





Tema 3: Características del lenguaje PHP



- Lenguajes embebidos en HTML.
- Tecnologías asociadas: PHP, ASP, JSP, Servlets, entre otras.
- Aplicaciones distribuidas basadas en componentes.
- Contenedores de servlets.
- Obtención del lenguaje de marcas a mostrar en el cliente.
- Etiquetas para inserción de código.
- Bloques de código.
- Directivas.
- Tipos de datos. Conversiones entre tipos de datos.
- Variables.
- Ámbito de utilización de las variables.



Tema 3: Características del lenguaje PHP



- Tomas de decisión.
 - Bucles.
 - Comentarios de cliente y de servidor.
 - Tipos de datos compuestos.
 - Arrays.
 - Funciones.
 - Paso de parámetros. Devolución de valores.
 - Recuperación y utilización de información proveniente del cliente Web.
 - Interacción con el usuario: formularios.
 - Procesamiento de la información introducida en un formulario.
-



Tema 3: Características del lenguaje PHP



Bloque de código PHP:

Bloque de código

<http://php.net/manual/es/>

```
<?php instrucciones y comentarios ?>
```

Instrucción

```
instrucción ;
```

Conjunto de instrucciones

```
{ instruccion1 ;  
  instruccion2 ;  
  ...  
}
```

Comentario (no se envían al navegador, los de HTML sí)

```
//Comentario de una línea
```

```
#Comentario de una línea
```

```
/*Comentario de varias líneas*/
```



Tema 3: Características del lenguaje PHP



Tipos de datos:

php.net

Simples

boolean integer float string

Compuestos

array object

Especiales

NULL resource



Tema 3: Características del lenguaje PHP



Variables:

\$nombrevariable

nombrar [declarar] [inicializar] usar

&\$nombrevariable

paso por referencia

=

operador de asignación de valor a una variable

Las variables mantienen su valor dentro de la misma página en los distintos bloques de código

Ámbito de utilización de las variables

Visibilidad de las variables

php.net

Variables locales a una función

Variables estáticas **static** \$variable para que no se pierda en sucesivas ejecuciones de una función



Tema 3: Características del lenguaje PHP



Operadores:

Asignación

= += -= .=

Aritméticos

+ - * / % ** ++ --

Relacionales

== === != < > !== < > <= >= <=> ??

Lógicos

and or xor ! && ||

Operaciones con bits

& | ^

Operaciones con string

.

php.net

- Precedencia de operadores
- Operadores aritméticos
- Operadores de asignación
- Operadores bit a bit
- Operadores de comparación
- Operadores de control de errores
- Operadores de ejecución
- Operadores de incremento/decremento
- Operadores lógicos
- Operadores para strings
- Operadores para arrays
- Operadores de tipo



Tema 3: Características del lenguaje PHP



Salida HTML:

echo echo()

print print()

printf()

sprintf()

print_r()

Especificaciones de tipo
para printf y sprintf

php.net

Cadena de texto:

"cadena" 'cadena'

"cadena con **\$variable** sustituida por su
valor"

Secuencias de escape \\ \' \" \n ...

Concatenación .

Asignación y concatenación .=

Heredoc

\$variable = <<< micadena

cadena de texto

de varias líneas

micadena;

php.net



Tema 3: Características del lenguaje PHP



Funciones relacionadas con los tipos de datos:

php.net

`gettype()`

`is_array()`

`is_bool()`

`is_float()`

`is_integer()`

`is_null()`

`settype()`

`isset()`

`unset()`

`empty()`

`define("NOMBREDECONSTANTE", valor)`

Constantes

`date()`

`date_default_timezone_set()`

`getdate()`

Fecha-Hora



Tema 3: Características del lenguaje PHP



Variables superglobales: (están disponibles en todos los ámbitos a lo largo del script)

