# Package ‘pragma’

February 20, 2015

**Type** Package

**Title** Provides a pragma / directive / keyword syntax for R.

**Version** 0.1.3

**Date** 2011-08-21

**Author** Christopher Brown

**Maintainer** Christopher Brown <chris.brown@decisionpatterns.com>

**Depends** R (>= 2.13.0), utils, methods

**Description** pragma allows for the use of pragma (also sometimes called directives or keywords. These allow assigning arbitrary functionality to a word without requiring the standard function call syntax i.e. with parens.

**License** GPL (>= 2)

**URL** http://www.decisionpatterns.com

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2012-10-29 08:59:30

**NeedsCompilation** no

## R topics documented:

pragma-package *Provides a pragma / directive / keyword syntax for R*

### Description

R is a functional language. All action taken are made through function calls. These calls are denoted by the use of parens – even when no arguments are necessary. Very often these parameterless functions are used to set some behaviour of the program, script or package. These abbreviated, argumentless functions are called pragmas ( or directives or keywords ).

This pragma syntax is non-essential. The same functionality can be used through the use of a standard function calls. What `pragma` mostly offers is a way for package authors to provide their users with a distinctive and clean syntax for adjusting package options.

### Details

| | |
|---|---|
| Package: | pragma |
| Type: | Package |
| Version: | 0.1.3 |
| Date: | 2011-08-21 |
| License: | Decision Patterns |
| LazyLoad: | yes |

### Author(s)

Christopher Brown

Maintainer: Christopher Brown <chris.brown@decisionpatterns.com>

### References

http://en.wikipedia.org/wiki/Pragma

http://en.wikipedia.org/wiki/Directive_(programming)

### Examples

```
# See \code{\link{setPragma}}
```

## setPragma *Function that returns a pragma*

### Description

setPragma allows for the use of pragma (also sometimes called directives or keywords. These allow assigning arbitrary functionality without having to use R's standard function call syntax i.e. with parens, resulting in cleaner code.

### Usage

```
setPragma(name,fun)
```

### Arguments

| | |
|---|---|
| name | The name given to the pragma/directive. |
| fun | A function providing the functionality to be executed. The function should have no arguments. |

### Details

Pragmas are useful for implementing assigning a small bit code that requires no parameters. Pragmas are implemented as object in their own unique class. The default (show) method is set to the user-supplied function. When, during the course of the program flow, the pragma is used, the arbitrary functionality is executed.

setPragma departs significantly from standard R language conventions in which every action appears as a function and side-effects are assiduously avoided.

### Value

An object that when encountered in the flow of the program will execute the supplied function without the need for parenthesis. The object should be assigned to a KEYWORD that can then be exported from a package.

### Warning

setPragma departs significantly from standard R language conventions in which every action appears as a function and side-effects are assiduously avoided.

### Author(s)

Christopher Brown

### References

http://en.wikipedia.org/wiki/Pragma

http://en.wikipedia.org/wiki/Directive_(programming)

## See Also

Nothing to see here, move along.

## Examples

```

   ## Not run: 
     setPragma( "KEYWORD", function() { cat( "Do Something Special.\n" ) } )
     KEYWORD

## End(Not run)
```

# Index