

DAW2- Desarrollo Web en Entorno Servidor



UD-3 Lenguaje PHP

Sesión 1

Índice



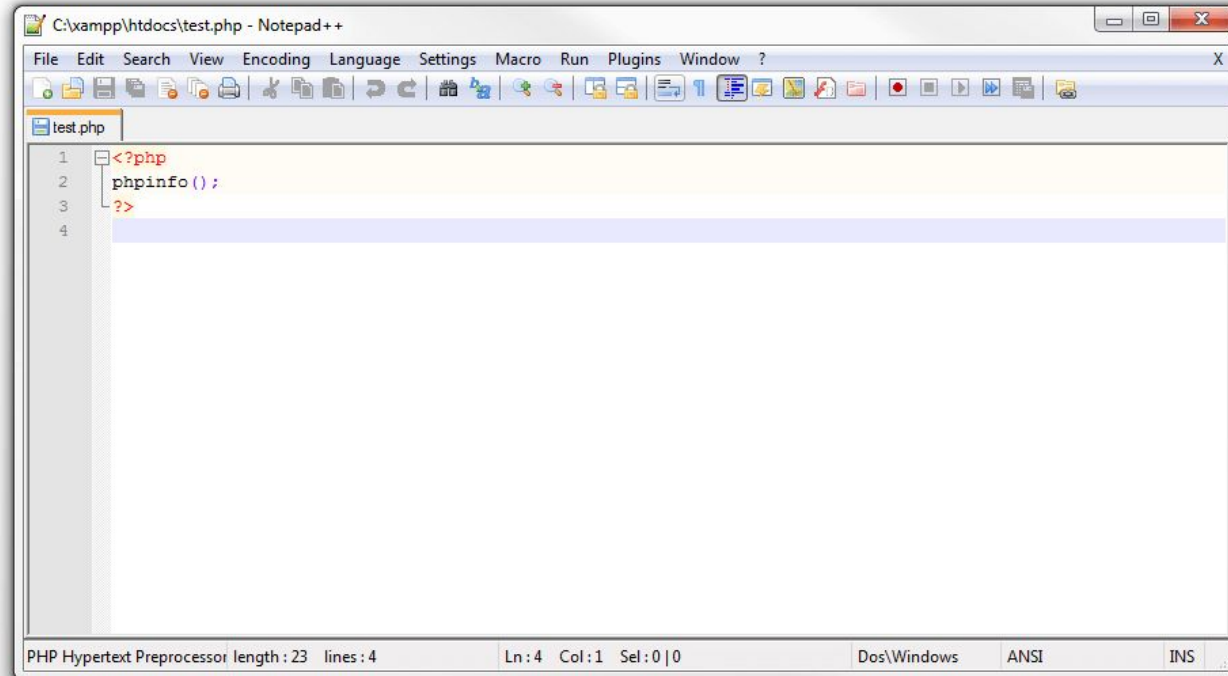
- Introducción Lenguaje PHP
- Sintaxis básica
 - Abrir Scripts
 - Comentarios
 - Variables
 - Constantes
 - Estructuras de control
 - Operadores
- Funciones auxiliares o nativas

Introducción Lenguaje PHP



- PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web.
- Es muy popular porque un gran número de páginas y portales web están creadas con PHP, además es de código abierto lo que significa que es de uso libre y gratuito para todos los programadores que quieran usarlo.
- Es un lenguaje rápido pese a ser interpretado, multiplataforma y dispone de una gran cantidad de bibliotecas para el desarrollo de aplicaciones Web.
- El estilo de programación con PHP es totalmente libre. Puedes usar tanto programación estructurada (funciones) como Programación Orientada a Objetos (clases y objetos). Incluso algunas características de la programación funcional están siendo incorporadas actualmente. Es por ello que cualquier tipo de programador puede sentirse cómodo con PHP.

