

Class 20: Naming Local Values

Held: Wednesday, 30 September 2009

Summary: We consider why and how you name values within a procedure, and limit access to those names to within the procedure.

Related Pages:

- EBoard.
- Lab: Local Bindings.
- Reading: Local Bindings.
- Due: Assignment 4: Exploring Colors.

Notes:

- I will not have office hours today. Sorry.
- Please keep your partners from Monday.
- Assignment 5 is now ready.
- EC for attending some part of the Rosenfield anniversary symposium on Thursday.
- Don't forget about the history of programming languages talk at noon on Friday in 3821.
- Reading for Friday: Boolean Values and Predicate Procedures.

Overview:

- Why name things.
- Naming things with `let`.
- Naming things with `let*`.
- Naming procedures.
- Lab.

The Problem: Naming Values

- As we've seen in many problems, it helps to name the values that we use within our procedure. Why?
 - It can make the code more readable because the name tells us something about the role the value plays.
 - It can make the code more efficient, because it allows us to avoid recomputing a value.
- Another reasons to name things is that we might want to create helper procedures and only make them available to the current procedure.

Naming Things with `let`

- You name things with `let`.
- `let` has the form

```
(let ((name1 exp1)
      (name2 exp2)
      ...
      (namen expn))
  body)
```

- `let` has the meaning:
 - Evaluate all the expressions.
 - Update the binding table to associate each name with the corresponding value.
 - Evaluate *body* using the updated binding table.
 - Eliminate all the bindings just created.
- You can use `let` in a simple expression:

```
(define values (list 1 4 2 4 1 5 9))
(let ((largest (max values))
      (smallest (min values)))
  (/ (+ largest smallest) 2))
```

- More frequently, we use `let` within a procedure.

Sequencing Bindings with `let*`

- If we want to bind some things in sequence, we need to use `let*` rather than `let`.
- `let*` has the form

```
(let* ((name1 exp1)
       (name2 exp2)
       ...
       (namen expn))
  body)
```

- `let*` has the meaning:
 - Evaluate *exp₁*.
 - Update the binding table to associate *name₁* with that value.
 - Evaluate *exp₂*.
 - Update the binding table to associate *name₂* with that value.
 - ...
 - Evaluate *exp_n*.
 - Update the binding table to associate *name_n* with that value.
 - Evaluate *body* using the updated binding table.
 - Eliminate all the bindings just created.

Copyright © 2007-9 Janet Davis, Matthew Kluber, Samuel A. Rebelsky, and Jerod Weinman. (Selected materials copyright by John David Stone and Henry Walker and used by permission.) This material is based upon work partially supported by the National Science Foundation under Grant No. CCLI-0633090. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. 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.