

1 Language Details

This appendix lists various language details.

1.1 Arithmetic operators on basic types

1 Language Details

Operator	leftOp	rightOp	Expression	Result	Description
leftOp + rightOp	ints	ints	5 + 2	7	Addition
	floats	floats	5.0 + 2.0	7.0	
	chars	chars	'a' + 'b'	'\195'	Addition of codes
	strings	strings	"ab" + "cd"	"abcd"	Concatenation
leftOp - rightOp	ints	ints	5 - 2	3	Subtraction
	floats	floats	5.0 - 2.0	3.0	
leftOp * rightOp	ints	ints	5 * 2	10	Multiplication
	floats	floats	5.0 * 2.0	10.0	
leftOp / rightOp	ints	ints	5 / 2	2	Integer division
	floats	floats	5.0 / 2.0	2.5	Division
leftOp % rightOp	ints	ints	5 % 2	1	Remainder
	floats	floats	5.0 % 2.0	1.0	
leftOp ** rightOp	floats	floats	5.0 ** 2.0	25.0	Exponentiation
leftOp && rightOp	bool	bool	true && false	false	boolean and
leftOp rightOp	bool	bool	true false	false	boolean or
leftOp &&& rightOp	ints	ints	0b1010 &&& 0b1100	0b1000	bitwise boolean and
leftOp rightOp	ints	ints	0b1010 0b1100	0b1110	bitwise boolean or
leftOp ^^^ rightOp	ints	ints	0b1010 ^^^ 0b1101	0b0111	bitwise boolean exclusive or
leftOp <<< rightOp	ints	ints	0b00001100uy <<< 2	0b00110000uy	bitwise shift left
leftOp >>> rightOp	ints	ints	0b00001100uy >>> 2	0b00000011uy	bitwise and
+op	ints		+3	3	identity
	floats		+3.0	3.0	
-op	ints		-3	-3	negation
	floats		-3.0	-3.0	
not op	bool		not true	false	boolean negation
~~~op	ints		~~~0b00001100uy	0b11110011uy	bitwise boolean negation

Table 1.1: Arithmetic operators on basic types. Ints, floats, chars, and strings means all built-in integer types etc. Note that for the bitwise operations, digits 0 and 1 are taken to be `true` and `false`.

Operator	leftOp	rightOp	Expression	Result	Description
leftOp < rightOp	bool	bool	true < false	false	Less than
	ints	ints	5 < 2	false	
	floats	floats	5.0 < 2.0	false	
	chars	chars	'a' < 'b'	true	
	strings	strings	"ab" < "cd"	true	
leftOp > rightOp	bool	bool	true > false	true	Greater than
	ints	ints	5 > 2	true	
	floats	floats	5.0 > 2.0	true	
	chars	chars	'a' > 'b'	false	
	strings	strings	"ab" > "cd"	false	
leftOp = rightOp	bool	bool	true = false	false	Equal
	ints	ints	5 = 2	false	
	floats	floats	5.0 = 2.0	false	
	chars	chars	'a' = 'b'	false	
	strings	strings	"ab" = "cd"	false	
leftOp <= rightOp	bool	bool	true <= false	false	Less than or equal
	ints	ints	5 <= 2	false	
	floats	floats	5.0 <= 2.0	false	
	chars	chars	'a' <= 'b'	true	
	strings	strings	"ab" <= "cd"	true	
leftOp >= rightOp	bool	bool	true >= false	true	Greater than or equal
	ints	ints	5 >= 2	true	
	floats	floats	5.0 >= 2.0	true	
	chars	chars	'a' >= 'b'	false	
	strings	strings	"ab" >= "cd"	false	
leftOp <> rightOp	bool	bool	true <> false	true	Not Equal
	ints	ints	5 <> 2	true	
	floats	floats	5.0 <> 2.0	true	
	chars	chars	'a' <> 'b'	true	
	strings	strings	"ab" <> "cd"	true	

Table 1.2: Comparison operators on basic types. Ints, floats, chars, and strings means all built-in integer types etc..

## 1.2 Basic arithmetic functions

Type	Function name	Example	Result	Description
