

## FUNCIONES DE STRINGS

URL	<a href="http://php.net/manual/es/ref.strings.php">http://php.net/manual/es/ref.strings.php</a>
<a href="#">echo</a>	<p>Muestra una o más cadenas.</p> <pre>echo cadena1 [, cadena2...];</pre> <pre>echo "Hola mundo"; echo "Hola ", "mundo";</pre> <pre>\$mensaje="Aquí hay una cadena"; echo \$mensaje;</pre> <pre>define(MENSAJE, "Hola Mundo"); echo MENSAJE;</pre> <pre>echo "Intercalamos texto con valor de variables \$mensaje";</pre> <pre>echo 'No me muestra la variable \$mensaje';</pre>
<a href="#">print</a>	<p>Muestra una cadena. No es realmente una función. No necesita paréntesis.</p> <pre>int <b>print</b> ( string \$arg )</pre> <pre>print "para escapar caracteres se \"hace así\".";</pre>
<a href="#">ord</a>	<p>Devuelve el valor ASCII de una carácter</p> <pre>int <b>ord</b> ( string \$string )</pre> <p>Devuelve el valor ASCII del primer carácter de la <i>string</i>.</p> <p>Esta función Complementa a <a href="#">chr()</a>.</p> <pre>&lt;?php</pre> <pre>\$str = "\n";</pre> <pre>if (ord(\$str) == 10) {</pre> <pre>    echo "El primer caracter de \"\$str\" es un salto de linea.\n";</pre> <pre>}</pre> <pre>?&gt;</pre>
<a href="#">explode</a>	<p>Divide una cadena en varias cadenas</p> <pre>array <b>explode</b> ( string \$delimiter , string \$string [, int \$limit ] )</pre> <p>Devuelve un array de cadenas, siendo cada una de ellas una subcadena del parámetro <i>string</i> formado por la división según los delimitadores indicados en el parámetro <i>delimiter</i>.</p> <p>Si el parámetro <i>limit</i> es positivo, el array devuelto contendrá el máximo de elementos en el <i>limit</i> y el último elemento contendrá el resto del <i>string</i>.</p> <p>Si el parámetro <i>limit</i> es negativo, se devolverán todos los componentes a excepción el último <i>-limit</i>.</p> <p>Si el parámetro <i>limit</i> es cero, actuará como si su valor fuera 1.</p> <pre>&lt;?php</pre> <pre>// Ejemplo 1</pre> <pre>\$pizza = "piece1 piece2 piece3 piece4 piece5 piece6";</pre> <pre>\$pieces = explode(" ", \$pizza);</pre> <pre>echo \$pieces[0]; // piece1</pre> <pre>echo \$pieces[1]; // piece2</pre> <pre>// Ejemplo 2</pre> <pre>\$data = "foo*:1023:1000:/home/foo:/bin/sh";</pre> <pre>list(\$user, \$pass, \$uid, \$gid, \$gecos, \$home, \$shell) = explode(":", \$data);</pre> <pre>echo \$user; // foo</pre> <pre>echo \$pass; // *</pre> <pre>?&gt;</pre>

```
<?php
$input1 = "hello";
$input2 = "hello,there";
var_dump( explode( ',', $input1 ) );
var_dump( explode( ',', $input2 ) );
?>
```

```
array(1)
(
    [0] => string(5) "hello"
)
array(2)
(
    [0] => string(5) "hello"
    [1] => string(5) "there"
)
```

```
<?php
$str = 'one|two|three|four';

// positive limit
print_r(explode('|', $str, 2));

// negative limit (since PHP 5.1)
print_r(explode('|', $str, -1));
?>
```

```
Array
(
    [0] => one
    [1] => two|three|four
)
Array
(
    [0] => one
    [1] => two
    [2] => three
)
```

## Ejercicio1: Uso de Explode

Dado vuestro "Apellido1 Apellido2, Nombre".  
Obtener las siguientes salidas usando la función EXPLODE:  
Los apellidos son: Apellido1 Apellido2  
El nombre es: Nombre  
El nombre y apellidos son: Nombre Apellido1 Apellido2.  
Posteriormente mostrar el array de cadenas. Usar print\_r y var\_dump.  
¿Qué diferencias hay entre print\_r y var\_dump?

