

## Class 18: Iteration

**Held:** Monday, 28 September 2009

**Summary:** We consider a technique for stepping through the values in a list, and doing some action (e.g., moving a turtle or drawing something with the GIMP tools) for each value in the list.

### Related Pages:

- EBoard.
- Lab: Iteration.
- Reading: Iteration.

### Notes:

- Does anyone need a group for Assignment 4?
- Are there any questions on that assignment?
- We'll go over a few things on the quiz.
- I was not able to get the exam graded for today. Hence, we'll start on the next topic (iteration) today and go over the exam later in the week. (I have it scheduled for Friday, but I'm hoping to move the discussion up a few days.)
- Starting today, I'd like you to work with someone new - someone you have not worked with previously.
- I will not have office hours on Wednesday at 11:00. I will try to be available on Wednesday afternoon.
- Reading for tomorrow: .

### Overview:

- A Problem: Drawing Spirals
- A Useful Tool: `for-each`

## The Problem: Making Spirals

- We ended the turtles lab trying to make spirals.
- Basically, you need to move forward a little and turn a little
- Three possible strategies:
  - Move the same amount each time and turn the same amount each time
  - Move a little more each time, and turn the same amount each time
  - Move the same amount each time, and turn a different amount each time
- The first strategy procedures a circle, and is possible to do with `repeat`.
- The other two strategies seem difficult to do without a *lot* of hand coding.
  - I was happy to hear that many of you thought of using `map`

## The `for-each` Function

- We'd like to use `map`, but there are two problems.
    - Officially, you don't know the order in which the function is applied when you do a `map`; it could be first to last, it could be last to first; it could be all at the same time.
    - We get back a list, which we really don't need.
  - Scheme provides `for-each` for situations like this.
  - Like `map`, but for sequences of actions.
    - `for-each` is guaranteed to work in order.
    - `for-each` doesn't return anything.
- 

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.