

User-Defined Procedures in Scheme

Summary: This lab provides practice with simple user-defined procedures.

- Exercises
 - Exercise 0: Preparation
 - Exercise 1: Fractional Parts
 - Exercise 2: Adding 2
 - Exercise 3: Converting Feet to Meters
 - Exercise 4: A Quadratic Polynomial
 - Exercise 5: Quadratic Roots
 - Exercise 6: Swapping List Elements
 - Exercise 7: Spherical Calculations
 - Exercise 8: `snoc`
- Extra Tasks for Those Who Finish Early
 - Extra 1: Rotate
 - Extra 2: Scoring Figure Skating
 - Extra 3: A Different `snoc`
 - Extra 4: Rounding Numbers
 - Extra 5: Multiple Roots

Exercises

Exercise 0: Preparation

- a. Open the reading in a separate window or tab.
- b. Start DrScheme and make sure that you're in "Pretty Big" Scheme mode.

Exercise 1: Fractional Parts

Copy the `fracpart` procedure from the reading on procedures and make sure that it works as advertised.

```
;;; Procedure:
;;;   fracpart
;;; Parameters:
;;;   val, a real number
;;; Purpose:
;;;   Express the fractional part of val as a fraction.
;;; Produces:
;;;   rat, a rational number.
;;; Preconditions:
;;;   val cannot be complex.
;;; Postconditions:
;;;   0 <= rat < 1.
```

```

;;; (- val rat) is a whole number (or a a close approximation).
;;; rat is exact.
;;; rat is approximately equal to the fractional part of val
;;; (within some unknown level of accuracy).
;;; rat is the ratio of two integers.
(define fracpart
  (lambda (val)
    (inexact->exact (- val (truncate val)))))

```

Exercise 2: Adding 2

Write a Scheme procedure (`addtwo a`) that returns the sum $a+2$.

In this exercise, and in the subsequent ones, you need not document the procedure unless I explicitly tell you to do so.

Exercise 3: Converting Feet to Meters

a. Define a Scheme procedure, (`feet->meters ft`) that takes one argument, a real number representing a length measured in feet, and returns the number that represents the same length as measured in meters. Note that one foot is equal to exactly $761/2500$ meters.

b. Use this procedure to determine the number of meters in one mile (5280 feet).

c. How would you use this procedure to determine the number of feet in a 1000-meter race? (And no, “It depends on the number of runners” is not an acceptable answer.) *Note that this is a thought question. It does not ask you to compute that value, simply how you would compute the value.*

d. Add `feet->meters` to your library of procedures.

Exercise 4: A Quadratic Polynomial

a. Define a procedure, (`poly1 x`), that corresponds to the polynomial $5x^2 - 8x + 2$.

b. Test your procedure on the values 0, 1, 2, 3, 4.

Exercise 5: Quadratic Roots

In homework 2, you wrote a pair of definitions for the roots of the quadratic polynomial ax^2+bx+c . The disadvantage of those definitions was that you had to redefine a , b , and c whenever you wanted to recompute roots. You can now make the root computations a procedure.

a. Write a procedure (`quadratic-root a b c`) that finds one root of the following quadratic equation by using the quadratic formula. (I don’t care which root you find.)

$$ax^2+bx+c = 0$$

In case you've forgotten, the quadratic formula is

$$(-b \pm \sqrt{b^2 - 4ac})/2a$$

b. Test your procedure by computing

```
(quadratic-root 1 -5 6)
(quadratic-root 2 -10 12)
(quadratic-root 1 4 4).
```

c. Use algebra to check these answers.

d. What are `(quadratic-root 1 0 1)` and `(quadratic-root 1 0 2)`?

Exercise 6: Swapping List Elements

a. Write a procedure, `(swap-first-two lst)`, that, given a list as an argument, creates a new list that interchanges the first two elements of the original list, leaving the rest of the list unchanged. Thus,

```
> (swap-first-two (list 'a 'b 'c 'd 'e))
(b a c d e)
```

In this problem, assume that the list given to `swap-first-two` has at least two elements; do not worry about the possibility that `swap-first-two` might be applied to numbers, symbols, empty lists, or lists with only one element.

b. Add this procedure to your library.

Exercise 7: Spherical Calculations

The volume of a sphere of radius r is $\frac{4}{3}$ times π times r^3 .

The circumference of a sphere of radius r is 2 times π times r .

a. Write a procedure, `(sphere-volume r)`, that takes as its argument the radius of a sphere (in, say, centimeters) and returns its volume (in, say, cubic centimeters).

b. Write a procedure, `(sphere-circ->radius circ)`, that converts the circumference of a sphere to its radius.

c. Use these procedures to compute the volume of a standard softball, which has a circumference of 12 inches.

d. Use these procedures to compute the volume of a Chicago-style softball, which has a circumference of 16 inches.

e. Compute the volumes of each kind of softball using centimeters instead of inches.

Exercise 8: snoc

a. Define a procedure `snoc` (“cons backwards”) that takes two arguments, of which the second should be a list. `snoc` should return a list just like its second argument, except that the first argument has been added at the right end:

```
> (snoc 'alpha (list 'beta 'gamma 'delta))
(beta gamma delta alpha)
> (snoc 1 (list 2 3 4 5 6))
(2 3 4 5 6 1)
> (snoc 'first null)
(first)
```

Hint: There are at least two ways to define this procedure. One uses calls to `reverse` and `cons`; the other uses calls to `append` and `list`.

b. Add `snoc` to your library.

Extra Tasks for Those Who Finish Early

If you find that you have extra time, you might want to attempt the following tasks.

Extra 1: Rotate

Write a procedure, (`rotate lst`), that, given a nonempty list of elements (e.g., (`a b c`)), creates a new list with the original first element moved to the end .

For example,

```
> (rotate (list 'a 'b 'c))
(b c a)
> (rotate (list 1 2))
(2 1)
> (rotate (rotate '(first second third fourth)))
(third fourth first second)
```

Extra 2: Scoring Figure Skating

In a figure-skating competition, judges have observed the competitors’ performances and awarded three separate scores to each competitor: one for accuracy, one for style, and one for the difficulty of the chosen routine. Each score is in the range from 0 to 10. The rules of the competition specify that a competitor’s three scores are to be combined into a weighted average, in which accuracy counts three times as much as difficulty and style counts twice as much as difficulty. The overall result should be a single number in the range from 0 to 10.

a. Write a comment in which you describe the nature and purpose of a procedure that takes three arguments -- a competitor’s accuracy, style, and difficulty scores -- and returns their weighted average.

b. Define the procedure that you have described.

c. Test your procedure, looking for cases in which the weighted average is computed incorrectly. (If you find any, make corrections in your definition.)

Extra 3: A Different `snoc`

Write `snoc` (exercise 8) in two different ways. (You should have already written it one way; find another way.)

Extra 4: Rounding Numbers

Implement `(round-to-n-places val p)`, which rounds `val` to `p` places after the decimal point.

Note that you may have described the steps for `round-to-n-places` in the lab on numeric values.

Extra 5: Multiple Roots

Write a procedure, `(quadratic-roots a b c)` which computes *both* roots of the quadratic equation and returns them in a list.

$$ax^2+bx+c = 0$$

For example,

```
> (quadratic-roots 3 5 2)
(1 2/3)
```

Copyright © 2007 Samuel A. Rebelsky. This work is licensed under a Creative Commons Attribution-NonCommercial 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.