

Tema 2: Introducción al PHP

EI1042 - Tecnologías y Aplicaciones Web

EI1036- Tecnologías Web para los Sistemas de Información (2020/2021)

Profesorado: Dra. Dolores María Llidó Escrivá



[Universitat Jaume I.](#)

Índice

- Introducción a PHP
 - Paginas dinámicas con php
 - PHP avanzado
 - Wordpres y PHP
-

1. Introducción al PHP



- Creado por Rasmus Lerdorf para uso personal en 1994
- PHP es un lenguaje de script del lado del servidor.
- PHP: Hypertext Preprocessor
- Versión actual: PHP 7
- Es potente, fácil de aprender, de libre distribución, permite el acceso a bases de datos y otras funcionalidades orientadas a la red
- Dispone de abundante soporte en la Web

Manual PHP: <http://php.net/manual/es/>

1.2. Sintaxis básica PHP

- PHP es sensible a mayúsculas/minúsculas **solo para las variables**.
- Las instrucciones se separan con ";"
- Espacios en blanco y cambios de línea no se tienen en cuenta.
- PHP interpreta entre comillas dobles pero no entre comillas simples.
- Se utiliza codificación UTF-8.

- Comprobar que el editor no ha generado un BOM (*byte order mark* de unicode en la primera línea)

Guía Estilos: <https://www.php-fig.org/psr/psr-2/>

--

Usar: YodaStyle



Comparaciones al revés para evitar equivocarnos.

Ejecución PHP

- Consola

```
php.exe "./bienvenidos.php"
```

- Entorno php:

```
>>php -a  
$hola="Adios";  
$echo $hola
```

- Servidor web: <https://asociacionpiruleta.cloudaccess.host/PHP/T2/holaMundo.php>
- Servidor web local:

```
php [options] -S <addr>:<port> [-t docroot]  
>> php -S localhost:8080
```

--

Ejemplo fichero PHP

```
<?php
$var = "test";
echo "$var"; // Salida:"test"
echo "\$var"; // Salida:" \"$var"
echo '$var'; // Salida:" "test"
# otro comentario hasta el final de la línea
// este también es de línea
/* comento
varias líneas */
?>
```

1.3. Tipos de datos

- Tipos escalares: boolean, integer, double, string
- Tipos compuestos: array, object
- Tipos especiales: resource, NULL
- Las variables no declaradas se asocian a NULL y se imprimen como cadenas vacías.
- El tipo resource no lo veremos este curso por ser un tópico avanzado

--

Array Asociativo

Sintaxis: array ([clave =>] valor, ...)

```
$medidas = array (10, 25, 15);
echo $medidas[0];

$color = array ('rojo'=>101, 'verde'=>51, 'azul'=>255);

#Acceso:
echo $color['rojo']; // No olvidar las comillas
echo array_keys($color);
```

1.4. Salida estándar

¿Cómo enviar mensajes a la salida estándar?

- echo — Muestra una o más cadenas separada por ','.
- print — Mostrar solo una cadena.
- printf — Imprimir una cadena con formato.
- print_r — Imprime información legible para humanos.
- var_dump— Vuelca información sobre una variable. La información y su tipo

--

Diferencias echo o print

- **Void** echo (string argument1[,...string argumentN])
- **Int** print (argument)
 - print solo tiene un argumento (echo puede tener varios)
 - si print devuelve 1 (significa que ha generado la salida)

No es obligatorio el uso de paréntesis ya que no son realmente una función.

```
echo "Hola mundo";
echo "Hola ", "mundo"; //dos argumentos
print "Hola mundo";
print "Hola ". "mundo"; // el operador '.' concatena cadenas
```

1.5. Variables

- No se declara el tipo de las variables.
- Las variables se pueden asignar
 - Por valor
 - Por referencia (con &)
 - Creación de nombres de variables dinámico.

```
$x='equis';
$_x = &$x; //referencia a $x
$_x = 'x';
echo $x; //salida: x
echo $_x; //salida: x
$a = "hola";
$$a = "mundo"; //nombrar una variable con otra variable
print "$a $hola\n"; //salida: hola mundo
print "$a ${a}s\n"; //salida: hola hola //utilizaremos ${} para evitar errores
print "$a ${$a}"; //salida: hola mundo //podemos acceder al valor de una variable ($mundo) nombrada en otra ($a)
```

--

¿De qué tipo es la variable?

