Anurag ⌄

BUILT-IN FILTERS

ADMIN_MODIFY

ADMIN_URLS

DEBUGGER_TAGS

HIGHLIGHTING

I18N

JET_DASHBOARD_TAGS

JET_TAGS

BUILT-IN FILTERS

add
Add the arg to the value.

addslashes
Add slashes before quotes. Useful for escaping strings in CSV, for example. Less useful for escaping JavaScript; use the `escapejs` filter instead.

capfirst
Capitalize the first character of the value.

center
Center the value in a field of a given width.

cut
Remove all values of arg from the given string.

date
Format a date according to the given format.

default
If value is unavailable, use given default.

default_if_none
If value is None, use given default.

dictsort
Given a list of dicts, return that list sorted by the property given in the argument.

dictsortreversed
Given a list of dicts, return that list sorted in reverse order by the property given in the argument.

divisibleby
Return True if the value is divisible by the argument.

escape
Mark the value as a string that should be auto-escaped.

escapejs
Hex encode characters for use in JavaScript strings.

filesizeformat
Format the value like a 'human-readable' file size (i.e. 13 KB, 4.1 MB, 102 bytes, etc.).

first
Return the first item in a list.

floatformat
Display a float to a specified number of decimal places.

If called without an argument, display the floating point number with one decimal place -- but only if there's a decimal place to be displayed:

- num1 = 34.23234
- num2 = 34.00000
- num3 = 34.26000
- {{ num1|floatformat }} displays "34.2"
- {{ num2|floatformat }} displays "34"
- {{ num3|floatformat }} displays "34.3"

If arg is positive, always display exactly arg number of decimal places:

- {{ num1|floatformat:3 }} displays "34.232"
- {{ num2|floatformat:3 }} displays "34.000"
- {{ num3|floatformat:3 }} displays "34.260"

If arg is negative, display arg number of decimal places -- but only if there are places to be displayed:

- {{ num1|floatformat:"-3" }} displays "34.232"
- {{ num2|floatformat:"-3" }} displays "34"
- {{ num3|floatformat:"-3" }} displays "34.260"

If the input float is infinity or NaN, display the string representation of that value.

force_escape
Escape a string's HTML. Return a new string containing the escaped characters (as opposed to "escape", which marks the content for later possible escaping).

get_digit

Given a whole number, return the requested digit of it, where 1 is the right-most digit, 2 is the second-right-most digit, etc. Return the original value for invalid input (if input or argument is not an integer, or if argument is less than 1). Otherwise, output is always an integer.

iriencode

Escape an IRI value for use in a URL.

join

Join a list with a string, like Python's `str.join(list)`.

json_script

Output value JSON-encoded, wrapped in a <script type="application/json"> tag.

last

Return the last item in a list.

length

Return the length of the value - useful for lists.

length_is

Return a boolean of whether the value's length is the argument.

linebreaks

Replace line breaks in plain text with appropriate HTML; a single newline becomes an HTML line break (`<br>`) and a new line followed by a blank line becomes a paragraph break (`</p>`).

linebreaksbr

Convert all newlines in a piece of plain text to HTML line breaks (`<br>`).

linenumbers

Display text with line numbers.

ljust

Left-align the value in a field of a given width.

lower

Convert a string into all lowercase.

make_list

Return the value turned into a list.

For an integer, it's a list of digits. For a string, it's a list of characters.

phone2numeric

Take a phone number and converts it in to its numerical equivalent.

pluralize

Return a plural suffix if the value is not 1, '1', or an object of length 1. By default, use 's' as the suffix:

- If value is 0, vote{{ value|pluralize }} display "votes".
- If value is 1, vote{{ value|pluralize }} display "vote".
- If value is 2, vote{{ value|pluralize }} display "votes".

If an argument is provided, use that string instead:

- If value is 0, class{{ value|pluralize:"es" }} display "classes".
- If value is 1, class{{ value|pluralize:"es" }} display "class".
- If value is 2, class{{ value|pluralize:"es" }} display "classes".

If the provided argument contains a comma, use the text before the comma for the singular case and the text after the comma for the plural case:

- If value is 0, cand{{ value|pluralize:"y,ies" }} display "candies".
- If value is 1, cand{{ value|pluralize:"y,ies" }} display "candy".
- If value is 2, cand{{ value|pluralize:"y,ies" }} display "candies".

pprint

A wrapper around pprint.pprint -- for debugging, really.

random

Return a random item from the list.

rjust

Right-align the value in a field of a given width.

safe

Mark the value as a string that should not be auto-escaped.

safeseq

A "safe" filter for sequences. Mark each element in the sequence, individually, as safe, after converting them to strings. Return a list with the results.

slice

Return a slice of the list using the same syntax as Python's list slicing.

slugify

Convert to ASCII. Convert spaces to hyphens. Remove characters that aren't alphanumerics, underscores, or hyphens. Convert to lowercase. Also strip leading and trailing whitespace.

stringformat

Format the variable according to the arg, a string formatting specifier.

This specifier uses Python string formatting syntax, with the exception that the leading "%" is dropped.

