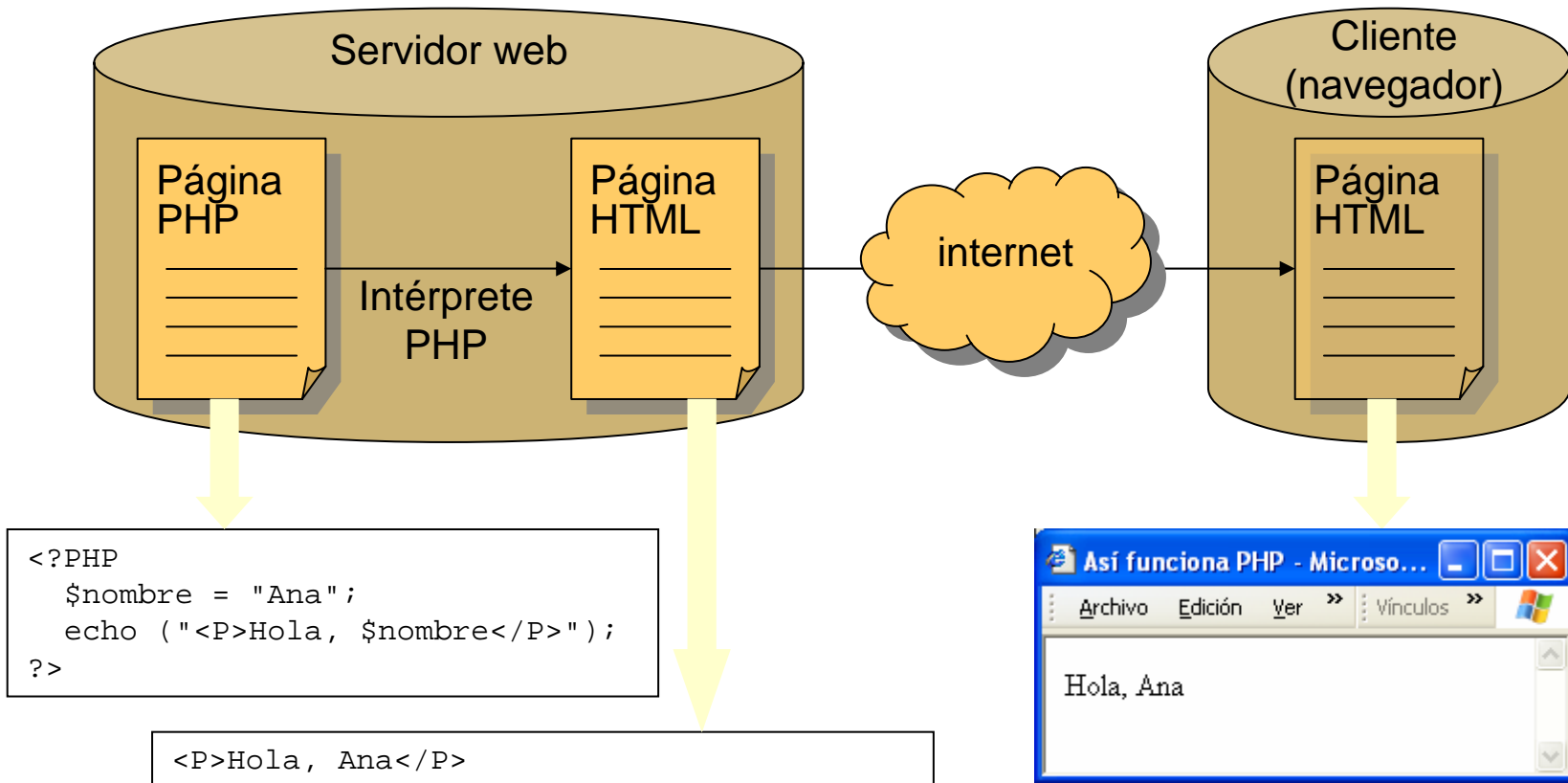
● ¿Cómo funciona PHP? (1)

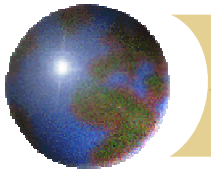




Introducción a PHP

● ¿Cómo funciona PHP? (2)

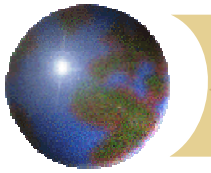




Introducción a PHP

✚ Breve historia de PHP

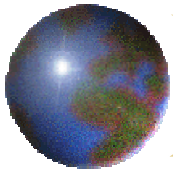
- ✚ Creado por Rasmus Lerdorf para uso personal en 1994
- ✚ PHP = *Personal Hypertext Processor*
- ✚ Versión actual: PHP 5
- ✚ Es un módulo que se añade al servidor web y fue concebido inicialmente para Apache



Introducción a PHP

✚ ¿Por qué PHP?

