

## PHP STRINGS FUNCTION

**PHP:** indicates the earliest version of PHP that supports the function.

Function	Description
addslashes()	<p>Returns a string with backslashes in front of the specified characters</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello, my name is Kai Jim."; echo \$str."&lt;br /&gt;"; echo addslashes(\$str,'m')."&lt;br /&gt;"; echo addslashes(\$str,'K')."&lt;br /&gt;"; ?&gt;</pre> <p>Output Hello, my name is Kai Jim. Hello, \my na\me is Kai Ji\m. Hello, my name is \Kai Jim.</p>
addslashes()	<p>Returns a string with backslashes in front of predefined characters</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Who's Kai Jim?"; echo \$str . " This is not safe in a database query.&lt;br /&gt;"; echo addslashes(\$str) . " This is safe in a database query."; ?&gt;</pre> <p>Output Who's Kai Jim? This is not safe in a database query. Who\'s Kai Jim? This is safe in a database query.</p>
bin2hex()	<p>Converts a string of ASCII characters to hexadecimal values</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "a"; echo bin2hex(\$str) . "&lt;br /&gt;"; echo pack("H*",'61') . "&lt;br /&gt;"; ?&gt;</pre> <p>Output <b>61</b> a</p>
chop()	<p>Alias of rtrim()</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello World!\n\n";</pre>

	<pre>echo \$str; echo chop(\$str); ?&gt;</pre> <p>OutputHello World! Hello World!</p>
chr()	<p>Returns a character from a specified ASCII value</p> <p><b>Example</b></p> <pre>&lt;?php echo chr(52)."&lt;br /&gt;"; echo chr(052)."&lt;br /&gt;"; echo chr(0x52)."&lt;br /&gt;"; ?&gt;</pre> <p>Output4 * R</p>
chunk_split()	<p>Splits a string into a series of smaller parts</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello world!"; echo chunk_split(\$str,1,"."); ?&gt;</pre> <p>OutputH.e.l.l.o. .w.o.r.l.d.!</p>
convert_cyr_string()	<p>Converts a string from one Cyrillic character-set to another</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello world! æøå"; echo \$str."&lt;br /&gt;"; echo convert_cyr_string(\$str,'w','a'); ?&gt;</pre> <p>OutputHello world! æøå Hello world! ❖.❖❖❖</p>
convert_uuencode()	<p>Decodes a uuencoded string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = ",2&amp;5L;&amp;\@=V]R;&amp;0A `"; echo convert_uuencode(\$str); ?&gt;</pre>

	<p>OutputHello world!</p>
convert_uuencode()	<p>Encodes a string using the uuencode algorithm</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello world!"; echo convert_uuencode(\$str); ?&gt;</pre> <p>Output,2&amp;5L;&amp;\@=V]R;&amp;0A `</p>
count_chars()	<p>Returns how many times an ASCII character occurs within a string and returns the information</p> <p><b>Example</b></p> <pre>&lt;?php \$str="aaaaabbbbcc";         print_r(count_chars(\$str,1)); ?&gt;</pre> <p>Output</p> <p>Array ( [97] =&gt; 5 [98] =&gt; 4 [99] =&gt; 2 )</p>
crc32()	<p>Calculates a 32-bit CRC for a string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = crc32("Hello world!"); echo 'Without %u: '.\$str."&lt;br /&gt;"; echo 'With %u: '; printf("%u",\$str); ?&gt;</pre> <p>Output</p> <p>Without %u: 461707669 With %u: 461707669</p>
crypt()	<p>One-way string encryption (hashing)</p> <p><b>Example</b></p> <pre>&lt;?php if (CRYPT_STD_DES == 1){ echo "Standard DES: ".crypt("hello world")."\n&lt;br /&gt;"; } else{</pre>

