

NAME

`finite`, `finitef`, `finitel`, `isinf`, `isinf`, `isinfl`, `isnan`, `isnanf`, `isnanl` - BSD floating-point classification functions

SYNOPSIS

```
#include <math.h>

int finite(double x);
int finitef(float x);
int finitel(long double x);

int isinf(double x);
int isinf(float x);
int isinfl(long double x);

int isnan(double x);
int isnanf(float x);
int isnanl(long double x);
```

Feature Test Macro Requirements for glibc (see [feature_test_macros\(7\)](#)):

```
finite(), finitef(), finitel():
    _BSD_SOURCE || _SVID_SOURCE
isinf():
    _BSD_SOURCE || _SVID_SOURCE || _XOPEN_SOURCE >= 600 || _ISOC99_SOURCE;
    or cc -std=c99
isinf(), isinfl():
    _BSD_SOURCE || _SVID_SOURCE
isnan():
    _BSD_SOURCE || _SVID_SOURCE || _XOPEN_SOURCE || _ISOC99_SOURCE;
    or cc -std=c99
isnanf(), isnanl():
    _BSD_SOURCE || _SVID_SOURCE || _XOPEN_SOURCE >= 600
```

DESCRIPTION

The **finite()**, **finitef()**, and **finitel()** functions return a nonzero value if *x* is neither infinite nor a not-a-number (NaN) value, and 0 otherwise.

The **isnan()**, **isnanf()**, and **isnanl()** functions return a nonzero value if *x* is a NaN value, and 0 otherwise.

The **isinf()**, **isinf()**, and **isinfl()** functions return 1 if *x* is positive infinity, -1 if *x* is negative infinity, and 0 otherwise.

ATTRIBUTES

Multithreading (see [pthreads\(7\)](#))

The **finite()**, **finitef()**, **finitel()**, **isinf()**, **isinf()**, **isinfl()**, **isnan()**, **isnanf()**, and **isnanl()** functions are thread-safe.

NOTES

Note that these functions are obsolete. C99 defines macros **isfinite()**, **isinf()**, and **isnan()** (for all types) replacing them. Further note that the C99 **isinf()** has weaker guarantees on the return value. See [fpclassify\(3\)](#).

SEE ALSO

[fpclassify\(3\)](#)

COLOPHON

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <http://www.kernel.org/doc/man-pages/>.