(define (foo y)
  (+ 6 y)
)

Higher-Order Procedures

Procedures as returned values:

(SUM-N 3)
0+1+2+3

(COND (() _)
(() _)
(() _)
(else _))

```
(define (plus-list x)
  (cond ((number? x)
         (lambda (y) (+ (sum-n x) y)))
        ((list? x)
         (lambda (y) (+ (sum-list x) y)))
        (else (lambda (x) x))
  ))
1 ]=> ((plus-list 3) 4)
;Value: 10
1 ]=> ((plus-list '(1 3 5)) 5)
;Value: 14
```

((plus-list 3) 4)
((lambda (y) (+ (sum-n 3) y)) 4)
((lambda (y) (+ 6 y)) 4)

((plus-list '(1 3 5)) 5)
((lambda (y) (+ (sum-list '(1 3 5)) y)) 5)
((lambda (y) (+ 9 y)) 5)