Bibliografía:

Implantación de Aplicaciones Web, Jorge Sánchez Asenjo

PHP con acceso a bases de datos, Félix Mateos.

TEMA 2. FUNDAMENTOS DEL LENGUAJE

Nociones básicas

- Un script php se escribe en un documento html dentro de la siguiente etiqueta:

  <?php

  Contenido del script

  ?>
- El final de cada instrucción PHP se designa con un punto y coma (;).
- Los bloques de instrucciones deben ir entre llaves {}.
- El intérprete PHP ignora los espacios, los tabuladores y los saltos de línea; considera que todo lo que escribamos corresponde a una misma instrucción hasta que encuentre un punto y coma;.
- Al intérprete PHP no le importa que escribamos los nombres de las instrucciones y de las funciones en mayúsculas o minúsculas o mezclando mayúsculas o minúsculas; por ejemplo, echo funciona igual de bien que ECHO o eChO. Sin embargo los nombres de variables y constantes debemos escribirlos siempre del mismo modo. Se aconseja utilizar mayúsculas para los nombres de las constantes y minúsculas para los nombres de las variables.
- // sirve para crear comentarios de una sola línea.
- /* y */ sirven para crear comentarios de varias líneas.

Ejemplo script php

```
<html>
<body>
 <?php
 $myvar = "Hola. Este es mi primer script en PHP \n";
 /*Esto es un comentario es mi primer script en PHP \n";
Termina aquí*/
//Esto es otro comentario
echo $myvar;
```

```
?>
</body>
</html>
```

Variables

En PHP todas las variables deben empezar con el signo $, seguido de una letra o un signo de subrayado, seguido de cualquier combinación de letras, números y signos de subrayado.

PHP es sensible al uso de mayúsculas/minúsculas en las variables; por ejemplo, $nombre y $Nombre se interpretan como dos variables diferentes.

No estamos obligados a especificar el tipo de datos. Una misma variable puede contener un número en un punto del script y una cadena de caracteres en otro punto diferente. Cuando definimos una variable asignándole un valor, le atribuye un tipo. Si por ejemplo definimos una variable entre comillas, la variable será considerada de tipo cadena:

```
$variable="5"; //esto es una cadena
```

Sin embargo si pedimos en nuestro script realizar una operación matemática con esta variable, no obtendremos un mensaje de error sino que la variable cadena se convierte en numérica

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

Se puede asignar el valor de una variable a otra:

```
$a=7;
$b=$a;
echo $a; // Muestra 7
$b=$b+8;
echo $b; // Muestra 15
echo $a; // Muestra 7
```

No se puede usar una variable que no ha sido definida, por ejemplo

```
<?php
echo $hola;
?>
```

Mostraría el error: `undefined variable $hola in line …`

Se puede usar la función isset() para saber si una variable ha sido definida:

`isset($a)` devuelve true si la variable ha sido definida y false si no.

Cadenas de *caracteres*

En PHP un dato de tipo cadena de caracteres debe ir escrito (preferiblemente) entre comillas simples, pero también puede ir entre comillas dobles.

Para incluir en una cadena de caracteres el carácter de delimitación ' tendremos que escribirlo precedido por el carácter \ . La anteposición del carácter \ a otro carácter se denomina escapado. Para incluir el carácter \ tendremos que escribirlo \\

\n salto de línea

\t salto de tabulador

También se pueden usar comillas dobles, en este caso el intérprete busca dentro de la cadena nombres de variables y, si los encuentra, los sustituye por sus correspondientes valores.

Al usar comillas dobles, los caracteres que requieren escapado para poder mostrarse correctamente son \\, \" y \$.

Si no necesitamos esta funcionalidad de sustitución de variables dentro de una cadena de caracteres es preferible usar comillas simples porque se interpretan más rápido.

Para concatenar dos cadenas de caracteres se utiliza el operador punto (.)

```
<?php
$cadena1='hola';
$cadena2=' que tal';
echo "$cadena1"."$cadena2"; // Muestra:hola que tal
echo '$cadena1'.'$cadena2'; //Muestra:$cadena1$cadena2
?>
```

Se pueden incrustar etiquetas html dentro del código PHP

```
<?php
    echo "Estoy estudiando <h1>Programación</h1>"
?>
```

Tipos de datos

- **Booleanos** o lógicos (bool): sólo pueden tener el valor true o false
- **Números enteros** (integer, int): Cualquier número que pertenezca al grupo de los enteros (cualquier número positivo o negativo incluido el cero), dentro de los límites de representación que permita la arquitectura del ordenador.

