Cadenas PHP

Son una secuencia de caracteres.

Características:

Pueden tener cualquier longitud, sin más límite que la memoria disponible.
No tienen índice, pero sí posición.
Pueden ir encerradas entre comillas simples(') o dobles.(””).

Cadenas: ejemplos

Ejemplos de cadenas:

```
$string_1 = "Esto es una cadena con comillas dobles";
$string_2 = 'Mundo';                          // Cadena con comillas simples
$string_3 = "Hola, $string_2!<br/>"; //Muestra “Hola Mundo!”
$string_4 = 'Hola, $string_2!<br/>';  // Muestra “Hola, $string_2!”
$string_5 = "Uso un \ttabulador";     // Muestra“Uso un      tabulador”
$string_6 = 'Uso un \ttabulador';     // Muestra “Uso un \ttabulador”
$string_7 = “”;                               // Cadena con cero caracteres
```

Cadenas: 'simple'

Las cadenas delimitadas por comillas simples (como 'esta') son tratadas 'casi' literalmente

Ejemplo:

```
$variable = “nombre”;
$literal = 'Mi $variable no se imprimirá\\n';
$literal2= 'Mi $variable no se imprimirá\n';
```

Ambas mostrarán 'Mi $variable no se imprimirá\n';

Nota:
¿Cómo mostraríamos la siguiente cadena simple?
'Muestra literalmente los caracteres \' '

Cadenas: “dobles”

Las cadenas delimitadas por comillas dobles (como “esta”) son procesadas de este modo:

- Ciertas secuencias de caracteres empezando con la contrabarra (\) son sustituidas por caracteres especiales.

- Las variables (comienzan con $) son sustituidas por la representación de su valor convertido a cadena.

Cadenas: “dobles”

Ejemplo:

```
<?php
   $variable = "nombre";

   $literal = "Mi $variable se mostrará\\n";

   print($literal);
?>
```

Salida:
Mi nombre se mostrará

Cadenas “dobles”

Las secuencias de escape más comunes son:

Secuencia	Significado
\n	Avance de línea (ASCII 10)
\r	Retorno de carro (ASCII 13)
\t	Tabulación horizontal (ASCII 9
\v	Tabulación vertical (ASCII 11)
\f	Carácer de avance (ASCII 12)
\\	Barra invertida (a diferencia del inicio de una secuencia de escape
\$	Símbolo $ (no indica inicio de nombre de variable.
\”	Comilla doble (para que no marque el final de una cadena doble)
\'	Comilla simple (para que no marque el final de una cadena simple

Cadenas: concatenación

Para concatenar cadenas usamos el operador punto (.)

```
<?php
   $string1="Hola Mundo";
   $string2="1234";

   echo $string1 . " " . $string2;
?>
```

Resultado:
Hola Mundo 1234

Cadenas multilínea

Podemos especificar cadenas multilínea en PHP, simplemente insertando nuevas líneas entre las comillas:

```
<?php
$multilinea = “Esto es una cadena
multilínea, aunque si usas un IDE, tenderá a
provocar una concatenación por línea.
Pruébalo en tu IDE.”;
echo  $multilinea;
?>
```

Sintaxis heredoc

```
$str = <<<EOD
Ejemplo de una cadena
expandida en varias líneas
empleando la sintaxis heredoc.
EOD;
```

Sintaxis nowdoc

```
$str = <<<'EOD'
Ejemplo de un string
expandido en varias líneas
empleando la sintaxis nowdoc.
EOD;
```

Cadenas: Funciones

Existen en PHP un elevado número de funciones para facilitar su manipulación. Ver cualquiera de los siguiente enlaces

http://in.php.net/manual/en/ref.strings.php

http://www.w3schools.com/php/php_ref_string.asp

¡Casi 100 funciones!

Cadenas: Funciones

Entre las más útiles se encuentran estas:

strtolower
strtoupper
trim
ucfirst
ucwords
str_repeat
strlen
strstr
str_replace
substr
strpos
strchr

Función strlen()

Calcula la longitud de una cadena

```
<?php
   $mivar = “Hola”;
   echo strlen($mivar);              // Muestra 4
   echo strlen("Hola Mundo!");   // Muestra 11
?>
```

Resultado:
11

Observación: Si sustituimos “Hola” por '¡Hola' (un carácter más), ¿cuál sería el resultado?

La longitud de una cadena es muy utilizada en bucles y otras funciones, por ejemplo, para saber cuando termina la cadena.

Función substr()

Extrae una subcadena de caracteres de una cadena.

Parámetros:

- cadena de la que extraer la subcadena.
- Posición en la que comenzar a extraer los caracteres. Si el valor es negativo, substr() cuenta hacia atrás desde el final de la cadena.
- Número de caracteres a extraer (opcional). Si el valor es negativo se pierden tantos caracteres como indica este parámetro, a contar desde el final de la cadena.

