



APRENDERAPROGRAMAR.COM

FUNCIONES PHP PARA
CADENAS: STR_REPLACE,
STRTOLOWER, TOUPPER,
COUNT_CHARS, STRPOS,
TRIM, STR_REPEAT,
STRSTR, CHR. EJEMPLOS.
(CU00828B)

Sección: Cursos

Categoría: Tutorial básico del programador web: PHP desde cero

Fecha revisión: 2029

Resumen: Entrega nº28 del Tutorial básico "PHP desde cero".

Autor: Enrique González Gutiérrez

FUNCIONES BÁSICAS EN PHP

PHP además de permitirnos definir nuestras propias funciones, dispone de una gran cantidad de funciones propias del lenguaje a las que se denomina "funciones nativas" o "funciones del core o núcleo php". Ya conocemos algunas de ellas como strlen y substr. Vamos a ver otras funciones muy usadas relacionadas con el manejo de cadenas de caracteres (texto).



FUNCIÓN STR_REPLACE

Esta función se utiliza para reemplazar caracteres dentro de una cadena de caracteres. Es decir, nos permite definir una cadena que debe ser reemplazada con otra dentro de una frase o palabra. La función devuelve la frase original con todas las apariciones de la cadena a buscar reemplazadas con una cadena de reemplazo especificada.

La sintaxis básica habitual para esta función es:

```
str_replace ("cadena a buscar", "cadena de reemplazo", $variableOFraseOriginal)
```

Veámoslo con un ejemplo para entenderla mejor. Escribe este código y guárdalo con un nombre de archivo como ejemplo1.php. A continuación, sube el fichero al servidor y visualiza el resultado.

```
<?php //Ejemplo funciones básicas aprenderaprogramar.com
$texto = "Donde dije digo digo Diego.";
echo str_replace("Diego", "recortes", $texto);
echo "<br />";
echo $texto;
?>
```



Como podemos observar, en la cadena de caracteres \$texto, hemos sustituido la palabra o conjunto de caracteres "Diego" por "Recortes" y hemos devuelto el resultado sin modificar la variable de entrada.

Fijate que en el código que hemos escrito no hemos reemplazado el valor de la variable \$texto por un nuevo contenido. Únicamente hemos impreso por pantalla el resultado que nos devuelve la función. Si hubiéramos escrito lo siguiente sí hubiéramos cambiado el contenido de la variable: \$texto = str_replace("Diego", "recortes", \$texto);

La función permite pasar opcionalmente un parámetro adicional que permite saber a posteriori el número de reemplazos que se han realizado:

```
str_replace("cadena a buscar", "cadena de reemplazo", $variableOFraseOriginal, $numReemplazos)
```

Veámoslo con un ejemplo para entenderlo mejor:

```
<?php //Ejemplo funciones básicas aprenderaprogramar.com
$texto = "El dijo: es tarde ahora, pero es mejor si hay tres";
echo str_replace("es", "***", $texto, $reemplazos );
echo '<br/>Se han realizado: '.$reemplazos. ' reemplazos<br/>';
echo $texto;
?>
```

El resultado esperado es:

El dijo: ** tarde ahora, pero ** mejor si hay tr**
Se han realizado: 3 reemplazos
El dijo: es tarde ahora, pero es mejor si hay tres

Fijate que se reemplaza la cadena "es" indistintamente según esté separada por espacios o dentro de una palabra. Si sólo quisiéramos reemplazar la cadena si está separada por espacios podríamos indicar los espacios dentro de la cadena: echo str_replace(" es ", " ** ", \$texto, \$reemplazos);

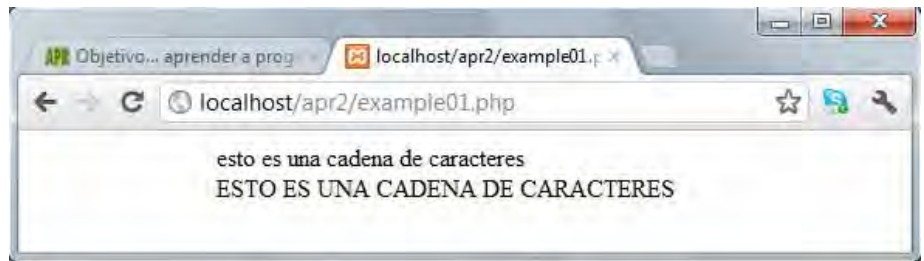
Aquí hemos indicado que se reemplace espacio-e-s-espacio con espacio-*-*espacio.

Si fuera necesario establecer reglas complejas de reemplazo, por ejemplo "reemplazar todas las palabras que empiecen por a y terminen por a con otra palabra, tendríamos que usar lo que se denominan expresiones regulares, pero no vamos a estudiar ahora la forma de definir y emplear expresiones regulares con PHP.