- ✚ Por sus ventajas: es potente, fácil de aprender, de libre distribución, permite el acceso a bases de datos y otras funcionalidades orientadas a la red.
- ✚ Dispone de abundante soporte en la Web.

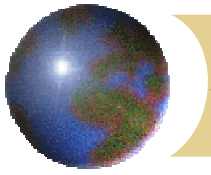


Introducción a PHP

Requisitos

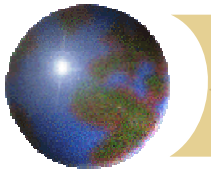
- Servidor web Apache u otro (www.apache.org)
- con el módulo PHP (www.php.net)
- y la base de datos MySQL (www.mysql.com) si se desea crear páginas dinámicas.





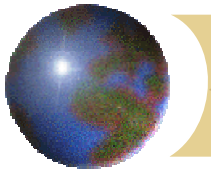
Lenguaje PHP básico

1. Sintaxis básica
2. Tipos de datos
3. Variables
4. Constantes
5. Expresiones y operadores



Sintaxis básica

- ⊕ PHP es sensible a las mayúsculas
- ⊕ ¿Cómo se incrusta en la página web?
 - `<?PHP ... ?>`
recomendado, siempre disponible
 - `<?= expresión ?>`
equivale a `<? echo expresión ?>`
- ⊕ Las instrucciones se separan con un ; como en C. La marca final `?>` implica un ;
- ⊕ Comentarios: como en C, `/* ... */` y `//`



Sintaxis básica

✚ Para imprimir: **echo** y **print**

echo: muestra una o más cadenas

echo cadena1 [, cadena2...]; // no es una función

```
echo "Hola mundo";
```

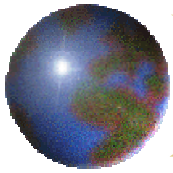
```
echo "Hola ", "mundo";
```

print: muestra una cadena

print cadena; // no es una función

```
print "Hola mundo";
```

```
print "Hola " . "mundo";
```



Sintaxis básica

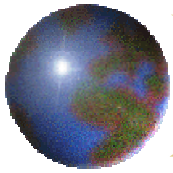
● Ejemplo:

```
<HTML>
<HEAD>
<TITLE>Mi primer programa en PHP</TITLE>
</HEAD>

<BODY>

<?PHP
    echo ("<P>Hola mundo</P>");
?>

</BODY>
</HTML>
```



Sintaxis básica

- ⊕ Uso de \n para generar código HTML legible
- ⊕ a) Sin \n

Código PHP

```
print ( "<P>Párrafo 1</P>" );  
print ( "<P>Párrafo 2</P>" );
```

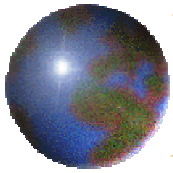
Código HTML

```
<P>Párrafo 1</P><P>Párrafo 2</P>
```

Salida

Párrafo 1

Párrafo 2



Sintaxis básica

- ⊕ Uso de `\n` para generar código HTML legible
- ⊕ b) Con `\n`

Código PHP

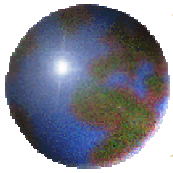
```
print ( "<P>Párrafo 1</P>\n" );  
print ( "<P>Párrafo 2</P>\n" );
```

Código HTML

```
<P>Párrafo 1</P>  
<P>Párrafo 2</P>
```

Salida

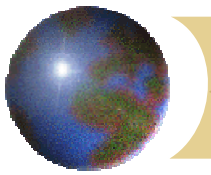
```
Párrafo 1  
  
Párrafo 2
```



Conceptos PHP

- Hay cuatro conjuntos de **etiquetas** que **delimitan el código PHP**:

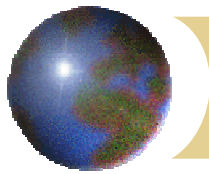
```
<?php y ?>  
  <? y ?>  
  <% y %>  
<script language="php"> y </script>
```