	<pre> echo "Standard DES not supported.\n&lt;br /&gt;"; } if (CRYPT_EXT_DES == 1){ echo "Extended DES: ".crypt("hello world")."\n&lt;br /&gt;"; } else{ echo "Extended DES not supported.\n&lt;br /&gt;"; } if (CRYPT_MD5 == 1){ echo "MD5: ".crypt("hello world")."\n&lt;br /&gt;"; } else{ echo "MD5 not supported.\n&lt;br /&gt;"; } if (CRYPT_BLOWFISH == 1){ echo "Blowfish: ".crypt("hello world"); } else{ echo "Blowfish DES not supported."; } ?&gt;  OutputStandard DES: \$1\$xc3.m61.\$m64e6837acBtlUebIt3L. Extended DES not supported. MD5: \$1\$1i2..22.\$Eko3fsmLI50.m5Ej8Sieh0 Blowfish DES not supported. </pre>
<p>echo()</p>	<p>Outputs strings</p> <p><b>Example</b></p> <pre> &lt;?php \$str = "Who's Kai Jim?"; echo \$str; echo "&lt;br /&gt;"; echo \$str."&lt;br /&gt;I don't know!"; ?&gt;  OutputWho's Kai Jim? Who's Kai Jim? I don't know! </pre>
<p>explode()</p>	<p>Breaks a string into an array</p> <p><b>Example</b></p> <pre> &lt;?php \$str = "Hello world. It's a beautiful day."; print_r (explode(" ", \$str)); ?&gt; </pre>

	<p>OutputArray ( [0] =&gt; Hello [1] =&gt; world. [2] =&gt; It's [3] =&gt; a [4] =&gt; beautiful [5] =&gt; day. )</p>
<p>fprintf()</p>	<p>Writes a formatted string to a specified output stream</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello"; \$number = 123; \$file = fopen("test.txt","w"); echo fprintf(\$file,"%s world. Day number %u",\$str,\$number); ?&gt;</pre> <p>Output27</p>
<p>get_html_translation_table()</p>	<p>Returns the translation table used by htmlspecialchars() and htmlentities()</p> <p><b>Example</b></p> <pre>&lt;?php print_r (get_html_translation_table()); echo "&lt;br /&gt;"; print_r (get_html_translation_table(HTML_ENTITIES)); ?&gt;</pre> <p>Output</p> <pre>Array ( [""] =&gt; " [&lt;] =&gt;&lt; [&gt;] =&gt;&gt; [&amp;] =&gt;&amp; ) Array ( [ ] =&gt; [ ] =&gt; i [ ] =&gt; ç [ ] =&gt; £ [ ] =&gt; ¤ [ ] =&gt; ¥ [ ] =&gt; ¦ [ ] =&gt; § [ ] =&gt; ¨ [ ] =&gt; © [ ] =&gt; ª [ ] =&gt; « [ ] =&gt; ¬ [ ] =&gt; [ ] =&gt; ® [ ] =&gt; ¯ [ ] =&gt; ° [ ] =&gt; ± [ ] =&gt; ² [ ] =&gt; ³ [ ] =&gt; ´ [ ] =&gt; µ [ ] =&gt; ¶ [ ] =&gt; · [ ] =&gt; ¸ [ ] =&gt; ¹ [ ] =&gt;º [ ] =&gt; » [ ] =&gt; ¼ [ ] =&gt; ½ [ ] =&gt; ¾ [ ] =&gt; ¿ [ ] =&gt; À [ ] =&gt; Á [ ] =&gt; Â [ ] =&gt; Ã [ ] =&gt; Ä [ ] =&gt; Å [ ] =&gt; Æ [ ] =&gt; Ç [ ] =&gt; È [ ] =&gt; É [ ] =&gt; Ê [ ] =&gt; Ë [ ] =&gt; Ì [ ] =&gt; Í [ ] =&gt; Î [ ] =&gt; Ï [ ] =&gt; Ð [ ] =&gt; Ñ [ ] =&gt; Ò [ ] =&gt; Ó [ ] =&gt; Ô [ ] =&gt; Õ [ ] =&gt; Ö [ ] =&gt; × [ ] =&gt; Ø [ ] =&gt; Ù [ ] =&gt; Ú [ ] =&gt; Û [ ] =&gt; Ü [ ] =&gt; Ý [ ] =&gt; Þ [ ] =&gt; ß [ ] =&gt; à [ ] =&gt; á [ ] =&gt; â [ ] =&gt; ã [ ] =&gt; ä [ ] =&gt; å [ ] =&gt; æ [ ] =&gt; ç [ ] =&gt; è [ ] =&gt; é [ ] =&gt; ê [ ] =&gt; ë [ ] =&gt; ì [ ] =&gt; í [ ] =&gt; î [ ] =&gt; ï [ ] =&gt; ð [ ] =&gt; ñ [ ] =&gt; ò [ ] =&gt; ó [ ] =&gt; ô [ ] =&gt; õ [ ] =&gt; ö [ ] =&gt; ÷ [ ] =&gt; ø [ ] =&gt; ù [ ] =&gt; ú [ ] =&gt; û [ ] =&gt; ü [ ] =&gt; ý [ ] =&gt; þ [ ] =&gt; ÿ [""] =&gt; " [&lt;] =&gt;&lt; [&gt;] =&gt;&gt; [&amp;] =&gt;&amp; )</pre>
<p>hebrew()</p>	<p>Converts Hebrew text to visual text</p> <p><b>Example</b></p>