Función substr()

Ejemplo:

```
$string = “Hola, Mundo”;
echo substr($string, 0, 5);          // Muestra 'Hola,”
echo substr($string, 6) . '<br/>';  // Muestra “Mundo”
echo substr($string, -1);            // Muestra 'o'
echo substr($string, -5, -1);        // Muestra 'Mund'
```

Función str_word_count()

Calcula el número de palabras que hay en una cadena

```
<?php
   echo str_word_count("¡Hola Mundo!");
?>
```

Resultado:
2

Función strrev()

Invierte una cadena

```
<?php
   echo strrev("¡Hola Mundo!");
?>
```

Resultado:
!odnuM aloH¡

Cadenas: mayúsculas y minúsculas

Función	Descripción
strtolower()	Convierte a minúsculas todos los caracteres de una cadena
strtoupper()	Convierte a mayúsculas todos los caracteres de una cadena
ucfirst()	Convierte a mayúsculas la primera letra de una cadena
lcfirst()	Convierte a minúsculas la primera letra de una cadena
ucwords()	Convierte a mayúsculas la primera palabra de una cadena

Cadenas: Buscar

Para buscar una cadena dentro de otra tenemos las siguientes funciones:

Función	Descripción
strstr()	
strpos()	
strrpos()	
substr()	
strpbrk()	Busca una cadena para cualquier lista de caracteres.

Cadenas-buscar: strstr()

Toma dos parámetros: la cadena en la que buscar y el texto a encontrar dentro de esa cadena. Retorna una subcadena, o *false* si no encuentra ninguna coincidencia.

Existe un tercer parámetro opcional cuyo default es ***false***.

En este caso, si encuentra el texto, la función devuelve la subcadena desde el inicio del texto encontrado.

```
$myString= “Hola, mundo cruel”;
Echo strstr($myString, “mun”);  // Devuelve 'mundo cruel'
Echo $strstr($myString, “xyz”);  // Devuelve false
```

Si el valor es ***true***, devuelve la subcadena que va desde el inicio de la cadena hasta el texto encontrado.

```
$myString= “Hola, mundo”;
Echo strstr($myString, “mun”, true);  // Devuelve 'Hola, '
```

Cadenas-buscar: strpos()

Devuelve el índice del primer carácter del texto buscado; sino encuentra el texto, devuelve false.

```
$myString= “Hola, mundo cruel”;
echo strpos($myString, “mun”);  // Muestra 6
echo strpos($myString, “xyz”);   // Muestra ' ' false
```

Un error muy común se produce cuando el texto buscado ocurre al inicio de la cadena; en este caso devuelve 0 (índice del primer carácter del texto encontrado) y puede confundirse con '***false***'

```
$myString= “Hola, mundo cruel”;
If ( !strpos ($myString, 'Hol') ) echo 'No encontrado'; //muestra 'No
encontrado'
```

Versión sin error del ejemplo anterior:

```
$myString= “Hola, mundo cruel”;
If ( strpos ($myString, 'Hol') === false) echo 'No encontrado';
//muestra 'No encontrado'
```

Cadenas-buscar: strpos()

La función admite un tercer argumento opcional: el índice de la cadena a partir del cual se inicia la búsqueda.

```
$myString= “Hola mundo cruel”;
echo strpos($myString, “o”);  // Muestra '1'
echo strpos($myString, “o”, 5);   // Muestra '9'
```

Combinando este argumento con el valor devuelto por la función, podemos encontrar todas las ocurrencias del texto de búsqueda.

```
$myString=”Hola mundo cruel”;
$pos=0;
While ( $pos= strpos($myString, “o”, $pos) ) !== false {
echo  “La letra 'o' fue encontrada en la posición: “ . $pos . '<br/>';
$pos++;
}
```

Cadenas-buscar: strrpos()

Similar a strpos(). La diferencia es que encuentra la última coincidencia en la cadena, en lugar de la primera.

```
<?php
$string = “Hola mundo”;
echo strpos($string, “o”) . '<br/>';   // Muestra '1'
echo strrpos($string, “o”) . '<br/>';  // Muestra '9'
?>
```

Pregunta: ¿Qué sucede si añadimos un número negativo como tercer parámetro?

Cadenas-buscar: strpbrk()

Devuelve una subcadena a partir del primer carácter encontrado perteneciente a un grupo de caracteres, o *false* si no contiene ninguno de los caracteres pasados como 2º parámetro.

Parámetros:

- Cadena en la que buscaremos cualquier carácter de entre un grupo de caracteres.
- Grupo de caracteres a buscar en la cadena.

ejemplo strpbrk()

```
<?php
$texto = 'Este es un texto Simple.';

// esto imprime "e es un texto Simple." ya que 'e' coincide primero
echo strpbrk($texto, 'me');

// esto imprime "Simple." ya que los caracteres son sensibles a mayúsculas/minúsculas
echo strpbrk($texto, 'S');
?>
```

Reemplazar: str_replace()

Reemplazar: substr_replace()

Reemplazar: strtr()

Cadenas-buscar: trim()

Cadenas: funciones no sensibles

La mayoría de las funciones de búsqueda y sustitución son sensibles a minúsculas y mayúsculas, pero también existen versiones no sensibles a mayúsculas y minúsculas.

Función	Equivalente no sensible a mayúsculas y minúsculas
strstr()	stristr()
strpos()	stripos()
strrpos()	strripos()
str_replace()	str_ireplace()

Función strchr()

Función str_repeat()