**Text Helper**

The Text Helper file contains functions that assist in working with text.

* [Loading this Helper](https://codeigniter.com/user_guide/helpers/text_helper.html#loading-this-helper)
* [Available Functions](https://codeigniter.com/user_guide/helpers/text_helper.html#available-functions)

[**Loading this Helper**](https://codeigniter.com/user_guide/helpers/text_helper.html#id1)

This helper is loaded using the following code:

$this**->**load**->**helper('text');

[**Available Functions**](https://codeigniter.com/user_guide/helpers/text_helper.html#id2)

The following functions are available:

**word\_limiter(*$str*[, *$limit = 100*[, *$end\_char = '&#8230;'*]])**

|  |  |
| --- | --- |
| **Parameters:** | * **$str** (*string*) – Input string * **$limit** (*int*) – Limit * **$end\_char** (*string*) – End character (usually an ellipsis) |
| **Returns:** | Word-limited string |
| **Return type:** | string |

Truncates a string to the number of *words* specified. Example:

$string **=** "Here is a nice text string consisting of eleven words.";

$string **=** word\_limiter($string, 4);

*// Returns: Here is a nice*

The third parameter is an optional suffix added to the string. By default it adds an ellipsis.

**character\_limiter(*$str*[, *$n = 500*[, *$end\_char = '&#8230;'*]])**

|  |  |
| --- | --- |
| **Parameters:** | * **$str** (*string*) – Input string * **$n** (*int*) – Number of characters * **$end\_char** (*string*) – End character (usually an ellipsis) |
| **Returns:** | Character-limited string |
| **Return type:** | string |

Truncates a string to the number of *characters* specified. It maintains the integrity of words so the character count may be slightly more or less than what you specify.

Example:

$string **=** "Here is a nice text string consisting of eleven words.";

$string **=** character\_limiter($string, 20);

*// Returns: Here is a nice text string*

The third parameter is an optional suffix added to the string, if undeclared this helper uses an ellipsis.

**Note**

If you need to truncate to an exact number of characters please see the [ellipsize()](https://codeigniter.com/user_guide/helpers/text_helper.html#ellipsize) function below.

**ascii\_to\_entities(*$str*)**

|  |  |
| --- | --- |
| **Parameters:** | * **$str** (*string*) – Input string |
| **Returns:** | A string with ASCII values converted to entities |
| **Return type:** | string |

Converts ASCII values to character entities, including high ASCII and MS Word characters that can cause problems when used in a web page, so that they can be shown consistently regardless of browser settings or stored reliably in a database. There is some dependence on your server’s supported character sets, so it may not be 100% reliable in all cases, but for the most part it should correctly identify characters outside the normal range (like accented characters).

Example:

$string **=** ascii\_to\_entities($string);

**convert\_accented\_characters(*$str*)**

|  |  |
| --- | --- |
| **Parameters:** | * **$str** (*string*) – Input string |
| **Returns:** | A string with accented characters converted |
| **Return type:** | string |

Transliterates high ASCII characters to low ASCII equivalents. Useful when non-English characters need to be used where only standard ASCII characters are safely used, for instance, in URLs.

Example:

$string **=** convert\_accented\_characters($string);

**Note**

This function uses a companion config file *application/config/foreign\_chars.php* to define the to and from array for transliteration.

**word\_censor(*$str*, *$censored*[, *$replacement = ''*])**

|  |  |
| --- | --- |
| **Parameters:** | * **$str** (*string*) – Input string * **$censored** (*array*) – List of bad words to censor * **$replacement** (*string*) – What to replace bad words with |
| **Returns:** | Censored string |
| **Return type:** | string |

Enables you to censor words within a text string. The first parameter will contain the original string. The second will contain an array of words which you disallow. The third (optional) parameter can contain a replacement value for the words. If not specified they are replaced with pound signs: ####.

Example:

$disallowed **=** **array**('darn', 'shucks', 'golly', 'phooey');

$string **=** word\_censor($string, $disallowed, 'Beep!');

**highlight\_code(*$str*)**

|  |  |
| --- | --- |
| **Parameters:** | * **$str** (*string*) – Input string |
| **Returns:** | String with code highlighted via HTML |
| **Return type:** | string |

Colorizes a string of code (PHP, HTML, etc.). Example:

$string **=** highlight\_code($string);

The function uses PHP’s highlight\_string() function, so the colors used are the ones specified in your php.ini file.

**highlight\_phrase(*$str*, *$phrase*[, *$tag\_open = '<mark>'*[, *$tag\_close = '</mark>'*]])**

|  |  |
| --- | --- |
| **Parameters:** | * **$str** (*string*) – Input string * **$phrase** (*string*) – Phrase to highlight * **$tag\_open** (*string*) – Opening tag used for the highlight * **$tag\_close** (*string*) – Closing tag for the highlight |
| **Returns:** | String with a phrase highlighted via HTML |
| **Return type:** | string |

Will highlight a phrase within a text string. The first parameter will contain the original string, the second will contain the phrase you wish to highlight. The third and fourth parameters will contain the opening/closing HTML tags you would like the phrase wrapped in.

Example:

$string **=** "Here is a nice text string about nothing in particular.";

**echo** highlight\_phrase($string, "nice text", '<span style="color:#990000;">', '</span>');

The above code prints:

Here is a **<**span style**=**"color:#990000;"**>**nice text**</**span**>** string about nothing in particular**.**

**Note**

This function used to use the <strong> tag by default. Older browsers might not support the new HTML5 mark tag, so it is recommended that you insert the following CSS code into your stylesheet if you need to support such browsers:

mark {

background**:** *#ff0;*

color**:** *#000;*

};

**word\_wrap(*$str*[, *$charlim = 76*])**

|  |  |
| --- | --- |
| **Parameters:** | * **$str** (*string*) – Input string * **$charlim** (*int*) – Character limit |
| **Returns:** | Word-wrapped string |
| **Return type:** | string |

Wraps text at the specified *character* count while maintaining complete words.

Example:

$string **=** "Here is a simple string of text that will help us demonstrate this function.";

**echo** word\_wrap($string, 25);

*// Would produce:*

*// Here is a simple string*

*// of text that will help us*

*// demonstrate this*

*// function.*

**ellipsize(*$str*, *$max\_length*[, *$position = 1*[, *$ellipsis = '&hellip;'*]])**

|  |  |
| --- | --- |
| **Parameters:** | * **$str** (*string*) – Input string * **$max\_length** (*int*) – String length limit * **$position** (*mixed*) – Position to split at (int or float) * **$ellipsis** (*string*) – What to use as the ellipsis character |
| **Returns:** | Ellipsized string |
| **Return type:** | string |

This function will strip tags from a string, split it at a defined maximum length, and insert an ellipsis.

The first parameter is the string to ellipsize, the second is the number of characters in the final string. The third parameter is where in the string the ellipsis should appear from 0 - 1, left to right. For example. a value of 1 will place the ellipsis at the right of the string, .5 in the middle, and 0 at the left.

An optional forth parameter is the kind of ellipsis. By default, &hellip; will be inserted.

Example:

$str **=** 'this\_string\_is\_entirely\_too\_long\_and\_might\_break\_my\_design.jpg';

**echo** ellipsize($str, 32, **.**5);

Produces:

this\_string\_is\_e**&**hellip;ak\_my\_design**.**jpg