PHP: Hypertext Preprocessor

PHP

- Lenguaje de scripting
- Se ejecuta en el servidor
- Permite crear contenidos dinámicos e interactivos
- Extensión por defecto .php
- Palabras reservadas, clases, funciones no son sensibles a mayúsculas y minúsculas
- En general es un lenguaje débilmente tipado

```php
<?php
$color = "rojo";
echo "Color " . $color;
Echo "Color " . $color;
Echo "Color " . $Color;
?>
```

PHP - Comentarios

- Varias formas de comentar el código

```
<?php
// Comentario de una línea
# También comentario de una línea
/*
Comentario
De varias
líneas
*/
?>
```

PHP – Variables

- Comienzan por $, seguido por el nombre de la variable
- El nombre de las variable comienza por una letra o guion bajo y sólo puede contener números, letras y guiones bajos
- Sensibles a mayúsculas y minúsculas
- Asociación automática del tipo de dato dependiendo de su valor
- Ámbitos: local, global, estático

PHP – Variables

Ámbito global

- Declaradas fuera de función
- Accedidas fuera de función

```php
<?php
$x = 5; // ámbito global

function test() {
  // producirá un error usar x dentro de la función
  echo "<p>Variable x dentro de la función: $x</p>";
}
test();

echo "<p>Variable x fuera de la función: $x</p>";
?>
```

PHP – Variables

Ámbito local

- Declaradas dentro de función
- Accedidas dentro de función
- Uso de global para acceder a variables globales desde función
- $GLOBALS[index] contiene todas variables globales. El array es accesible dentro de funciones

```php
<?php
function test() {
  $x = 5; // Ámbito local
  echo "<p>Variable x dentro de la función: $x</p>";
}
test();

// producirá un error usar x fuera de la función
echo "<p>Variable x fuera de la función: $x</p>";
?>
```

PHP – Variables

Ámbito local

- Declaradas dentro de función
- Accedidas dentro de función
- Uso de global para acceder a variables globales desde función
- $GLOBALS[index] contiene todas variables globales. El array es accesible dentro de funciones

```php
<?php
$x = 5;
$y = 15;

function test() {
  global $x, $y;
  $y = $x + $y;
}

test();
echo $y; // muestra 20
?>
```

PHP – Variables

Ámbito local

- Declaradas dentro de función
- Accedidas dentro de función
- Uso de global para acceder a variables globales desde función
- $GLOBALS[index] contiene todas variables globales. El array es accesible dentro de funciones

```php
<?php
$x = 5;
$y = 15;

function test() {
  $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
}

test();
echo $y; // muestra 20
?>
```

PHP – Variables

Ámbito estático

- Ser requiere el uso de static
- La variable persiste tras ejecutarse la función

```
<?php
function test() {
  static $x = 0;
  echo $x;
  $x++;
}

test();
test();
test();
?>
```

PHP – Variables

Tipos de datos

- Cadena de caracteres
- Entero
- Flotante
- Booleano
- Array
- Objecto
- Nulo
- Recurso

PHP – Variables

Tipos de datos

- Cadena de caracteres

```php
<?php
echo 'Esto es una cadena de caracteres';
echo ”Esto también”;
?>
```

PHP – Variables

Tipos de datos

- Entero

```php
<?php
$a = 1234; // número decimal
$a = -123;  // número negativo
$a = 0123; // número octal (equivale a 83 decimal)
$a = 0x1A; // número hexadecimal (equivale a 26 decimal)
$a = 0b11111111; // número binario (equivale al 255 decimal)
?>
```

PHP – Variables

Tipos de datos

- Flotante

```php
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

PHP – Variables

Tipos de datos

- Booleano

```php
<?php
$verdad = true; // asigna el valor TRUE a $verdad
?>
```

PHP – Variables

Tipos de datos

- Array

```php
<?php
$coches = array("Volvo", "BMW", "Toyota");
$ejemplo = array(
    "foo" => "bar",
    "bar" => "foo",
    100   => -100,
    -100  => 100,
);
var_dump($coches);
var_dump($ejemplo);
?>
```

PHP – Variables

Tipos de datos

- Objecto

