|  |  |
| --- | --- |
| rep {base} | R Documentation |

**Replicate Elements of Vectors and Lists**

**Description**

rep replicates the values in x. It is a generic function, and the (internal) default method is described here.

rep.int and rep\_len are faster simplified versions for two common cases. They are not generic.

**Usage**

rep(x, ...)

rep.int(x, times)

rep\_len(x, length.out)

**Arguments**

|  |  |
| --- | --- |
| x | a vector (of any mode including a list) or a factor or (for rep only) a POSIXct or POSIXlt or Date object; or an S4 object containing such an object. |
| ... | further arguments to be passed to or from other methods. For the internal default method these can include:  times  A integer vector giving the (non-negative) number of times to repeat each element if of length length(x), or to repeat the whole vector if of length 1. Negative or NA values are an error.  length.out  non-negative integer. The desired length of the output vector. Other inputs will be coerced to an integer vector and the first element taken. Ignored if NA or invalid.  each  non-negative integer. Each element of x is repeated each times. Other inputs will be coerced to an integer vector and the first element taken. Treated as 1 if NA or invalid. |
| times | see .... |
| length.out | non-negative integer: the desired length of the output vector. |

**Details**

The default behaviour is as if the call was

rep(x, times = 1, length.out = NA, each = 1)

. Normally just one of the additional arguments is specified, but if each is specified with either of the other two, its replication is performed first, and then that implied by times or length.out.

If times consists of a single integer, the result consists of the whole input repeated this many times. If times is a vector of the same length as x (after replication by each), the result consists of x[1] repeated times[1] times, x[2] repeated times[2] times and so on.

length.out may be given in place of times, in which case x is repeated as many times as is necessary to create a vector of this length. If both are given, length.out takes priority and times is ignored.

Non-integer values of times will be truncated towards zero. If times is a computed quantity it is prudent to add a small fuzz or use [round](http://127.0.0.1:14695/library/base/help/round). And analogously for each.

If x has length zero and length.out is supplied and is positive, the values are filled in using the extraction rules, that is by an NA of the appropriate class for an atomic vector (0 for raw vectors) and NULL for a list.

**Value**

An object of the same type as x.

rep.int and rep\_len return no attributes (except the class if returning a factor).

The default method of rep gives the result names (which will almost always contain duplicates) if x had names, but retains no other attributes.

**Note**

Function rep.int is a simple case which was provided as a separate function partly for S compatibility and partly for speed (especially when names can be dropped). The performance of rep has been improved since, but rep.int is still at least twice as fast when x has names.

The name rep.int long precedes making rep generic.

Function rep is a primitive, but (partial) matching of argument names is performed as for normal functions.

For historical reasons rep (only) works on NULL: the result is always NULL even when length.out is positive.

Although it has never been documented, these functions have always worked on [expression](http://127.0.0.1:14695/library/base/help/expression) vectors.

**References**

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

**See Also**

[seq](http://127.0.0.1:14695/library/base/help/seq), [sequence](http://127.0.0.1:14695/library/base/help/sequence), [replicate](http://127.0.0.1:14695/library/base/help/replicate).

**Examples**

rep(1:4, 2)

rep(1:4, each = 2) # not the same.

rep(1:4, c(2,2,2,2)) # same as second.

rep(1:4, c(2,1,2,1))

rep(1:4, each = 2, len = 4) # first 4 only.

rep(1:4, each = 2, len = 10) # 8 integers plus two recycled 1's.

rep(1:4, each = 2, times = 3) # length 24, 3 complete replications

rep(1, 40\*(1-.8)) # length 7 on most platforms

rep(1, 40\*(1-.8)+1e-7) # better

## replicate a list

fred <- list(happy = 1:10, name = "squash")

rep(fred, 5)

# date-time objects

x <- .leap.seconds[1:3]

rep(x, 2)

rep(as.POSIXlt(x), rep(2, 3))

## named factor

x <- factor(LETTERS[1:4]); names(x) <- letters[1:4]

x

rep(x, 2)

rep(x, each = 2)

rep.int(x, 2) # no names

rep\_len(x, 10)