Desarrollo de sitios web con PHP y MySQL

Tema 2: Lenguaje PHP básico

Ana Mª Fermoso García
afermosoga@upsa.es

Tema 2: Lenguaje PHP básico

1. Sintaxis básica
2. Tipos de datos
3. Variables
4. Constantes
5. Expresiones y operadores
6. Estructuras de control
7. Funciones
8. Tablas
9. Bibliotecas de funciones

Sintaxis básica

- PHP es sensible a las mayúsculas
- ¿Cómo se incrusta en la página web?

```
<?php ... ?>
```

- Las instrucciones se separan con un ; como en C. La marca final ?> implica un ;
- Comentarios: como en C, /* … */ (varias líneas) y // (una línea)

```
/* Comentario de
varias líneas */
print "hola"; // Comentario de una línea
```

Sintaxis básica

- Para imprimir: **echo** y **print**

  echo: muestra una o más cadenas
  echo cadena1 [, cadena2…];

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

  print: muestra una cadena
  print cadena;

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

Sintaxis básica

- Ejemplo:

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

 <BODY>

 <?PHP
    print (“<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
```

Sintaxis básica

- Inclusión de ficheros externos:
  - **include**()
  - **require**()
- Ambos incluyen y evalúan el fichero especificado
- Diferencia: en caso de error include() produce un warning y require() un error fatal
- Se usará require() si al producirse un error debe interrumpirse la carga de la página
- Ejemplo:

Sintaxis básica

```
<HTML>
<HEAD>
   <TITLE>Título</TITLE>
<?PHP
// Incluir bibliotecas de funciones
   require ("conecta.php");
   require ("fecha.php");
   require ("cadena.php");
   require ("globals.php");
?>
</HEAD>
<BODY>
<?PHP
   include ("cabecera.html");
?>
// Código HTML + PHP
. . .
<?PHP
   include ("pie.html");
?>
</BODY>
</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, $_COOKIES, $_FILES, $_ENV, $_REQUEST, $_SESSION
- Ámbito: globales al fichero (excepto funciones) o locales a una función
- Ejemplo:

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

Resultado:
   El valor es: 5
```

Variables globales predefinidas

- $GLOBALS: matriz con todas las variables globales. Al igual que la palabra clave "global" nos permite acceder a las variables globales dentro de una función: $GLOBALS['nom_variable']
- $_SERVER: matriz con las variables del entorno del servidor
- $_GET: matriz con las variables pasadas en la secuencia de comandos con el método GET
- $_POST: matriz con las variables pasadas a la secuencia de comandos con el método POST
- $_COOKIE: contiene las variables de cookies
- $_FILES: matriz relacionada con las variables para carga de ficheros
- $_ENV: matriz de variables de entorno
- $_REQUEST: matriz con todas las variables de entrada de usuario incluyendo las de $_GET, $_POST y $_COOKIE
- $_SESSION: matriz con las variables de sesión

Variables

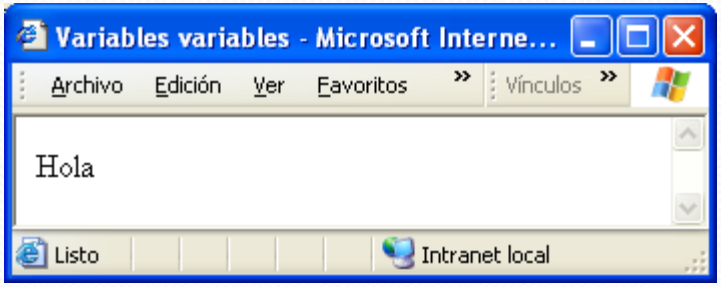
- Variables variables
  - Se pueden crear nombres de variables dinámicamente
  - La variable variable toma su nombre del valor de otra variable previamente declarada
  - Ejemplo:

```
$a = "hola";
$$a = "mundo";

print "$a $hola\n";
print "$a ${$a}";

Resultado:
      hola mundo
      hola mundo
```

Variables

Ejemplo de variables variables: página internacionalizada (1)

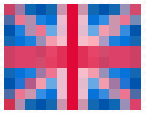
```
<?PHP
    $mensaje_es="Hola";
    $mensaje_en="Hello";
    $idioma = "es";
    $mensaje = "mensaje_" . $idioma;
    print $$mensaje;
?>
```



Variables

- Ejemplo de variables variables: página internacionalizada (2)

```
<?PHP
    $mensaje_es="Hola";
    $mensaje_en="Hello";
    $idioma = "en";
    $mensaje = "mensaje_" . $idioma;
    print $$mensaje;
?>
```



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; → a vale 8
  $b = "hola "; $b .= "mundo"; → b vale "hola mundo"
  → Equivale a $b = $b . "mundo";
- 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
- Operadores lógicos:
  and (&&), or (||), !, xor
  and/&& y or/|| tienen diferentes prioridades
- Operadores de cadena:
  concatenación: . (punto)
  asignación con concatenación: .=

Expresiones y operadores

- Precedencia de operadores (de mayor a menor):

```
++, --
*, /, %
+,-
<, <=, >, >=
==, !=
&&
||
and
or
```

Estructuras de control

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

Estructuras de control

- 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
```

- Mismo comportamiento que en C
- Las sentencias compuestas se encierran entre llaves
- elseif puede ir todo junto

Estructuras de control

- Ejemplo de estructura selectiva if-else:

```
<?PHP
   if ($sexo == 'M')
         $saludo = "Bienvenida, ";
   else
         $saludo = "Bienvenido, ";
   $saludo = $saludo . $nombre;
   print ($saludo);
?>
```

