

TEMA 6. FUNCIONES INTRÍNSECAS: CADENAS, FECHA/HORA, NÚMEROS Y DATOS

Cadenas de caracteres

A continuación se describen algunas de las funciones más interesantes para trabajar con cadenas de caracteres:

explode *array explode (string \$delimitador , string \$cadena)*

Devuelve un array construido a partir de la \$cadena dividiéndola en fragmentos delimitados por el carácter \$delimitador.

Echo Explode(“;”, “lunes;martes;miercoles”); → Array
(0=>lunes, 1=>martes, 2=>miércoles)

ltrim *String ltrim (string \$cadena)*

Devuelve una copia de la cadena con los espacios que pudiera tener al principio eliminados.

Echo ltrim(‘ Hola’); → ‘Hola’

rtrim *String rtrim (string \$cadena)*

Devuelve una copia de cadena con los espacios que pudiera tener al final eliminados.

echo rtrim(‘Hola ’); → ‘Hola’

nl2br *string nl2br (string \$string)*

Devuelve una copia de la cadena con todos los saltos de línea (\r\n, o \n\r, o \n o \r) convertidos en saltos de línea HTML (
).

```
$cadena=<<<'IDENTIFICATIVO'
```

```
Hola
```

```
Adios
```

```
IDENTIFICATIVO;
```

```
echo nl2br($cadena);
```

parse_str *void parse_str (string \$cadena)*

Interpreta cadena como si contuviera la cadena de consulta generada por los datos enviados desde un formulario con el método GET, y crea

automáticamente variables para cada uno de ellos.

```
parse_str('pagina=1&seccion=2');
echo $pagina.' '.$seccion;
```

str_replace *mixed str_replace (mixed \$aguja , mixed \$reemplazo , mixed \$pajar)*

Devuelve una copia de la cadena pajar con todas las ocurrencias de la cadena aguja reemplazadas por la cadena reemplazo.

```
echo str_replace('lunes','martes','lunes, ¿quién ha dicho lunes?'); → 'martes,
¿quién ha dicho martes?'
```

strlen *Int strlen (string \$cadena)*

Devuelve el número de caracteres que contiene una cadena.

```
echo strlen('cadena'); → 6
```

strpos *Int strpos (string \$pajar , mixed \$aguja)*

Devuelve la posición de la primera ocurrencia de la cadena aguja dentro de la cadena pajar. Si no se encuentra, devuelve FALSE, no 0. Si quiere usar la respuesta de esta función en una comparación recuerde utilizar el operador de comparación exacto === (valor y tipo), pues de otro modo podría estar confundiendo con la aguja se encuentra en la posición 0 de pajar, con que aguja no se encuentra en pajar.

```
echo strpos('Hola, buenas','buenas'); → 6
```

strtolower *String strtolower (string \$cadena)*

Devuelve una copia de cadena con todos los caracteres en minúsculas.

```
echo strtolower('TEXTO EN MAYÚSCULAS'); → 'texto en mayúsculas'
```

strtoupper *String strtoupper (string \$cadena)*

Devuelve una copia de cadena con todos los caracteres en mayúsculas.

```
echo strtoupper('hola'); → 'HOLA'
```

substr *string substr (string \$cadena , int \$inicio, int \$longitud)*

Devuelve los caracteres de cadena a partir de la posición inicio (incluido) y hasta un máximo de longitud caracteres.

```
echo substr('probando',2,5); → 'oband'
```

abs	<i>Number abs (\$numero)</i> Devuelve el valor absoluto de número. echo abs(-6.64); →6,64
ceil	<i>Float ceil (float \$valor)</i> Devuelve el valor entero inmediatamente superior a valor. echo ceil(3.14159265);→4
floor	<i>Float floor (float \$valor)</i> Devuelve el valor entero inmediatamente inferior a valor. echo floor(-3.45);→ -3
max	Devuelven el valor máximo o mínimo de los números almacenados en un array.
min	echo max(array(1,2,3,4,5));→5
pow	<i>number pow (number \$base , number \$exp)</i> Devuelve el valor de la potencia de base elevado a exponente. echo pow(2,3);→8
rand	<i>int rand (int \$minimo , int \$maximo)</i>
mt_rand	<i>int mt_rand (int \$minimo , int \$maximo)</i> Devuelve un número entero pseudo-aleatorio elegido entre mínimo y máximo. La variante mt_rand aplica el algoritmo MersenneTwister, que es bastante más rápido que el utilizado por rand. echo mt_rand(0,45);(probarlo)
sqrt	<i>Float sqrt (float \$numero)</i> Devuelve la raíz cuadrada de número. echo sqrt(4); →2

Fechas y horas

En PHP las fechas se cuentan como el número (entero) de segundos transcurridos desde la época Unix, que es el origen del sistema de fechas. La época UNIX corresponde a las 00:00:00 GMT del 1 de enero de 1970. El intervalo de fechas que puede manejar PHP(en 32 bits) es:

- **Desde:** viernes, 13 de diciembre de 1901 a las 20:45:52

- **Hasta:** jueves, 19 de enero de 2038 a las 03:14:07 GMT.