	<pre>&lt;?php echo hebrevc("á çùãï äúùñâ"); ?&gt;</pre> <p>Output á çùãï äúùñ</p>
hebrevc()	<p>Converts Hebrew text to visual text and new lines (\n) into &lt;br /&gt;</p> <p><b>Example</b></p> <pre>&lt;?php echo hebrevc("á çùãï äúùñâ\ná çùãï äúùñâ"); ?&gt; //Output á çùãï äúùñ á çùãï äúùñ</pre>
html_entity_decode()	<p>Converts HTML entities to characters</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Jane &amp; #039;Tarzan#039;"; echo html_entity_decode(\$str); echo "&lt;br /&gt;"; echo html_entity_decode(\$str, ENT_QUOTES); echo "&lt;br /&gt;"; echo html_entity_decode(\$str, ENT_NOQUOTES); ?&gt;</pre> <p>Output</p> <p>Jane &amp; 'Tarzan'</p> <p>Jane &amp; 'Tarzan'</p> <p>Jane &amp; 'Tarzan'</p>
htmlentities()	<p>Converts characters to HTML entities</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Jane &amp; 'Tarzan'"; echo htmlentities(\$str, ENT_COMPAT); echo "&lt;br /&gt;"; echo htmlentities(\$str, ENT_QUOTES); echo "&lt;br /&gt;"; echo htmlentities(\$str, ENT_NOQUOTES); ?&gt;</pre> <p>Output</p> <p>Jane &amp; 'Tarzan'</p> <p>Jane &amp; 'Tarzan'</p> <p>Jane &amp; 'Tarzan'</p>
htmlspecialchars_decode()	<p>Converts some predefined HTML entities to characters</p>

	<p><b>Example</b></p> <pre>&lt;?php \$str = "Jane &amp; #039;Tarzan#039;"; echo htmlspecialchars_decode(\$str); echo "&lt;br /&gt;"; echo htmlspecialchars_decode(\$str, ENT_QUOTES); echo "&lt;br /&gt;"; echo htmlspecialchars_decode(\$str, ENT_NOQUOTES); ?&gt;</pre> <p>Output Jane &amp; 'Tarzan' Jane &amp; 'Tarzan' Jane &amp; 'Tarzan'</p>
htmlspecialchars()	<p>Converts some predefined characters to HTML entities</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Jane &amp; 'Tarzan'"; echo htmlspecialchars(\$str, ENT_COMPAT); echo "&lt;br /&gt;"; echo htmlspecialchars(\$str, ENT_QUOTES); echo "&lt;br /&gt;"; echo htmlspecialchars(\$str, ENT_NOQUOTES); ?&gt;</pre> <p>Output Jane &amp; 'Tarzan' Jane &amp; 'Tarzan' Jane &amp; 'Tarzan'</p>
implode()	<p>Returns a string from the elements of an array</p> <p><b>Example</b></p> <pre>&lt;?php \$arr = array('Hello','World!','Beautiful','Day!'); echo implode(" ",\$arr); ?&gt;</pre> <p>Output Hello World! Beautiful Day!</p>
join()	<p>Alias of implode()</p> <p><b>Example</b></p> <pre>&lt;?php \$arr = array('Hello','World!','Beautiful','Day!');</pre>