- `gettype()` devuelve el tipo de una variable
- `is_type()` comprueba si una variable es de un tipo dado:

```
is_array(), is_bool(), is_null(), is_object(), is_resource(),
is_scalar(), is_string(), is_float(), is_integer(), is_numeric(), is_nan()
```

```
--
```

Ámbito de variables

- Local: Variable definida en una función
 - Está limitada a dicha función.
 - Se elimina al acabar la ejecución de la función
 - Salvo si la variable se declara como **static** .
- Global:
 - No se puede definir dentro de las funciones a menos que :
 - se declare en la función con la palabra clave 'global'
 - O que se acceda con el array \$GLOBALS[indice]
 - Existen durante todo el tiempo de proceso del fichero
 - Al acabar de procesar el fichero se eliminan las variables globales

1.6. Funciones

```
function suma ($x, $y)
{
    $s = $x + $y;
    return $s;
}
```

Salida:

```
$a=1;
$b=2;
$c=suma ($a, $b);
print $c;
```

```
--
```

Paso parámetros

- Por defecto paso parámetros por valor
- Paso por referencia:

```
function incrementa (&$a)
{
    $a = $a + 1;
}
$a=1;
incrementa ($a);
print $a; // Muestra un 2
```

--

Argumentos por defecto

- Los argumentos con valores por defecto deben ser siempre los últimos:

```
function muestranombre ($nombre, $titulo= "Sr.")
{
    print "Estimado $titulo $nombre:\n";
}
muestranombre ("Fernández");
muestranombre ("Fernández", "Prof.");
```

Salida:

```
Estimado Sr. Fernández:
Estimado Prof. Fernández:
```

1.7. OOP en PHP

- PHP no es un lenguaje 100% Orientado a Objetos.
- Soporta:
 - Encapsulamiento.
 - Tipos Abstractos de Datos y ocultamiento de la Información.
 - Herencia.
 - Polimorfismo.

--

Clases y Objetos

```
class MyClass{

    const CONST_VALUE = 10;
    public $numero=5;
    function dameNumero(){
        return self::CONST_VALUE*$this->numero;
    }
    function llamoDame(){ return self::dameNumero();}
}
$classname = 'Myclass';
echo "<p>",$classname::CONST_VALUE,</p>"; // A partir de PHP 5.3.0
echo "<p>",$classname::CONST_VALUE,</p>";
```

```
$datos=new MyClass();
$datos->numero=15;
echo "<p>",$datos->dameNumero(),"</p>";
echo "<p>",$datos->llamoDame(),"</p>"; //Da error
echo "<p>",$datos->dameNumero();
```

- `::` es el operador de Resolución de Ámbito, el cual permite acceder a elementos estáticos/constantes y sobrescribir propiedades o métodos de una clase.
- `->` permite acceder a las propiedades y métodos de un objeto.

--

`$this`, `self`, `parent`

- `$this` es una variable especial que auto-referencia al objeto para acceder a sus métodos y propiedades.
- `self` y `parent`: son pseudo-variables para acceder a una propiedad o método de una clase.
- Usamos `$this->` para hacer referencia al objeto (instancia) actual, y `self::` para referenciar a la clase actual.

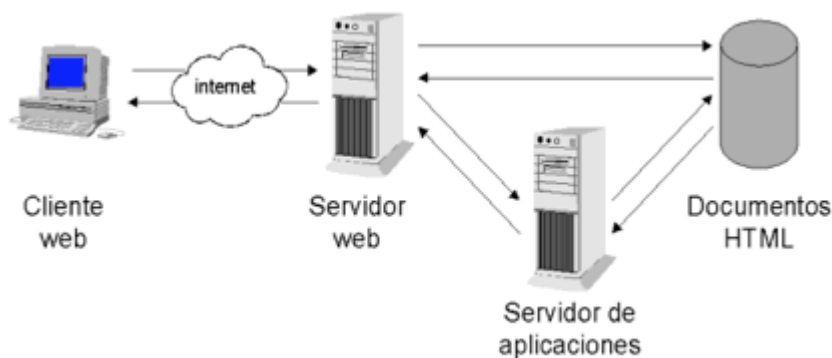
```
$this->nombre
self::nombres
```

--

2. Páginas web dinámicas con PHP

- El cliente no ve el código PHP sino los resultados que produce en la salida estándar.
- Apache ejecuta php no como un CGI sino como un módulo.