Estructuras de control

- Estructura selectiva **switch**

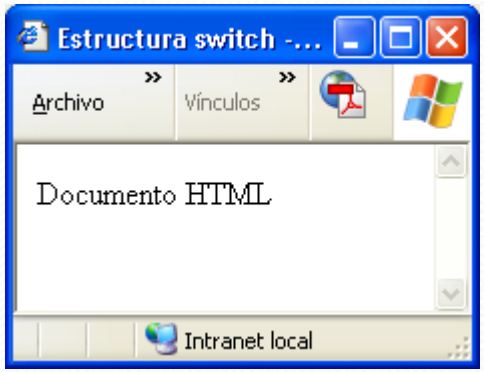
```
switch (expresión)
{
   case valor_1:
         sentencia 1
         break;
   case valor_2:
         sentencia 2
         break;
   …
   case valor_n:
         sentencia n
          break;
   default
         sentencia n+1
}
```

- Mismo comportamiento que en C, sólo que la expresión del case puede ser integer, float o string

Estructuras de control

- Ejemplo de estructura selectiva switch:

```
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);
```

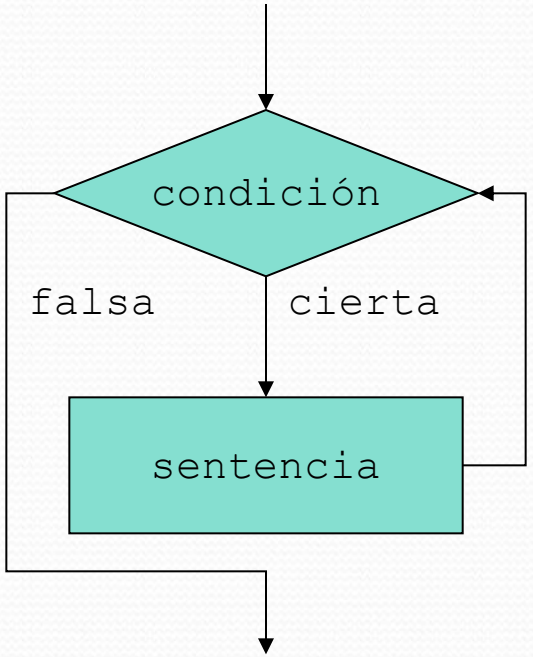


Estructuras de control

- Estructura repetitiva **while**

```
while (condición)
   sentencia
```

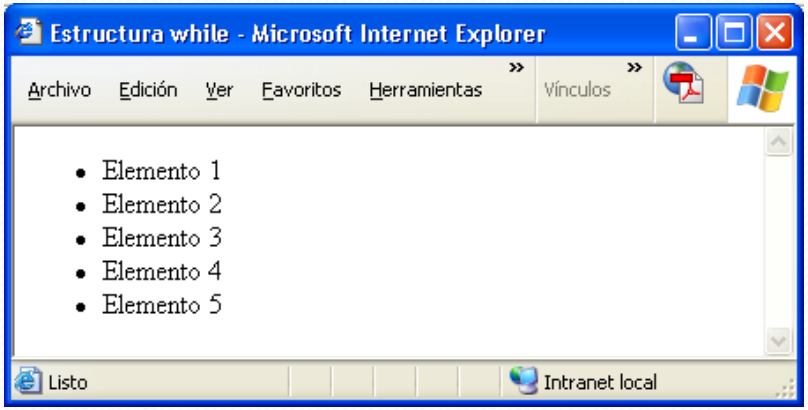
- Mismo comportamiento que en C



Estructuras de control

- Ejemplo de estructura repetitiva while:

```
<?PHP
   print ("<UL>\n");
   $i=1;
   while ($i <= 5)
   {
         print ("<LI>Elemento $i</LI>\n");
         $i++;
   }
   print ("</UL>\n");
?>
```

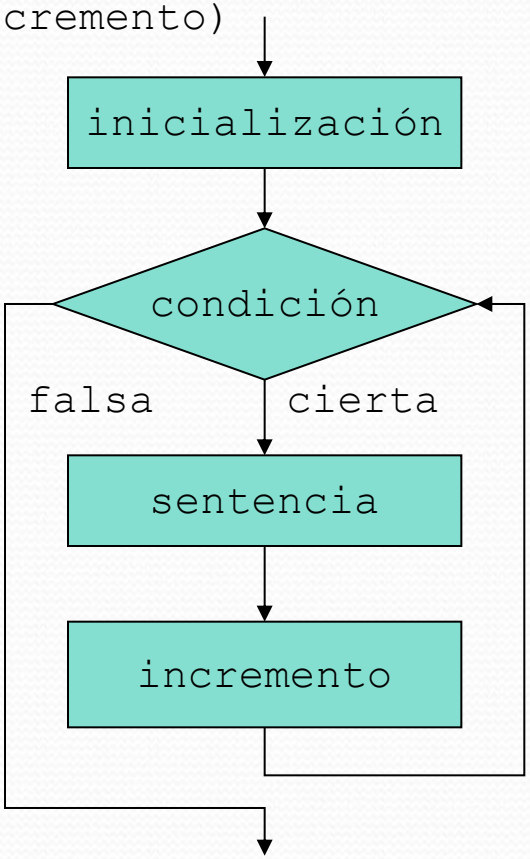


Estructuras de control

- Estructura repetitiva **for**

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

- Mismo comportamiento que en C



Estructuras de control

- Ejemplo de estructura repetitiva for:

```
<?PHP
   print ("<UL>\n");
   for ($i=1; $i<=5; $i++)
         print ("<LI>Elemento $i</LI>\n");
   print ("</UL>\n");
?>
```