See https://docs.python.org/library/stdtypes.html#printf-style-string-formatting for documentation of Python string formatting.

striptags

Strip all [X]HTML tags.

time

Format a time according to the given format.

timesince

Format a date as the time since that date (i.e. "4 days, 6 hours").

timeuntil

Format a date as the time until that date (i.e. "4 days, 6 hours").

title

Convert a string into titlecase.

truncatechars

Truncate a string after arg number of characters.

truncatechars_html

Truncate HTML after arg number of chars. Preserve newlines in the HTML.

truncatewords

Truncate a string after arg number of words. Remove newlines within the string.

truncatewords_html

Truncate HTML after arg number of words. Preserve newlines in the HTML.

unordered_list

Recursively take a self-nested list and return an HTML unordered list -- WITHOUT opening and closing <ul> tags.

Assume the list is in the proper format. For example, if var contains: ['States', ['Kansas', ['Lawrence', 'Topeka'], 'Illinois']], then {{ var|unordered_list }} returns:

```
<li>States
<ul>
        <li>Kansas
        <ul>
                <li>Lawrence</li>
                <li>Topeka</li>
        </ul>
        </li>
        <li>Illinois</li>
</ul>
</li>
```

upper

Convert a string into all uppercase.

urlencode

Escape a value for use in a URL.

The safe parameter determines the characters which should not be escaped by Python's quote() function. If not provided, use the default safe characters (but an empty string can be provided when *all* characters should be escaped).

urlize

Convert URLs in plain text into clickable links.

urlizetrunc

Convert URLs into clickable links, truncating URLs to the given character limit, and adding 'rel=nofollow' attribute to discourage spamming.

wordcount

Return the number of words.

wordwrap

Wrap words at arg line length.

yesno

Given a string mapping values for true, false, and (optionally) None, return one of those strings according to the value:

VALUE	ARGUMENT	OUTPUTS
True	"yeah,no,maybe"	yeah
False	"yeah,no,maybe"	no
None	"yeah,no,maybe"	maybe
None	"yeah,no"	"no" (converts None to False if no mapping for None is given.

ADMIN_MODIFY

To use these filters, put `{% load admin_modify %}` in your template before using the filter.

cell_count

Return the number of cells used in a tabular inline.

ADMIN_URLS

To use these filters, put `{% load admin_urls %}` in your template before using the filter.

admin_urlname

admin_urlquote

DEBUGGER_TAGS

To use these filters, put `{% load debugger_tags %}` in your template before using the filter.

ipdb

Interactive Python debugger filter.

pdb

Python debugger filter.

wdb

Web debugger filter.

HIGHLIGHTING

To use these filters, put `{% load highlighting %}` in your template before using the filter.

parse_template

I18N

To use these filters, put `{% load i18n %}` in your template before using the filter.

language_bidi

language_name

language_name_local

language_name_translated

JET_DASHBOARD_TAGS

To use these filters, put `{% load jet_dashboard_tags %}` in your template before using the filter.

format_change_message

JET_TAGS

To use these filters, put `{% load jet_tags %}` in your template before using the filter.

jet_append_version

jet_is_checkbox

jet_select2_lookups

L10N

To use these filters, put `{% load l10n %}` in your template before using the filter.

localize

Force a value to be rendered as a localized value, regardless of the value of `settings.USE_L10N`.

unlocalize

Force a value to be rendered as a non-localized value, regardless of the value of `settings.USE_L10N`.

SYNTAX_COLOR

To use these filters, put `{% load syntax_color %}` in your template before using the filter.

colorize

colorize_noclasses

colorize_table

TZ

To use these filters, put {% load tz %} in your template before using the filter.

localtime

Convert a datetime to local time in the active time zone.

This only makes sense within a {% localtime off %} block.

timezone

Convert a datetime to local time in a given time zone.

The argument must be an instance of a tzinfo subclass or a time zone name.

Naive datetimes are assumed to be in local time in the default time zone.

utc

Convert a datetime to UTC.

WIDONT

To use these filters, put {% load widont %} in your template before using the filter.

widont

Add an HTML non-breaking space between the final two words of the string to avoid "widowed" words.

Examples:

```
>>> print(widont('Test   me   out'))
Test   me out

>>> print("'",widont('It works with trailing spaces too  '), "'")
' It works with trailing spaces too   '

>>> print(widont('NoEffect'))
NoEffect
```

widont_html

Add an HTML non-breaking space between the final two words at the end of (and in sentences just outside of) block level tags to avoid "widowed" words.

Examples:

```
>>> print(widont_html('<h2>Here is a simple  example  </h2> <p>Single</p>'))
<h2>Here is a simple example  </h2> <p>Single</p>

>>> print(widont_html('<p>test me<br /> out</p><h2>Ok?</h2>Not in a p<p title="test me">and this<, '))
<p>test me<br /> out</p><h2>Ok?</h2>Not in a p<p title="test me">and this</p>

>>> print(widont_html('leading text  <p>test me out</p>  trailing text'))
leading text  <p>test me out</p>  trailing text
```