

Funcons-beta: Floats *

The PLaNCompS Project

Floats.cbs | PLAIN | PRETTY

OUTLINE

Floats

- Conversions
- Comparison
- Arithmetic
- Rounding
- Miscellaneous

*Suggestions for improvement: plancomps@gmail.com.
Reports of issues: <https://github.com/plancomps/CBS-beta/issues>.

Floats

[<i>Datatype</i>	float-formats
<i>Funcon</i>	binary32
<i>Funcon</i>	binary64
<i>Funcon</i>	binary128
<i>Funcon</i>	decimal64
<i>Funcon</i>	decimal128
<i>Type</i>	floats
<i>Funcon</i>	float
<i>Funcon</i>	quiet-not-a-number
<i>Alias</i>	qNaN
<i>Funcon</i>	signaling-not-a-number
<i>Alias</i>	sNaN
<i>Funcon</i>	positive-infinity
<i>Alias</i>	pos-inf
<i>Funcon</i>	negative-infinity
<i>Alias</i>	neg-inf
<i>Funcon</i>	float-convert
<i>Funcon</i>	decimal-float
<i>Funcon</i>	float-equal
<i>Funcon</i>	float-is-less
<i>Funcon</i>	float-is-less-or-equal
<i>Funcon</i>	float-is-greater
<i>Funcon</i>	float-is-greater-or-equal
<i>Funcon</i>	float-negate
<i>Funcon</i>	float-absolute-value
<i>Funcon</i>	float-add
<i>Funcon</i>	float-subtract
<i>Funcon</i>	float-multiply
<i>Funcon</i>	float-multiply-add
<i>Funcon</i>	float-divide
<i>Funcon</i>	float-remainder
<i>Funcon</i>	float-sqrt
<i>Funcon</i>	float-integer-power
<i>Funcon</i>	float-float-power
<i>Funcon</i>	float-round-ties-to-even
<i>Funcon</i>	float-round-ties-to-infinity
<i>Funcon</i>	float-floor
<i>Funcon</i>	float-ceiling
<i>Funcon</i>	float-truncate
<i>Funcon</i>	float-pi
<i>Funcon</i>	float-e
<i>Funcon</i>	float-log
<i>Funcon</i>	float-log10
<i>Funcon</i>	float-exp
<i>Funcon</i>	float-sin
<i>Funcon</i>	float-cos
<i>Funcon</i>	float-tan
<i>Funcon</i>	float-asin
<i>Funcon</i>	float-acos
<i>Funcon</i>	float-atan
<i>Funcon</i>	float-sinh

Floating-point numbers according to the IEEE 754 Standard (2008).

See:

- <http://doi.org/10.1109/IEEESTD.2008.4610935>
- https://en.wikipedia.org/wiki/IEEE_754

Datatype `float-formats ::= binary32 | binary64 | binary128 | decimal64 | decimal128`

Built-in Type `floats(_ : float-formats)`

Note that for distinct formats FF_1 , FF_2 , the types `floats(FF_1)` and `floats(FF_2)` are not necessarily disjoint.

Built-in Funcon `float(
 FF : float-formats,
 _ : bounded-integers(0, 1), _ : natural-numbers, _ : integers)
 : \Rightarrow floats(FF)`

Each finite number is described by three integers: * s = a sign (zero or one), * c = a significand (or 'coefficient'), * q = an exponent. The numerical value of a finite number is $(-1)^s * c * b^q$ where b is the base (2 or 10), also called radix.

The possible finite values that can be represented in a format are determined by the base b , the number of digits in the significand (precision p), and the exponent parameter $emax$: * c must be an integer in the range zero through $(b^p)-1$ (e.g., if $b=10$ and $p=7$ then c is 0 through 9999999); * q must be an integer such that $1-emax \leq q+p-1 \leq emax$ (e.g., if $p=7$ and $emax=96$ then q is -101 through 90).

Note that `float(FF , S , C , Q)` is not a 1-1 operation.

