

Programación III

PHP

Clase 1

Maximiliano Neiner

Temas a Tratar

- ❖ Introducción a PHP

Temas a Tratar

Introducción a PHP

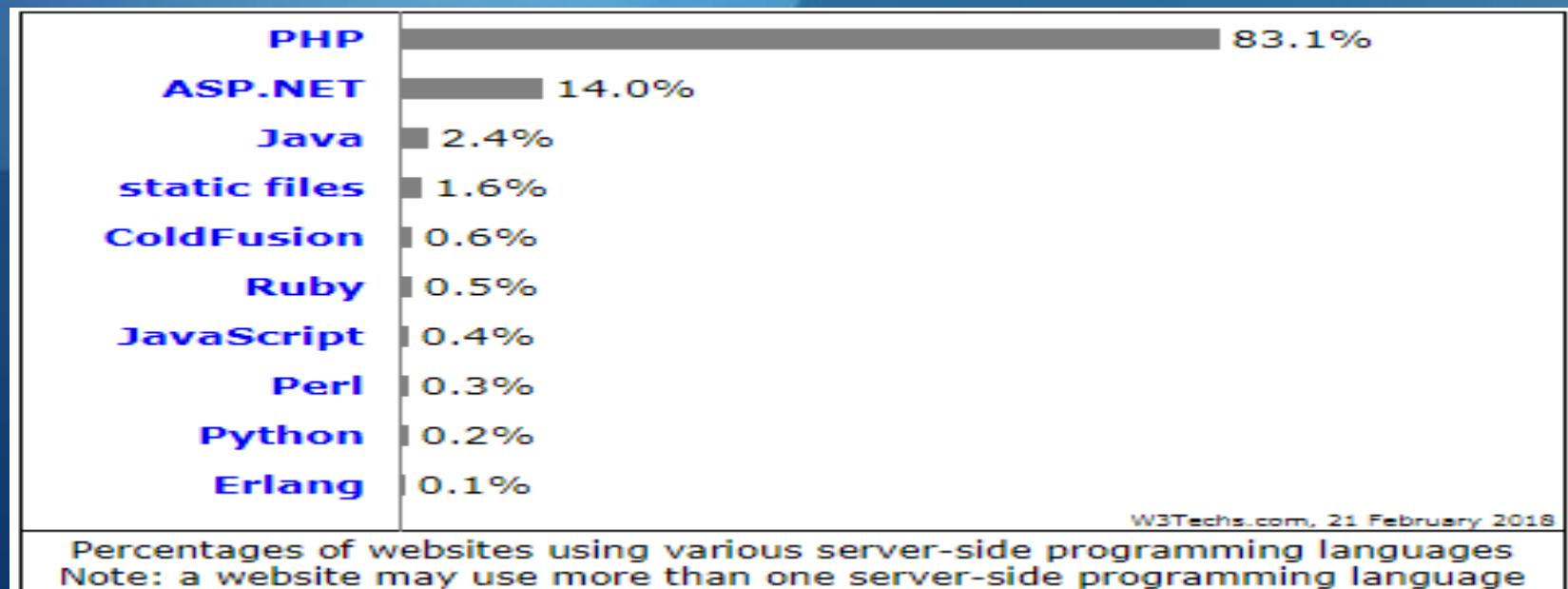
- Temas Generales
- Definición e inicialización de variables
- Operadores
- Estructuras de Control
- Arrays

PHP (PHP: Hypertext Pre-Processor)

- ❖ PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo Web y que puede ser incrustado en HTML.
- ❖ Popular: porque un gran número de páginas y portales Web están creadas con PHP (*).
- ❖ Código abierto: significa que es de uso libre y gratuito para todos los programadores que quieran usarlo.
- ❖ Incrustado en HTML: significa que en un mismo archivo vamos a poder combinar código PHP con código HTML.

PHP (PHP: Hypertext Pre-Processor)

- Es multiplataforma (Windows, Linux, Mac)
- El código PHP es ‘invisible’ al navegador.
- Capacidad de conexión a base de datos.
- Capacidad de expandir su potencial con plugins.



Cliente - Servidor

- ❖ Llamamos servidores a ordenadores generalmente potentes, con un software y hardware especial, que se encargan de resolver peticiones que le hacen otros ordenadores.
- ❖ Hablamos de clientes para referirnos a cualquier dispositivo (ordenadores tipo PC, Note Books, Tablets, Smart Phones, etc.) que generan peticiones hacia los servidores.

