

cat(base)

R Documentation

Concatenate and Print

Description

Outputs the objects, concatenating the representations. `cat` performs much less conversion than [print](#).

Usage

```
cat(... , file = "", sep = " ", fill = FALSE, labels = NULL,
     append = FALSE)
```

Arguments

- `...` `R` objects (see ‘Details’ for the types of objects allowed).
- `file` A connection, or a character string naming the file to print to. If "" (the default), `cat` prints to the standard output connection, the console unless redirected by [sink](#).
- `sep` a character vector of strings to append after each element.
- `fill` a logical or (positive) numeric controlling how the output is broken into successive lines. If `FALSE` (default), only newlines created explicitly by "`\n`" are printed. Otherwise, the output is broken into lines with print width equal to the option `width` if `fill` is `TRUE`, or the value of `fill` if this is numeric. Non-positive `fill` values are ignored, with a warning.
- `labels` character vector of labels for the lines printed. Ignored if `fill` is `FALSE`.
- `append` logical. Only used if the argument `file` is the name of file (and not a connection or "`|cmd`"). If `TRUE` output will be appended to `file`; otherwise, it will overwrite the contents of `file`.

Details

`cat` is useful for producing output in user-defined functions. It converts its arguments to character vectors, concatenates them to a single character vector, appends the given `sep=` string(s) to each element and then outputs them.

No linefeeds are output unless explicitly requested by "`\n`" or if generated by filling (if argument `fill` is `TRUE` or numeric.)

Currently only atomic vectors (and so not lists) and [names](#) are handled. Character strings are output ‘as is’ (unlike [print.default](#) which escapes non-printable characters and backslash — use [encodeString](#) if you want to output encoded strings using `cat`). (Character strings with embedded nuls are truncated at the first nul.) Other types of `R` object should be converted (e.g. by [as.character](#) or [format](#)) before being passed to `cat`.

`cat` converts numeric/complex elements in the same way as `print` (and not in the same way as [as.character](#) which is used by the S equivalent), so [options](#) "digits" and "scipen" are relevant. However, it uses the minimum field width necessary for each element, rather than the same field width for all elements.

Value

None (invisible `NULL`).

Note

Despite its name and earlier documentation, `sep` is a vector of terminators rather than separators, being output after every vector element (including the last). Entries are recycled as needed.

References

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

See Also

[print](#), [format](#), and [paste](#) which concatenates into a string.

Examples

```
iter <- stats::rpois(1, lambda=10)
## print an informative message
cat("iteration = ", iter <- iter + 1, "\n")

## 'fill' and label lines:
cat(paste(letters, 100* 1:26), fill = TRUE,
    labels = paste("{", 1:10, "}: ", sep=""))
```

[Package *base* version 2.7.1 [Index](#)]