FUNCIONES STRTOLOWER Y STRTOUPPER

Las funciones strtolower y strtoupper transforman una cadena de caracteres en la misma cadena en minúsculas o mayúsculas respectivamente. Escribe ahora este código y guárdalo con un nombre de archivo como ejemplo4.php. A continuación, sube el fichero al servidor y visualiza el resultado.

```
<?php //Ejemplo funciones básicas aprenderaprogramar.com
$cadena = "EstO eS UnA cadeNA de CARacteres";
echo strtolower($cadena);
echo "<br />";
echo strtoupper($cadena);
?>
```



Como vemos el comportamiento de las funciones es sencillo. Simplemente devuelven la cadena de caracteres pasada como argumento en minúsculas o mayúsculas respectivamente.

Ten en cuenta que muchas veces para realizar comparaciones o para almacenar datos será interesante uniformizar la información que se almacena. Por ejemplo, si se pide una ciudad, un usuario puede introducir Buenos Aires, otro Buenos aires, otro BUENOS AIRES. Si usamos estas funciones, podemos uniformizar y hacer que siempre se muestren o guarden de una misma manera, lo cual facilitará el trabajo posterior.

FUNCIÓN COUNT_CHARS Y SUBSTR_COUNT

La función count_chars sirve para contar el número de apariciones de un carácter en una cadena. La sintaxis a emplear es la siguiente:

```
count_chars ( $cadena, $opcModo)
```

\$opcModo es un entero opcional. Si no se especifica vale 0 por defecto. Sus valores admitidos son:

- 0: se devolverá un array con el valor numérico ascii como índice y la frecuencia de cada carácter ascii como valor.
- 1: se devolverá un array con el valor numérico ascii como índice y la frecuencia de cada carácter que aparezca al menos una vez como valor.
- 2: se devolverá un array de caracteres que no aparecen en la cadena, con el valor numérico ascii como índice y la frecuencia de cada carácter ascii que no aparece como valor.

3: devuelve una cadena que contiene todos los caracteres únicos.

4: devuelve una cadena que contiene todos los caracteres no utilizados.

Nota: los códigos numéricos ascii van de 0 a 255 y no todos son visibles por pantalla.

Ejemplo:

```
<?php //Ejemplo count_chars aprenderaprogramar.com
$cadena = 'es jueves';
$miArray = count_chars ( $cadena, 1);
foreach ($miArray as $indiceNum => $veces) {
    echo 'Letra: '.chr($indiceNum).' , encontrada '. $veces.' veces<br/>';
}
?>
```

El resultado esperado es: Letra: , encontrada 1 veces. Letra: e , encontrada 3 veces. Letra: j , encontrada 1 veces. Letra: s , encontrada 2 veces. Letra: u , encontrada 1 veces. Letra: v , encontrada 1 veces

Otra función útil es **substr_count**(\$cadena, \$subcadena). Esta función nos devuelve el número de veces que aparece la subcadena dentro de la cadena. Por ejemplo:

```
<meta charset="utf-8">
<?php //Ejemplo curso PHP aprenderaprogramar.com
$cadena1 = 'Pedro Juan Luis Marco Luis Pedro Juan Luis Pedro';
$cadena2 = 'Luis';
echo 'Luis aparece '.substr_count($cadena1, $cadena2). ' veces'
?>
```

El resultado esperado es <<Luis aparece 3 veces>>

OTRAS FUNCIONES PARA CADENAS DE TEXTO

PHP define numerosas funciones nativas para el manejo de cadenas de texto. Vamos a ver algunas:

| FUNCIÓN | UTILIDAD | EJEMPLOS aprenderaprogramar.com |
|---|--|-----------------------------------|
| strlen (\$cadena) | Devuelve la longitud o número de caracteres de la cadena | Ver entregas anteriores del curso |
| substr (\$cadena, \$inicio, \$opcNumCar) | Si no se especifica \$opcNumCar devuelve la subcadena entre la posición \$inicio y el fin de cadena siendo la posición inicial la cero. Si se especifica \$opcNumCar extrae \$opcNumCar caracteres desde la posición \$inicio (inclusive). | Ver entregas anteriores del curso |

