

CSC151.02 2014S, Class 34: Iteration

Overview

- Preliminaries.
 - Admin.
 - Questions.
- Reflecting on procedures and side-effects.
- A useful tool for repetition: `for - each`.
- Contrasting `map`, `for - each`, `repeat`, and recursion.
- Lab.

Preliminaries

Upcoming Work

- Homework for Wednesday: HW 7: GitHub
- Today's lab writeup: 4c and 4d.
- Reading for tomorrow: Geometric Art through Numeric Recursion
- HW 8 is also posted.

Admin

- If our Web server is not working correctly, you should be able to get to today's lab at the URL
- Continue your partner assignments from Monday. If you missed class on Monday, work with someone else who missed class on Monday.
- CS T shirts soon to be available. Vote if you plan to buy one by sending email to sepc-cs@grinnell.edu. The list of possibilities is at <http://imgur.com/a/P0haH#0>.
 - You should encourage the SEPC to make a "Curried Lamb-Duh" T-shirt
- Extra credit:
 - Convocation, Wednesday, noon: Elizabeth Kolbert (the Sixth Extinction)
 - Technology in liberal arts education, lots of talks on Thursday
 - No CS extra this Thursday.
 - CS table Friday: The API Performance Contract Robert F. Sproull and Jim Waldo. 2014. The API performance contract. Commun. ACM 57, 3 (March 2014), 45-51. DOI=10.1145/2576792 <http://doi.acm.org/10.1145/2576792>
<http://delivery.acm.org/10.1145/2580000/2576792/p45-sproull.pdf>

Questions

Reflecting on procedures and side-effects

- A transition in how we think about procedures (not necessarily positive)
- It used to be, we called procedures to return a result
 - And nothing else
- Now, procedures have "side effects" - they change "the world" (or themselves, or an image)
- When you're doing `(+ (square 5) (square 7))`, it doesn't matter which order we do the square
- If you do `(turtle-forward! t 10) (turtle-turn! t 10)`
- Order of operations is much more important!
- We didn't tell you, but `map` does not necessarily do the transformations in order from left to right
 - `(map square (list 1 2 5 7 9))`
- `map` is not a good tool for side-effecting operations, since order matters

What is a procedure with a side-effect?

- One view: The procedure changes something external (e.g., an image, the implicit position of the turtle, ...)
- Another view: We are not assured of getting the same behavior each time we call the procedure on the same parameters

Examples

- Side effects: `turtle-forward!`, `image-select-ellipse!`, ...
- No side effects (pure): `square`, `increment`, `iota`, `rgb-redder`, `irgb-new`, ...
- Badly named: `image-show`

A useful tool for repetition: `for-each`

- When you want to apply the same function to one thing and then another and then another (as in `map`)
- But you care about the order
- To different values
- and we do this more for the side effects than the resulting list
- USE `for-each`

`(for-each PROCEDURE LIST-OF-PARAMETERS)`

Contrasting map, for - each, repeat, and recursion

- Are we doing the work with side effects or not?
 - With side effects: for-each, repeat, recursion
 - Without: map, recursion
- What does our result look like?
 - A list of transformed values: map
 - Nothing: for-each, repeat
 - Variable: recursion
- Do we do something different at each step?
 - No: repeat, (recursion)
 - Yes: for-each, map, recursion

Lab

Samuel A. Rebelsky, rebelsky@grinnell.edu

Copyright (c) 2007-2014 Janet Davis, Samuel A. Rebelsky, and Jerod Weinman. (Selected materials are copyright by John David Stone or Henry Walker and are used with permission.)



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.