Introducción Lenguaje PHP



A screenshot of a Notepad++ window titled "C:\xampp\htdocs\test.php - Notepad++". The window displays a PHP file named "test.php" with the following code:

```
1 <?php
2 phpinfo();
3 ?>
4
```

The status bar at the bottom indicates "PHP Hypertext Processor", "length: 23", "lines: 4", "Ln: 4 Col: 1 Sel: 0 | 0", "Dos\Windows", "ANSI", and "INS".

Sintaxis básica: Abrir Scripts

- El código PHP habitualmente está incrustado entre medio del código HTML. Por tanto debemos indicar de alguna forma que dejamos de escribir en código HTML y empezamos a escribir en código PHP. También debemos indicar dónde acaba el código PHP y volvemos a escribir en código HTML. Esto se hace:

```
<?php .... código php ... ?>
```

- También podemos indicar que estamos escribiendo en PHP mediante la etiqueta:

```
<script language="php">.... código php ... </script>
```

- Hay otras etiquetas para delimitar el código PHP, pero no funcionan bien con todas las versiones de "Apache" o de "XAMPP". estas son:

```
<? .... código php ... ?>
```

```
<% .... código php ... %>
```

Sintaxis básica: Delimitar sentencias



- Otra característica general de los scripts en PHP es la forma de separar las distintas instrucciones. Para hacerlo, hay que acabar cada instrucción con un punto y coma ";". Para la última expresión, la que va antes del cierre de etiqueta, este formalismo no es necesario:
 - `<?php código php ... ?>`

La función echo



- Muestra una o más cadenas
- Muestra todos los parámetros. No se añade ninguna nueva línea adicional.
- echo no es realmente una función (es una construcción del lenguaje), por lo que no se requiere el uso de paréntesis con él
- Las cadenas pueden ser pasadas individualmente como varios argumentos o concatenadas como un único argumento

Sintaxis básica: Comentarios

- Muy similar a Java, C o Javascript:

```
<?php
$mensaje="Tengo hambre!!"; //Comentario de una línea
echo $mensaje; #Este comentario también es de una línea
/*En este caso
mi comentario ocupa
varias líneas, ¿lo ves? */
?>
```


Sintaxis básica: Ejemplo completo página



```
<!doctype html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Primera página PHP</title>
</head>
<body>
    <h1>Esto es HTML</h1>
    <?php
    echo '<p>Esto viene de PHP</p>';
    ?>
</body>
</html>
```

Sintaxis básica: Variables

- El nombre de las variables:
 - Empiezan por el símbolo dólar (\$) y es quizás una de las características más fuertes del lenguaje.
 - Pueden estar compuestas por caracteres alfanuméricos o el carácter subrayado “_”.
 - Se distinguen entre mayúsculas y minúsculas
 - De forma predeterminada, las variables siempre se asignan por valor (se copia) pero también se podría hacer por referencia.

```
<?php
```

```
$var = 'Roberto';  
$Var = 'Juan';  
echo "$var, $Var";      // imprime "Roberto, Juan"
```

```
$4site = 'aun no';      // inválido; comienza con un número  
$_4site = 'aun no';     // válido; comienza con un carácter de subrayado  
$täyte = 'mansikka';    // válido; 'ä' es ASCII (Extendido) 228
```

```
?>
```

Sintaxis básica: Variables

- Declaración de variables
 - De forma predeterminada, las variables siempre se asignan por valor (se copia) pero también se podría hacer por referencia.
- En caso de que se omita el valor inicial, la variable será inicializada a cero si corresponde a un tipo numérico; a una cadena de carácter vacío si es del tipo String; al valor null si es del tipo Object, y a false si es del tipo bool.

```
<?php
```

```
$foo = 'Bob';           // Asigna el valor 'Bob' a $foo
$bar = &$foo;           // Referenciar $foo vía $bar.
$bar = "Mi nombre es $bar"; // Modifica $bar...
echo $bar;
echo $foo;              // $foo también se modifica.
```

```
?>
```

Sintaxis básica: Variables

- Variables asignadas por referencia:
 - En PHP también podemos asignar variables por referencia, aunque a decir verdad no es una característica que se use mucho.
 - La notación para asignar por referencia es colocar un "&" antes del nombre de la variable.

