

Resumen Funciones de cadena.

1.- Algunas funciones usuales

strlen(cad): Devuelve la longitud de una cadena

strstr(cad, cad_bus), ***strchr(cad, cad_bus)***: Busca la aparición de una cadena dentro de otra y devuelve la subcadena comprendida entre la primera aparición y el final de cadena. Es case sensitive. Si no la encuentra devuelve una cadena vacía.

stristr(cad, cad_bus): Igual que la anterior. No es case sensitive.

strchr(cad, car_bus): Busca la aparición de un carácter y devuelve la subcadena comprendida entre la primera aparición y el final de cadena. Si no la encuentra devuelve una cadena vacía.. Es case sensitive

strpos(cad1, cad2, [, desplz]): Encuentra la primera posición de aparición de una cadena a partir de cierta posición (si no se indica el valor es 0). Es case sensitive.

strrpos(cad, carácter): Devuelve la posición de la última aparición de un carácter determinado en una cadena. Si no lo encuentra devuelve false. Es case sensitive

strspn(cadena, máscara): Obtenemos la longitud de la subcadena más larga que está formada sólo por caracteres contenidos en la máscara. Una vez encontrado un carácter que NO esté en la máscara se abandona la búsqueda. Es case sensitive

strcspn(cad, máscara): Obtenemos la longitud de la subcadena más larga que está formada sólo por caracteres NO contenidos en la máscara. Es case sensitive.

2.- Comparación de Cadenas

== vs ===

strcmp(cad1, cad2): Compara dos cadenas y devuelve un valor <0, si la segunda cadena es menor que la primera, mayor que 0 en caso contrario y 0 si son iguales. Es case sensitive

strcasecmp(cad1, cad2): Igual que la anterior NO case sensitive

strncmp(cad1, cad2, num): Funciona como strcmp, solo que permite comparar los num primeros caracteres de dos cadenas. Si alguna cadena es menor que num se usará su longitud como num para la comparación.

strnatcmp(cad1, cad2): Igual que strcmp pero utiliza la comparación natural de cadenas alfanuméricas. Es case sensitive

strnatcasecmp(cad1, cad2): Igual que la anterior NO case-sensitive

3.- Operar con Subcadenas

substr(cad, inicio [, tamaño]): Devuelve la subcadena que se encuentra a partir de la posición dada y hasta el final si no se indica el tamaño o hasta el tamaño.

Los argumentos pueden ser negativos de modo que:

Una posición de inicio negativa significa que debe empezarse desde el final.

Un tamaño negativo indica cuantos caracteres del final de cadena ignoraremos

substr_replace(cad1, cad2, inicio [, tamaño]): Devuelve una cadena que es el resultado de sustituir parte de la cadena original (indicado por la posición de inicio y opcionalmente por el tamaño) por el contenido de otra cadena. Es case sensitive y los argumentos de inicio y tamaño pueden ser negativos.

str_replace(cad1, cad2, cad_or): Devuelve una cadena que es el resultado de sustituir todas las apariciones de cad1 por cad2 en cad_or. Case sensitive

strtr(cadena, cadBus, cadRem): Devuelve una cadena que es el resultado de la sustitución de cada una de las apariciones, en la cadena original, de cada uno de los caracteres de una subcadena, por sus correspondientes caracteres dentro de una cadena de sustitución dada. Case sensitive

substr_count(cadena, patron): Devuelve el número de apariciones de una subcadena dentro de una cadena. Case sensitive

4.- Modificación del contenido

4.1.- Limpieza

chop(cad) rtrimcad(): quita espacios en blanco y finalización de a la derecha (final)

ltrimcad(): lo mismo a la izquierda(principio)

trim(cad): limpia al principio y al final

4.2.- Relleno

str_pad(cad, long): Rellena una cadena con un carácter de relleno (por defecto es el espacio en blanco) hasta que la cadena resultante tenga la longitud deseada. Opcionalmente se puede indicar el modo de relleno con los siguientes valores:

- STR_PAD_RIGHT: Rellena por la dcha. Opción por defecto
- STR_PAD_LEFT: Rellena por la izda
- STR_PAD_BOTH: Intenta rellenar por ámbos lados

4.3.- Conversión entre may y min

strtolower(cad): Convierte una cadena a minúsculas

strtoupper(cad): Convierte cadena a mayúsculas

ucfirst(cad): Convierte a mayúsculas el primer carácter de una cadena, siempre ue sea alfabético

ucwords(cad): Convierte a mayúsculas el primr carácter de cada palabra que forma una cadena

4.4.- Enmascaramiento de caracteres

addslashes(cad): Devuelve una cadena igual que la original en la que se han escapado los caracteres especiales “, ‘ y \

stripslashes(cad): Contrariamente a la función anterior elimina ‘\’ de escape

addcslashes(cad): Nos permite enmascarar cualquier carácter que desee el usuario

stripcslashes(cadena): Realiza lo contrario a lo anterior

quotemeta(cadena): Devuelve una cadena de caracteres igual que la original en la que se han escapado los caracteres especiales: . \ + * ? [^] (\$)

4.5.- División de Cadenas

strtok(cadena, divisor): Divide una cadena en diferentes subcadenas

chunk_split(): (cadena [, long [, separador]]): Permite marcar una cadena en porciones diferenciadas de tamaño menor. Podemos indicarle el tamaño de estas porciones e, incluso, el carácter o cadena con el que las separará. No modificará la cadena original.

split(patron, cadena): Devuelve un array resultado de dividir una cadena en diferentes subcadenas.

implode() explode: Lo veremos cuando veamos arrays

5.- Relacionadas con HTML

htmlspecialchars(cad): Se encarga de convertir los caracteres con significado especial en HTML en entidades HTML según lo siguiente:

& → &	< → <
“ → "	> → >

htmlentities(cad): Similar a la anterior pero traduce todos los caracteres a su entidad HTML

6.- Otras Funciones

chr(entero): Recibe el carácter ASCII asociado al entero

count_chars(cad [, modo]): Cuenta el número de apariciones de cada carácter dentro de una cadena. La manera que devuelve la información depende de un parámetro opcional cuyos valores son:

- 0: Devuelve una matriz asociativa con el valor del carácter como clave y su frecuencia como valor
- 1: Como el 0 pero sólo los caracteres que aparecen alguna vez
- 2: Como el 0 pero sólo los caracteres que no aparecen
- 3: devuelve una cadena que contiene los caracteres utilizados
- 4: Devuelve una cadena que contiene los caracteres NO utilizados

strrev(cad): Da la vuelta a una cadena

str_repeat(cad, veces): devuelve cad repetidas tantas veces como indique veces