## Ejercicio2: Uso de Explode

```
$pizza = "piece1 piece2 piece3 piece4 piece5 piece6";
$pieces = explode(" ", $pizza);
echo $pieces[0]; // piece1
echo $pieces[1]; // piece2
```

Dado el ejercicio anterior, mostrar todos los piezas una en cada fila.  
Lo vamos a realizar independientemente del número de piezas.

# Las Cadenas en PHP

	<p><b>Ejercicio3: Uso de Explode</b></p> <pre>&lt;?php \$str = 'one two three four';  // positive limit print_r(explode(' ', \$str, 2));</pre> <p>Realizar:</p> <p>Modifica la variable \$str para que llegue hasta 10. Mostrar cada número en un elemento del array y el último elemento del array contenga del cinco hasta el final. Mostrar cada número en un elemento del array excepto los dos últimos. Comprobar con print_r.</p>
strstr	<p>Encuentra la primera aparición de un string. string strstr ( string \$haystack , mixed \$needle [, bool \$before_needle = false ] ) Devuelve parte del string haystack iniciando desde e incluyendo la primera aparición de needle (aguja) hasta el final del haystack. Si no encuentra la subcadena devuelve false.</p> <pre>\$email = 'name@example.com'; \$domain = strstr(\$email, '@'); echo \$domain; // mostrará @example.com echo"&lt;br/&gt;";  \$user = strstr(\$email, '@', true); // Desde PHP 5.3.0 echo \$user; // mostrará name echo"&lt;br/&gt;";  \$user2 = strstr(\$email, 's', true); // Desde PHP 5.3.0 echo \$user2; // No devuelve nada echo"&lt;br/&gt;";</pre> <p><b>Ejercicio 4: strstr</b></p> <p>Buscar el carácter "s" en la cadena \$email del ejemplo anterior. Si no lo encuentra mostrar "No ha encontrado 's' dentro del email".</p>
stristr	<p>strstr() insensible a mayúsculas y minúsculas.</p> <pre>\$email = 'user@example.com'; echo strstr(\$email, 'E');// No devuelve nada (false) echo"&lt;br/&gt;"; echo stristr(\$email, 'E');// Devuelve er@example.com echo"&lt;br/&gt;"; echo strstr(\$email, 'e', true); // A partir de PHP 5.3.0, salida us echo"&lt;br/&gt;";</pre>
strchr	<p>Es un alias a strstr</p>

strrchr	<p>Encuentra la última aparición de un caracter en un string</p> <p><b>string strrchr ( string \$haystack , mixed \$needle )</b></p> <p>Esta función devuelve la porción de haystack la cual comienza en la última aparición de needle y que va hasta el final de haystack. Si needle contiene más de un caracter, sólo se utiliza el primero. Este comportamiento es diferente que el de strstr().</p> <p>Si needle no es un string, se convierte en un entero y se aplica como el valor ordinal de un caracter.*/</p> <pre>// obtiene todo lo que está después de la última nueva línea \$text = "Line 1\nLine 2\nLine 3";  \$last = substr(strrchr(\$text,10),1); echo \$last;//Devuelve Line 3 echo"&lt;br/&gt;";  \$last = substr(strrchr(\$text,"\n"),1); echo \$last;//Devuelve Line 3 echo"&lt;br/&gt;";</pre> <p><b>Ejercicio 5:</b> Comprobar la equivalencia entre 10 y \n, mediante el uso de alguna función. Razona la respuesta.</p>
strpos	<p>Devuelve la posición de la primera aparición de una cadena a partir de un desplazamiento determinado (si se omite, es cero). La búsqueda diferencia entre mayúsculas y minúsculas.</p> <p>int <b>strpos</b> ( string \$haystack , string \$needle [, int \$offset = 0 ] )</p> <p><b>haystack</b> El string donde buscar.</p> <p><b>Needle</b> Si <b>needle</b> no es un string, se convierte en un integer y se aplica como el valor ordinal de un caracter.</p> <p><b>Offset</b> Si se especifica, la búsqueda iniciará en éste número de caracteres contados desde el comienzo del string. Si el valor es negativo, entonces la búsqueda iniciará desde esa cantidad de caracteres desde el final del string, buscando hacia atrás.</p> <pre>\$findme    = 'A'; \$string1   = 'xyz'; \$string2   = 'ABC';  \$pos1 = strpos(\$string1, \$findme); \$pos2 = strpos(\$string2, \$findme);  if (\$pos1 == false) {     echo "The string '\$findme' was not found in the string '\$string1'"; } echo"&lt;br/&gt;";// Nótese el uso de === //Usar sólo == no funcionará como se espera // debido a que la posición de 'a' es el 0º (primer) caracter. if (\$pos2 != false) {     echo "We found '\$findme' in '\$string2' at position \$pos2"; }</pre>