	<pre>echo join(" ",\$arr); ?&gt;</pre> <p>OutputHello World! Beautiful Day!</p>
levenshtein()	<p>Returns the Levenshtein distance between two strings</p> <p><b>Example</b></p> <pre>&lt;?php echo levenshtein("Hello World","ello World"); echo "&lt;br /&gt;"; echo levenshtein("Hello World","ello World",10,20,30); ?&gt;</pre> <p>Output</p> <p>1 30</p>
localeconv()	<p>Returns locale numeric and monetary formatting information</p> <p><b>Example</b></p> <pre>&lt;?php setlocale(LC_ALL, 'US'); \$locale_info = localeconv(); print_r(\$locale_info); ?&gt;</pre> <p>Output</p> <pre>Array ( [decimal_point] =&gt; . [thousands_sep] =&gt; , [int_curr_symbol] =&gt; USD [currency_symbol] =&gt; \$ [mon_decimal_point] =&gt; . [mon_thousands_sep] =&gt; , [positive_sign] =&gt; [negative_sign] =&gt; - [int_frac_digits] =&gt; 2 [frac_digits] =&gt; 2 [p_cs_precedes] =&gt; 1 [p_sep_by_space] =&gt; 0 [n_cs_precedes] =&gt; 1 [n_sep_by_space] =&gt; 0 [p_sign_posn] =&gt; 3 [n_sign_posn] =&gt; 0 [grouping] =&gt; Array ( [0] =&gt; 3 ) [mon_grouping] =&gt; Array ( [0] =&gt; 3 ) )</pre>
ltrim()	<p>Strips whitespace from the left side of a string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "  Hello World!"; echo "Without ltrim: " . \$str; echo "&lt;br /&gt;"; echo "With ltrim: " . ltrim(\$str); ?&gt;</pre> <p>Output</p>



	<p>Without ltrim: Hello World!          With ltrim: Hello World!</p>
md5()	<p>Calculates the MD5 hash of a string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello"; echo md5(\$str); ?&gt;</pre> <p>Output8b1a9953c4611296a827abf8c47804d7</p>
md5_file()	<p>Calculates the MD5 hash of a file</p> <p><b>Example</b></p> <pre>&lt;?php \$filename = "test.txt"; \$md5file = md5_file(\$filename); echo \$md5file; ?&gt;</pre> <p>Output8d75db5b6b096ff2096f895a57627b98</p>
metaphone()	<p>Calculates the metaphone key of a string</p> <p><b>Example</b></p> <pre>&lt;?php echo metaphone("world"); ?&gt;</pre> <p>OutputWRLT</p>
money_format()	<p>Returns a string formatted as a currency string</p> <p><b>Example</b></p> <pre>&lt;?php \$number = 1234.56; setlocale(LC_MONETARY, "en_US"); echo money_format("The price is %i", \$number); ?&gt;</pre> <p>OutputThe price is USD 1,234.56</p> <p>Note: -The money_format() function does not work on Windows platforms.</p>
nl2br()	<p>Inserts HTML line breaks in front of each newline in a string</p>

	<p><b>Example</b></p> <pre>&lt;?php echo nl2br("One line.\nAnother line."); ?&gt;</pre> <p>Output</p> <p>One line. Another line.</p>
<p>number_format()</p>	<p>Formats a number with grouped thousands</p> <p><b>Example</b></p> <pre>&lt;?php echo number_format("1000000"); echo "&lt;br /&gt;"; echo number_format("1000000",2); echo "&lt;br /&gt;"; echo number_format("1000000",2,"","."); ?&gt;</pre> <p>Output</p> <p>1,000,000 1,000,000.00 1.000.000,00</p>
<p>ord()</p>	<p>Returns the ASCII value of the first character of a string</p> <p><b>Example</b></p> <pre>&lt;?php echo ord("h")."&lt;br /&gt;"; echo ord("hello")."&lt;br /&gt;"; ?&gt;</pre> <p>Output</p> <p>104 104</p>
<p>parse_str()</p>	<p>Parses a query string into variables</p> <p><b>Example</b></p> <pre>&lt;?php parse_str("id=23&amp;name=Kai%20Jim"); echo \$id."&lt;br /&gt;"; echo \$name; ?&gt;</pre>

	<p>Output23 Kai Jim</p>
print()	<p>Outputs a string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Who's Kai Jim?"; print \$str; print "&lt;br /&gt;"; print \$str."&lt;br /&gt;I don't know!"; ?&gt;</pre> <p><b>Output</b></p> <p>Who's Kai Jim? Who's Kai Jim? I don't know!</p>
printf()	<p>Outputs a formatted string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello"; \$number = 123; printf("%s world. Day number %u",\$str,\$number); ?&gt;</pre> <p>OutputHello world. Day number 123</p>
quoted_printable_decode()	<p>Decodes a quoted-printable string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello=0Aworld."; echo quoted_printable_decode(\$str); ?&gt;</pre> <p>OutputHello world.</p>
quotemeta()	<p>Quotes meta characters</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello world. (can you hear me?)";</pre>

