You can type the name of any function listed in this document in the help desk search box to find out the number of arguments, output type, and general usage examples. Literals are atomic values such as integers, floating-point numbers, strings (text inside quotation marks), Booleans, and characters. NOTE: all functions that end with ? return a boolean value, true or false.

**Atomic data type NUMBER: functions & constants**

Functions for Arithmetic with numbers: (these take numbers as args and return a number)

+  -  /  *

quotient  remainder  modulo  add1  sub1  abs  max  min  gcd  lcm  round  floor  ceiling  truncate  numerator  denominator  exact->inexact  inexact->exact
(also see functions for powers and roots below).

Functions for Number Comparison: (these take numbers as args and return a boolean)

=  <  <=  >  >=

Functions for conversion of number to String and from String to number:

number->string  string->number

Functions for Powers and Roots: (these take numbers as args and return a number)

sqrt  expt  exp  log  sqr

Trig functions: (these take numbers as args and return numbers)

sin  cos  tan  asin  acos  atan

Function to generate Random numbers: (take a number as input and return a number)

random

Functions that are Type checkers for numbers: (take numbers as args and return a boolean)


Constants: (these are decimal numbers)

pi  e

Literals: (atomic primitive types)

13  56.868686868  #i2343.5674  ...
Atomic data type **BOOLEAN**: functions and literals

Functions:


Literals:

| #t | #f | true | false |

Atomic data type **EMPTY LIST**: functions, literals, & constants

Type checker functions:

| null? | empty? |

Literal:

| '(' |

Constants:

| empty | null |

Atomic data type **QUOTED SYMBOL**: functions and literals

Quoted symbols should not be confused with symbols that are named values in the global environment. Named values written in the global environment are placeholders for data and functions and are evaluated by looking them up in the GE. Quoted symbols are just labels and are not placeholders for any values nor are they written into the GE.

Function for type conversion to string:

| symbol->string |

Functions for equality checking and type checking:

| symbol=? | symbol? |

Literals:

Any sequence of characters, preceded by a single quote (apostrophe), that does not start with a number, a string, a boolean, or any other atomic primitive type. For example: 'abc help 'SOS 'peace988bce 'sqYOUare '(5 3 8 5) '. Spaces signal the end of the quoted symbol.
Atomic data type CHARACTER: functions and literals

Function for type conversion to integer:

char->integer

Functions for specific type checking:


Functions for character comparison (ci stands for case insensitive; each returns a boolean):


Functions for character conversion to upper/lower case:

char-downcase  char-upcase

Literals:

#a  #\A  #b  #\B  ... Any single keyboard character preceded by #\
Compound data type STRING: functions and literals

String Constructors (look up proper arguments)
make-string  string  build-string

String type checker:
string?

Frequently used String Functions
string-length  substring  string-append

Functions for string conversion to upper/lower case:
string-upcase  string-downcase

Function for string type conversions:
string->list  list->string

Functions for string comparisons (ci stands for Case Insensitive):
  string-ci<=?  string-ci>?  string-ci=>?

Literals:
"Hello world"  "I <3 Vassar"  "*(&*(%" ... Almost any sequence of characters inside ""s (quotation marks) except for "s inside ""s.

Compound data type NON-EMPTY LIST: functions and literals

List Constructors and literals
list  cons  empty  null

List Selectors
first  rest

Frequently used List Functions
length  list-ref  list-tail  append  reverse  remove  sort  list->string  member

Iterative or Higher Order List Functions
map  andmap  ormap  for-each  filter

List type checkers
cons?  empty?  null?