¿Cómo funciona?

- ❖ Cuando el cliente hace una petición, el servidor ejecuta el intérprete de PHP (para nuestro caso).
- ❖ Este compila el código fuente, que genera el sitio Web.
- ❖ El resultado es enviado al navegador del cliente.



Estructura Básica

```
<?php  
    echo "HOLA MUNDO";  
?>
```

- ❖ Todo código va entre <?php y ?>.
- ❖ Con ‘echo’ (*) se muestra el mensaje en el navegador.
- ❖ Al igual que en C o C#, toda instrucción se debe finalizar con punto y coma (;)
- ❖ La extensión del archivo fuente debe ser .php

Comentarios

- PHP soporta dos tipos de comentarios

```
<?php  
  
// Comentario de una sola línea  
  
# Otro comentario de una línea  
  
/* Comentario con más  
de una línea */  
  
?>
```

Temas a Tratar

Introducción a PHP

- Temas Generales
- Definición e inicialización de variables
- Operadores
- Estructuras de Control
- Arrays

Variables (1/2)

- ✖ PHP soporta ocho tipos primitivos.
- ✖ Cuatro tipos escalares:
 - Boolean
 - Integer
 - Float
 - String
- ✖ Dos tipos compuestos
 - Array
 - Object
- ✖ Dos especiales
 - Resource
 - NULL

Variables (2/2)

```
<?php
```

```
$nombre = "Juan";  
$edad = 25;  
$sueldo = 8500.33;  
  
print("nombre: $nombre");  
echo "edad:", $edad;  
printf("sueldo:%f", $sueldo);  
  
?>
```

- ❖ Comienzan con el símbolo **\$**.
- ❖ Son **case sensitivity** (no así las palabras claves).
- ❖ Los tipos se definen cuando se les asigna un valor.

Conversión de Tipos

- ❖ Las conversiones las realiza automáticamente PHP dependiendo del contenido de las variables.
- ❖ Sin embargo, si se desea explícitamente realizar una conversión de tipos:
 - (int), (integer) -> convierte a entero
 - (bool), (boolean) -> convierte a booleano
 - (float), (double), (real) -> convierte a decimal
 - (string) -> convierte a cadena de caracteres
 - (array) -> convierte a array
 - (object) -> convierte a objeto
 - (unset) -> convierte a nulo

Funciones de Cadenas

- ✖ **strlen()** Retorna la cantidad de caracteres de una cadena.
- ✖ **strcmp()** Compara dos cadenas (case sensitive).
- ✖ **strtolower()** Convierte una cadena a minúsculas.
- ✖ **strtoupper()** Convierte una cadena a mayúsculas.
- ✖ **substr()** Retorna una porción de la cadena.
- ✖ **ucfirst()** Convierte el primer carácter de la cadena a mayúscula.
- ✖ **ucwords()** Convierte el primer carácter de cada palabra de la cadena en mayúsculas.

Temas a Tratar

Introducción a PHP

- Temas Generales
- Definición e inicialización de variables
- Operadores
- Estructuras de Control
- Arrays

Operadores

- ❖ PHP divide a los operadores en grupos
 - Operadores Aritméticos (Ídem C - C#)
 - Operadores de Asignación (Ídem C - C#)
 - Operadores Comparación (Ídem C - C#) (1)
 - Incremento/Decremento (Ídem C - C#)
 - Operadores Lógicos (Ídem C - C#) (2)
 - Operadores de cadena (3)
 - Operadores de Array (4)

Temas a Tratar

Introducción a PHP

- Temas Generales
- Definición e inicialización de variables
- Operadores
- Estructuras de Control
- Arrays

Sentencias condicionales (1/2)

● PHP: sentencia if con varios formatos

Sentencias condicionales (2/2)

● PHP: sentencia case

```
$a = 0;

switch ($a) {
    case 1: //CODIGO 1
    break;

    case 2: //CODIGO 2
    break;

    default: //CODIGO DEFAULT
    break;
}
```

Sentencia For

- PHP: la sentencia for consta de tres partes

```
//Partes: declaración, prueba, acción
for ($i=0; $i < 10; $i++)
{
    echo "<br/>", $i + 1;
}
```

