

UD-02a: Introducción a PHP

Desarrollo Web en Entorno Servidor

Curso 2020/2021

Introducción

PHP es un lenguaje que se ejecuta en el ordenador servidor.
NO se ejecuta en el ordenador cliente.

Editores que se podrían utilizar para escribir PHP:

- **Windows:** Sublime Text, Notepad++, NetBeans,...
- **Mac:** TextMate, Bbedit, Sublime Text,...
- **Linux:** Gedit, Kate,...

Introducción

Delimitadores `<?php` y `?>`

En una misma página, normalmente habrá varios bloques de código php, que siempre se diferencian mediante estos delimitadores.

Si el bloque genera una salida, se colocará dentro del HTML en el lugar donde queremos que aparezca.

Introducción

El bloque de php siempre tiene que ir entre las etiquetas <body> y </body>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Documento sin título</title>
</head>
<body>

<?php                               ← Inicio de Bloque de php
//bloque de php
//bloque de php
//bloque de php
//bloque de php
//bloque de php
//bloque de php

?>                                  ← Final del Bloque de php

</body>
</html>
```

Introducción

Ejemplo 1. Impresión en pantalla de la información de la versión de php instalada.

```
<?php  
    phpinfo();  
?>
```

Ejemplo 2. Impresión de un texto por pantalla.

```
<?php  
    echo "esta es la información del módulo php";  
?>
```

Ejemplo 3. Impresión de variables por pantalla .

```
<?php  
    $nombre = "Juan";  
    $edad = 18;  
    echo $nombre, $edad;  
?>
```

Características básicas del lenguaje

Comentarios

- Comentarios de una línea → `// texto aquí`
- Comentarios de una línea (otra forma) → `# texto aquí`
- Comentarios de varias líneas → `/* texto aquí`
`texto aquí */`

`|`
No se mostrarán en la web que se envíe al navegador.

Características básicas del lenguaje

Variables en PHP

- Siempre comienzan con el signo \$
- Sus nombres deben empezar por una letra o por el carácter _
- También pueden contener números
- Al declararla, no se especifica el tipo de dato, simplemente se le asigna directamente un valor con el operador =

```
$mi_variable = 7;
```

Características básicas del lenguaje

Ejemplo

// Al asignarle el valor 7, la variable es de tipo “entero”

\$mi_variable = 7;

// Si le cambiamos el contenido la variable puede cambiar de tipo

// En este caso pasará a ser de tipo “cadena”

\$mi_variable = “siete”;

Características básicas del lenguaje

- **booleano** (*boolean*). Sus posibles valores son true y false. Además, cualquier número entero se considera como true, salvo el 0 que es false.
- **entero** (*integer*). Cualquier número sin decimales. Se pueden representar en formato decimal o hexadecimal (comenzando por 0x).
- **real** (*float*). Cualquier número con decimales.
- **cadena** (*string*). Conjuntos de caracteres delimitados por comillas simples o dobles.
- **null**. Tipo especial que se usa para indicar que la variable no tiene valor.

Características básicas del lenguaje

Ejemplos

\$mi_booleano = false;

\$mi_entero = 0x2A;

\$mi_real = 7.3e-1;

\$otro_real = 5.2;

\$mi_cadena = "texto";

\$mi_variable = Null;

// La constante Null es insensible a mayúsculas o minúsculas

Características básicas del lenguaje

Ejemplo:

```
<?php
    $mi_entero = 3;
    $mi_real = 2.3;
    $resultado = $mi_entero + $mi_real;
    echo "Mi entero es " . $mi_entero . " y mi real es " . $mi_real . "<br/>";
    echo "Resultado de la suma =" . $resultado;
?>
```

// La variable \$resultado es de tipo real (valor 5.3)

Si realizas una operación con variables de distintos tipos, ambas se convierten primero a un tipo común.

Características básicas del lenguaje

Ejemplo:

```
<?php
    $mi_entero = 3;
    $mi_real = 2.3;
    $resultado = $mi_entero + (int)$mi_real;
    echo "Mi entero es " . $mi_entero . " y mi real es " . $mi_real . "<br />";
    echo "Resultado de la suma = " . $resultado;
?>

