LISP :

Lisp expression : (op arg1 ... argn) Prefix notation

- Numeric
- Symbolic
- Boolean
- Branching
- Functions

operator
- function
  - ↳ user-defined
  - ↳ built-in
- special operator

```
> (+ 137 349)
> 486
```

```
> (* (* 3 5 1) (- 10 6))
> 60
```

```
> (*)
> 1
```

```
> (+)
> 0
```

SPECIAL OPERATOR :

QUOTE :

> (quote (+ 3 1))

> (+ 3 1)

> '(+ 3 1)

> (+ 3 1)

SETQ :

> (setq x 3)

> 3

> x

> 3

> (+ x y)

> error

LIST MANIPULATIONS:

* SELECTORS
* CONSTRUCTORS

SELECTORS

(a b c d) → head: a

→ tail: (b c d)

→ car (first)

→ cdr (rest)

```
> (setq x '(a b c d))
    > (a b c d)

> (car x)
    > a

> (cdr x)
    > (b c d)

> (car (cdr x))
    > b
```

→ Get me c

> (caddr x)

First d then d then a

(car (cdr (cdr x)))

> (setq x '(a (b c) d e))

> (a (b c) d e)

→ Get me b

> (caadr x)

> b

→ Get me c

> (cadadr x)

> c

Constructors

* cons
* list

```
> (setq a 7)
> (cons 'a '(b c))
      > (a b c)
> (cons a '(b c))
      > (7 b c)
> (cons 1 (cons 'a '(3)))
      (cons 1 '(a 3))
      > (1 a 3)
> (cons 1 '(cons 'a '(3)))
      > (1 cons 'a '(3))
```

EMPTY LIST:

> (setq x '(3))

> (3)

> (car x)

> 3

> (cdr x)

> NIL

↳ both a list and an atom.

NIL → empty list

→ false

> (car NIL)

> NIL

> (cdr NIL)

> NIL.