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:
+  -  /  *
quotient remainder modulo add1 sub1 abs max min gcd lcm round floor
ceiling truncate numerator denominator
exact->inexact  inexact->exact

Functions for Number Comparison: (these take numbers as args and return boolean)
=  <  <=  >  >=

Functions for conversion of number to String and from String to number:
number->string  string->number

Functions for Powers and Roots:
sqrt  expt  exp  log  sqr

Trig functions:
sin  cos  tan  asin  acos  atan

Function to generate Random numbers:
random

Functions that are Type checkers for numbers:

Constants:
pi  e

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

Functions:
- boolean?
- not
- equal?
- eqv?
- eq?
- boolean=?
- false?

Special forms:
- and
- or

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

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 'sqUare '(5 3 8 5) '()
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):

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

Literals:
#\a #\A #\b #\B ... Any single keyboard character preceded by #\n
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):

Literals:
"Hello world" "I love 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
list  cons

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?

Literals:
empty  null

...and many, many more.