`$_SERVER[]`

php.net

`$_REQUEST[]` `$_GET[]` `$_POST[]` `$_COOKIE[]`

`$_ENV[]`

`$_FILES[]`

`print_r()`

`$_SESSION[]`

`$GLOBALS[]`

`$GLOBALS`
`$_SERVER`
`$_GET`
`$_POST`
`$_FILES`
`$_COOKIE`
`$_SESSION`
`$_REQUEST`
`$_ENV`



Tema 3: Características del lenguaje PHP



Estructuras de control alternativas:

```
if ( Condicion )  
{ ConjuntoDeInstrucciones }
```

```
if ( Condicion )  
{ ConjuntoDeInstrucciones }  
[elseif ( Condicion )  
{ ConjuntoDeInstrucciones }]  
else  
{ ConjuntoDeInstrucciones }
```

```
switch ( Expresión ) {  
case valor1:  
    ConjuntoDeInstrucciones  
    break;  
case valor2:  
    ConjuntoDeInstrucciones  
    break;  
...  
Default:  
    ConjuntoDeInstrucciones  
}
```

php.net



Tema 3: Características del lenguaje PHP



Estructuras de control repetitivas:

```
while ( Condicion )  
{ ConjuntoDeInstrucciones }
```

```
foreach ( array as $valor )  
{ ConjuntoDeInstrucciones }
```

```
do  
{ ConjuntoDeInstrucciones }  
while ( Condicion );
```

```
foreach ( array as $clave => $valor )  
{ ConjuntoDeInstrucciones }
```

```
for ( Inicializacion; CondicionDeSalida; Incremento )  
{ ConjuntoDeInstrucciones }
```

php.net



Tema 3: Características del lenguaje PHP



Estructuras de control incondicionales:

`break;`

`continue;`

`goto Etiqueta;`

`return [Valor];`

`php.net`



Tema 3: Características del lenguaje PHP



Funciones:

Extensiones que incluyen librerías de funciones

`pecl install nombre-extensión`

activarlas en `php.ini`

php.net

Inclusión de ficheros externos

`include`

(`include_path` en `php.ini`)

php.net

`include_once`

(extensión no oficial `.ini.php` para ficheros incluidos)

`require`

`require_once`

Usar funciones

`nombrefuncion ():`

`function_exists()`

Definir funciones

```
function nombrefuncion (argumentos)
{definicion
[return valor;]
}
```

Argumentos

`&$variable` pasa el argumento por referencia

`$variable = valordefecto`
todos a la derecha en la definición de la función



Tema 3: Características del lenguaje PHP



Tipos de datos compuestos:

Array

Objeto (lo vemos más adelante)

php.net

Array (Colección - vector - matriz)

array miembro del array valor clave (numérico o asociativo)

array numérico (por defecto el índice numérico comienza en 0)

array asociativo

Definición de un array

```
$nombrearray = array (valorclave => valormiembro, ...);
```

Array unidimensional - vector

Array bidimensional - matriz

Inicialiar un array \$nombrearray[] = valor (comienza en 0 por defecto)

Recorrer un array print_r() foreach(){}

Las cadenas de texto se pueden tratar como arrays en PHP (\$cadenatexto[7])



Tema 3: Características del lenguaje PHP



Array (Colección - vector - matriz)

php.net

Recorrer un array

```
foreach($variablearray as $miembroarray)
```

```
{
```

```
//tratamiento de cada elemento (miembro) del array
```

```
}
```

```
foreach($variablearray as $codigoindice => $miembroarray)
```

```
{
```

```
//tratamiento de cada valor indice y elemento (miembro) del array
```

```
}
```




Tema 3: Características del lenguaje PHP



Array (Colección - vector - matriz)

php.net

Recorrer un array con funciones

<code>reset()</code>	Situa el puntero interno al comienzo del array
<code>next()</code>	Avanza el puntero interno una posición
<code>prev()</code>	Mueve el puntero interno una posición hacia atrás
<code>end()</code>	Sitúa el puntero interno al final del array
<code>current()</code>	Devuelve el elemento de la posición actual
<code>key()</code>	Devuelve la clave de la posición actual
<code>each()</code>	Devuelve <u>un array</u> con la clave y el elemento de la posición actual. Además avanza el puntero interno una posición.



Tema 3: Características del lenguaje PHP



Array (Colección - vector - matriz)

php.net

Definir un array con funciones

`array()`

`unset()`

Otras funciones con arrays

`array_values()`

`is_array()`

`count()`

`in_array()`

`array_search()`

`array_key_exists()`



Tema 3: Características del lenguaje PHP



Formularios (HTML):

```
<form name= "nombreform" action="fichActionWhenSubmit" method= "post"/"get">
```

...

```
<input type="text"/"radio"/"submit" name="..." value="..." >
```

```
<select>
```

```
<texarea>
```

```
<button>
```

```
<datalist>
```

```
<keygen>
```

```
<output>
```

(HTML5)

...

```
</form>
```



Tema 3: Características del lenguaje PHP



Formularios (HTML) - action

Una página para el formulario y otra página distinta para procesar los datos.

