



Linux Ubuntu 26.04 Manual Pages on command 'expf.3'

\$ man expf.3

exp(3) Library Functions Manual exp(3)

NAME

exp, expf, expl - base-e exponential function

LIBRARY

Math library (libm, -lm)

SYNOPSIS

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

```
double exp(double x);
```

```
float expf(float x);
```

```
long double expl(long double x);
```

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

expf(), expl():

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

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

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

DESCRIPTION

These functions return the value of e (the base of natural logarithms) raised to the power of x.

RETURN VALUE

On success, these functions return the exponential value of x.

If x is a NaN, a NaN is returned.

If x is positive infinity, positive infinity is returned.

If x is negative infinity, +0 is returned.

If the result underflows, a range error occurs, and zero is returned.

If the result overflows, a range error occurs, and the functions return +HUGE_VAL, +HUGE_VALF, or +HUGE_VALL, respectively.

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:

Range error, overflow

errno is set to ERANGE. An overflow floating-point exception (FE_OVERFLOW) is raised.

Range error, underflow

errno is set to ERANGE. An underflow floating-point exception (FE_UNDERFLOW) is raised.

ATTRIBUTES

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

[illegible]

?	Interface	?	Attribute	?	Value	?
---	-----------	---	-----------	---	-------	---

[illegible]

? exp(), expf(), expl()	? Thread safety ? MT-Safe ?
-------------------------	-----------------------------

[illegible]

STANDARDS

C11, POSIX.1-2008.

HISTORY

C99, POSIX.1-2001.

The variant returning double also conforms to SVr4, 4.3BSD, C89.

SEE ALSO

cbrt(3), cexp(3), exp10(3), exp2(3), expm1(3), sqrt(3)

Linux man-pages 6.17

2026-02-08

 $\exp(3)$