<?php

```
$foo = 'Bob'; // Asigna el valor 'Bob' a $foo
$bar = &$foo; // Referencia $foo vía $bar.
$bar = "Mi nombre es $bar"; // Modifica $bar...
echo $foo; // $foo también se modifica.
echo $bar;
```

?>

Sintaxis básica: Tipos de datos



- Tipos simples:
 - boolean para valores lógicos
 - integer para números enteros
 - float (número de coma flotante, también conocido como double)
 - string para cadenas de caracteres
- Tipos compuestos:
 - array para mapas, matrices y vectores
 - object para instancias de clases
- Tipos especiales:
 - resource para referencia a datos externos
 - NULL para representar datos sin valor

Sintaxis básica: Tipos de datos

- Tipos numéricos:
 - Integer (entero): números enteros (1, 2, 3, 4, 5, 6, 7, 8, 9...).

```
<?php
```

```
$a = 1234; // número decimal
```

```
$a = -123; // un número negativo
```

```
$a = 0123; // número octal (equivalente a 83 decimal)
```

```
$a = 0x1A; // número hexadecimal (equivalente a 26 decimal)
```

```
?>
```

- float (reales): números en coma flotante (con decimales) (1,45 3,89875 8,7724 etc).

```
<?php
```

```
$a = 1.234;
```

```
$b = 1.2e3;
```

```
$c = 7E-10;
```

```
?>
```

Sintaxis básica: Tipos de datos



- Tipos alfanuméricos o cadenas
 - Un string es una serie de caracteres donde un carácter es lo mismo que un byte. Esto significa que PHP solo soporta el conjunto de 256 caracteres y por lo tanto no tiene soporte nativo Unicode.
 - Puede ser especificado de cuatro formas diferentes:
 - Comillas simples '...'
 - Comillas dobles "..."
 - Sintaxis heredoc (comillas dobles en bloque)
 - Sintaxis nowdoc (comillas simples en bloque)

Sintaxis básica: Tipos de datos

- Tipos alfanuméricos o cadenas
 - Comillas simples "
 - Esta sintaxis no se puede utilizar para representar caracteres de escape como avance de línea, salto de carro, tabuladores, etc...

```
<?php
// Cadena de caracteres
$cadena='Cadena de caracteres';
// Me llamo 'Luis Roca' y mi web es minubeinformatica.com
$cadena='Me llamo \'Luis Roca\' y mi web es minubeinformatica.com';
// El archivo se encuentra en c:\wamp64\www
$cadena='El archivo se encuentra en c:\\wamp64\\www';
?>
```


Sintaxis básica: Tipos de datos

- Tipos alfanuméricos o cadenas
 - Comillas dobles ""
 - Si un string está encerrado entre comillas dobles (""), PHP interpretará más sentencias de escape como caracteres especiales.

```
<?php
echo "Esto es una cadena sencilla";

// Resultado: El nombre de mi web es: "MiNubeInformatica"
echo "El nombre de mi web es: \"MiNubeInformatica\"";
$var1 = 10;
$var2 = 5;
$suma = $var1+$var2;
// Resultado: Las variables $var1 y $var2 contienen 10 y 5. Sumadas dan como resultado: 15
echo "Las variables \$var1 y \$var2 contienen $var1 y $var2. Sumadas dan como resultado: $suma";
?>
```

Sintaxis básica: Tipos de datos



- Secuencias de escape:

<code>\n</code>	Salto de línea
<code>\r</code>	Retorno de carro
<code>\t</code>	Tabulador horizontal
<code>\v</code>	Tabulador vertical (incluido en PHP 5.2.5)
<code>\e</code>	Tecla de escape (incluido en PHP 5.4.4)
<code>\f</code>	Tecla de avance de página (incluido en PHP 5.2.5)
<code>\\</code>	Contrabarra o barra invertida
<code>\\$</code>	Símbolo de dólar
<code>\"</code>	Comillas dobles

Sintaxis básica: Tipos de datos

- Tipos booleanos
 - Este es el tipo más simple. Un boolean expresa un valor lógico. Puede ser true (verdadero) o false (falso).

```
<?php
```

```
$Correcto = true; // Asigna true a la variable $Correcto
```

```
?>
```