Conceptos PHP

✿ Ejemplos de embeber código php y html

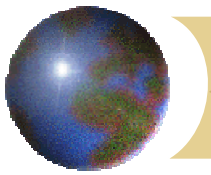
```
1 <HTML>
2 <HEAD>
3     <TITLE>Etiquetas PHP</TITLE>
4 </HEAD>
5 <BODY>
6 <CENTER>
7 <?php
8     $mensaje="Estas etiquetas son las más adecuadas para embeber código PHP";
9     echo($mensaje);
10 ?>
11 </CENTER>
12 </BODY>
13 </HTML>
```



Conceptos PHP

✿ Ejemplos de embeber código php y html

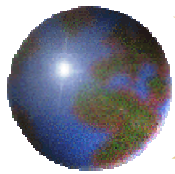
```
1 <HTML>
2 <HEAD>
3     <TITLE>Etiquetas PHP</TITLE>
4 </HEAD>
5 <BODY>
6 <CENTER>
7 <?
8     $mensaje="El formato corto no siempre está disponible";
9     echo ($mensaje);
10 ?>
11 </CENTER>
12 </BODY>
13 </HTML>
```



Conceptos PHP

✿ Ejemplos de embeber código php y html

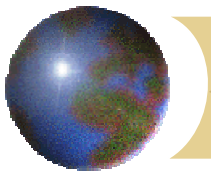
```
1 <HTML>
2 <HEAD>
3     <TITLE>Etiquetas PHP</TITLE>
4 </HEAD>
5 <BODY>
6 <CENTER>
7     <%
8     $mensaje="Opcionalmente, puedes usar las etiquetas ASP";
9     echo ($mensaje);
10    %>
11</CENTER>
12</BODY>
13</HTML>
```

Conceptos PHP

✚ Ejemplos de embeber código php y html

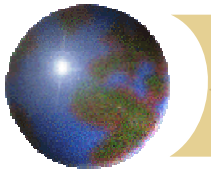
```
1 <HTML>
2 <HEAD>
3     <TITLE>Etiquetas PHP</TITLE>
4 </HEAD>
5 <BODY>
6 <CENTER>
7 <script language="php">
8     $mensaje="Muchos editores, como FrontPage,
9     no aceptan instrucciones de procesado";
10    echo ($mensaje);
11</script>
12</CENTER>
13</BODY>
14</HTML>
```



Conceptos PHP

⊕ Combinar Tags.

```
<html>
  <head>
    <title>PHP – Hello World</title>
  </head>
  <body>
    <b> <?php print("Hello World"); ?> </b>
  </body>
</html>
```



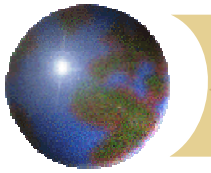
Conceptos PHP

- ⊕ Combinar Tags.- insertar tags html dentro de php

```
<html>
  <head>
    <title>PHP - Hello World</title>
  </head>
  <body>

    <?php
      print("<b>Hello</b> <i>World</i>");
    ?>

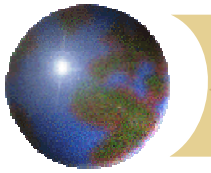
  </body>
</html>
```



Conceptos PHP

⊕ Solo php

```
<?php
    print ("<html>");
    print ("<head>");
    print ("    <title>PHP – Hello World</title>");
    print ("    </head>");
    print ("    <body>");
    print ("        <p>Hello World!</p>");
    print ("    </body>");
    print ("</html>");
?>
```

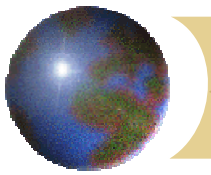


Separación de instrucciones

La separación de instrucciones en PHP se hace terminando cada declaración con un punto y coma ";".

Para la ultima instrucción, la que va antes del cierre de etiqueta, este formalismo no es necesario

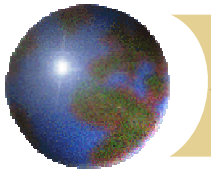
```
1 <HTML>
2 <HEAD>
3     <TITLE>Separación de instrucciones</TITLE>
4 </HEAD>
5 <BODY>
6 <CENTER>
7 <?php
8     $mensaje="Esto es una prueba";
9     echo($mensaje);
10 ?>
11 </CENTER>
12 </BODY>
13 </HTML>
```



Comentario

Un comentario es un texto de carácter informativo que incluimos en el código para comprenderlo fácilmente al volverlo a leer un tiempo después. El servidor ignora los comentarios ya que no van dirigidos a él sino a nosotros mismos.

