

## # Basic List Manipulation functions in LISP →

Variables in LISP are symbolic i.e. nonnumeric atoms. They may be assigned values by setq function.

setq function → It bounds the second element to the first variable.

setq takes two arguments the first of which must be a variable. first variable is never evaluated and should not be in quotation-marks. The second argument is evaluated.

Examples →

(i) → (setq x 10)

(ii) → (setq x (+ 35))

(iii) → (setq x' (+ 35))  
(+ 35)

(iv) → y  
unbound variable: y

Basic List Manipulation function →

function call	value returned	Remarks
---------------	----------------	---------

(i) (car '(abc))	a	car takes one
------------------	---	---------------

(22)

argument, or list & returns the first element

(ii) (cdr '(abc)) (bc)  
cdr takes one argument, a list and returns a list with the first element removed.

(iii) (cons 'a '(bc)) (abc)  
cons takes two arguments, an element, and a list & returns a list with the element inserted at the beginning.

(iv) (list 'a '(bc)) (a (bc))  
list take any no. of arguments & returns a list with the arguments as elements

Example →

(i) → (cons '(x23) '(1))

~~(x23)(1)~~ ((x23)1)

(ii) → (cons (x23) '(1))  
(61)

The syntax for a function-call is —

(function-name arg1 arg2 ... -)

When any no. of arguments may be used, when a function is called the arguments are first evaluated from left to right.

Some other Definition  $\rightarrow$

(19)

i) Top-element  $\rightarrow$  The element which is consist by list, called Top-elements.

for example  $\rightarrow$  (a b (c d) e (f)) is a list  
The Top-element of above - list are -

a, b, (c, d), e, (f)

and Top-element of sub-list (c, d) is c, d

e). Symbolic-expression or S-expressions  $\rightarrow$

Atom, lists and strings are the only valid objects in LISP. They are called Symbolic-expression or S-expressions.

$\rightarrow$  LISP Programs run either on an interpreter or as compiled code.

$\rightarrow$  The interpreter examines source programs in a repeated loop, called the "read-evaluate-print loop". This loop reads the program code, evaluates it and prints the values returned by the program. ' $\rightarrow$ ' symbol is used as a prompt.

for example  $\rightarrow$  To find the sum of the three nos. 5, 6 & 9, we type after the prompt, we can write

$\rightarrow$  (+ 5 6 9)

20

$\rightarrow$

## # LISP Function:-

(20)

i) Predefined Numerical function  $\rightarrow$

The basic numeric operations are +, -, \* and /. Arguments may be real or integer values and function takes different no. of arguments.

for example  $\Rightarrow$

+ & \* normally takes two or more arguments while - & / take two arguments. Nil is also the same as the empty-list.

Predefined Numeric Functions  $\rightarrow$

function call	Value Return	Remarks
(i) (+ 3 5 8 4)	20	+ takes zero or more arguments. The sum of zero argument is 0.
(ii) (- 10 12)	2	- takes two arguments
(iii) (* 2 3 4)	24	* takes zero or more arguments
(iv) (/ 25 2)	12.5	/ takes two arguments