- **Números de coma flotante** (float,double): Cualquier número que pertenezca al grupo de los reales. Se usa punto para los decimales, por ejemplo $n=213.4;
- **Cadenas de caracteres** (string).

PHP nos ofrece la función var_dump para obtener información sobre una variable, dato o expresión. Devuelve el tipo del dato y su valor, y NULL si la variable no está definida.

Podemos forzar la conversión (casting) de tipos. Se escribe entre paréntesis delante del dato, variable o expresión el tipo al que queremos forzarlo.

```
<?php
$real=4.5;
$entera=(int)$real;
echo $entera;  // Muestra 4

var_dump($real); // Muestra float:4.5
var_dump($entera); //Muestra int:4
?>
```

Reglas automáticas de conversión de tipos de datos

PHP nos permite combinar datos de distintos tipos en una misma operación. Esto es porque de forma interna y automática se lleva a cabo una conversión de datos.

Se comentan algunas de las reglas automáticas de conversión que sí conviene recordar porque pueden ahorrarnos escribir mucho código:

- En operaciones lógicas, los datos NULL, 0, '0' y '' se consideran FALSE. Cualquier otro dato se considera TRUE (incluida la cadena 'FALSE').
- En operaciones aritméticas no unitarias las cadenas se intentan leer como números y, si no se puede, se convierten en 0, TRUE se convierte en 1, y FALSE se convierte en 0.
- En operaciones de comparación, si un operando es un número, el otro también se convertirá en un número. Sólo si ambos operandos son cadenas se compararán como cadenas de caracteres.

Constantes

Las constantes son nombres que se refieren a un dato que no cambia a lo largo de la ejecución del script. Son reconocidas en el script en el que son definidas tanto dentro como fuera de las funciones. Son globales.

No se asignan con el operador =, sino con la función define

define(*nombre_constante_entre_comillas*,*dato_constante*)

```
<?php
define("PI", 3.14);
echo PI;
?>
```

Los nombres de las constantes no van precedidos del signo $, por convenio se escriben completamente en mayúsculas.

PHP incluye algunas constantes predefinidas, y entre ellas destacan algunas denominadas popularmente *constantes mágicas*, cuyo valor es asignado por el propio intérprete y varía en función de dónde sean utilizadas (no son verdaderas constantes), como:

- __FILE__: Contiene la ruta completa del archivo en el que se encuentra el script.
- __DIR__: Contiene el nombre de la carpeta en la que se encuentra el archivo del script.
- __LINE__: Contiene la línea que se está ejecutando en ese momento exacto del fichero actual
- __FUNCTION__ :Contiene el nombre de la función que se está ejecutando actualmente
- PHP_INT_MAX: Contiene el máximo entero utilizable por el sistema.

Ejecutar el siguiente script:

```
<?php
echo __FILE__;
echo '<br>';
echo __DIR__;
echo PHP_INT_MAX;?>
```

Operadores básicos

Las operaciones no se realizarán de izquierda a derecha sino que se realizarán siguiendo un orden de preferencia que PHP asigna a cada operando, estas son las preferencias de PHP en cuanto a los operadores:

new
[
! ~ ++ -- (int) (float) (string) (array) (object) @
* / %
+ - .
<< >>

< <= > >=
== != === !==
&
^
|
&&
||
? :
= += -= *= /= .= %= &= |= ^= <<= >>=
and
or

- Aritméticos :
  - +: Suma
  - -: Resta
  - *: Multiplicación
  - /: División (float)
  - %: módulo
  - ++: Incremento unitario
  - --: Decremento unitario
  - +=, -=, *=, /=

- De comparación:
  - == : Igualdad (compara valor y no tipo)
  - ===: idéntico (compara valor y tipo)
  - !=, <>: Desigualdad
  - ¡==: no idénticos
  - <: Menor que
  - >: Mayor que
  - <=: Menor o igual que
  - >=: Mayor o igual que

- Lógicos :
  - !: negación
  - and, &&: Y
  - or,||: O

- Operador asignación =
- Operador concatenación.
- (.=), este operador concatena a una variable un valor dado de la siguiente manera

```
<?php
$html  = "<html>\n";
$html .= "<head>\n";
$html .= "<title>Mi página</title>\n";
$html .= "</head>\n";
$html .= "<body>\n";
$html .= "Este es el contenido de mi\n";
$html .= " página\n";
$html .= "</body>\n";
$html .= "</html>\n";
echo $html;
?>
```