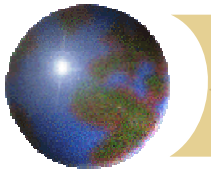
```
1 <HTML>
2 <HEAD>
3   <TITLE>Comentarios</TITLE>
4 </HEAD>
5 <BODY>
6 <CENTER>
7 <?php
8   $mensaje="Probando los comentarios";// Este es un comentario de una línea.
9   echo($mensaje); #Este es otro comentario de una línea.
10  /*En cambio, este otro
11   comentario ocupa varias
12   líneas*/
13 >><BR>
14 <?PHP
15   echo("Todo funciona según lo esperado");
16 >>
17 </CENTER>
18 </BODY>
19 </HTML>
```



Tipos de datos

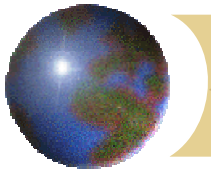
- PHP soporta 8 **tipos de datos primitivos**:
 - Tipos escalares: boolean, integer, double, string
 - Tipos compuestos: array, object
 - Tipos especiales: resource, NULL
- El tipo de una variable no se suele especificar. Se decide en tiempo de ejecución en función del contexto y puede variar
- Funciones de interés:
 - La función `gettype()` devuelve el tipo de una variable
 - Las funciones `is_type` comprueban si una variable es de un tipo dado:

`is_array(), is_bool(), is_float(), is_integer(), is_null(),
is_numeric(), is_object(), is_resource(), is_scalar(),
is_string()`
 - La función `var_dump()` muestra el tipo y el valor de una variable. Es especialmente interesante con los arrays



Tipos de datos

- ⊕ Tipo **integer** (números enteros)
 - ⊞ 27, -5, 0
- ⊕ Tipo **double** (números reales)
 - ⊞ 1.234, -5.33
- ⊕ Tipo **boolean** (lógico)
 - ⊞ Valores: *true*, *false* (insensibles a las mayúsculas)
 - ⊞ El 0 y la cadena vacía tienen valor *false*

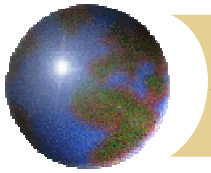


Tipos de datos

• Tipo string:

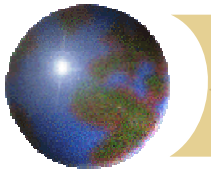
- Las cadenas se encierran entre comillas simples o dobles:
 - 'simples': admite los caracteres de escape \' (comilla simple) y \ (barra). Las variables **NO** se expanden
 - "dobles": admite más caracteres de escape, como \n, \r, \t, \\\, \\$, \". Los nombres de variables **SÍ** se expanden
 - Ejemplos:

```
$a = 9;
print 'a vale $a\n';
// muestra a vale $a\n
print "a vale $a\n";
// muestra a vale 9 y avanza una línea
print "<IMG SRC='logo.gif'>";
// muestra <IMG SRC='logo.gif'>
print "<IMG SRC=\"logo.gif\">";
// muestra <IMG SRC="logo.gif">
```
- Acceso a un carácter de la cadena:
 - La forma es \$inicial = \$nombre{0};



Variables

- Las variables siempre van precedidas de un \$
- El nombre es sensible a las mayúsculas
- Comienzan por letra o subrayado, seguido de letras, números o subrayado
- Variables predefinidas:
 \$GLOBALS, \$_SERVER, \$_GET, \$_POST,
 \$_COOKIE, \$_FILES,
 \$_ENV, \$_REQUEST, \$_SESSION
- Ámbito: globales al archivo (excepto funciones) o locales a una función



Variables

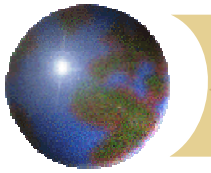
- El ámbito de una variable es el contexto dentro del cual la variable está definida.

- Ejemplo:**

```
$valor = 5;  
print "El valor es: " . $valor . "\n";  
print "El valor es: $valor\n"; // ojo:  
    comillas dobles
```

Resultado:

```
El valor es: 5
```



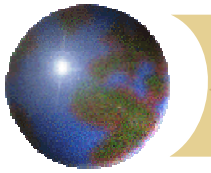
Constantes

- Definición de constantes:

```
define ("CONSTANTE", "hola");  
print CONSTANTE;
```

- No llevan \$ delante

- Sólo se pueden definir constantes de los tipos escalares (boolean, integer, double, string)



Expresiones y operadores

- ⊕ Operadores aritméticos:

$+$, $-$, $*$, $/$, $\%$, $++$, $--$

- ⊕ Operador de asignación:

$=$

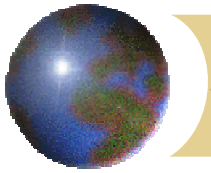
operadores combinados: $.=$, $+=$, etc

$\$a = 3$; $\$a += 5$; \rightarrow a vale 8

$\$b = \text{"hola "}$; $\$b .= \text{"mundo"}$;

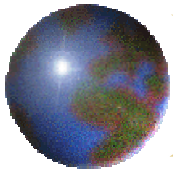
\rightarrow b vale "hola mundo"

\rightarrow Equivale a $\$b = \$b . \text{"mundo"}$;



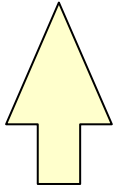
Expresiones y operadores

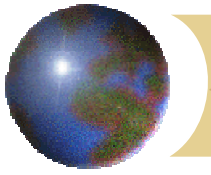
Sintaxis	Nombre	Descripción
$\$a + \b	Adición	Suma de $\$a$ y $\$b$.
$\$a - \b	Substracción	Diferencia entre $\$a$ y $\$b$
$\$a * \b	Multiplicación	Producto de $\$a$ y $\$b$
$\$a / \b	División	Cociente de $\$a$ entre $\$b$
$\$a \% \b	Módulo	Resto de la división de $\$a$ entre $\$b$



Expresiones y operadores

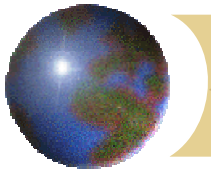
```
1 <?php
2 $a = ($b = 4) + 5; // La variable $a vale 9 y $b vale 4.
3 ?>
```





Expresiones y operadores

- ⊕ Operadores de comparación:
==, !=, <, >, <=, >= y otros
- ⊕ Operador de control de error: @.
Antepuesto a una expresión, evita cualquier mensaje de error que pueda ser generado por la expresión



Expresiones y operadores

Sintaxis	Nombre	Descripción
$\$a == \b	Igualdad	Cierto si \$a es igual a \$b.
$\$a === \b	Identidad	Cierto si \$a es igual a \$b y si son del mismo tipo (sólo PHP4)
$\$a != \b	Desigualdad	Cierto si \$a no es igual a \$b.
$\$a < \b	Menor que	Cierto si \$a es estrictamente menor que \$b.
$\$a > \b	Mayor que	Cierto si \$a es estrictamente mayor que \$b.
$\$a <= \b	Menor o igual que	Cierto si \$a es menor o igual que \$b.
$\$a >= \b	Mayor o igual que	Cierto si \$a es mayor o igual que \$b.



Expresiones y operadores

Operadores lógicos:

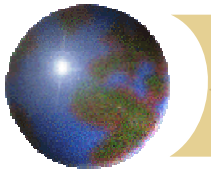
and (&&), or (||), !, xor

and/&& y or/|| tienen diferentes prioridades

✚ Operadores de cadena:

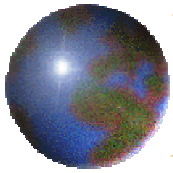
concatenación: . (punto)

asignación con concatenación: .=



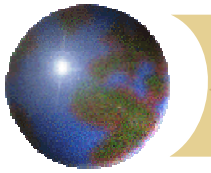
Expresiones y operadores

Sintaxis	Nombre	Descripción
++\$a	Preincremento	Incrementa \$a en uno y después devuelve \$a.
\$a++	Postincremento	Devuelve \$a y después incrementa \$a en uno.
--\$a	Predecremento	Decrementa \$a en uno y después devuelve \$a.
\$a--	Postdecremento	Devuelve \$a y después decrementa \$a en uno.



Expresiones y operadores

Sintaxis	Nombre	Descripción
$\$a$ and $\$b$ ó $\$a$ && $\$b$	Y	Cierto si tanto $\$a$ como $\$b$ son ciertos.
$\$a$ or $\$b$ ó $\$a$ $\$b$	O	Cierto si $\$a$ o $\$b$ son ciertos.
$\$a$ xor $\$b$	O exclusiva	Cierto si $\$a$ es cierto o $\$b$ es cierto, pero no ambos a la vez.
! $\$a$	Negación	Cierto si $\$a$ no es cierto.