Built-in Funcon `quiet-not-a-number(FF : float-formats) : floats(FF)`
Alias `qNaN = quiet-not-a-number`

Built-in Funcon `signaling-not-a-number(FF : float-formats) : floats(FF)`
Alias `sNaN = signaling-not-a-number`

Built-in Funcon `positive-infinity(FF : float-formats) : floats(FF)`
Alias `pos-inf = positive-infinity`

Built-in Funcon `negative-infinity(FF : float-formats) : floats(FF)`
Alias `neg-inf = negative-infinity`

Conversions

Built-in Funcon `float-convert(
 FF_1 : float-formats,
 FF_2 : float-formats, F : floats(FF_1)
 : \Rightarrow floats(FF_2)`

Built-in Funcon `decimal-float(
 FF : float-formats,
 _ : strings, _ : strings, _ : strings)
 : \Rightarrow floats(FF)`

`decimal-float(F, "M", "N", "E")` is an approximation in `floats(FF)` to the value of 'M.N' times 10 to the power 'E', where "M.N" is decimal notation (optionally-signed) for a fixed-point number and "E" is decimal notation (optionally signed) for an integer. When any argument string is invalid, the result is `quiet-not-a-number(F)`.

Comparison

Built-in Funcon `float-equal(`
 `FF : float-formats,`
 `_ : floats(FF), _ : floats(FF)`
 `: ⇒ booleans`

Built-in Funcon `float-is-less(`
 `FF : float-formats,`
 `_ : floats(FF), _ : floats(FF)`
 `: ⇒ booleans`

Built-in Funcon `float-is-less-or-equal(`
 `FF : float-formats,`
 `_ : floats(FF), _ : floats(FF)`
 `: ⇒ booleans`

Built-in Funcon `float-is-greater(`
 `FF : float-formats,`
 `_ : floats(FF), _ : floats(FF)`
 `: ⇒ booleans`

Built-in Funcon `float-is-greater-or-equal(`
 `FF : float-formats,`
 `_ : floats(FF), _ : floats(FF)`
 `: ⇒ booleans`

Arithmetic

Built-in Funcon `float-negate(FF : float-formats, _ : floats(FF)) : ⇒ floats(FF)`

Built-in Funcon `float-absolute-value(FF : float-formats, _ : floats(FF)) : ⇒ floats(FF)`

Built-in Funcon `float-add(FF : float-formats, _ : floats(FF), _ : floats(FF)) : ⇒ floats(FF)`

Built-in Funcon `float-subtract(FF : float-formats, _ : floats(FF), _ : floats(FF)) : ⇒ floats(FF)`

Built-in Funcon `float-multiply(FF : float-formats, _ : floats(FF), _ : floats(FF)) : ⇒ floats(FF)`

Built-in Funcon `float-multiply-add(`
 `FF : float-formats,`
 `_ : floats(FF), _ : floats(FF), _ : floats(FF)`
 `: ⇒ floats(FF)`

Built-in Funcon `float-divide(FF : float-formats, _ : floats(FF), _ : floats(FF)) : ⇒ floats(FF)`

Built-in Funcon float-remainder($FF : \text{float-formats}$, $_ : \text{floats}(FF)$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-sqrt($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-integer-power($FF : \text{float-formats}$, $_ : \text{floats}(FF)$, $_ : \text{integers}$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-float-power($FF : \text{float-formats}$, $_ : \text{floats}(FF)$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Rounding

Built-in Funcon float-round-ties-to-even($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{integers}$

Built-in Funcon float-round-ties-to-infinity($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{integers}$

Built-in Funcon float-floor($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{integers}$

Built-in Funcon float-ceiling($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{integers}$

Built-in Funcon float-truncate($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{integers}$

Miscellaneous

Built-in Funcon float-pi($FF : \text{float-formats}$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-e($FF : \text{float-formats}$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-log($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-log10($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-exp($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-sin($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-cos($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-tan($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-asin($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-acos($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-atan($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-sinh($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-cosh($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-tanh($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-asinh($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-acosh($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-atanh($FF : \text{float-formats}$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$

Built-in Funcon float-atan2($FF : \text{float-formats}$, $_ : \text{floats}(FF)$, $_ : \text{floats}(FF)$) : $\Rightarrow \text{floats}(FF)$