

Tema 2: Introducción al PHP

▷ EI1042 - Tecnologías y Aplicaciones Web

EI1036- Tecnologías Web para los Sistemas de Información (2018/2019)

Professora: Dra. Dolores M^a Llidó Escrivá


[Universitat Jaume I.](#)

(Drive/T2) <https://drive.google.com/drive/folders/1U8unry7CEj77DF9-dpq9JuAJ4ZQkjYWf>

▷ Índice

- Introducción a PHP
- Sintaxis básica
- Tipos de datos
- Salida Estándar
- Variables Expresiones y Operadores
- Formularios
- Funciones
- Objetos

▷ 1. Introducción al PHP

- Creado por Rasmus Lerdorf para uso personal en 1994  [Image of Rasmus](#)
- 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

⁝ Ficheros con PHP

- Fichero PHP: hola.php

```
<?php
echo "Hola Mundo";
?>
```

- Fichero con HTML: bienvenido.php

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

⁝ Ejecución PHP

- Consola
- php.exe "../bienvenidos.php"
- Entorno php:

```
php.exe
$hola="Adios";
$echo $hola
```

-Servidor web: <https://piruletas.000webhostapp.com/Teo/T2/holaMundo.php>

▸ Configuración php

- Fichero: php.ini
- Visualizar la configuración del servidor: función phpinfo()

<https://piruletas.000webhostapp.com/teoria/T2/form0Info.html>

▸ 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

 [Image of La World Wide Web](#)

▸ Cuestiones

<https://piruletas.000webhostapp.com/Lab/P1/includes/portal.php?action=listar>

Analiza la URL.

- ¿Cual es el nombre servidor?
- ¿Cual es el recurso?

- ¿Que parametros requiere el programa php?

▸ Servidor WEB Local

Local:

- Instalar PHP
- Ejecutar: `php -S localhost`

`php [options] -S : [-t docroot]`

<http://localhost:8088/T2/tutoPhp.php>

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

--

▸ Cuestiones

- ¿Que es el docroot del servidor web?
- ¿Donde ponemos el fichero tutoPhp.php en el servidor?

▸ 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 (www.phpmyadmin.net)

Apache 2.3 <http://httpd.apache.org/docs/current/es/>

▸ Servidor WEB en desarrollo: XAMPP

- 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. Sintaxis básica PHP

- PHP es sensible a las mayúsculas/minúsculas.
- 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 utf8.
- Comprobar que el editor no ha generado un BOM (*byte order mark* de unicode en la primera línea)

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

--

▸ Ejemplo fichero PHP

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

--

› Usar: YodaStyle



Escribir comparaciones al revés (Yoda habla al revés).

En ocasiones podemos equivocarnos *if (\$value = true)* y ...simplemente estamos asignado SIEMPRE true a nuestra variable. Nos costaría identificar el error en nuestro código.

› 3. Tipos de datos

- Tipos escalares: boolean, integer, double, string
- Tipos compuestos: array, object
- Tipos especiales: resource, NULL

--

› 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)
```

4. Salida estándar

¿Cómo enviar mensajes a la salida estándar?

- echo — Muestra una o más cadenas.
- print — Mostrar 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 sólo tiene un argumento (echo puede tener varios)
 - 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";  
print "Hola mundo";  
print "Hola ". "mundo";
```

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; //x  
echo $_x; //x  
$a = "hola";  
$$a = "mundo";  
print "$a $hola\n";  
//hola mundo  
print "$a ${$a}";  
//hola mundo 18
```

¿Que 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_float(), is_integer(), is_null(), is_numeric(), is_object(), is_resource(),  
is_scalar(), is_string()
```

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 \$GLOBAL[indice]
 - Existen durante todo el tiempo de proceso del fichero
 - Al acabar de procesar el fichero se eliminan las variables globales

› Superglobal

- Variables predefinidas en PHP
- Están disponibles en todos los ámbitos.

Ejemplos:

- \$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
- \$_SESSION — Variables de sesión
- \$_COOKIE— Variables con datos de la cookie
- \$_ENV— Variables del entorno

› Ejercicio

<https://piruletas.000webhostapp.com/teoria/T2/tutoPhp.php>

- Analiza este fichero y mira el funcionamiento.

6. Formularios

Ejercicios

```
RADIO
<form action='procesar.php' method="get">
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:

- ¿Cuál es la petición al servidor al pulsar submit?
- ¿Que aparecerá por pantalla? <https://piruletas.000webhostapp.com/teoria/T2/form0.html>

5. Formularios desde PHP

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");  
?>
```

› Formularios desde PHP

BUTTON

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

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

› Formularios desde PHP

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"});
?>
```

7. Funciones

Ejemplo:

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

Salida:

```
$a=1;
$b=2;
```

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

› Funciones

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

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

› Funciones

- 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:

⁝ Ejemplo funciones y sesiones

```
<?php
function activarSession()
{
    if (!isset($_SESSION["activo"])) {
        $_SESSION = array();
        setcookie(session_name(), '', time() + 10);
        $_SESSION["activo"] = 1;
        print "<h2>Hola</h2>";
        $_SESSION["usuario"] = "visitante";
        return 0;
    } else {
        if ($_SESSION['last_action'] < time() - 60 /* be a little tolerant here */ ) {
            session_destroy();
        } // destroy the session and quit

        print "<h2>Bienvenido de nuevo " . $_SESSION["usuario"] . "\n</H2>";
        return 1;
    }
}

print "</p>Cookies</p>";
print_r($_COOKIE);
session_start();
if (1 == activarSession()) {
    print "<p> Ya tenías una sesión activa</p>";
}
print "</p>Session</p>";
```

```
print_r($_SESSION);
```

```
?>
```

<https://piruletas.000webhostapp.com/teoria/T2/session.php>

8. 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 $classname::CONST_VALUE,"</p>"; // A partir de PHP 5.3.0
echo MyClass::CONST_VALUE,"</p>";
```

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

- :: Operador de Resolución de Ámbito es un token que 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 l objeto para acceder a sus métodos y propiedades.
- self y parent: son pseudo-variables para acceder a una propiedad o método de una clase.
- Usa \$this-> para hacer referencia al objeto (instancia) actual, y se utiliza self:: para referenciar a la clase actual.

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

▷ Bibliografía

-
- PHP: <http://es.php.net/manual/es/> Guia Estilos: <https://www.php-fig.org/psr/psr-2/>
 - Apache: <http://httpd.apache.org/docs/2.3/es/>