CSC 151 Spring 2020
Topic: Higher Order Procedures

April 8, 2019

Day 28

Left-Section, Right-Section

In the reading we introduced:

left-section and right-section

What's going on!?!?

1. Sorry that came out of nowhere – the readings used to be structured differently.
2. How does (left-section binary-proc arg1) work:

   A way to define a new procedure - essentially using section without the <>

Left-Section, Right-Section

```
#lang racket
(require csc151)

(define subtract2-1
  (section - <> 2))

(define subtract2-2
  (left-section - 2))

(define subtract2-3
  (right-section - 2))
```

```
> (subtract2-1 1.0)
-1.0
> (subtract2-2 1.0)
1.0
> (subtract2-3 1.0)
-1.0
```

HOP

Think about the answers to these questions:

1. **What is a higher-order procedure?**
2. **What are some examples of higher-order procedures?**
3. **Why do we use higher-order procedures?**

HOP

1. What is a higher-order procedure?

A procedure that takes another procedure as a parameter or returns a procedure as its final value.

2. What are some examples of higher-order procedures?

Map – takes a procedure and a list, applies the procedure to each item in the list

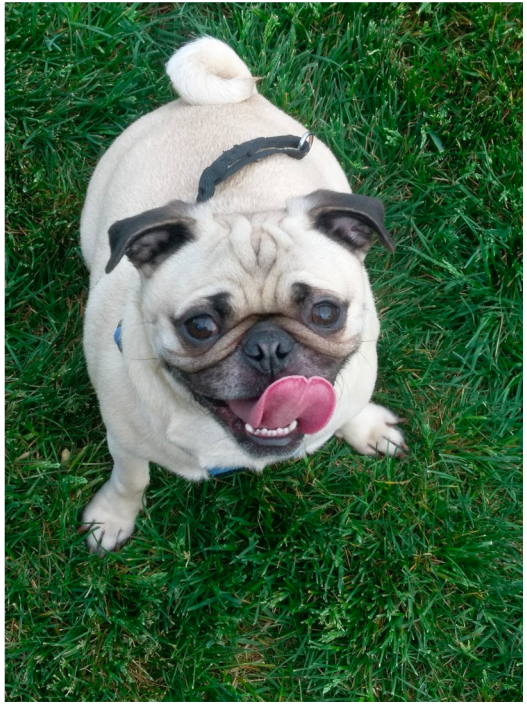
Reduce – applies a procedure to pairs of values in a list to combine with all vals

Apply – run a procedure with a list of values used as parameters

Left-section – fill in one parameter for a procedure

3. Why do we use higher-order procedures?

It's efficient for the programmer, and maybe also the computer. We might make fewer mistakes when we reuse code! Sometimes it's how we create procedures.



Lab Time

Ezra wants to know, is it warm in your part of the world? He loves playing outside!

Upcoming Work

Submit all problems before the extras by Friday 4/10 at 10:30pm Central Time.
Include 6P's for: 3, 4, 5,

To: csc-151-02-grader@grinnell.edu
Subject: CSC 151-02 Lab 4/8

Assignment 6 due Thursday April 9

Exam 2 resubmission due April 10 (optional)

Exam 3 will be posted on Friday