Scheme Basics

Getting Started

For each Scheme expression below, what value results when the expression is evaluated?

42

"Hello World"

(8 + 9)

(+ 8 9)

(define a 10)

a

b

(define b a)

b

(* a b)

Nested Expressions

For each Scheme expression below, what value results when the expression is evaluated?

(* (- 8 4) (+ 1 10))

(define foo 100)

(define bar (* 10 foo))

(+ (- (- 2010 (/ bar foo)) (* foo (- (/ bar foo) 3))) 37)
Hello, λ

For each Scheme expression below, what value results when the expression is evaluated?

(lambda (x) (/ x 1024))

(((lambda (x) (/ x 1024)) 4096))

(lambda () 1)

(((lambda () 1))

(((lambda () 1) 5)

(lambda (y z) (+ z y))

(((lambda (y z) (+ z y)) 5 4))

(((lambda (y z) (+ z y)) x 7))

What’s in a name?

Assume that you’ve already evaluated the following Scheme expressions:

(define x 1)
(define y -1)
(define foo (lambda (a b) (+ a b)))
(define bar (lambda (x) x))
(define baz (lambda () 1))
(define quux (lambda (p) (foo p 5)))

Alright, now to what value do each of these Scheme expressions evaluate?

x
foo
(foo 1 2)
(foo 1)
(foo)
(baz)
(bar 10)
(quux (foo (baz) (bar y)))

Short and sweet: Syntactic Sugar

For each Scheme expression below, write an equivalent Scheme expression that doesn’t explicitly use lambda.

(define foo (lambda (a b) (+ a b)))
(define bar (lambda () 1))