| FUNCIÓN | UTILIDAD | EJEMPLOS aprenderaprogramar.com |
|---|---|---|
| ucfirst(\$cadena); | Devuelve la cadena con la primera letra en mayúsculas | <pre>\$saludo = "hoy es viernes"; ucfirst(\$saludo); //Hoy es viernes</pre> |
| ucwords(\$cadena); | Devuelve la cadena con cada una de sus palabras con la primera letra en mayúsculas | <pre>\$saludo = "hoy es viernes"; echo ucwords(\$saludo); //Hoy Es Viernes</pre> |
| strpos (\$cadena , \$subCadena, \$opcPosInicio) | Devuelve la posición en que comienza la subcadena a partir de la posición inicial (cero) si no se especifica \$opcPosInicio, o a partir de la posición \$opcPosInicio si se especifica. Si no se encuentra la subcadena devuelve false. | <pre>\$saludo = "hoy es viernes"; echo strpos(\$saludo, "es"); //4</pre> |
| trim (\$cadena, \$opcCaracteres) | Si no se especifica \$opcCaracteres, devuelve la cadena eliminando espacios en blanco, tabuladores, saltos de línea y retornos de carro del principio y final de la cadena. Si se especifica \$opcCaracteres, se eliminan los caracteres especificados. | <pre>\$saludo = "**** hoy es viernes ****"; echo trim(\$saludo, "*"); //hoy es viernes</pre> <p>Nota: puede no funcionar como se espera debido a juego de caracteres, configuración local y configuración del servidor.</p> |
| ltrim (\$cadena, \$opcCaracteres) | Si no se especifica \$opcCaracteres, devuelve la cadena eliminando espacios en blanco, tabuladores, saltos de línea y retornos de carro del principio de la cadena. Si se especifica \$opcCaracteres, se eliminan los caracteres especificados. | <pre>\$saludo = "**** hoy es viernes ****"; echo ltrim(\$saludo, "*"); //hoy es viernes ***</pre> <p>Nota: puede no funcionar como se espera debido a juego de caracteres, configuración local y configuración del servidor.</p> |
| rtrim (\$cadena, \$opcCaracteres) | Si no se especifica \$opcCaracteres, devuelve la cadena eliminando espacios en blanco, tabuladores, saltos de línea y retornos de carro del final de la cadena. Si se especifica \$opcCaracteres, se eliminan los caracteres especificados. | <pre>\$saludo = "**** hoy es viernes ****"; echo rtrim(\$saludo, "*"); //**** hoy es viernes</pre> <p>Nota: puede no funcionar como se espera debido a juego de caracteres, configuración local y configuración del servidor.</p> |
| chop | Igual que rtrim | |
| str_repeat(\$cadena, \$numVeces) | Devuelve la cadena repetida \$numVeces siendo éste un número entero. | <pre>\$saludo = 'hola '; str_repeat(\$saludo, 3); //hola hola hola</pre> |
| strstr (\$cadena, \$desde, \$opcBoolean) | Devuelve la subcadena desde la primera aparición de \$desde (incluida) hasta el final si no se especifica \$opcBoolean ó este es false. Si \$opcBoolean es true devuelve la subcadena desde el inicio hasta la primera aparición de \$desde (excluida). | <pre>\$saludo = "hoy es viernes"; echo strstr(\$saludo, "es", false); //es viernes</pre> |
| strchr | Igual que strstr | |
| chr (\$ascii) | Devuelve el carácter correspondiente al código numérico entero \$ascii según el código ascii | <pre>echo chr(65). ' es la salida'; // A es la salida</pre> |

Existen más funciones para cadenas en PHP pero los programadores no las suelen conocer todas, ya que algunas son para tareas muy específicas. No te preocupes por memorizar las funciones ni por conocerlas todas, simplemente sé consciente de su existencia y aprende a buscarlas cuando te resulten necesarias. Si en algún momento necesitas realizar tareas muy específicas con cadenas en PHP puedes conocer todas las funciones existentes sobre cadenas acudiendo a la documentación oficial de PHP sobre funciones de cadenas, disponible en <http://php.net/manual/es/ref.strings.php>

EJERCICIO

Crear archivos php que ejecuten la tarea indicada:

- a) Dada la cadena "El abecedario completo es algo largo y detallarlo exhaustivamente es costoso", indicar cuántas veces aparece cada vocal usando funciones para cadenas PHP.
- b) Dada la cadena "El abecedario completo es algo largo y detallarlo exhaustivamente es costoso", indicar qué consonantes son las que aparecen y cuántas veces aparecen usando funciones para cadenas PHP.
- c) Dada la cadena "El abecedario completo es algo largo y detallarlo exhaustivamente es costoso", mostrar la cadena donde todas las a hayan sido sustituidas por el símbolo *. Por ejemplo en lugar de <<El abecedario...>> se habrá de mostrar <<El *beced*rio...>>
- d) Dada la cadena "El abecedario completo es algo largo y detallarlo exhaustivamente es costoso", mostrar la subcadena desde la primera aparición de <<completo>> excluyendo completo, es decir, deberá mostrarse << es algo largo y detallarlo exhaustivamente es costoso>>. Para ello usa una o varias funciones para cadenas PHP.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU00829B

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=70&Itemid=193