

LISP(1958)

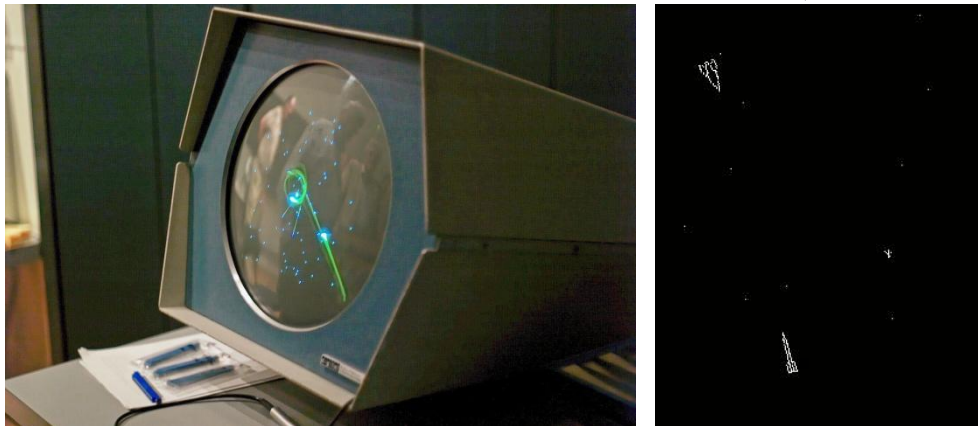
プログラミング言語論
マツキン

LISP

- 1958年、FORTRANに続く、2番目に古い高級言語（高水準言語）
- LISt Processing（リスト処理）に由来
- S式を用いた関数型言語

プログラム用数式表現

- John McCarthy(ジョン・マッカーシー)による発明
- 着想は、1956のダートマス会議(The Dartmouth Summer Research Project on Artificial Intelligence)に遡る
- Steve Russell (世界初のビデオゲーム Spacewar! の発明者) が IBM704用にLISPインタプリタの核となるeval関数をマシン語で実装



Spacewar! on PDP-1

LISPの特徴

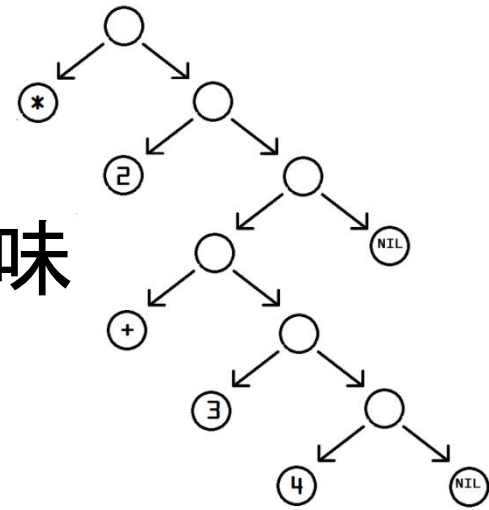
- インタプリタ型（後にコンパイラも1962に作成）
- REPL (Read-Eval-Print Loop)環境
- 関数型言語
- 動的な型付け
- 前置記法（ポーランド記法 Polish Notation）
- LISPはLISPで書かれている
- 人工知能用言語として広く利用された
- IF-THEN-ELSE が初めて実装された

LISPの方言

- LISPは非常に多くの方言が存在した
- 1984にCommon Lisp として統合が目指された
- 1994にANSI Common Lisp として規定
- 今でも広く利用されているのが Common Lisp と Scheme(1975)
- Emacs のマクロ言語としても emacs lispが利用されている

LISP コードの書き方

- S式(S-expression)とは、
(命令 引数 …)
のように記述する。SはSymbolの意味
- 引数に命令を入れることもできる
(命令 (命令 引数) …)
- このようにLISPはあっという間にかっこだらけになる。



Lots of Irritating Superfluous Parentheses

前置記法 (ポーランド法)

- $1 + 2 \Rightarrow + 1 2$
- $2 + 3 - 1 \Rightarrow - + 2 3 1$
- $(1 + 2) * (4 - 3) \Rightarrow (* (+ 1 2) (- 4 3))$

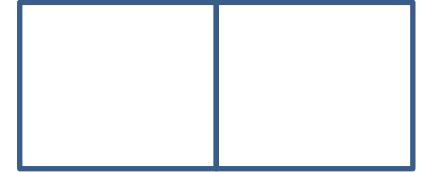
Lispの関数

- (関数 [引数] [引数] ...)
- (+ 1 2)

Lispでの演算

- `(+ 1 2 3 4 5)`
- `(- 100 10 1)`
- `(* 1 2 3 4 5)`
- `(/ 100 10 2)`
- `(/ 2 4)`
- `(+ 1/2 1/4)`
- `(+ (* 5 3) 5)`

CONS (Construct)