Sentencia Foreach

- Foreach permite recorrer arrays y objetos.

```
$vec = array(1,2,3);
foreach($vec as $valor)
{
    // $valor es un elemento de la colección
}
```

```
$vec = array("uno" => 1, "dos" => 2, "tres" => 3);
foreach($vec as $k => $valor)
{
    // $k posee la clave y $valor el elemento
}
```

Sentencia While

- PHP:

```
$condicion = true;  
  
while($condicion == true) {  
    //En algún momento poner $condicion = false  
}
```

```
$condicion = true;  
  
do {  
    //En algún momento poner $condicion = false  
}while($condicion == true);
```

Temas a Tratar

Introducción a PHP

- Temas Generales
- Definición e inicialización de variables
- Operadores
- Estructuras de Control
- Arrays

Arrays (1/4)

- ❖ Un array en PHP es realmente un mapa ordenado.
Un mapa es un tipo de datos que
asocia *valores* con *claves*.
- ❖ PHP tiene tres tipos de arrays
 - Arrays indexados. Índices numéricos.
 - Arrays asociativos. Índices nombrados.
 - Arrays multidimensionales. Arrays que contienen otros arrays.

Arrays (2/4)

- En PHP los arrays pueden ser creados con el constructor del lenguaje array().

```
$vec = array(1,2,3);  
var_dump($vec);  
/*  
Salida:  
array(3) { [0]=>int(1) [1]=>int(2) [2]=>int(3) }  
*/
```

- O simplemente

```
$vec[0] = 1; $vec[1] = 2; $vec[2] = 3;  
var_dump($vec);  
/*  
Salida:  
array(3) { [0]=>int(1) [1]=>int(2) [2]=>int(3) }  
*/
```

Arrays (3/4)

● Arrays asociativos con el constructor array().

```
$vec = array ("Juan"=>22, "Romina"=>12, "Uriel"=>8) ;  
var_dump ($vec) ;  
/*  
Salida:  
array(3) { ["Juan"]=>int(22) ["Romina"]=>int(12)  
["Uriel"]=>int(8) }  
*/
```

● 0

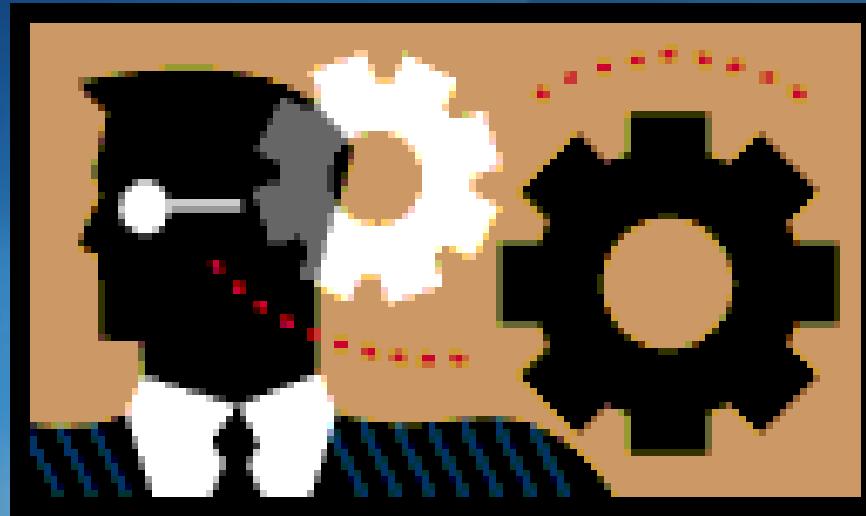
```
$vec ["Hugo"] = 15; $vec ["Juana"] = 36;  
var_dump ($vec) ;  
/*  
Salida:  
array(2) { ["Hugo"]=>int(15) ["Juana"]=>int(36) }  
*/
```

Arrays (4/4)



Funciones para ordenar Arrays

- **sort()** Ordena un array ascendentemente
- **rsort()** Ordena un array descendentemente
- **asort()** Ordena un array asociativo ascendentemente, por su valor.
- **ksort()** Ordena un array asociativo ascendentemente, por su clave.
- **arsort()** Ordena un array asociativo descendentemente, por su valor.
- **krsort()** Ordena un array asociativo descendentemente, por su clave.



Ejercitación

Ejercicios de Programación

- ❖ Realizar los primeros ejercicios de la guía.
- ❖ Aplicar las recomendaciones estándares PSR-1.
 - <https://www.php-fig.org/psr/psr-1/>