Sin embargo, en MYSQL existen tres tipos de datos para fechas (DateTime, Date y Timestamp), y los tres utilizan la notación YYYY-MM-DD HH:MM:

Afortunadamente en PHP contamos con la función `strtotime`, que sirve para convertir la notación de fechas de MySQL a la de PHP, y con `date` que sirve para convertir la notación de fechas de PHP a la de MySQL.

Y en MySQL disponemos de la función `FROM_UNIXTIME`, que sirve para convertir la notación de fechas de PHP a la de MySQL, y con `UNIX_TIMESTAMP` que sirve para convertir la notación de MySQL a la de PHP.

En la siguiente tabla se describen algunas de las funciones de fecha y hora más importantes de PHP.

<code>checkdate</code>	<i>Bool checkdate (int \$mes , int \$día , int \$año)</i> Devuelve un booleano que indica si la fecha expresada es válida (obsérvese que el mes, día y año deben expresarse en formato numérico). En este caso el rango del año puede ir del 1 al 32767.
------------------------	---

`echo checkdate(2,29,2012);→true`

<code>date</code>	<i>string date (string \$formato [, int \$fecha = time()])</i> Devuelve la fecha expresada con el formato indicado. Si se omite la fecha se toma la actual. Los formatos se construyen con comodines;
-------------------	---

Los más frecuentes

- d representa el día con 2 dígitos.
- m representa el mes con dos dígitos
- y representa el año con cuatro dígitos
- l representa el día de la semana en inglés
- h horas
- i minutos
- s segundos

El resto de los formatos pueden consultarse en la dirección

<http://www.php.net/manual/es/function.date.php>

<code>getdate</code>	<i>array getdate ([int \$timestamp = time()])</i> Devuelve un array con los valores de la fecha (día, mes, año, ...)
----------------------	---

almacenados en sus elementos..

```
var_dump(getdate());
// Saca por pantalla
array (size=11)
  'seconds' => int 54
  'minutes' => int 36
  'hours' => int 19
  'mday' => int 24
  'wday' => int 4
  'mon' => int 1
  'year' => int 2022
  'yday' => int 23
  'weekday' => string 'Thursday' (length=8)
  'month' => string 'January' (length=7)
  0 => int 1359056214
```

mktime *Int mktime (int \$hora, int \$minuto, int \$segundo, int \$mes, int \$dia, int \$año)*

Devuelve la fecha PHP correspondiente a los datos introducidos.

```
echo mktime(0,0,0,7,7,2011);
```

strftime *string strftime (string \$formato [, int \$fecha = time()])*

Como la función date, devuelve la fecha con el formato indicado, pero es sensible a la información de localización (información regional) establecida con setlocale, de modo que si, por ejemplo, activamos la configuración regional de España podríamos ver los nombres de los días y de los meses en castellano. Los comodines del formato pueden consultarse en <http://www.php.net/manual/es/function.strftime.php>.

```
setlocale(LC_ALL,'esp');//Funciona en Windows
```

```
echo strftime('%A, %d de %B de %Y');
```

strtotime *Int strtotime (string \$fecha[,int \$fecha=time()])*

Convierte fecha expresada en forma de cadena de texto (por ejemplo, en el formato propio de MySQL) en una fecha expresada en época UNIX.

Cómo String se puede usar:

```
tomorrow, yesterday, next Saturday, +2 Months, - 3 days
```

Puede consultar en

<http://es.php.net/manual/es/datetime.formats.compound.php> todos los formatos que reconoce.

time

int time(void)

Devuelve la fecha PHP correspondiente al instante actual.

```
echo time();
```

Gestión de arrays

En la siguiente tabla se describen algunas de las funciones que nos ofrece PHP para gestionar datos de tipo array.

array_pop

mixed array_pop (array &\$matriz)

Extrae y devuelve el último elemento del array. Obsérvese que esta función actúa sobre el array original como indica el hecho de que reciba el argumento implícitamente por referencia.

```
$matriz=array('lunes','martes','miércoles','jueves','viernes','sábado','domingo');
```

```
echo array_pop($matriz);
```

```
var_dump($matriz);
```

array_push

Int array_push (array &\$matriz , mixed \$var, ...)