Expresiones y operadores

- ⊕ Precedencia de operadores (de mayor a menor):

++, --

*, /, %

+, -

<, <=, >, >=

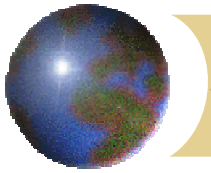
==, !=

&&

||

and

or



Colocar en el cuaderno y dar solución

1. Cual es la salida de las variables \$a, \$b y \$c

\$a=5;

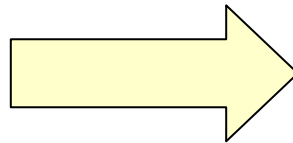
\$b=9;

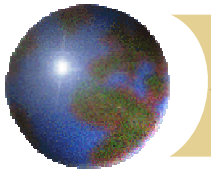
\$c=3:

\$b+=4;

\$a-=1;

(\$a>=\$b) || (\$c<=\$a)





Colocar en el cuaderno y dar solución

2. Cual es la salida de las variables \$a, \$b y \$c

\$a=18;

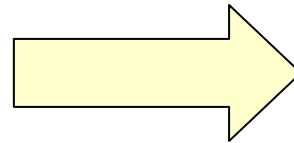
\$b=22;

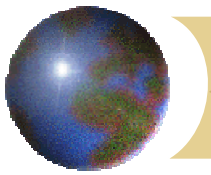
\$c=11:

\$b+=c+4;

\$a-=3;

(\$a>=\$b) && (\$c<=\$a)

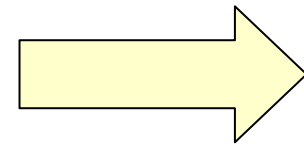


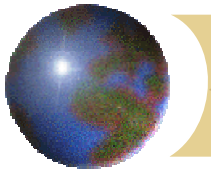


Colocar en el cuaderno y dar solución

3. Cual es la salida de las variables \$a, \$b y \$c

```
1  <?  
2  $a="ESTAMOS";  
3  $b="  EN CLASE";  
4  $c=11;  
5  
6  $b.=$A." ";  
7  echo $b;  
8  $a=$a." lunes";  
9  echo $a.$b;  
10  
11 ?>
```





Colocar en el cuaderno y dar solución

4. Cual es la salida de las variables \$a, \$b y \$c

```
$suma=26;
```

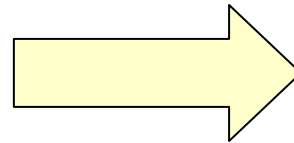
```
$resta=58;
```

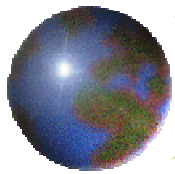
```
$resultado=61:
```

```
$resta+=resultado+4-($suma+3);
```