# Las Cadenas en PHP

	<pre>echo"&lt;br/&gt;";</pre> <p><b>Ejercicio 6: strpos</b> Comprobar el ejercicio anterior, y arreglarlo para que funcione. No devuelve " We found ...." Razona la respuesta.</p>
stripos	<p>Igual que strpos pero éste es insensible a mayúsculas y minúsculas.</p> <p><b>Ejercicio 7: stripos</b> Realiza un ejemplo basándote en el caso anterior con stripos.</p>
strrpos	Encuentra la posición de la última aparición de un substring en un string
stripos	Encuentra la posición de la última aparición de un substring insensible a mayúsculas y minúsculas en un string
strlen	<p>Obtiene la longitud de un string</p> <pre>int strlen ( string \$string )</pre> <pre>\$str = 'abcdef'; echo "La longitud de \\$str es ".strlen(\$str); // 6 echo"&lt;br/&gt;";</pre> <pre>\$str2 = ' ab cd '; echo "La longitud de \\$str2 es ".strlen(\$str2); // 7</pre>
trim	<p>Elimina espacio en blanco (u otro tipo de caracteres) del inicio y el final de la cadena</p> <pre>string trim ( string \$str [, string \$charlist = " \t\n\r\0\x0B" ] )</pre> <p>Esta función devuelve una cadena con los espacios en blanco eliminados del inicio y final del str. sin el segundo parámetro, trim() eliminará estos caracteres:</p> <pre>" " (ASCII 32 (0x20)), espacio simple. "\t" (ASCII 9 (0x09)), tabulación. "\n" (ASCII 10 (0x0A)), salto de línea. "\r" (ASCII 13 (0x0D)), retorno de carro. "\0" (ASCII 0 (0x00)), el byte NUL. "\x0B" (ASCII 11 (0x0B)), tabulación vertical.</pre> <pre>\$text  = "\t\tThese are a few words :) ... "; \$text2 = "These are a few words :) ... "; \$trimmed = trim(\$text); var_dump(\$trimmed); echo"&lt;br/&gt;"; \$trimmed2 = trim(\$text2); var_dump(\$trimmed2); echo"&lt;br/&gt;";echo"&lt;br/&gt;";</pre>