Sintaxis básica: Tipos de datos

- Tipo array:
 - Un array puede ser creado con el constructor del lenguaje array(). Éste toma cualquier número de parejas clave => valor como argumentos.
 - array(clave => valor, clave2 => valor2, clave3 => valor3, ...)

```
<?php
```

```
$array = array(  
    "foo" => "bar",  
    "bar" => "foo",  
);
```

```
// a partir de PHP 5.4
```

```
$array = [  
    "foo" => "bar",  
    "bar" => "foo",  
];  
?>
```

Sintaxis básica: Tipos de datos



- Tipo Objeto:
 - En PHP a parte de datos simples podemos trabajar con objetos. Objetos que tengan sus propiedades y métodos. Podemos crear un objeto en PHP a partir de una clase, como se haría de una manera formal, si bien podemos crear un objeto en PHP de forma directa, sin clase asociada.
 - Para crear un nuevo object, utilice la sentencia new para instanciar una clase:

```
$libro = new stdClass();
```

Sintaxis básica: Tipos de datos



- Tipo Nulo:
 - El valor especial NULL representa una variable sin valor. NULL es el único valor posible del tipo null.
 - Una variable es considerada null si:
 - Se le ha asignado la constante NULL.
 - No se le ha asignado un valor todavía.
 - Se ha destruido con unset().

```
<?php  
$var = NULL;  
?>
```

Sintaxis básica: Tipos de datos

- PHP tiene tipado dinámico
 - A diferencia de otros lenguajes, PHP posee una gran flexibilidad a la hora de operar con variables. En efecto, cuando definimos una variable asignándole un valor, el ordenador le atribuye un tipo. Si por ejemplo definimos una variable entre comillas, la variable será considerada de tipo cadena:

```
<?
$cadena="5"; //esto es una cadena
$entero=3; //esto es un entero
echo $cadena+$entero
?>
```

- Este script dará como resultado "8"

Sintaxis básica: Constantes

- Las constantes
 - Existen casos en los que necesitamos almacenar un valor en concreto, pero sabemos que ese valor no se va a modificar en ningún momento de la ejecución del programa. Para ello usamos una constante.
 - La forma de declararla es la siguiente:

```
<?php
define ("constante", "valor");
echo (constante);
?>

<?php
define ("pi", "3,14159265358979323846");
echo (pi);
?>
```


Sintaxis básica: Enumerados

- Las enumeraciones
 - PHP no tiene un tipo de enumeración nativo. Hay diferentes formas de hacerlo pero una popular es:

```
<?php
abstract class DaysOfWeek
{
    const Sunday = 0;
    const Monday = 1;
    // etc.
}

$today = DaysOfWeek::Sunday;
?>
```

Sintaxis básica: Estructuras de control



- Instrucción if
 - La instrucción if selecciona una instrucción para ejecución en base al valor de una expresión Boolean.
 - Sintaxis:

```
if (condición)  
{  
instrucciones  
}  
else  
{  
instrucciones  
}
```

Sintaxis básica: Estructuras de control



- Instrucción Switch
 - La instrucción switch es una instrucción de control que controla múltiples selecciones y enumeraciones pasando el control a una de las instrucciones case de su cuerpo.
 - Sintaxis:

```
switch (expresion)  
{  
  case constante 1:  
    instrucciones
```

```
  case constante 2:  
    instrucciones
```

```
  default;  
    instrucciones  
  break;  
}
```

Sintaxis básica: Estructuras de control

- Instrucción While
 - La instrucción while ejecuta una instrucción o un bloque de instrucciones repetidamente hasta que una expresión especificada se evalúa como false.
 - Sintaxis

while (expr):
sentencias

...