	<pre>echo quotemeta(\$str); ?&gt;</pre> <p>OutputHello world\. \.(can you hear me\?)</p>
rtrim()	<p>Strips whitespace from the right side of a string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello World! "; echo "Without rtrim: " . \$str; echo "&lt;br /&gt;"; echo "With rtrim: " . rtrim(\$str); ?&gt;</pre> <p>OutputWithout rtrim: Hello World! With rtrim: Hello World!</p>
setlocale()	<p>Sets locale information</p> <p><b>Example</b></p> <pre>&lt;?php echo setlocale(LC_ALL,"En-US"); echo "&lt;br /&gt;"; echo setlocale(LC_ALL,NULL); ?&gt;</pre> <p>OutputEnglish_United States.1252</p>
sha1()	<p>Calculates the SHA-1 hash of a string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = 'Hello'; echo sha1(\$str); ?&gt;</pre> <p><b>Output</b>f7ff9e8b7bb2e09b70935a5d785e0cc5d9d0abf0</p>
sha1_file()	<p>Calculates the SHA-1 hash of a file</p> <p><b>Example</b></p> <pre>&lt;?php \$filename = "test.txt"; \$sha1file = sha1_file(\$filename); echo \$sha1file; ?&gt;</pre>

similar_text()	<p>Calculates the similarity between two strings</p> <p><b>Example</b></p> <pre>&lt;?php echo similar_text("Hello World","Hello Peter"); ?&gt;</pre> <p>Output 7</p>
soundex()	<p>Calculates the soundex key of a string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "hello"; echo soundex(\$str); ?&gt;</pre> <p>OutputH400</p>
sprintf()	<p>Writes a formatted string to a variable</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello"; \$number = 123; \$txt = sprintf("%s world. Day number %u",\$str,\$number); echo \$txt; ?&gt; //OutputHello world. Day number 123</pre>
sscanf()	<p>Parses input from a string according to a format</p> <p><b>Example</b></p> <pre>&lt;?php \$string = "age:30 weight:60kg"; sscanf(\$string,"age:%d weight:%dkg",\$age,\$weight); // show types and values var_dump(\$age,\$weight); ?&gt;</pre> <p>Outputint(30) int(60)</p>
str_ireplace()	<p>Replaces some characters in a string (case-insensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo str_ireplace("WORLD","Peter","Hello world!"); ?&gt;</pre> <p>OutputHello Peter!</p>

<p>str_pad()</p>	<p>Pads a string to a new length</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello World"; echo str_pad(\$str,20,"."); ?&gt;</pre> <p><b>Output</b>Hello World.....</p>
<p>str_repeat()</p>	<p>Repeats a string a specified number of times</p> <p><b>Example</b></p> <pre>&lt;?php echo str_repeat("p",13); ?&gt;</pre> <p>Outputpppppppppppppp</p>
<p>str_replace()</p>	<p>Replaces some characters in a string (case-sensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo str_replace("world","Peter","Hello world!"); ?&gt;</pre> <p>OutputHello Peter!</p>
<p>str_rot13()</p>	<p>Performs the ROT13 encoding on a string</p> <p><b>Example</b></p> <pre>&lt;?php echo str_rot13("Hello World"); echo "&lt;br /&gt;"; echo str_rot13("Uryyb Jbeyq"); ?&gt;</pre> <p>OutputUryyb Jbeyq Hello World</p>
<p>str_shuffle()</p>	<p>Randomly shuffles all characters in a string</p> <p><b>Example</b></p> <pre>&lt;?php echo str_shuffle("Hello World"); ?&gt;</pre>
<p>str_split()</p>	<p>Splits a string into an array</p> <p><b>Example</b></p>

	<pre>&lt;?php print_r(str_split("Hello")); ?&gt;</pre> <p>OutputArray ( [0] =&gt; H [1] =&gt; e [2] =&gt; l [3] =&gt; l [4] =&gt; o )</p>
str_word_count()	<p>Count the number of words in a string</p> <p><b>Example</b></p> <pre>&lt;?php echo str_word_count("Hello world!"); ?&gt;</pre> <p>Output 2</p>
strcasecmp()	<p>Compares two strings (case-insensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo strcasecmp("Hello world!","HELLO WORLD!"); ?&gt;</pre> <p>Output 0</p>
strchr()	<p>Finds the first occurrence of a string inside another string (alias of strstr())</p> <p><b>Example</b></p> <pre>&lt;?php echo strchr("Hello world!","world"); ?&gt;</pre> <p>Outputworld!</p>
strcmp()	<p>Compares two strings (case-sensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo strcmp("Hello world!","Hello world!"); ?&gt;</pre> <p>Output 0</p>
strcoll()	<p>Locale based string comparison</p> <p><b>Example</b></p> <pre>&lt;?php setlocale (LC_COLLATE, 'NL'); echo strcoll("Hello World!","Hello WORLD!"); echo "&lt;br /&gt;"; setlocale (LC_COLLATE, 'en_US');</pre>

