

# 프로그래밍의 원리, 가을 2012

## 실습 준비

### Scheme의 소개

강동욱, 최민아  
프로그래밍 연구실, 서울대학교

2012년 9월 6일

1. 아래 나와있는 항목들은 Scheme에서의 표현식입니다. 모든 Scheme 표현식은 값을 되돌려줍니다. 아래 나와있는 표현식들을 차례로 Scheme 실행기에 입력했을 때 각 항목들이 무슨 값으로 계산되는 지를 알아봅시다. 더불어 값의 종류에는 어떤 것들이 있을 지도 생각해 봅시다.

6

"Principles of Programming"

(+ 5 3 4)

(+ (\* 2 4) (- 4 6))

(\* 3.141592 2)

(= 3 4)

(< 3 4)

(if (and (> 4 3) (< 4 (\* 3 4)))

4

3)

(+ 2 (if (> 4 3) 4 3))

(car (cons (cons 1 "abc") (+ 2 3)))

```
(cdr (cons (cons 1 "abc") (+ 2 3)))
```

```
((lambda (x y) (+ x y)) 3 4)
```

```
((lambda (t) (* 5/9 (- t 32))) -40)
```

## 2. define, 리스트

Scheme에서는 define을 사용하여 값이나 함수에 이름을 붙일 수 있습니다. 또 ‘리스트’라는 자료구조를 만들 수 있습니다. 리스트는 여러 값들을 순서있게 나열한 것입니다. 리스트는 항상 앞에서만 값을 뽑아낼 수 있습니다. 아직 수업을 나가지는 않았지만 숙제를 할때 유용할 수 있는 define 과 리스트 사용법에 대해서 알아보시다.

### (a) define 사용방법

```
(define a 3)
```

```
a
```

```
(define b (+ a 1))
```

```
(+ a (* a b))
```

```
(= a b)
```

```
(< a b)
```

```
(if (and (> b a) (< b (* a b)))
```

```
  b
```

```
  a)
```

```
(define (f2c t) (* 5/9 (- t 32)))
```

```
(f2c -40)
```

```
(f2c 212)
```

### (b) 리스트를 만드는 방법

```
(define a '(1 2 3))
```

```
(define b (list 1 2 'orange 4))
```

```
(define c ())
```

(c) 리스트의 앞에 원소를 추가하는 방법

```
(define d (cons 100 a))  
(define e (cons 'apple ()))
```

(d) 가장 앞에 있는 원소를 꺼내는 방법

```
(car a)  
(car d)  
(car e)
```

(e) 리스트에서 앞의 원소를 제거하는 방법

```
(cdr a) (cdr c)  
(car (cdr (cdr b)))
```

(f) 빈 리스트인지 알아보는 방법

```
(null? d) (null? c)
```

(g) 두 리스트를 합치는 방법

```
(append '(1 2 3) '(4 5 6))
```

(h) 이러면 왜 안될까를 생각해 봅시다.

```
(cons '(1 2 3) 4)
```

3. 주어진 수의 부호 알아내는 함수를 만들어 봅시다. 이 함수는 양수에 대해서 1을 돌려주고, 음수는 -1, 0일 경우에는 0을 돌려줍니다.

```
((lambda (n) (...)) -2)  
-1
```

4. 주어진 수의 절대값을 알아내는 함수를 만들어 봅시다. 이 함수는 한 수를 입력으로 받아서 그 수의 절대값을 돌려 줍니다. 예를 들자면 다음과 같습니다.

```
((lambda (n) (...)) -2)  
2
```

5. 가장 큰 수를 구하는 함수를 만들어 봅시다. 이 함수는 세 개의 수를 입력으로 받아서 그 결과를 돌려줍니다.

```
((lambda (x y z) (...)) -50 7 16)  
16
```