<?php

```
$i = 1;
while ($i <= 10) {
    echo $i++;
}
?>
```

Estructuras de control



- Instrucción For
 - La instrucción for ejecuta una instrucción o un bloque de instrucciones mientras una expresión booleana especificada se evalúa como true.
 - Sintaxis:

for (initializer; condition; iterator)
body

```
<?php
for ($i = 1; $i <= 10; $i++) {
    echo $i;
}
```

Sintaxis: Estructuras de control

- Instrucción Foreach, in
 - El constructor foreach proporciona un modo sencillo de iterar sobre arrays. foreach funciona sólo sobre arrays y objetos.
 - Sintaxis:

```
foreach (expresión_array as $valor)  
sentencias  
foreach (expresión_array as $clave => $valor)  
sentencias
```

```
<?php  
$array = array(1, 2, 3, 4);  
foreach ($array as &$amp;valor) {  
    $valor = $valor * 2;  
}
```

Sintaxis: Operadores



- Asignación
- Aritméticos
- Comparación
- Incremento y decremento
- Lógicos
- De concatenación

Sintaxis: Operadores

- Operador de asignación
 - El operador básico de asignación es "=".
 - El valor de una expresión de asignación es el valor asignado. Es decir, el valor de "\$a = 5" es de 5.
- Además del operador básico de asignación, existen "operadores combinados" para todos los de aritmética binaria, unión de arrays y operadores de strings que permiten usar un valor en una expresión:

```
<?php
$a = 3;
// Asigna el valor 8 a la variable $a, como si se hubiéramos hecho: $a = $a + 5;
$a += 5;
$b = 'Hola ';
// Establece $b en 'Hola ahí!', utilizando el operador de concatenación de strings al igual que
$b=$b.'ahí!';
$b .= 'ahí!';
?>
```


Sintaxis: Operadores



Operadores aritméticos

| Ejemplo | Nombre | Resultado |
|----------|----------------|---|
| $-a$ | Negación | Opuesto de a . |
| $a + b$ | Adición | Suma de a y b . |
| $a - b$ | Sustracción | Diferencia de a y b . |
| $a * b$ | Multiplicación | Producto de a y b . |
| a / b | División | Cociente de a y b . |
| $a \% b$ | Módulo | Resto de a dividido por b |
| $a ** b$ | Exponenciación | Resultado de elevar a a la potencia b |

Sintaxis: Operadores



| Operadores de comparación | | |
|-------------------------------|-------------------|--|
| Ejemplo | Nombre | Resultado |
| <code>\$a == \$b</code> | Igual | TRUE si \$a es igual a \$b después de la manipulación de tipos. |
| <code>\$a === \$b</code> | Idéntico | TRUE si \$a es igual a \$b, y son del mismo tipo. |
| <code>\$a != \$b</code> | Diferente | TRUE si \$a no es igual a \$b después de la manipulación de tipos. |
| <code>\$a <> \$b</code> | Diferente | TRUE si \$a no es igual a \$b después de la manipulación de tipos. |
| <code>\$a !== \$b</code> | No idéntico | TRUE si \$a no es igual a \$b, o si no son del mismo tipo. |
| <code>\$a < \$b</code> | Menor que | TRUE si \$a es estrictamente menor que \$b. |
| <code>\$a > \$b</code> | Mayor que | TRUE si \$a es estrictamente mayor que \$b. |
| <code>\$a <= \$b</code> | Menor o igual que | TRUE si \$a es menor o igual que \$b. |
| <code>\$a >= \$b</code> | Mayor o igual que | TRUE si \$a es mayor o igual que \$b. |

Sintaxis: Operadores



Operadores de incremento / decremento

| Ejemplo | Nombre | Efecto |
|--------------------|-----------------|--|
| <code>++\$a</code> | Pre-incremento | Incrementa a en uno, y luego retorna a . |
| <code>\$a++</code> | Post-incremento | Retorna a , y luego incrementa a en uno. |
| <code>--\$a</code> | Pre-decremento | Decrementa a en uno, luego retorna a . |
| <code>\$a--</code> | Post-decremento | Retorna a , luego decrementa a en uno. |

