

Package ‘humanize’

July 22, 2025

Version 0.2.0

Title Create Values for Human Consumption

Description An almost direct port of the 'python' 'humanize' package <<https://github.com/jmoiron/humanize>>.

This package contains utilities to convert values into human readable forms.

Encoding UTF-8

LazyData true

ByteCompile true

RoxygenNote 6.0.1

Suggests testthat, purrr

Imports lubridate, assertthat, glue

License MIT + file LICENSE

BugReports <https://github.com/newtux/humanize/issues>

URL <https://newtux.github.io/humanize/index.html>,
<https://github.com/newtux/humanize>

NeedsCompilation no

Author Gerry Manóim [aut, cre]

Maintainer Gerry Manóim <gerrymanoim@gmail.com>

Repository CRAN

Date/Publication 2018-04-04 04:16:58 UTC

Contents

count_as_ap	2
count_as_ordinal	2
count_as_word	3
natural_date	3
natural_day	4
natural_size	4

natural_time	5
number_as_comma	5
seconds_to_natural_delta	6

Index	7
--------------	----------

count_as_ap	<i>Convert to AP Number</i>
-------------	-----------------------------

Description

Convert to AP Number

Usage

count_as_ap(value)

Arguments

value A single positive integer

Value

For numbers 1-9, returns the number spelled out. Otherwise, returns the number as a string.

Examples

count_as_ap(3)
count_as_ap(20)

count_as_ordinal	<i>Transform a count to an ordinal string</i>
------------------	---

Description

Transform a count to an ordinal string

Usage

count_as_ordinal(value)

Arguments

value A single positive integer

Value

A string with the ordinal representation of a number

Examples

```
count_as_ordinal(1)
count_as_ordinal(111)
```

count_as_word	<i>Convert Large Counts into Friendly Text</i>
---------------	--

Description

Note - currently limited to .Machine\$integer.max.

Usage

```
count_as_word(value, fmt = "%.1f")
```

Arguments

value	A single positive integer
fmt	Extra number formatting supplied to sprintf

Value

Returns a string with the power of a number replaced by the appropriate word.

Examples

```
count_as_word(100)
count_as_word(1000000)
count_as_word(1200000000)
```

natural_date	<i>Natural Date</i>
--------------	---------------------

Description

Like natural day, but will append a year for dates that are a year or more in the past or future

Usage

```
natural_date(value)
```

Arguments

value	A Date value
-------	--------------

See Also

natural_day

Examples

```
natural_date(Sys.Date())
natural_date(Sys.Date()-10)
```

natural_day	<i>Natural Day</i>
-------------	--------------------

Description

For date values that are tomorrow, today or yesterday compared to present day returns representing string. Otherwise, returns a string formatted according to fmt

Usage

```
natural_day(value, fmt = "%b %d")
```

Arguments

- value A date value
- fmt Optional formatting string for dates not yesterday, today, tomorrow

Value

A nicely formatted date

Examples

```
natural_day(Sys.Date())
natural_day(Sys.Date()-10)
```

natural_size	<i>Convert bytes to a more natural representation</i>
--------------	---

Description

Convert bytes to a more natural representation

Usage

```
natural_size(bytes, suffix_type = "decimal", fmt = "%.1f")
```

Arguments

bytes	Number of bytes
suffix_type	One of 'decimal', 'binary', 'gnu'
fmt	Extra number formatting

Examples

```
natural_size(3000)
```

natural_time	<i>Convert times to natural values relative to now.</i>
--------------	---

Description

Given a datetime or a number of seconds, return a natural representation of that resolution that makes sense. Ago/From now determined by positive or negative values.

Usage

```
natural_time(value, use_months = TRUE)
```

Arguments

value	a datetime or a number of seconds
use_months	Boolean whether we should (imprecisely) use months as a unit

Examples

```
natural_time(Sys.time()-1)  
natural_time(Sys.time()-100)
```

number_as_comma	<i>Convert an number to a string with comma separation</i>
-----------------	--

Description

Just a wrapper around format with defaults for full digits

Usage

```
number_as_comma(value)
```

Arguments

value	A numeric
-------	-----------

Value

A string with comma separation every three digits

Examples

```
number_as_comma(1000)
number_as_comma(10000)
```

```
seconds_to_natural_delta
```

Takes in a number of seconds and computes a "human" delta

Description

Takes in a number of seconds and computes a "human" delta

Usage

```
seconds_to_natural_delta(seconds, use_months = TRUE)
```

Arguments

seconds	A positive number of seconds
use_months	Boolean whether we should (imprecisely) use months as a unit

See Also

natural_time

Index

count_as_ap, [2](#)
count_as_ordinal, [2](#)
count_as_word, [3](#)

natural_date, [3](#)
natural_day, [4](#)
natural_size, [4](#)
natural_time, [5](#)
number_as_comma, [5](#)

seconds_to_natural_delta, [6](#)