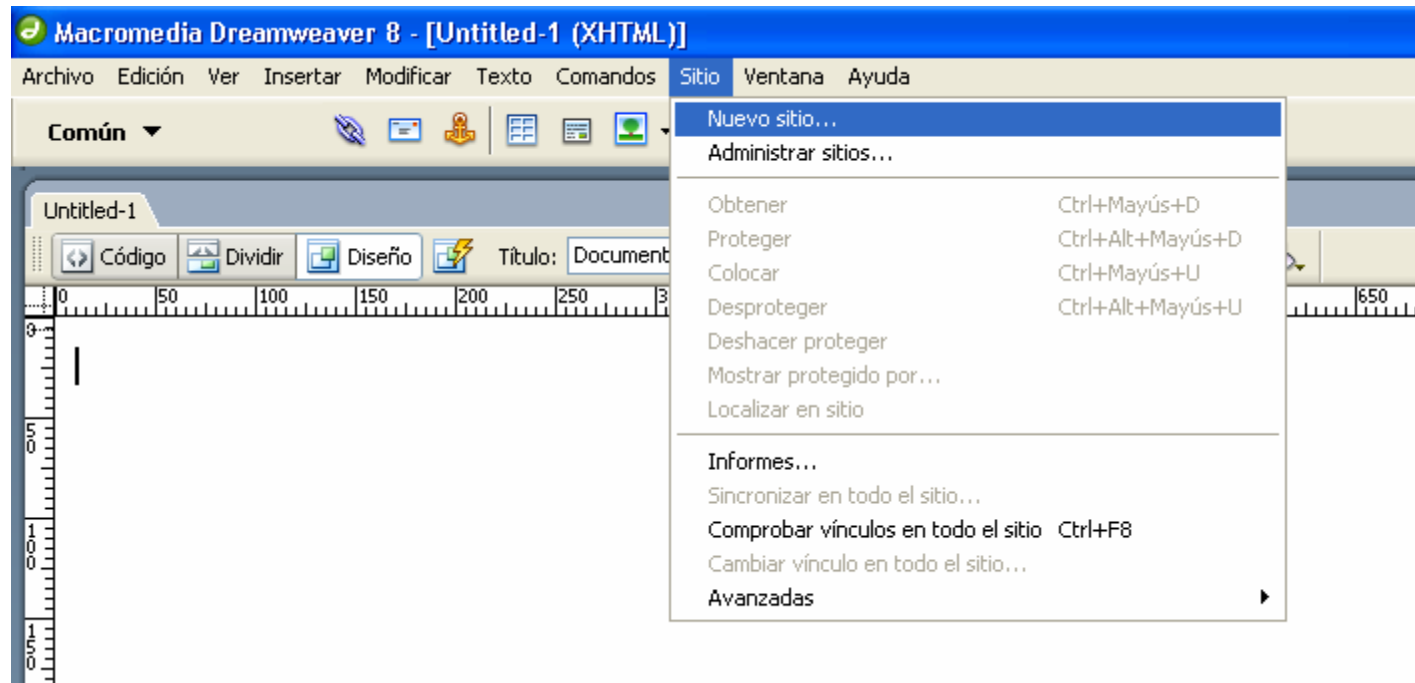
```
$suma-=3;
```

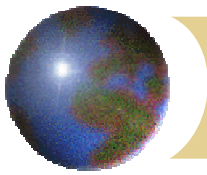
```
($resta>=$resultado) && ($suma<=$resta)
```





Ahora a crear un sitio web con Tecnología Servidor





Ahora a crear un sitio web con Tecnología Servidor

Definición del sitio para clase

Básicas | Avanzadas

Categoría

- Datos locales
- Datos remotos
- Servidor de prueba
- Ocultación
- Design Notes
- Mapa de diseño del sitio
- Columnas vista archivo
- Contribute

Datos locales

Nombre del sitio:

Carpeta raíz local:

☒ Actualizar lista de archivos locales automáticamente

Carpeta predeterminada de imágenes:

Vínculos relativos a: ☒ Documento ☐ Raíz del sitio

Dirección HTTP:

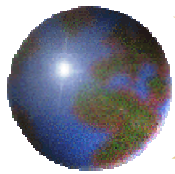
Esta dirección se utiliza para vínculos relativos al sitio y para que Verificador de vínculos detecte los vínculos HTTP referidos a su propio sitio

Vínculos con distinción entre mayúsculas y minúsculas: ☐ Usar verificación de vínculos con distinción entre mayúsculas y minúsculas

Caché: ☒ Activar caché

El caché mantiene la información de archivos y activos del sitio. Esto acelera las funciones de panel de Activos, administración de vínculos y Mapa del sitio.

Aceptar Cancelar Ayuda



Ahora a crear un sitio web con Tecnología Servidor

Básicas Avanzadas

Categoría

- Datos locales
- Datos remotos
- Servidor de prueba**
- Ocultación
- Design Notes
- Mapa de diseño del sitio
- Columnas vista archivo
- Contribute

