|  |  |
| --- | --- |
| ifelse {base} | R Documentation |

**Conditional Element Selection**

**Description**

ifelse returns a value with the same shape as test which is filled with elements selected from either yes or no depending on whether the element of test is TRUE or FALSE.

**Usage**

ifelse(test, yes, no)

**Arguments**

|  |  |
| --- | --- |
| test | an object which can be coerced to logical mode. |
| yes | return values for true elements of test. |
| no | return values for false elements of test. |

**Details**

If yes or no are too short, their elements are recycled. yes will be evaluated if and only if any element of test is true, and analogously for no.

Missing values in test give missing values in the result.

**Value**

A vector of the same length and attributes (including dimensions and "class") as test and data values from the values of yes or no. The mode of the answer will be coerced from logical to accommodate first any values taken from yes and then any values taken from no.

**Warning**

The mode of the result may depend on the value of test (see the examples), and the class attribute (see [oldClass](http://127.0.0.1:14695/library/base/help/oldClass)) of the result is taken from test and may be inappropriate for the values selected from yes and no.

Sometimes it is better to use a construction such as

(tmp <- yes; tmp[!test] <- no[!test]; tmp)

, possibly extended to handle missing values in test.

Further note that if(test) yes else no is much more efficient and often much preferable to ifelse(test, yes, no) whenever test is a simple true/false result, i.e., when length(test) == 1.

**References**

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

**See Also**

[if](http://127.0.0.1:14695/library/base/help/if).

**Examples**

x <- c(6:-4)

sqrt(x) #- gives warning

sqrt(ifelse(x >= 0, x, NA)) # no warning

## Note: the following also gives the warning !

ifelse(x >= 0, sqrt(x), NA)

## ifelse() strips attributes

## This is important when working with Dates and factors

x <- seq(as.Date("2000-02-29"), as.Date("2004-10-04"), by = "1 month")

## has many "yyyy-mm-29", but a few "yyyy-03-01" in the non-leap years

y <- ifelse(as.POSIXlt(x)$mday == 29, x, NA)

head(y) # not what you expected ... ==> need restore the class attribute:

class(y) <- class(x)

y

## ==> Again a case where it is better \*not\* to use ifelse(), but

## both more efficient and clear:

y2 <- x

y2[as.POSIXlt(x)$mday != 29] <- NA

stopifnot(identical(y2, y))

## example of different return modes:

yes <- 1:3

no <- pi^(0:3)

typeof(ifelse(NA, yes, no)) # logical

typeof(ifelse(TRUE, yes, no)) # integer

typeof(ifelse(FALSE, yes, no)) # double