/* La variable $mi_real se convierte a entero (valor 2) antes de sumarse. */
// La variable $resultado es de tipo entero (valor 5)
```

Similar al caso anterior, pero esta vez forzamos la conversión.

Características básicas del lenguaje

Conversiones de tipo posibles y los resultados obtenidos:

<http://php.net/manual/es/language.types.type-juggling.php>

Características básicas del lenguaje

Operadores

Muy similares a otros lenguajes. Todas las expresiones deben contener al menos un operando y un operador.

\$a = \$b + \$c;

\$valor++;

\$x += 5;

Características básicas del lenguaje

Operadores

Existen operadores para:

- **Realizar operaciones aritméticas:** suma, resta, multiplicación, división...
- **Realizar asignaciones:** además de =, operadores con los que realizar operaciones y asignaciones en un único paso (+=, -=,...).
- **Comparar operandos:** además de (>, >=,...), operadores para probar igualdad (==, ===) y para probar diferencia (<>, != y !==).

\$a == \$b	Igual	TRUE si \$a es igual a \$b después de la manipulación de tipos.
---------------	-------	---

\$a === \$b	Idéntico	TRUE si \$a es igual a \$b, y son del mismo tipo.
----------------	----------	---

- **Comparar expresiones booleanas:** realizar un Y lógico (operadores and o &&), O lógico (operadores or o ||), negación lógica (operador !).

Características básicas del lenguaje

Operadores disponibles y forma de usarlos:

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

Características básicas del lenguaje

Ámbito de utilización de las variables

En nuestro código, dependiendo de dónde aparezca una variable por primera vez, ésta podrá ser utilizada en una parte en concreto (esto se llama visibilidad de la variable).

Local: Declarada dentro de una función. Visible y accesible dentro de la función.

Global: Declarada en cualquier lugar del código PHP (dentro o fuera de una función). Visible y accesible desde dentro de cualquier lugar del código.

Superglobal/De Sistema: Se declara como Array. Visible/accesible desde fuera del script PHP.

Características básicas del lenguaje

Ejemplo

```
<?php
    $a = 1;
    function prueba() {
        $a = 18;
    }
    echo $a;
?>
```

**El resultado
sería 1 o 18?**

Ejemplo

```
<?php
    $nombre = "Juan";
    function minombre () {
        $nombre = "Lucia";
    }
    minombre();
    echo $nombre;
?>
```

**El resultado es
Juan o Lucia?**

Recuerda: Dentro de una función, NO puedo acceder a una variable que está fuera.

Características básicas del lenguaje

Ejemplo:

```
<?php
    $a = 1;
    function prueba() {
        global $a;
        $b = $a + 18;
        echo $b;
    }
    prueba();
?>
```

→ De esta forma le decimos a PHP que utilice la variable ya existente.

¿Qué se mostrará por pantalla?

Características básicas del lenguaje

Las variables locales a una función desaparecen cuando acaba la función y su valor se pierde. Si queremos mantener el valor de una variable local entre distintas llamadas a la función, debemos declarar la variable como estática utilizando la palabra **static**.

```
function contador()  
{  
    static $a = 0;  
    $a++;  
    // Cada vez que se ejecuta la función, se incrementa el valor de $a  
}
```

De esta forma, igual que anteriormente, también le decimos a PHP que no cree una nueva variable local, sino que utilice la ya existente.

Características básicas del lenguaje

Más ejemplos de ámbitos de las variables:

<http://php.net/manual/es/language.variables.scope.php>

EJERCICIOS



1.- Crea una página llamada *ejer01.php* cuyo título sea *Ejercicio01*

En ella aparecerá primero una frase de saludo y tras un salto de línea se mostrará el resultado de la función `phpinfo()`. Por último se mostrará una frase de despedida.

2.- Crea una página llamada *ejer02.php* cuyo título sea *Ejercicio02*

En ella crearás tres variables: una de tipo entero cuyo valor es **3**, una de tipo real cuyo valor es **7 ' 2** y una cuyo valor en hexadecimal es **0x0C**. Súmalas y muestra el resultado.

Resultado: 22,2



El resultado de `phpinfo()` es bastante largo...

PHP License

This program is free software; you can redistribute it and/or modify it under the terms of the PHP License as published by the PHP Group. For more details, see the full text of the license in the distribution in the file: LICENSE

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

If you did not receive a copy of the PHP license, or have any questions about PHP licensing, please contact license@php.net.

Adiós