

Class **Float** < Numeric

Float objects represent real numbers using the native architecture’s double-precision floating-point representation.

Class constants

DIG	Precision of Float (in decimal digits)
EPSILON	The smallest Float such that $1.0 + \text{EPSILON} \neq 1.0$
MANT_DIG	The number of mantissa digits (base RADIX)
MAX	The largest Float
MAX_10_EXP	The maximum integer x such that 10^x is a finite Float
MAX_EXP	The maximum integer x such that $\text{FLT_RADIX}^{(x-1)}$ is a finite Float
MIN	The smallest Float
MIN_10_EXP	The minimum integer x such that 10^x is a finite Float
MIN_EXP	The minimum integer x such that $\text{FLT_RADIX}^{(x-1)}$ is a finite Float
RADIX	The radix of floating-point representations
ROUNDS	The rounding mode for floating-point operations; possible values include <ul style="list-style-type: none"> -1 if the mode is indeterminate 0 if rounding is toward zero 1 if rounding is to nearest representable value 2 if rounding is toward $+\infty$ 3 if rounding is toward $-\infty$

Instance methods

Arithmetic operations

Performs various arithmetic operations on *flt*.

<i>flt</i>	+	<i>numeric</i>	Addition
<i>flt</i>	-	<i>numeric</i>	Subtraction
<i>flt</i>	*	<i>numeric</i>	Multiplication
<i>flt</i>	/	<i>numeric</i>	Division
<i>flt</i>	%	<i>numeric</i>	Modulo
<i>flt</i>	**	<i>numeric</i>	Exponentiation
<i>flt</i>	-@		Unary minus

Comparisons

Compares *flt* to other numbers.

<, <=, ==, >=, >.

<=> *flt* <=> *numeric* → -1, 0, +1

Returns -1, 0, or +1 depending on whether *flt* is less than, equal to, or greater than *numeric*.

Float

== *flt == obj* → true or false

Returns true only if *obj* has the same value as *flt*. Contrast this with `Float#eql?`, which requires *obj* to be a Float.

```
1.0 == 1.0      # => true
(1.0).eql?(1.0) # => true
1.0 == 1       # => true
(1.0).eql?(1)  # => false
```

abs *flt.abs* → numeric

Returns the absolute value of *flt*.

```
(-34.56).abs # => 34.56
-34.56.abs   # => 34.56
```

ceil *flt.ceil* → int

Returns the smallest Integer greater than or equal to *flt*.

```
1.2.ceil # => 2
2.0.ceil # => 2
(-1.2).ceil # => -1
(-2.0).ceil # => -2
```

divmod *flt.divmod(numeric)* → array

See `Numeric#divmod` on page 617.

eql? *flt.eql?(obj)* → true or false

Returns true only if *obj* is a Float with the same value as *flt*. Contrast this with `Float#==`, which performs type conversions.

```
1.0.eql?(1) # => false
1.0 == 1     # => true
```

fdiv *flt.fdiv(number)* → float

1.9 Returns the floating-point result of dividing *flt* by *number*. Alias for `Float#quo`.

```
63.0.fdiv(9)      # => 7.0
1234.56.fdiv(3.45) # => 357.84347826087
```

finite? *flt.finite?* → true or false

Returns true if *flt* is a valid IEEE floating-point number (it is not infinite, and `nan?` is false).

```
(42.0).finite? # => true
(1.0/0.0).finite? # => false
```

floor*flt.floor* → *int*Returns the largest integer less than or equal to *flt*.

```
1.2.floor      # =>  1
2.0.floor      # =>  2
(-1.2).floor   # => -2
(-2.0).floor   # => -2
```

infinite?*flt.infinite?* → *nil, -1, +1*Returns *nil*, *-1*, or *+1* depending on whether *flt* is finite, $-\infty$, or $+\infty$.

```
(0.0).infinite? # =>  nil
(-1.0/0.0).infinite? # => -1
(+1.0/0.0).infinite? # =>  1
```

magnitude*flt.magnitude* → *float***1.9**Returns the magnitude of *flt* (the distance of *flt* from the origin of the number line). Synonym for *Float#abs*. See also *Complex#magnitude*.**modulo***flt.modulo(numeric)* → *numeric*Synonym for *Float#%*.

```
6543.21.modulo(137)      # =>  104.21
6543.21.modulo(137.24)  # =>  92.9299999999996
```

nan?*flt.nan?* → *true or false*Returns true if *flt* is an invalid IEEE floating-point number.

```
(-1.0).nan?      # =>  false
(0.0/0.0).nan?   # =>  true
```

quo*flt.quo(number)* → *float***1.9**Returns the floating-point result of dividing *flt* by *number*.

```
63.0.quo(9)      # =>  7.0
1234.56.quo(3.45) # =>  357.84347826087
```

round*flt.round(digits=0)* → *numeric***1.9**Rounds *flt* to the nearest integer if the parameter is omitted or zero or rounds to the given number of digits.

```
1.5.round      # =>  2
(-1.5).round   # => -2
3.14159.round  # =>  3
3.14159.round(4) # =>  3.1416
3.14159.round(2) # =>  3.14
```

to_f*flt.to_f* → *flt*Returns *flt*.

to_i *flt.to_i* → *int*

Returns *flt* truncated to an Integer.

```
1.5.to_i      # =>  1
(-1.5).to_i  # => -1
```

to_int *flt.to_int* → *int*

Synonym for Float#to_i.

to_r *flt.to_r* → *number*

1.9 / Converts *flt* to a rational number.

```
1.5.to_r      # =>  3/2
(1.0/3).to_r  # => 6004799503160661/18014398509481984
```

to_s *flt.to_s* → *string*

Returns a string containing a representation of *flt*. As well as a fixed or exponential form of the number, the call may return NaN, Infinity, and -Infinity.

truncate *flt.truncate* → *int*

Synonym for Float#to_i.

zero? *flt.zero?* → true or false

Returns true if *flt* is 0.0.