Ints and floats	<code>abs</code>	<code>abs -3</code>	3	Absolute value
Floats	<code>acos</code>	<code>acos 0.8</code>	0.6435011088	Inverse cosine
Floats	<code>asin</code>	<code>asin 0.8</code>	0.927295218	Inverse sinus
Floats	<code>atan</code>	<code>atan 0.8</code>	0.6747409422	Inverse tangent
Floats	<code>atan2</code>	<code>atan2 0.8 2.3</code>	0.3347368373	Inverse tangentvariant
Floats	<code>ceil</code>	<code>ceil 0.8</code>	1.0	Ceiling
Floats	<code>cos</code>	<code>cos 0.8</code>	0.6967067093	Cosine
Floats	<code>cosh</code>	<code>cosh 0.8</code>	1.337434946	Hyperbolic cosine
Floats	<code>exp</code>	<code>exp 0.8</code>	2.225540928	Natural exponent
Floats	<code>floor</code>	<code>floor 0.8</code>	0.0	Floor
Floats	<code>log</code>	<code>log 0.8</code>	-0.2231435513	Natural logarithm
Floats	<code>log10</code>	<code>log10 0.8</code>	-0.09691001301	Base-10 logarithm
Ints, floats, chars, and strings	<code>max</code>	<code>max 3.0 4.0</code>	4.0	Maximum
Ints, floats, chars, and strings	<code>min</code>	<code>min 3.0 4.0</code>	3.0	Minimum
Ints	<code>pown</code>	<code>pown 3 2</code>	9	Integer exponent
Floats	<code>round</code>	<code>round 0.8</code>	1.0	Rounding
Ints and floats	<code>sign</code>	<code>sign -3</code>	-1	Sign
Floats	<code>sin</code>	<code>sin 0.8</code>	0.7173560909	Sinus
Floats	<code>sinh</code>	<code>sinh 0.8</code>	0.8881059822	Hyperbolic sinus
Floats	<code>sqrt</code>	<code>sqrt 0.8</code>	0.894427191	Square root
Floats	<code>tan</code>	<code>tan 0.8</code>	1.029638557	Tangent
Floats	<code>tanh</code>	<code>tanh 0.8</code>	0.6640367703	Hyperbolic tangent

Table 1.3: Predefined functions for arithmetic operations

Name	Example	Description
<code>fst</code>	<code>fst (1, 2)</code>	
<code>snd</code>	<code>snd (1, 2)</code>	
<code>failwith</code>	<code>failwith</code>	
<code>invalidArg</code>	<code>invalidArg</code>	
<code>raise</code>	<code>raise</code>	
<code>reraise</code>	<code>reraise</code>	
<code>ref</code>	<code>ref</code>	
<code>ceil</code>	<code>ceil</code>	

Table 1.4: Built-in functions.

## **1.3 Precedence and associativity**

- boolean or
- boolean and

Operator	Associativity	Description
ident "<" types ">"	Left	High-precedence type application
ident "(" expr ")"	Left	High-precedence application
"."	Left	
prefixOp	Left	All prefix operators
" rule	Left	Pattern matching rule
ident expr, "lazy" expr, "assert" epxr	Left	
"**" opChar	Right	Exponent like
"*" opChar, "/" opChar, "%" opChar	Left	Infix multiplication like
"-" opChar, "+" opChar	Left	Infix addition like
":?"	None	
"::"	Right	
"^" opChar	Right	
"!=" opChar, "<" opChar, ">" opChar, "=", " " opChar, "&" opChar, "\$" opChar	Left	Infix addition like
":>", ":?>"	Right	
"&", "&&"	Left	Boolean and like
"or", "  "	Left	Boolean or like
","	None	
":="	Right	
"->"	Right	
"if"	None	
"function", "fun", "match", "try"	None	
"let"	None	
";"	Right	
" "	Left	
"when"	Right	
"as"	Right	

Table 1.5: Precedence and associativity of operators. Operators in the same row has same precedence. See ?? for the definition of `prefixOp`