Modelo de servidor: PHP MySQL

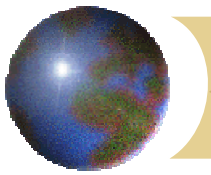
Acceso: Local/red

Carpeta de servidor: C:\AppServ\www\clase\

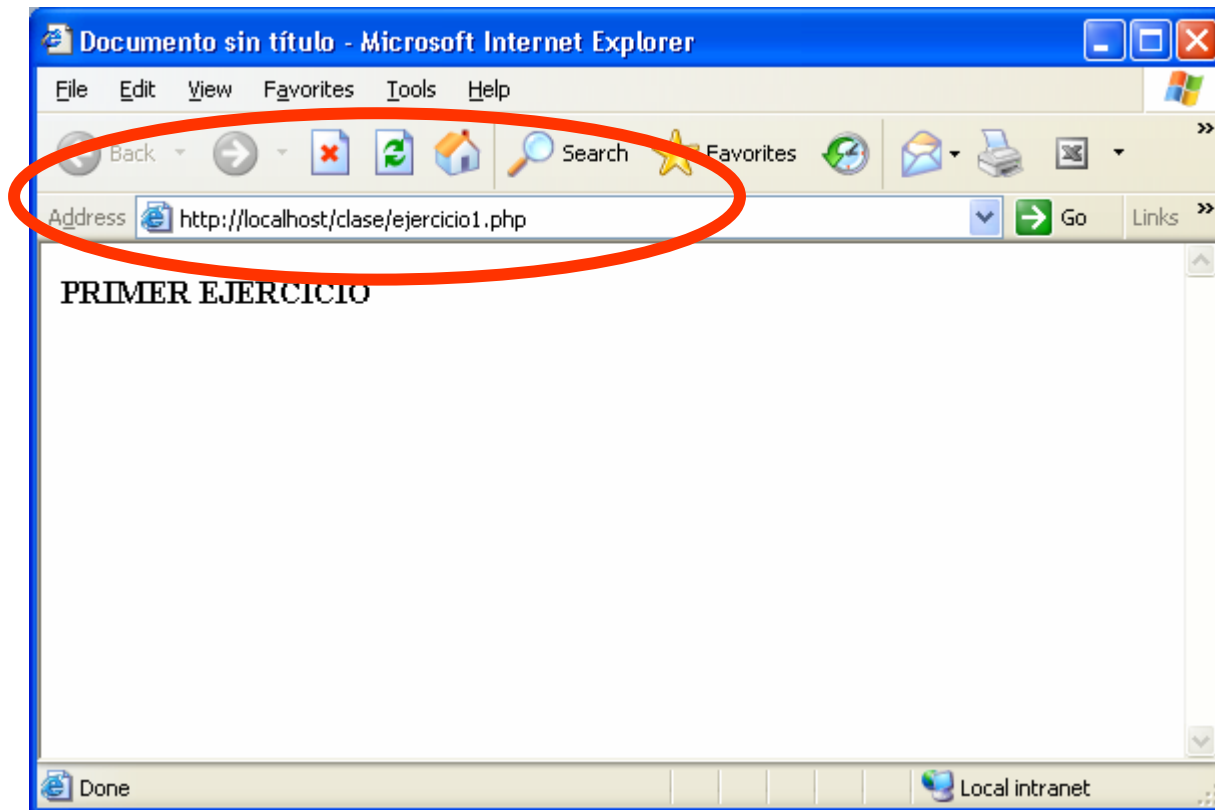
☒ Actualizar automáticamente lista de

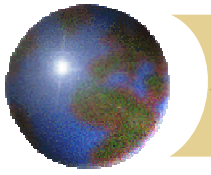
Prefijo de URL: http://localhost/clase/

El Prefijo de URL es la carpeta raíz del sitio en el servidor de prueba.



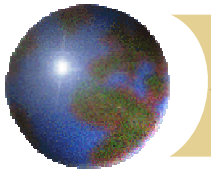
Ahora a crear un sitio web con Tecnología Servidor





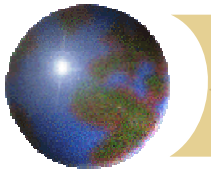
Expresiones y operadores

EJERCICIO 1



Colocar en el cuaderno y dar solución

2. Crear 4 variables, aplicarles valores correlativos, en base al valor de la primera variable, se debe mantener una razón aritmética entre variables de 2 unidades. Mostrar en pantalla el cuadrado de cada una de ellas.



Colocar en el cuaderno y dar solución

3. Realizar un ejercicio que, a partir de **SOLO** tres variables

Vproducto: es el producto de venta

Vcosto: es el precio unitario de venta

Vcantidad: es la cantidad de venta del producto.

Se permita obtener el siguiente reporte por pantalla:

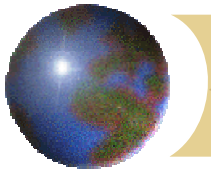
El producto: xxxxxxxxxxx **La cantidad:** xxxxx **El precio Uni:** yyyyyy

Sub total: xxxxxxxxxxx

Descuento: es el 20% del sub total.

Recargo : es el 15% del saldo del subtotal y el descuento.

Total neto: es el subtotal menos descuento mas recargo.



Colocar en el cuaderno y dar solución

PLANTEADO....4. Realizar un ejercicio que, a partir de dos variables enteras llamadas \$num1 y \$num2, asigne valor a estas variables y muestre por pantalla los siguientes mensajes en función del valor asignado a las variables. Si por ejemplo asignamos los valores 3 y 7 a \$num1 y \$num2 respectivamente se debe mostrar por pantalla:

El valor de la variable \$num1 es: 3

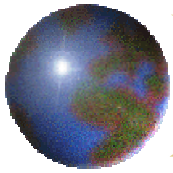
El valor de la variable \$num2 es: 7

La suma de 3 y 7 es: 10

El producto de 3 por 7 es: 21

El cociente de 3 entre 7 es: 0.42857142857143

El resto de dividir 3 entre 7 es: 3



Colocar en el cuaderno y dar solución

PLANTEADO.....5. Realizar el ejercicio siguiente:

Numero	Cuadrado	Cubo
Num1		
Num2		
num3		

Detalle: el num1 es cualquier valor, el num2 es el doble del num1 mas 3 unidades, y el num3 es el num2-num1