```php
<?php
class Coche {
  public $color;
  public $modelo;
  public function __construct($color, $modelo) {
    $this->color = $color;
    $this->modelo = $modelo;
  }
  public function mensaje() {
    return "Mi " . $this->modelo . " es de color " . $this->color;
  }
}
$miCoche = new Coche("Negro", "Volvo");
echo $miCoche -> mensaje();
?>
```

PHP – Variables

Tipos de datos

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

- Nulo

PHP – Variables

Tipos de datos

```php
<?php
$fp = fopen("test.txt", "w");
var_dump($fp);
?>
```

- Recurso

PHP - Funciones

- Multitud de funciones incluidas en el lenguaje
- Por defecto, paso de parámetros por valor
- Uso de & para pasar parámetros por referencia
- Uso de valores de parámetros predeterminados
- Uso de comprobación estricta de tipos con declaraciones de tipo en parámetros y valores de retorno en funciones. Uso de strict_types con escalares.

```php
<?php
echo cos(3);
?>
```

PHP - Funciones

- Multitud de funciones incluidas en el lenguaje
- Por defecto, paso de parámetros por valor
- Uso de & para pasar parámetros por referencia
- Uso de valores de parámetros predeterminados
- Uso de comprobación estricta de tipos con declaraciones de tipo en parámetros y valores de retorno en funciones. Uso de strict_types con escalares.

```php
<?php
function concatenar(&$cadena)
{
    $cadena .= 'concatenada.';
}
$resultado = 'Esto es una cadena ';
concatenar($resultado);
echo $resultado;    // imprime 'Esto es una cadena concatenada.'
?>
```

PHP - Funciones

- Multitud de funciones incluidas en el lenguaje
- Por defecto, paso de parámetros por valor
- Uso de & para pasar parámetros por referencia
- Uso de valores de parámetros predeterminados
- Uso de comprobación estricta de tipos con declaraciones de tipo en parámetros y valores de retorno en funciones. Uso de strict_types con escalares.

```php
<?php
function hacerZumo($tipo = "naranja")
{
    return "Hacer un zumo de $tipo.\n";
}
echo hacerZumo();
echo hacerZumo(null);
echo hacerZumo("melocotón");
?>
```

PHP - Funciones

- Multitud de funciones incluidas en el lenguaje
- Por defecto, paso de parámetros por valor
- Uso de & para pasar parámetros por referencia
- Uso de valores de parámetros predeterminados
- Uso de comprobación estricta de tipos con declaraciones de tipo en parámetros y valores de retorno en funciones. Uso de strict_types con escalares.

```php
<?php
class Persona {
    private $nombre;
    public function __construct($nombre) {
        $this->nombre = $nombre;
    }
    public function nombre() { return $this->nombre; }
}
function saludar(Persona $a, Persona $b) {
    return $a->nombre() . " saluda a " . $b->nombre();
}
echo saludar("Antonio", "Manuel");
$antonio = new Persona("Antonio");
$manuel = new Persona("Manuel");
echo saludar($antonio, $manuel);
?>
```

PHP - Funciones

- Multitud de funciones incluidas en el lenguaje
- Por defecto, paso de parámetros por valor
- Uso de & para pasar parámetros por referencia
- Uso de valores de parámetros predeterminados
- Uso de comprobación estricta de tipos con declaraciones de tipo en parámetros y valores de retorno en funciones. Uso de strict_types con escalares.

```php
<?php declare(strict_types=1); // requisito comprobación


function suma(int $a, int $b) {
  return $a + $b;
}
echo suma(20, "10 coches");
//10 coches no es un entero y por tanto producirá un error
?>
```

PHP - Clases

- Palabras reservadas para visibilidad de propiedades y funciones
- Creación de objetos mediante la palabra reservada new
- Herencia mediante la palabra reservada extends. Sólo es posible heredar de una clase base

```php
<?php
class Base
{
    public $publica = 'valor publico';
    protected $protegida = 'valor protegido';
    private $privada = 'valor privado';

    public function mostrarVariable() {
        echo $this->publica;
    }
}
?>
```

PHP - Clases

- Palabras reservadas para visibilidad de propiedades y funciones
- Creación de objetos mediante la palabra reservada new
- Herencia mediante la palabra reservada extends. Sólo es posible de heredad de una clase base

```php
<?php
class Extendida extends Base
{
    function mostrarVariable()
    {
        echo "Clase extendida\n";
        parent::mostrarVariable();
    }
}
$objeto = new Extendida();
$objeto->mostrarVariable();
?>
```