	<pre>echo strcoll("Hello World!","Hello WORLD!"); ?&gt;</pre> <p>Output:</p> <pre>1 1</pre>
strcspn()	<p>Returns the number of characters found in a string before any part of some specified characters are found</p> <p><b>Example</b></p> <pre>&lt;?php echo strcspn("Hello world!","w"); ?&gt;</pre> <p>Output 6</p>
strip_tags()	<p>Strips HTML and PHP tags from a string</p> <p><b>Example</b></p> <pre>&lt;?php echo strip_tags("Hello &lt;b&gt;world!&lt;/b&gt;"); ?&gt;</pre> <p>OutputHello world!</p>
stripslashes()	<p>Unquotes a string quoted with addslashes()</p> <p><b>Example</b></p> <pre>&lt;?php echo stripslashes("Hello, \my na\me is Kai Ji\m."); ?&gt;</pre> <p>OutputHello, my name is Kai Jim.</p>
stripslashes()	<p>Unquotes a string quoted with addslashes()</p> <p><b>Example</b></p> <pre>&lt;?php echo stripslashes("Who\'s Kai Jim?"); ?&gt;</pre> <p>OutputWho's Kai Jim?</p>
strpos()	<p>Returns the position of the first occurrence of a string inside another string (case-insensitive)</p> <p><b>Example</b></p>



	<pre>&lt;?php echo strpos("Hello world!","WO"); ?&gt;</pre> <p>Output 6</p>
stristr()	<p>Finds the first occurrence of a string inside another string (case-insensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo stristr("Hello world!","WORLD"); ?&gt;</pre> <p>Outputworld!</p>
strlen()	<p>Returns the length of a string</p> <p><b>Example</b></p> <pre>&lt;?php echo strlen("Hello world!"); ?&gt;</pre> <p>Output 12</p>
strnatcasecmp()	<p>Compares two strings using a "natural order" algorithm (case-insensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo strnatcasecmp("2Hello world!","10Hello world!"); echo "&lt;br /&gt;"; echo strnatcasecmp("10Hello world!","2Hello world!"); ?&gt;</pre> <p>Output-1 1</p>
strnatcmp()	<p>Compares two strings using a "natural order" algorithm (case-sensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo strnatcasecmp("2Hello world!","10Hello world!"); echo "&lt;br /&gt;"; echo strnatcasecmp("10Hello world!","2Hello world!");</pre>

	<p>?&gt;</p> <p><b>Output</b></p> <p>-1</p> <p>1</p>
strncasecmp()	<p>String comparison of the first n characters (case-insensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo strncasecmp("Hello world!","hello earth!",6); ?&gt;</pre> <p>Output 0</p>
strncmp()	<p>String comparison of the first n characters (case-sensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo strncmp("Hello world!","Hello earth!",6); ?&gt;</pre> <p>Output 0</p>
strpbrk()	<p>Searches a string for any of a set of characters</p> <p><b>Example</b></p> <pre>&lt;?php echo strpbrk("Hello world!","oe"); ?&gt;</pre> <p>Outputello world!</p>
strpos()	<p>Returns the position of the first occurrence of a string inside another string (case-sensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo strpos("Hello world!","wo"); ?&gt;</pre> <p>Output 6</p>
strrchr()	<p>Finds the last occurrence of a string inside another string</p> <p><b>Example</b></p>

	<pre>&lt;?php echo strrchr("Hello world!","world"); ?&gt;</pre> <p>Outputworld!</p>
strrev()	<p>Reverses a string</p> <p><b>Example</b></p> <pre>&lt;?php echo strrev("Hello World!"); ?&gt;</pre> <p>Output!dlroW olleH</p>
strrpos()	<p>Finds the position of the last occurrence of a string inside another string (case-insensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo strrpos("Hello world!","WO"); ?&gt;</pre> <p>Output 6</p>
strrpos()	<p>Finds the position of the last occurrence of a string inside another string (case-sensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo strrpos("Hello world!","wo"); ?&gt;</pre> <p>Output 6</p>
strspn()	<p>Returns the number of characters found in a string that contains only characters from a specified charlist</p> <p><b>Example</b></p> <pre>&lt;?php echo strspn("Hello world!","kHlleo"); ?&gt;</pre> <p>Output 5</p>
strstr()	<p>Finds the first occurrence of a string inside another string (case-sensitive)</p> <p><b>Example</b></p>

