CSC151.02 2014S, Class 06: Writing Your Own Procedures

Overview

- Preliminaries.
  - Admin.
  - Work to do.
  - Questions on the homework.
- Why define your own procedures?
- How to define your own procedures.
- Lab(s).

Preliminaries

Admin

- Continue partners from yesterday for the rest of this week!
- Since yesterday’s lab went more smoothly than I expected, we’re combining the drawings lab and procedure lab days. We’ll continue with procedures tomorrow.
- Today is a changing of the guard, of sorts.
  - Thanks to our guest mentors, Toby B and Alex G
  - Welcome to our new Tuesday mentor, Ajuna K
- Extra credit:
  - Thursday at 4:15: Spencer Liberto, Lea Marolt Sonnenschein, and Daniel Torres on Ushahidi and more.
  - CS Table Friday at Noon: The ACM Code of Ethics.
  - Convo Feb. 5. (I’ll give my "Why go to convo" lecture closer to the date.)
  - EC for attending tonight’s climbing club meeting at 7:30 at the rock wall.
  - Others?

Work to do

- Finish HW 2 (due tonight).
- Read "How Scheme Evaluates Expressions (version 2)".
- Drawings lab writeup (due Friday): Exercise 4, parts a, c, d, f, and g
  - Email subject: CSC 151 Writeup 3: Drawings as Values (YOUR NAME(s) HERE)
- Procedures lab writeup (also due Friday): Exercises 3 and 6
  - Email subject: CSC 151 Writeup 4: Procedures (YOUR NAME(s) HERE)
Why define your own procedures?

- We often want to do the same (or similar) computations again and again and again.

\[
\begin{align*}
(* \ x \ x) \\
(* \ y \ y) \\
(* \ 2 \ 2)
\end{align*}
\]

It would helpful to be able to name that code, and to take different inputs

How to define your own procedures

A fascinating syntax

\[(\text{define} \ \text{NAME-OF-PROCEDURE} \ (\lambda \text{INPUTS} \ \text{EXPRESSION})\)\]

\[(\text{define} \ \text{square} \ (\lambda \text{val} \ (* \text{val} \ \text{val}))\)\]

We can call this like any other procedure already in Scheme

\[(\text{square} \ 5)\]

Like

\[(\text{define} \ \text{val} \ 5) \\
(* \ \text{val} \ \text{val})\]

Similarly,

\[(\text{define} \ \text{weighted-average} \ (\lambda \text{grade1 grade2 grade3 grade4 grade5} \ (/- \ (- \ (+ \text{grade1 grade2 grade3 grade4 grade5}) \ (\text{min} \text{grade1 grade2 grade3 grade4 grade5}) \ (\text{max} \text{grade1 grade2 grade3 grade4 grade5})) \ 3)))\]

Call

\[(\text{weighted-average} \ 100 \ 90 \ 90 \ 10 \ 90)\]

Remember:

- The form of procedures
- Evaluation is a lot like "(define PARAM VAL-YOU-PROVIDED)a" then body.
Questions

- Can I reuse names between procedures?
  - Yes. Scheme is smart.

```scheme
(define sq (lambda (val) (* val val)))
(define dbl (lambda (val) (+ val val)))
(+ (sq 5) (dbl 2)) 29 (sq (dbl 2)) 16
```

Lab

Lessons from the drawings lab

- The various procedures make new drawings, rather than modifying an existing drawing.

```scheme
(define d1 (recolor-drawing "yellow" (scale-drawing 50 drawing-unit-circle))) (drawing-hshift 20 d1)
(image-show (drawing->image d1 200 200))
```
- Similar to

```scheme
(define x 5) (square x) x
```
- drawing-scale scales an image’s offset as well as its size.
  - See the whiteboard for an explanation