# Las Cadenas en PHP

	<pre>function trim_value(&amp;\$value) {     \$value = trim(\$value); }  \$fruit = array('apple','banana ', ' cranberry '); var_dump(\$fruit); echo"&lt;br/&gt;";echo"&lt;br/&gt;"; array_walk(\$fruit, 'trim_value'); var_dump(\$fruit);</pre>
Array_walk	<pre>\$frutas = array("d" =&gt; "limón", "a" =&gt; "naranja", "b" =&gt; "banana",                "c" =&gt; "manzana");  function test_alter(&amp;\$elemento1, \$clave, \$prefijo) {     \$elemento1 = "\$prefijo: \$elemento1"; }  function test_print(\$elemento2, \$clave) {     echo "\$clave. \$elemento2&lt;br /&gt;\n"; }  echo "Antes ...:\n"; array_walk(\$frutas, 'test_print');  array_walk(\$frutas, 'test_alter', 'fruta'); echo "... y después:\n";  array_walk(\$frutas, 'test_print');</pre> <p><b>Ejercicio 8: array_walk</b> Ejecuta y razona qué es lo que hace. ¿Qué significa &amp;?</p>
rtrim	Retira los espacios en blanco (u otros caracteres) del final de un string
ltrim	Retira espacios en blanco (u otros caracteres) del inicio de un string
substr	<p>Devuelve parte de una cadena</p> <pre>string substr ( string \$string , int \$start [, int \$length ] ) Devuelve una parte del string definida por los parámetros start y length.</pre> <pre>echo substr("abcdef", -1);    // devuelve "f" echo"&lt;br/&gt;"; echo substr("abcdef", -2);    // devuelve "ef" echo"&lt;br/&gt;"; echo substr("abcdef", -3, 1); // devuelve "d" echo"&lt;br/&gt;"; echo substr('abcdef', 1);     // bcdef echo"&lt;br/&gt;"; echo substr('abcdef', 1, 3);  // bcd echo"&lt;br/&gt;"; echo substr('abcdef', 0, 4);  // abcd echo"&lt;br/&gt;"; echo substr('abcdef', 0, 8);  // abcdef</pre>

# Las Cadenas en PHP

	<pre>echo"&lt;br/&gt;"; echo substr('abcdef', -1, 1); // f echo"&lt;br/&gt;";  // El acceso a caracteres específicos en una cadena // se puede conseguir usando "corchetes" \$string = 'abcdef'; echo \$string[0];                // a echo"&lt;br/&gt;"; echo \$string[3]; echo"&lt;br/&gt;";                    // d echo \$string[strlen(\$string)-1]; // f echo"&lt;br/&gt;";</pre> <p><b>Ejercicio 9: substr</b> Realizar el ejercicio 1 sin usar la función Explode.</p>
<pre>\$var=&lt;&lt;&lt;EOD ... EOD;</pre>	<pre>\$cadena1="Esto es una cadena de texto"; \$cadena2= &lt;&lt;&lt;Pepe Esta es otra cadena escrita con la sintaxis de documento incrustado. Se escribe en varias líneas y tiene la sintaxis siguiente. Después de escribir el nombre de la variable y el signo igual se ponen los tres &lt;&lt;&lt; y un nombre cualquiera. En este caso, Pepe. Luego hay que saltar de línea y escribir el texto con las líneas que se desee, pero cuidado... a la hora de visualizar la cadena con la instrucción echo todo este texto se verá seguido ya que para que se visualizaran saltos de línea en una página web habría que poner las famosas etiquetas &lt;BR/&gt;. Se indica el final de la cadena escribiendo de nuevo el nombre asignado en la primera línea -Pepe- pero teniendo la precaución de escribirlo en una línea nueva al final de todo el texto... Así como lo ves en el código fuente. Pepe;</pre> <pre>\$cadena3= &lt;&lt;&lt;Pepa Esta es otra cadena con el nombre Pepa puedo escribir Pepa cuantas veces quiera porque el PHP no interpretará el final de documento incrustado hasta que no la escriba en una sola línea y seguida del punto y coma Pepa;</pre> <pre>echo \$cadena1,"&lt;BR/&gt;"; echo \$cadena2,"&lt;BR/&gt;"; echo \$cadena3,"&lt;BR/&gt;";</pre>

# Las Cadenas en PHP

---

Uniando cadenas	<pre>\$cadena1="Esto es una cadena de texto"; \$cadena2="Esta es una segunda cadena de texto"; \$cadena3=127; \$cadena4=257.89; \$union1=\$cadena1 . \$cadena2; \$union2=\$cadena1 . \$cadena3; \$union3=\$cadena3 . \$cadena4; echo \$union1,"&lt;br&gt;"; echo \$union2,"&lt;br&gt;"; echo \$union3,"&lt;br&gt;"; \$cadena3 .=" Este es el texto que se añadirá a la variable cadena3"; echo \$cadena3,"&lt;br&gt;"; \$cadena3 .=" &lt;&lt;&lt;Pepito     Ahora le añado a la cadena     este trocillo asignado con el "formato"     de documento incrustado     Pepito; echo \$cadena3,"&lt;br&gt;";</pre>
-----------------	---