	<pre>&lt;?php echo strstr("Hello world!","world"); ?&gt;</pre> <p>Outputworld!</p>
strtok()	<p>Splits a string into smaller strings</p> <p><b>Example</b></p> <pre>&lt;?php \$string = "Hello world. Beautiful day today."; \$token = strtok(\$string, " "); while (\$token != false) {     echo "\$token&lt;br /&gt;";     \$token = strtok(" "); } ?&gt;</pre> <p>OutputHello world. Beautiful day today.</p>
strtolower()	<p>Converts a string to lowercase letters</p> <p><b>Example</b></p> <pre>&lt;?php echo strtolower("Hello WORLD."); ?&gt;</pre> <p>Outputhello world.</p>
strtoupper()	<p>Converts a string to uppercase letters</p> <p><b>Example</b></p> <pre>&lt;?php echo strtoupper("Hello WORLD!"); ?&gt;</pre> <p>OutputHELLO WORLD!</p>
strtr()	<p>Translates certain characters in a string</p> <p><b>Example</b></p>

	<pre>&lt;?php echo strstr("Hilla World","ia","eo"); ?&gt;</pre> <p>OutputHello World</p>
substr()	<p>Returns a part of a string</p> <p><b>Example</b></p> <pre>&lt;?php echo substr("Hello world!",6); ?&gt;//Outputworld!</pre>
substr_compare()	<p>Compares two strings from a specified start position (binary safe and optionally case-sensitive)</p> <p><b>Example</b></p> <pre>&lt;?php echo substr_compare("Hello world","Hello world",0); ?&gt;</pre> <p>Output 0</p>
substr_count()	<p>Counts the number of times a substring occurs in a string</p> <p><b>Example</b></p> <pre>&lt;?php echo substr_count("Hello world. The world is nice","world"); ?&gt;</pre> <p>Output 2</p>
substr_replace()	<p>Replaces a part of a string with another string</p> <p><b>Example</b></p> <pre>&lt;?php echo substr_replace("Hello world","earth",6); ?&gt;</pre> <p>OutputHello earth</p>
trim()	<p>Strips whitespace from both sides of a string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = " Hello World! "; echo "Without trim: " . \$str; echo "&lt;br /&gt;";</pre>

	<pre>echo "With trim: " . trim(\$str); ?&gt;</pre> <p>Output Without trim: Hello World! With trim: Hello World!</p>
ucfirst()	<p>Converts the first character of a string to uppercase</p> <p><b>Example</b></p> <pre>&lt;?php echo ucfirst("hello world"); ?&gt;</pre> <p>Output Hello world</p>
ucwords()	<p>Converts the first character of each word in a string to uppercase</p> <p><b>Example</b></p> <pre>&lt;?php echo ucwords("hello world"); ?&gt;</pre> <p>Output Hello World</p>
fprintf()	<p>Writes a formatted string to a specified output stream</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello"; \$number = 123; \$file = fopen("test.txt","w"); echo fprintf(\$file,"%s world. Day number %u",array(\$str,\$number)); ?&gt;</pre> <p>Output 27</p> <p>(test.txt will be written "Hello world. Day number 123")</p>
vprintf()	<p>Outputs a formatted string</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello"; \$number = 123; vprintf("%s world. Day number %u",array(\$str,\$number)); ?&gt;</pre> <p>Output Hello world. Day number 123</p>

vsprintf()	<p>Writes a formatted string to a variable</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "Hello"; \$number = 123; \$txt = vsprintf("%s world. Day number %u",array(\$str,\$number)); echo \$txt; ?&gt;</pre> <p>OutputHello world. Day number 123</p>
wordwrap()	<p>Wraps a string to a given number of characters</p> <p><b>Example</b></p> <pre>&lt;?php \$str = "An example on a long word is: Supercalifragulistic"; echo wordwrap(\$str,15); ?&gt;</pre> <p>OutputAn example on a long word is: Supercalifragulistic</p>