ANSI Common Lisp Practice

ismdeep

December 29, 2019

Contents

1 chapter-01

1.1 sum

```
; (dotimes (i n s) () ...)
; i => [0, 1, ... , n]
; return value is s
; ... is operations

(defun sum (n)
   (let ((s 0))
      (dotimes (i n s)
         (incf s i))))

(format t "~D~%" (sum 10))
```

1.2 addn

```
; lambda ?
; I don't know how to use it yet. -_-

(defun addn (n)
   #'(lambda (x)
      (+ x n)))

(format t "~A~%" (addn 10))
```

2 chapter-02

2.1 Form

```
(format t "~A~%" (+ 1 2))
(format t "~A~%" (+ 1 2 3 4 5))
(format t "~A~%" (/ (- 7 1) (- 4 2)))
```

2.2 Evaluation

```
(format t "~A~%" (quote (+ 3 5)))
(format t "~A~%" '(+ 3 4))
```

2.3 Data

```
(format t "~A~%" 'Hello)
(format t "~A~%" '(my 3 "Sons"))
(format t "~A~%" (list 'my (+ 2 1) "Sons"))
(format t "~A~%" ())
(format t "~A~%" nil)
```

2.4 List Operations

```
(format t "~A~%" (cons 1 '(2 3 4)))
(format t "~A~%" (car '(1 2 3 4)))
(format t "~A~%" (cdr '(1 2 3 4)))
(format t "~A~%" (car (cdr (cdr '(1 2 3 4)))))
(format t "~A~%" (third '(1 2 3 4)))
```

2.5 Truth

```
(format t "~A~%" (listp '(1 2 3 4)))
(format t "~A~%" (null nil))
(format t "~A~%" (not nil))
(format t "~A~%" (if (listp '(a b c))
   (+ 1 2)
   (+ 5 6)))
```

2.6 Functions

```
(defun our-third (x)
   (car (cdr (cdr x))))

(format t "~A~%" (our-third '(a b c d)))
```

2.7 Recursion

```
(defun is-member (obj lst)
   (if (null lst)
      nil
      (if (eql (car lst) obj)
         T
         (is-member obj (cdr lst)))))

(format t "~A~%" (is-member 1 '(2 3 4 1 7 8)))
```

2.8 Reading Lisp

```
(defun our-member (obj lst) (if (null lst) nil
    (if
(eql (car lst) obj) lst (our-member obj (cdr
    lst)))))
```

2.9 Input and Output

```
(format t "~A plus ~A equals ~A. ~%" 2 3 (+ 2 3))

(defun askem (string)
   (format t "~A~%" string)
   (read))

(let ((age (askem "How old are you?")))
   (format t "I'm ~A year old.~%" age))
```