Class 22: Input and Output

Held: Tuesday, February 27, 2007

Summary: Today we consider why and how to make Scheme do explicit input and output.

Related Pages:
- EBoard.
- Lab: Input and Output.
- Reading: Output in Scheme and Input in Scheme.

Notes:
- Homework 8 is now due Friday.
- Homework 9 is now due Friday.

Overview:
- Interactive programs.
- Output in Scheme.
- Input in Scheme.

Interactive Programs
- In our traditional usage of Scheme, we define procedures and then ask Scheme to evaluate one or more expressions.
- Note that this model does not follow our traditional uses of computing, in which we start a program (e.g., by typing its name in the shell) and then interact with it.
- In the simplest interactive model, a program repeatedly queries someone for information, the person enters the information, and the program.
- To match this model, we need to figure out how to get input from the user and generate output for the user to read.

Program Output
- You're already seen one procedure for producing output: display.
  - (display val) displays a value, eliminating the quotation marks from around strings.
- We've also used the (newline) procedure that starts a new line of output.
- There's a third useful output procedure, write, that works a lot like display except that it also displays quotation marks around strings and displays characters with the #\.
- A fourth procedure, write-char, only works with characters.
Program Input

- What procedures are available to read input?
  - Primarily `read`, which reads one Scheme value, skipping whitespace as appropriate.
  - Also `read-char`, which reads a single character for input.
- `read` is an interesting procedure. In many cases, it will continue to read for more than one line.

Lab

- You will, of course, experiment with these issues in the lab.
- Be prepared to reflect.