

NAME

log, logf, logl - natural logarithmic function

SYNOPSIS

```
#include <math.h>
```

```
double log(double x);
```

```
float logf(float x);
```

```
long double logl(long double x);
```

Link with *-lm*.

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

```
logf(), logl():
```

```
_BSD_SOURCE || _SVID_SOURCE || _XOPEN_SOURCE >= 600 || _ISOC99_SOURCE ||
```

```
_POSIX_C_SOURCE >= 200112L;
```

```
or cc -std=c99
```

DESCRIPTION

The `log()` function returns the natural logarithm of *x*.

RETURN VALUE

On success, these functions return the natural logarithm of *x*.

If *x* is a NaN, a NaN is returned.

If *x* is 1, the result is +0.

If *x* is positive infinity, positive infinity is returned.

If *x* is zero, then a pole error occurs, and the functions return **-HUGE_VAL**, **-HUGE_VALF**, or **-HUGE_VALL**, respectively.

If *x* is negative (including negative infinity), then a domain error occurs, and a NaN (not a number) is returned.

ERRORS

See [math_error\(7\)](#) for information on how to determine whether an error has occurred when calling these functions.

The following errors can occur:

Domain error: *x* is negative

errno is set to **EDOM**. An invalid floating-point exception (**FE_INVALID**) is raised.

Pole error: *x* is zero

errno is set to **ERANGE**. A divide-by-zero floating-point exception (**FE_DIVBYZERO**) is raised.

CONFORMING TO

C99, POSIX.1-2001. The variant returning *double* also conforms to SVr4, 4.3BSD, C89.

BUGS

In glibc 2.5 and earlier, taking the `log()` of a NaN produces a bogus invalid floating-point (**FE_INVALID**) exception.

SEE ALSO

[cbrt\(3\)](#), [clog\(3\)](#), [log10\(3\)](#), [log1p\(3\)](#), [log2\(3\)](#), [sqrt\(3\)](#)

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/>.