Sintaxis: Operadores



| Operadores lógicos | | |
|---------------------------------|-------------------|---|
| Ejemplo | Nombre | Resultado |
| <code>\$a and \$b</code> | And (y) | TRUE si tanto <i>\$a</i> como <i>\$b</i> son TRUE. |
| <code>\$a or \$b</code> | Or (o inclusivo) | TRUE si cualquiera de <i>\$a</i> o <i>\$b</i> es TRUE. |
| <code>\$a xor \$b</code> | Xor (o exclusivo) | TRUE si <i>\$a</i> o <i>\$b</i> es TRUE, pero no ambos. |
| <code>! \$a</code> | Not (no) | TRUE si <i>\$a</i> no es TRUE. |
| <code>\$a && \$b</code> | And (y) | TRUE si tanto <i>\$a</i> como <i>\$b</i> son TRUE. Se evalúa antes que el operador and |
| <code>\$a \$b</code> | Or (o inclusivo) | TRUE si cualquiera de <i>\$a</i> o <i>\$b</i> es TRUE. Se evalúa antes que el operador or |

Sintaxis: Operadores



- Concatenación
 - En PHP nos encontramos sólo dos operadores para el trabajo con cadenas (string). El primero es el operador de concatenación ('.'), el cual retorna el resultado de concatenar sus argumentos derecho e izquierdo. El segundo es el operador de asignación sobre concatenación ('.='), el cual añade el argumento del lado derecho al argumento en el lado izquierdo.

Sintaxis: Operadores



```
<?php
```

```
$a = 'Mi casa';  
$b = ' es grande';  
$c = ' es pequeña';
```

```
echo '<h2>OPERADORES PARA STRINGS</h2>'  
    "<p>El valor de \$a es: '$a'<br />",  
    "El valor de \$b es: '$b'<br />",  
    "El valor de \$c es: '$c'</p>";
```

```
$d=$a.$c;  
$a.=$b;
```

```
echo "Hemos declarado la variable \$d como \$a.$c y su valor es: '$d'<br/>",  
    "Hemos redefinido la variable \$a como .=\$b y su valor es: '$a'<br/>";
```

```
?>
```

Funciones auxiliares o nativas



- Funciones para principiantes más usadas
 - <https://jesusfernandeztoledo.com/funciones-predefinidas-en-php-mas-importantes/>
- Listado de todas las funciones, manual oficial
 - <https://www.php.net/manual/es/indexes.functions.php>

Ejecutar ficheros PHP por consola

- Los archivos de PHP pueden ejecutarse desde la terminal.
 - La forma de hacerlo sería:
 - `php archivo.php`
- Al script le podemos pasar argumentos. Por ejemplo:
 - `php archivo.php --nombre="luis"`
 - `php archivo.php luis`
- Para acceder a los argumentos hacemos uso de la variable `$argv` y el número de argumentos que nos pasan está en la variable `$argc`.

```
<?php
echo "Hay $argc argumentos\n";
foreach ($argv as $argumento) {
    echo "Recibido un argumento: $argumento \n";
}
```


Entrada por consola

- `$argv` — Array de argumentos pasados a un script
- Si se desea leer una sola línea de *stdin*, puede usarse:

```
<?php
$line = trim(fgets(STDIN)); // lee una línea de STDIN
fscanf(STDIN, "%d\n", $number); // lee un número de STDIN
?>
```

- O bien la función `readline`:

```
<?php
$num = readline("Introduzca un número: ");
?>
```

Recursos



- <https://www.ionos.es/digitalguide/paginas-web/creacion-de-paginas-web/tutorial-de-php-fundamentos-basicos-para-principiantes/>
- <http://minubeinformatica.com/cursos/programacion-en-php/tipos-de-datos-variables-y-constantes>
- <https://www.php.net/manual/es/language.variables.basics.php>
- <https://www.php.net/manual/es/features.commandline.php>
- <https://www.hostinger.es/tutoriales/que-es-php/>
- <https://www.gestionatuweb.net/curso-de-php/3-operadores-en-php/>
- <https://desarrolloweb.com/manuales/manual-php.html>
- <https://diego.com.es/estructuras-de-control-en-php>
- <https://desarrolloweb.com/articulos/312.php>