

Package ‘ympes’

August 9, 2022

Type Package

Title Collection of Helper Functions

Version 0.1.1

Description Provides a collection of lightweight helper functions (imps) both for interactive use and for inclusion within other packages. These include minimal assertion functions with a focus on informative error messaging for both missing and incorrect function arguments as well as other functions for visualising colour palettes and quoting user input.

License GPL-2

Encoding UTF-8

RoxygenNote 7.2.1

Suggests clipr, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Tim Taylor [aut, cre, cph] (<<https://orcid.org/0000-0002-8587-7113>>)

Maintainer Tim Taylor <tim.taylor@hidden elephants.co.uk>

Repository CRAN

Date/Publication 2022-08-09 12:10:02 UTC

R topics documented:

assertions	2
cc	4
plot_palette	5

Index	6
--------------	----------

assertions

Argument assertions

Description

Assertions for function arguments. Motivated by `vctrs::vec_assert()` but with lower overhead at a cost of less informative error messages. Designed to make it easy to identify the top level calling function whether used within a user facing function or internally.

Usage

```
imp_assert_integer(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_int(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_double(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_dbl(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_numeric(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_num(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_logical(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_lgl(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_character(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_chr(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_data_frame(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_list(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_scalar_integer(
  x,
  arg = deparse(substitute(x)),
  call = sys.call(-1L)
)
imp_assert_scalar_int(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_scalar_double(x, arg = deparse(substitute(x)), call = sys.call(-1L))
imp_assert_scalar_dbl(x, arg = deparse(substitute(x)), call = sys.call(-1L))
```

```
imp_assert_scalar_numeric(  
  x,  
  arg = deparse(substitute(x)),  
  call = sys.call(-1L)  
)  
  
imp_assert_scalar_num(x, arg = deparse(substitute(x)), call = sys.call(-1L))  
  
imp_assert_scalar_logical(  
  x,  
  arg = deparse(substitute(x)),  
  call = sys.call(-1L)  
)  
  
imp_assert_scalar_lgl(x, arg = deparse(substitute(x)), call = sys.call(-1L))  
  
imp_assert_bool(x, arg = deparse(substitute(x)), call = sys.call(-1L))  
  
imp_assert_boolean(x, arg = deparse(substitute(x)), call = sys.call(-1L))  
  
imp_assert_scalar_character(  
  x,  
  arg = deparse(substitute(x)),  
  call = sys.call(-1L)  
)  
  
imp_assert_scalar_chr(x, arg = deparse(substitute(x)), call = sys.call(-1L))  
  
imp_assert_string(x, arg = deparse(substitute(x)), call = sys.call(-1L))
```

Arguments

x	Argument to check.
arg	Name of argument being checked (used in error message).
call	Call to use in error message.

Value

The input argument (invisibly) if the assertion succeeds (error otherwise).

Examples

```
# Use in a user facing function  
fun <- function(i, d, l, chr, b) {  
  imp_assert_scalar_int(i)  
  TRUE  
}  
fun(i=1L)
```

```
try(fun())
try(fun(i="cat"))

# Use in an internal function
internal_fun <- function(a) {
  imp_assert_string(a, arg = deparse(substitute(a)), call = sys.call(-1L))
  TRUE
}
external_fun <- function(b) {
  internal_fun(a=b)
}
external_fun(b="cat")
try(external_fun())
try(external_fun(a = letters))
```

cc

Quote names

Description

`cc()` quotes comma separated names whilst trimming outer whitespace. It is intended for interactive use only.

Usage

```
cc(..., .clip = getOption("imp.clipboard", FALSE))
```

Arguments

<code>...</code>	Unquoted names (separated by commas) that you wish to quote; empty arguments (e.g. third item in one, two, , four) will be returned as "".
<code>.clip</code>	Should the code to generate the constructed character vector be copied to your system clipboard; defaults to FALSE unless the option "imp.clipboard" is set to TRUE.

Value

A character vector of the quoted input.

Note

Copying to clipboard requires the availability of package `clipr`.

Examples

```
cc(dale, audrey, laura, hawk)
```

plot_palette	<i>Plot a colour palette</i>
--------------	------------------------------

Description

plot_palette() plots a palette from a vector of colour values (name or hex).

Usage

```
plot_palette(values, label = TRUE, square = FALSE)
```

Arguments

values	character vector of named or hex colours.
label	boolean. Do you want to label the plot or not? If values is a named vector the names are used for labels, otherwise, the values.
square	boolean. Display palette as square?

Value

The input (invisibly).

Examples

```
plot_palette(c("#5FE756", "red", "black"))  
plot_palette(c("#5FE756", "red", "black"), square=TRUE)
```

Index

assertions, [2](#)

cc, [4](#)

imp_assert_bool (assertions), [2](#)

imp_assert_boolean (assertions), [2](#)

imp_assert_character (assertions), [2](#)

imp_assert_chr (assertions), [2](#)

imp_assert_data_frame (assertions), [2](#)

imp_assert_dbl (assertions), [2](#)

imp_assert_double (assertions), [2](#)

imp_assert_int (assertions), [2](#)

imp_assert_integer (assertions), [2](#)

imp_assert_lgl (assertions), [2](#)

imp_assert_list (assertions), [2](#)

imp_assert_logical (assertions), [2](#)

imp_assert_num (assertions), [2](#)

imp_assert_numeric (assertions), [2](#)

imp_assert_scalar_character
(assertions), [2](#)

imp_assert_scalar_chr (assertions), [2](#)

imp_assert_scalar_dbl (assertions), [2](#)

imp_assert_scalar_double (assertions), [2](#)

imp_assert_scalar_int (assertions), [2](#)

imp_assert_scalar_integer (assertions),
[2](#)

imp_assert_scalar_lgl (assertions), [2](#)

imp_assert_scalar_logical (assertions),
[2](#)

imp_assert_scalar_num (assertions), [2](#)

imp_assert_scalar_numeric (assertions),
[2](#)

imp_assert_string (assertions), [2](#)

plot_palette, [5](#)