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.

**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:
```
=  <  <=  >  >=
```

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:

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 'sqYOUare
**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:

<table>
<thead>
<tr>
<th>char-downcase</th>
<th>char-upcase</th>
</tr>
</thead>
</table>

Literals:

| #\a | #\A | #\b | #\B | ... Any single keyboard character preceded by #\ |

**Compound data type STRING: functions and literals**

String Constructors

<table>
<thead>
<tr>
<th>make-string</th>
<th>string</th>
<th>build-string</th>
</tr>
</thead>
</table>

String type checker:

<table>
<thead>
<tr>
<th>string?</th>
</tr>
</thead>
</table>

Frequently used String Functions

<table>
<thead>
<tr>
<th>string-length</th>
<th>substring</th>
<th>string-append</th>
</tr>
</thead>
</table>

Functions for string conversion to upper/lower case:

<table>
<thead>
<tr>
<th>string-upcase</th>
<th>string-downcase</th>
</tr>
</thead>
</table>

Function for string type conversions:

<table>
<thead>
<tr>
<th>string-&gt;list</th>
<th>list-&gt;string</th>
</tr>
</thead>
</table>

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 "."