Version php S.O. != Apache



Seguridad: Ficheros sensibles estar fuera de la raíz (docroot-httpdocs) del servidor web.

Ejercicio 1

<https://asociacionpiruleta.cloudaccess.host/PHP/Lab/P1/portal.php?action=listar>

Analiza la URL.

- ¿Cual es el nombre servidor?
- ¿Cual es el recurso?
- ¿Que parametros requiere el programa php?

Fichero con HTML con php:

bienvenido.php

```
<body>
<p>Inicio</p>
<?php
$nombre = "Ana";
print(" <P>Hola, $nombre</P>");
?>
<p>Fin</p>
```

Ejercicio 2

```
>>php -S localhost:8080
```

- ¿Cuál es el docroot del servidor web?
- ¿Dónde ponemos el fichero tutoPhp.php en el servidor para acceder desde <http://localhost/teoria/T2/tutoPHP.php?>

--

Servidor WEB en Producción

- Servidor web Apache (<http://www.apache.org>) con el módulo PHP (<http://www.php.net>)
- Base de datos MySQL (<http://www.mysql.com>) si se desea crear páginas dinámicas
- Herramientas para la gestión de MySQL, como PHPMyAdmin (<http://www.phpmyadmin.net>)

Apache <http://httpd.apache.org/docs/current/es/>

--

Servidor WEB en desarrollo: XAMPP

<https://geekflare.com/es/lamp-lemp-mean-xampp-stack-intro/>

- XAMPP es una distribución de Apache que incluye MySQL, PHP y phpMyAdmin
- XAMPP es gratuito y fácil de instalar
- XAMPP es multiplataforma
- Precaución: la configuración por defecto no es segura, ni para un entorno de producción.

--

2.1 Variables en el servidor web con php

Superglobal

- Variables predefinidas en PHP
- Están disponibles en todos los ámbitos.
- \$GLOBALS — Array con todas las variables disponibles en el ámbito global
 - \$_SERVER — Información del entorno del servidor y de ejecución
 - \$_GET(POST) — Variables HTTP GET(POST)
 - \$_FILES — Variables de Carga de Archivos HTTP
 - \$_REQUEST — Variables HTTP Request: GET+POST+COOKIE
 - \$_SESSION — Variables de sesión
 - \$_COOKIE — Variables con datos de la cookie
 - \$_ENV — Variables del entorno

--

Ejercicio 3

<https://asociacionpiruleta.cloudaccess.host/PHP/T2/tutoPhp.php>

- Analiza este fichero y mira el funcionamiento.

2.2. Formularios

Ejemplo: radio

```
<form action="procesar.php" method="post">
Sexo:
  <INPUT TYPE="radio" NAME="sexo" VALUE="M" CHECKED >Mujer
  <INPUT TYPE="radio" NAME="sexo" VALUE="H">Hombre
  <INPUT TYPE="submit">
</form>
```

```
#Procesar.php
<?PHP
$sexo = $_REQUEST["sexo"];
print ($sexo);
?>
```

--

Ejercicio 4:

- ¿Cuál es la petición al servidor al pulsar submit/enviar?
- Añade el nombre de la persona en el formulario y muéstralo en Procesar.php.
- ¿Diferencia entre post y get?
- ¿Como probar que funciona el código php?

Simulando PETICIONES POST con parámetros

```
curl --data "param1=value1&param2=value2" http://hostname/resource
```

--

CHECKBOX

```
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="garaje" CHECKED>Garaje  
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="piscina">Piscina  
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="jardin">Jardín
```

```
<?php  
$extras = $_REQUEST["extras"];  
foreach ($extras as $extra)  
    print ("{$extra}<BR>\n");  
?>
```

--

BUTTON

```
<INPUT TYPE="button" NAME="actualizar" VALUE="Actualizar datos">
```

```
<?php  
$actualizar = $_REQUEST["actualizar"];  
if ($actualizar)  
    print ("Se han actualizado los datos");  
?>
```

--

SELECT múltiple

Idiomas:

```
<SELECT MULTIPLE SIZE="3" NAME="idiomas[]">
  <OPTION VALUE="ingles" SELECTED>Inglés
  <OPTION VALUE="francés">Francés
  <OPTION VALUE="alemán">Alemán
  <OPTION VALUE="holandés">Holandés
</SELECT>
```

```
<?php
$idiomas = $_REQUEST["idiomas"];
foreach ($idiomas as $idioma)
    print ("{$idioma}<BR>\n");
?>
```
