

02.1. Elementos básicos de PHP

1º DAW - Programación
David Soler Talens

Variables en PHP (I)



- **Debemos dar un nombre que**

- Una variable comienza con el signo \$, seguido del nombre de la variable
- El nombre de una variable debe comenzar con una letra o el carácter de subrayado
- Un nombre de variable no puede empezar con un número
- Un nombre de variable sólo puede contener caracteres alfanuméricos y guiones bajos (A-z, 0-9, y _)
- Los nombres de las variables distinguen entre mayúsculas y minúsculas (\$age y \$Age son dos variables diferentes)

Variables en PHP (II)



```
1  <?php
2
3      $var01 = 12;
4      $var02 = 15.3;
5      $var03 = "hola";
6      $var04 = 'Adios';
7      $var05 = true;
8      $var06 = false;
9
10 ?>
```

El signo “=” sirve para dar valor a una variable

Variable numérica

Variable numérica decimal

Variable cadena texto (*string*)

Variable cadena texto, se pueden usar las dobles comillas o las simples

Variables booleanas

ATENCIÓN! En PHP cada línea debe acabar con punto y coma

Constantes en PHP



- **Las constantes son como las variables, excepto que una vez definidas NO SE PUEDE CAMBIAR SU VALOR.**
- Los nombres de constante válido comienzan con una letra o un guion bajo (sin signo \$).
- Un ejemplo de constante seria el número PI
- Se definen de la siguiente manera:

```
1  <?php
2
3      define("PI", 3.14);
4
5  ?>
```

Operadores en PHP (I)



- **Procesan acciones básicas sobre datos**
- **tipos**
 - Operadores aritméticos o matemáticos
 - Operadores de asignación (dan valor a variables)
 - Operadores de comparación entre datos
 - Operadores para Incrementar/Disminuir valores
 - Operadores lógicos o de trabajo con booleanos
 - Operadores de cadenas de texto

Operadores en PHP (II)



• Operadores aritméticos

```
1  <?php
2
3  $x=7;
4  $y=5;
5
6  $resultado = $x + $y;
7  $resultado = $x - $y;
8  $resultado = $x * $y;
9  $resultado = $x / $y;
10 $resultado = $x % $y;
11 $resultado = $x ** $y;
12
13 ?>
```

\$resultado vale 12 (suma)

\$resultado vale 2 (resta)

\$resultado vale 35 (multiplicación)

\$resultado vale 1 (división entera)

\$resultado vale 2 (resto)

\$resultado vale 16807 (potencia)

Operadores en PHP (III)



• Operadores de incremento/decremento

```
1  <?php
2
3  $x=7;
4
5  $resultado = $x++;
6  $resultado = $x--;
7  $resultado = ++$x;
8  $resultado = --$x;
9
10 ?>
```

\$resultado vale:

→ 7, después \$x pasa a valer 8

→ 8, después \$x pasa a valer 7

→ 8, previo incremento de \$x

→ 7, previo decremento de \$x



Operadores en PHP (IV)

- **Operadores aritméticos que simplifican cálculos cuando actuamos sobre la misma variable**

1	<code><?php</code>	
2		
3	<code>\$x=11;</code>	
4		
5	<code>\$x += 3;</code>	→ \$x ahora vale 14 ($\$x=\$x+3$)
6	<code>\$x -= 3;</code>	→ \$x ahora vale 11 ($\$x=\$x-3$)
7	<code>\$x *= 3;</code>	→ \$x ahora vale 33 ($\$x=\$x*3$)
8	<code>\$x /= 3;</code>	→ \$x ahora vale 11 ($\$x=\$x/3$)
9	<code>\$x %= 3;</code>	→ \$x ahora vale 2 ($\$x=\$x\%3$)
10	<code>\$x **= 3;</code>	→ \$x ahora vale 8 ($\$x=\x^3)
11		
12	<code>?></code>	

Operadores en PHP (V)



• Operadores de cadena

```
1  <?php
2
3  $x="hola,";
4
5  $resultado = $x." que tal?";
6  $resultado .= " bien, gracias";
7
8  ?>
```

\$resultado vale:

→ "hola, que tal?"

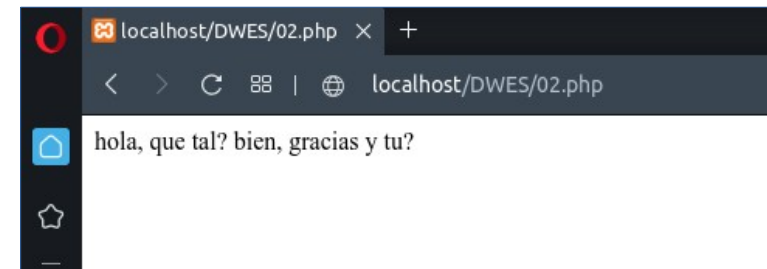
→ "hola, que tal? Bien, gracias"



A tener en cuenta en PHP (I)

- **Un script PHP comienza con `<?php` y termina con `?>`**
- **Cada linea termina en `;`**
- **Hay dos formas básicas de conseguir la salida por pantalla:**
 - *echo*
 - *print*

```
1  <?php
2
3  $x="hola,";
4
5  $resultado = $x." que tal?";
6  $resultado .= " bien, gracias";
7
8  echo $resultado;
9  print " y tu?";
10
11 ?>
```



Pantalla del navegador

A tener en cuenta en PHP (II)



- **Los comentarios son**

- Fragmentos de código que *NO se ejecuta*
- Su función es poder insertar texto entre las líneas de código
- Este texto ayuda a documentar qué es lo que hace el código
- Es de gran ayuda para poder volver sobre un programa que desarrollaste hace mucho y no recuerdas porque lo hiciste así
- También sirve de ayuda cuando el programa cambia de desarrollador.
- También los podemos usar para anular “*temporalmente*” un fragmento de código

- **ES MUY IMPORTANTE DOCUMENTAR EL CÓDIGO**

A tener en cuenta en PHP (III)



```
1  <?php
2
3  $x="hola,";
4
5  $resultado = $x." que tal?";
6  $resultado .= " bien, gracias";
7
8  echo $resultado; //salida por pantalla
9  print " y tu?"; /* print es equivalente
10                  y en este caso lo usamos
11                  de igual forma */
12
13  ?>
```

→ comentarios de 1 línea
→ comentarios en los que usamos varias líneas