

Package ‘felp’

December 6, 2019

Type Package

Title Functional Help for Functions, Objects, and Packages

Version 0.2.2

Author Atsushi Yasumoto [aut, cph, cre]
(<<https://orcid.org/0000-0002-8335-495X>>)

Maintainer Atsushi Yasumoto <atusy.rpkg@gmail.com>

Description Provides pseudo-postfix operators and more to enhance displaying documents. The `?.` pseudo-postfix operator and the `?`` prefix operator displays documents and contents (source or structure) of objects simultaneously to help understanding the objects. The `?p`` pseudo-postfix operator displays package documents, and is shorter than `help(package = foo)`.

URL <https://github.com/atusy/felp>

BugReports <https://github.com/atusy/felp/issues>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports prettycode

Suggests knitr, pkgdown, printr, roxygen2, testthat (>= 2.1.0), covr

VignetteBuilder knitr

Language en-US

NeedsCompilation no

Repository CRAN

Date/Publication 2019-12-06 15:40:02 UTC

R topics documented:

?	2
dummy	2
felp	3

Index**4**

? *Functional help with ? operator*

Description

Displays help and structure of an object, or help of a package. Two syntax sugars are added to enhance `utils::?`. One is `object?`, which works as if `?object`. Another is `package?p` which works as if `help(package = package)`

Usage

```
"?"(e1, e2)
```

Arguments

`e1` A topic of document. Refer to topic argument described in `utils::`?``.
`e2` `.` and `p` have special meanings as documented above. Otherwise, `e2` is same as type argument of `utils::`?``.

Examples

```
# Identical to help(identity); print(identity)
?identity
identity?. # The same

# Identical to help(iris); str(iris)
?iris
iris?. # The same

# Identical to help(package = MASS)
MASS?p
```

dummy *A dummy data for felp*

Description

A dummy data for felp

Usage

```
dummy
```

Format

An object of class `list` of length 2.

felp	<i>Functional help which displays structure of an object in addition to help</i>
------	--

Description

Structure of object is returned by `str()`. For a function, its source is returned instead of `str()`.

Usage

```
felp(topic, package = NULL, ...)
```

Arguments

topic	usually, a name or character string specifying the topic for which help is sought. A character string (enclosed in explicit single or double quotes) is always taken as naming a topic. If the value of <code>topic</code> is a length-one character vector the topic is taken to be the value of the only element. Otherwise <code>topic</code> must be a name or a reserved word (if syntactically valid) or character string. See ‘Details’ for what happens if this is omitted.
package	a name or character vector giving the packages to look into for documentation, or NULL. By default, all packages whose namespaces are loaded are used. To avoid a name being deparsed use e.g. <code>(pkg_ref)</code> (see the examples).
...	Arguments passed on to <code>utils::help</code>

lib.loc a character vector of directory names of \mathbb{R} libraries, or NULL. The default value of NULL corresponds to all libraries currently known. If the default is used, the loaded packages are searched before the libraries. This is not used for HTML help (see ‘Details’).

verbose logical; if TRUE, the file name is reported.

try.all.packages logical; see Note.

help_type character string: the type of help required. Possible values are "text", "html" and "pdf". Case is ignored, and partial matching is allowed.

Examples

```
# Identical to help(identity); print(identity)
felp(identity)

# Identical to help(iris); str(iris)
felp(iris)

# Identical to help(package = MASS)
felp(package = MASS)
```

Index

*Topic **datasets**

dummy, [2](#)

?, [2](#)

dummy, [2](#)

fe1p, [3](#)

name, [3](#)

reserved, [3](#)