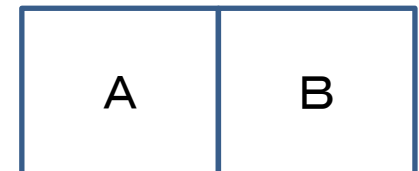
- データは2つの値のペアで、CONSと呼ぶ

- (CONS 'A 'B)



- CARで最初の値、CDRで二つ目の値を得る

- (CAR (CONS 'A 'B))



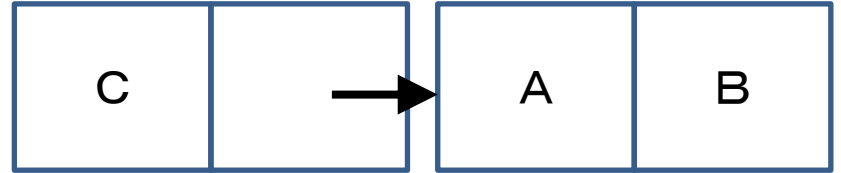
- (CDR (CONS 'A 'B))

car (Contents of the Address part of Register number)

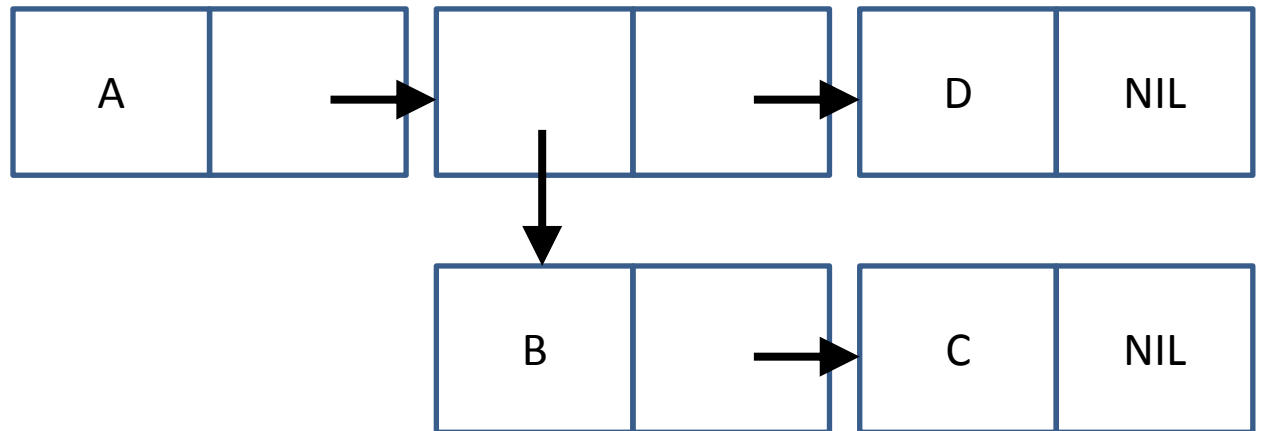
cdr (Contents of the Decrement part of Register number)

LIST

- (CONS 'C (CONS 'A 'B))



- (CONS 'A (CONS (CONS 'B (CONS 'C 'NIL)) (CONS 'D 'NIL)))



変数の代入(束縛)

- (setq a 10)
- (setq b 20)
- (+ a b)
- (setq c (+ a b))
- a
- b
- c

if

- `(setq x 10)`
- `(= x 10)`
- `(< x 10)`
- `(> x 10)`
- `(and (> x 5) (< x 7))`
- `(or (< x 3) (> x 9))`
- `(if (= x 10) (quote true) (quote false))`
- `(setq y 20)`
- `(if (> x y) (setq a x) (setq a y))`

ループ

- (setq i 1)
- (loop
 (if (= i 10)(return i))
 (setq i (+ i 1))
)

標準出力

- (format t “Hello, world!”)
- (setq a 10)
- (format t “a is ~d” a)
- (setq s “Hello world!”)
- (format t “s is ~a” s)

リスト処理

- `(cons 'a '(b c d))`
- `(car '(a b c))`
- `(cdr '(a b c))`
- `(setq x 'nil)`
- `(setq x (cons 1 x))`
- `(setq x (cons 2 x))`
- `(setq x (cons 3 x))`

関数定義

- (defun 関数 (引数)
プログラム
)
- (defun canon (a s)
 (format t "~d ~d" a s)
)

三角関数

- `(sin .7)`
- `(setq x .5)`
- `(cos x)`
- `(tan (* 2 x))`

その他

- (load “canon.lisp”)
- (bye)

課題

- 砲台ゲームをLispで作成する
- 着弾距離 $d = s^2 \sin(2a) / 9.8$

ヒント

- (defun canon (a s)
 (setq d (着弾距離を計算する...))
 (if 着弾距離dが490より大きく510より小さいなら format "HIT" さもなくば format "MISS")
)