```scheme
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; MIT 6.001 Spring/2005                                         ;;;;;
;;;;; Barb Cutler - Recitation 1: scheme expressions                ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; Student: Abrantes Araújo Silva Filho                          ;;;;;
;;;;; Date: 2019-02-10                                              ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;; Evaluate:
(* 2 3)
; 6

(*2 3)
; Error, there is no "*2" name binded to something

((*2 3))
; Error, could not apply (6)

(define a 5)
; bound the name "a" with the value of expression "5", wich is 5

a
; 5

b
; Error, there is no name "b"

(define b (+ a 2))
; bound the name "b" with the value 7

a
; 5

b
; 7

(define a 3)
; bound the name "a" with the value of expression "3", wich is 3

a
; 3

b
; 7

(a)
; Error, could not apply 3

(define define 4)
; bound the name "define" to value 4. Note: this will cause the lost
; of the bind "define -> procedure to name things"!

(and #t)
; #t

(and #t #f)
; #f

(and #t #f (4))
```

```
; #f. Note that (4) is an error because we can not apply 4, but the
; expression return #f because Scheme evaluate the "#f" subexpression and
; this alone was sufficient to determine the whole value of expression
; to be false.

(define and 5)
; bound the name "and" to value 5. Note: this will cause the lost of
; the boind "and -> boolean procedure and"!
```