Inserta los elementos \$var al final del array y devuelve el número de elementos que contiene el array aumentado.

```
$matriz=array('lunes','martes','miércoles','jueves','viernes','sábado');
```

```
echo (array_push($matriz,'domingo'));
```

```
var_dump($matriz);
```

array_rand

mixed array_rand (array \$matriz , int \$numero)

Devuelve un array con \$numero índices (no datos) de la matriz elegidos al azar.

```
$matriz=array('lunes','martes','miércoles','jueves','viernes','sábado','domingo');
```

```
var_dump(array_rand($matriz,4));
```

array_replace

array array_replace (array&\$matriz_destino , array&\$matriz_origen);

Devuelve un array que es el resultado de sobrescribir/añadir sobre

	<p>matriz destino los elementos de matriz origen (los que coinciden en índice se sobrescriben, y los que no se añaden). No afecta a las matrices que recibe como argumento.</p> <pre>\$matriz_destino=array('altura'=>185,'peso'=>85); \$matriz_origen=array('pelo'=>'moreno','peso'=>95); var_dump(array_replace(\$matriz_destino,\$matriz_origen));</pre>
array_shift	<p><i>mixed array_shift (array &\$matriz)</i></p> <p>Elimina el primer elemento de matriz y lo devuelve.</p> <pre>\$matriz=array('lunes','martes','miércoles','jueves','viernes','sábado','domingo'); echo array_shift(\$matriz); var_dump(\$matriz);</pre>
array_slice	<p><i>array array_slice (array \$matriz , int \$inicio, int \$longitud)</i></p> <p>Devuelve un array compuesto por longitud elementos de matriz contados a partir del elemento inicio. No afecta a la matriz original.</p> <pre>\$matriz=array('lunes','martes','miércoles','jueves','viernes','sábado','domingo'); var_dump (array_slice(\$matriz,1,2));</pre>
array_splice	<p><i>array array_splice (array \$matriz , int \$inicio, int \$longitud)</i></p> <p>Elimina de matriz longitud elementos contados a partir del elemento inicio, y los devuelve en un array. Si los índices son numéricos los reajusta.</p> <pre>\$matriz=array('lunes','martes','miércoles','jueves','viernes','sábado','domingo'); var_dump (array_splice(\$matriz,1,2)); var_dump(\$matriz); \$matriz=array('altura'=>185,'peso'=>85,'pelo'=>'moreno'); var_dump(array_splice(\$matriz,1,2)); var_dump(\$matriz);</pre>
array_unshift	<p><i>int array_unshift (array &\$matriz , mixed \$var, ...)</i></p> <p>Inserta al principio de matriz los elementos var y devuelve la nueva longitud del array ampliado. Sólo permite añadir elementos con índice numérico (no admite pares \$índice=>\$valor).</p> <pre>\$matriz=array('altura'=>185,'peso'=>85,'pelo'=>'moreno'); echo(array_unshift(\$matriz, 'español'));</pre>

	<code>var_dump(\$matriz);</code>
count	<code>intcount (mixed \$var)</code> Devuelve el número de elemento que contiene el array. <code>\$matriz=array('lunes','martes','miércoles','jueves','viernes','sábado','domingo');</code> <code>echo count(\$matriz);</code>
implode	<code>string implode (string \$delimitador , array \$matriz)</code> Convierte matriz en una cadena de caracteres separando sus elemento con la cadena indicada en delimitador. <code>\$matriz=array(7,'julio',2011);</code> <code>echo implode(' de ', \$matriz);</code>
sort	<code>boolsort (array&\$matriz)</code> Ordena los datos (no respeta los índices) de matriz de menor a mayor y devuelve TRUE. Afecta al array original. <code>\$matriz=array('lunes','martes','miércoles','jueves','viernes','sábado','domingo');</code> <code>var_dump(sort(\$matriz));</code> <code>var_dump(\$matriz);</code>
shuffle	<code>boolshuffle (array&\$matriz)</code> Ordena al azar los datos (no respeta los índices) de matriz y devuelve TRUE. Afecta a la matriz original. <code>\$matriz=array('altura'=>185,'peso'=>85,'pelo'=>'moreno');</code> <code>var_dump(shuffle(\$matriz));</code> <code>var_dump(\$matriz);</code>

Identificación de tipos de datos

PHP nos ofrece funciones específicas para detectar el tipo de datos que contienen las variables, pero también otras que nos ayudarán a saber si una variable ha sido asignada o cuáles son todas las variables definidas (reconocibles) en el ámbito actual.

En la siguiente tabla se describen estas funciones:

isset	<i>boolisset (mixed \$var)</i> Indica si la variable existe y no tiene asignado el valor NULL. Muy útil para averiguar si se ha recibido una variable por \$_GET o \$_POST. \$variable1=NULL; var_dump(isset(\$variable1)); var_dump(isset(\$variable2));
is_bool	<i>boolis_bool (mixed \$variable)</i>
is_float	Devuelve un booleano que indica si la variable es del tipo inquirido.
is_int	var_dump(is_int(M_PI));
is_string	
is_array	
get_defined_vars	<i>array get_defined_vars (void)</i> Devuelve un array asociativo con absolutamente todas (superglobales, globales y locales) las variables definidas en el ámbito donde se llame a la función. var_dump(get_defined_vars());