

**NAME**

log, logf, logl – natural logarithmic function

**LIBRARY**

Math library (*libm*, *-lm*)

**SYNOPSIS**

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

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

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

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

Feature Test Macro Requirements for glibc (see **feature\_test\_macros(7)**):

**logf(), logl():**

```
_ISOC99_SOURCE || _POSIX_C_SOURCE >= 200112L
```

```
    /* Since glibc 2.19: */ _DEFAULT_SOURCE
```

```
    /* glibc <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
```

**DESCRIPTION**

These functions return 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.

**ATTRIBUTES**

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
<b>log(), logf(), logl()</b>	Thread safety	MT-Safe

**STANDARDS**

C99, POSIX.1-2001, POSIX.1-2008.

The variant returning *double* also conforms to SVr4, 4.3BSD.

**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)**