Se ejecuta el fichero destino cuando se pulsa el elemento submit

Utilizamos en el fichero destino los nombres que asignamos a los elementos del formulario

Tratamiento de los datos en la página destino

`$_REQUEST[]` `$_POST[]` `$_GET[]`

```
$inputname = $_REQUEST['inputname'];
```

```
$inputname = $_POST['inputname'];
```

```
$inputname = $_GET['inputname'];
```



Tema 3: Características del lenguaje PHP



Formularios (HTML) - action

Procesamos los datos en la misma página donde está el formulario.

Una parte del código se ejecuta después de rellenar el formulario y enviarlo y la otra parte cuando no se ha enviado.

...

```
if (isset($_POST['submitname']))  
{//Código que se ejecuta cuando se envía el formulario  
}  
else  
{//Código que se ejecuta antes de rellenar el formulario  
<form ... action="<?php echo $_SERVER['PHP_SELF'];? ... >  
...  
}
```

...



Tema 3: Características del lenguaje PHP



Formularios (HTML) - validación de datos de entrada en el servidor

Procesamos los datos en la misma página donde está el formulario pero comprobando los datos y (si no son correctos) volvemos a mostrar el formulario con los datos tecleados correctamente hasta que rellene bien los incorrectos.

```
...
(!empty($_POST['inputname'])) //Condición todo bien y rellenado
...
value="<?php echo $_POST['inputname'];?>" //Forma de cargar los
datos tecleados anteriormente
...
(isset($_POST['submitname']) && empty($_POST['inputname']))
//Condición se ha enviado el formulario pero el campo está vacío
...
(isset($_POST['inputname']) && in_array("valor", $_POST['inputname']))
//Condición se ha enviado un valor y forma parte de los valores válidos
```



Tema 3: Características del lenguaje PHP



Formularios (HTML) - validación de datos de entrada en el servidor

Pseudocódigo resumen

//Inicialización de variables

si (BotonEnviarPulsado)

//Para cada campo del formulario: Validar entrada y actuar en consecuencia

sino

//El formulario no se ha rellenado nunca

finsi

si (FormularioBienRellenado)

//Tratamiento del formulario

sino

//Mostrar el formulario hasta que lo rellenemos correctamente

finsi



Tema 3: Características del lenguaje PHP



Formularios (HTML) - validación de datos de entrada en el servidor

//Iniciación de variables

```
$entradaOK = true           //Iniciación de la variable que nos indica que todo va bien
$errores = array (tex_nombre=>""...) //Iniciación del array donde recogemos los errores
//Cargar valores por defecto en los campos del formulario
si (empty/iset($_POST['enviar'])
```

//Para cada campo del formulario: Validar entrada y actuar en consecuencia

//Validar entrada

//Limpiar campos malos

//Construir mensajes de error

//Cargar variables con los datos correctos (si es necesario)

//\$entradaOK = false cuando encontremos un error

sino

//El formulario no se ha rellenado nunca

\$entradaOK = false

finsi

si (\$entradaOK)

//Tratamiento del formulario - Tratamiento de datos OK

sino

//Mostrar el formulario hasta que lo rellenemos correctamente

//Mostrar formulario

//Mostrar los datos tecleados correctamente en intentos anteriores

//Mostrar mensajes de error (si los hay y el formulario no se muestra por primera vez)

finsi



Tema 3: Características del lenguaje PHP



Funciones interesantes

`highlight_file()` `show_source()` //Utilizada para mostrar el contenido de un fichero remarcado con la sintaxis de PHP

`setlocale(LC_ALL,"es_ES")` //Establece la información del localismo para todas las funciones a español de España

`strftime()` //Formatea una fecha/hora según la configuración local

`preg_match("patron","cadena")` //localiza caracteres en una cadena

`nl2br("cadena")` //reemplaza secuencias de escape por marcas HTML



Tema 3: Características del lenguaje PHP



Funciones interesantes

filter_input() //Toma una variable externa concreta por su nombre y opcionalmente la filtra

filter_var() //Filtra una variable con el contenido que se indique

checkdate() //Comprueba la validez de una fecha formada por los argumentos. Una fecha se considera válida si cada parámetro está propiamente definido.

Tipos de filtro: filtros de validación - filtros de saneamiento

¿Preguntas?



Mapa conceptual

