**Funciones para cadenas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Sintaxis** | **Sirve para…** | **Ejemplo** |
| ctype\_alnum  ctype\_alpha  ctype\_cntrl  ctype\_digit  ctype\_graph  ctype\_lower  ctype\_print  ctype\_punct  ctype\_space | bool ctype\_...($string):  ctype\_alnum($string);  ctype\_alpha($string);  ctype\_cntrl($string);  ctype\_digit($string);  ctype\_graph($string);  ctype\_lower($string);  ctype\_print($string);  ctype\_punct($string);  ctype\_space($string); |  | $palabra="665fd65";  if(ctype\_alpha($palabra))  echo "tiene letras";  else  echo "no tiene letras";  //devuelve nontiene letras |
| strlen | int strlen ( $string); | Devuelve el numero de caracteres que se encuentrna en una cadena | $hola="holamundo";  $prub=strlen($hola);  echo $prub;  //da 9 |
| chr | string chr ( $ASCII ); | Convierte un número y solo un numero del codigo ascii en su respectivo caracter | $hola=112;  $prub=chr($hola);  echo $prub;  //devuelve p |
| ord | int ord ( $char ); | Devuelve el codigo ascii de un carácter en especifico | $hola="p";  $prub=ord($hola);  echo $prub;  //devuelve 112 |
| strtolower | string strtolower ( $string ); | Convierte todos los caracteres de un areglo en miniscula | $hola="SOYUNSTrING";  $prub=strtolower($hola);  echo $prub;  ///muesta soyunstring |
| strtoupper | string strtoupper ( $string ); | Convierte todos los careteres de un arreglo en mayúscula | $hola="SOYUNSTrING";  $prub=strtoupper($hola);  echo $prub;  ///muesta SOYUNASTRING |
| str\_shuffle | string str\_shuffle ( $string ); | Revuelve los caracteres contenidos en una string | $hola="SOYUNSTrING";  $prub=str\_shuffle($hola);  echo $prub;  ///muesta NONUYSSrTIG |
| strrev | string strrev ( $string ); | Invierte el orden de una cadena | $hola="SOYUNSTrING";  $prub= strrev ($hola);  echo $prub;  // devuelve GNIrTSNUYOS |
| ltrim | string ltrim ( $string[, $otros ] ); | Retira espacios en blanco (u otros caracteres) del inicio de un string | ltrim($text); |
| trim | string trim ( $string[, $otros ] ); | Elimina espacio en blanco (u otro tipo de caracteres) del inicio y el final de la cadena | trim($text); |
| rtrim | string rtrim ( $string[, $otros ] ); | Retira los espacios en blanco (u otros caracteres) del final de un string | rtrim($text); |
| substr | string substr ( $string , $ini [, int $long ] ); | Devuelve parte de una cadena | substr("abcdef", -1); // devuelve "f" |
| explode | array explode ( $str\_limite ,$cadena ); | Descompone una cadena en un arreglo | $pizza = "porción1 porción2 porción3 porción4 porción5 porción6";  $porciones = explode(" ", $pizza);  echo $porciones[0]; // porción1  echo $porciones[1]; // porción2 |
| implode | string implode ( $glue , $array );  string implode ( $array ); | Toma un arralgo y lo convierte en una cadena | $array = array('apellido', 'email', 'teléfono');  $separado\_por\_comas = implode(",", $array);  echo $separado\_por\_comas; // apellido,email,teléfono  // Devuelve un string vacío si se usa un array vacío:  var\_dump(implode('hola', array())); // string(0) |
| substr\_count | int substr\_count ( $pajar , $aguja ); | Cuenta el número de caracteres de una o más cadenas | $text = 'This is a test';  echo strlen($text); // 14  echo substr\_count($text, 'is'); // 2  // el string es reducido a 's is a test', así que muestra 1  echo substr\_count($text, 'is', 3); |
| str\_replace | mixed str\_replace ( $aguja , $reemplazo , $pajar ); | Devuelve parte de una cadena | // Produce: <body text='black'>  $bodytag = str\_replace("%body%", "black", "<body text='%body%'>"); |
| str\_ireplace | mixed str\_ireplace ( $aguja , $reemplazo , $pajar ); | Versión insensible a mayúsculas y minúsculas de str\_replace() | $bodytag = str\_ireplace("%body%", "black", "<body text=%BODY%>");  echo $bodytag; // <body text=black> |
| strpos | mixed strpos ( $pajar , $aguja ); | Encuentra la posición de la primera ocurrencia de un substring en un string | $mystring = 'abc';  $findme = 'a';  $pos = strpos($mystring, $findme);  if ($pos === false) {  echo "La cadena '$findme' no fue encontrada en la cadena '$mystring'";  } else {  echo "La cadena '$findme' fue encontrada en la cadena '$mystring'";  echo " y existe en la posición $pos";  } |
| stripos | mixed stripos ( $pajar , $aguja ); | Encuentra la posición de la primera aparición de un substring en un string sin considerar mayúsculas ni minúsculas | $findme = 'a';  $mystring1 = 'xyz';  $mystring2 = 'ABC';  $pos1 = stripos($mystring1, $findme);  $pos2 = stripos($mystring2, $findme);  // No, 'a' sin duda no está en 'xyz'  if ($pos1 === false) {  echo "El string '$findme' no se encontró en el strng '$mystring1'";  } |
| strrpos | mixed strrpos ( $pajar , $aguja ); | Encuentra la posición de la última aparición de un substring en un string | $pos = strrpos($mystring, "b");  if ($pos === false) { // nota: tres signos de igual  // no encontrado...  } |
| strripos | mixed strripos ( $pajar , $aguja ); | Encuentra la posición de la última aparición de un substring insensible a mayúsculas y minúsculas en un string |  |
| strstr | string strstr ( $pajar , $aguja ); | Encuentra la primera aparición de un string | $email = 'name@example.com';  $domain = strstr($email, '@');  echo $domain; // mostrará @example.com |
| stristr | string stristr ( $pajar , $aguja ); | strstr() insensible a mayúsculas y minúsculas | $email = 'USER@EXAMPLE.com';  echo stristr($email, 'e'); // salida ER@EXAMPLE.com  echo stristr($email, 'e', true); // A partir de PHP 5.3.0, salida US |
| strtr | string strtr ( $str , $origen , $trans );  string strtr ( $str , array($origen =>$trans) ); | Convierte caracteres o reemplaza substrings | $addr = strtr($addr, "äåö", "aao"); |