6.001 SICP – September 10

6001-Introduction

Trevor Darrell trevor@csail.mit.edu
32-D512

6.001 web page: http://sicp.ai.mit.edu/
section web page: http://www.ai.mit.edu/trevor/6001/

- how to use the tutor system
- problem sets and projects
- edwin/MIT scheme
- basic scheme
- read eval print

What is the focus of 6.001?

- This course is about Computer Science

Examples

7
8.5
(+ 2 4)
6
(* (- 5 3) (/ 9 3))
6
(7 - 4)

error
Edwin demo...

- What are the "two worlds" of a scheme program
- How do we get from one to the other?

Examples

(> 7 3)
#t
y
unbound
(+ y 1)
unbound
(define y 17)
y -> 17
y
17
(+ y 1)
18

Syntax

Things that make up scheme programs:

- self-evaluating 23, "hello", #t
- names +, pi
- combinations (+ 2 3)
  (* pi 4)
- special forms (define pi 3.14)
  (lambda ...)
  (if ...)

Note that special forms are not combinations, even though they are syntactically similar. A combination requires that the first subexpression be a procedure.

Evaluation Rules

- A numeral, string or boolean evaluates to itself (number, string, #t, #f).
- A name evaluates to the value associated with that name in the environment
- A combination is evaluated as follows:
  1. Evaluate the subexpressions in any order
  2. Apply the value of the operator subexpression to the values of the remaining subexpressions.

What about...

(+ 4)
  4
(- 3)
  -3
(/ 5)
  .2
(/ 60 5 2 3)
  2
(+)
  0
(*)
  1
(-)
*error*

Define

(define x 4)
  :Value: "x --> 4"
x
  4
(+ x 5)
  9
(define y (+ x 2))
  :Value: "y --> 6"
y
  6
(define x 2)
  :Value: "x --> 4"
y
  ??
Remember...

- Register on 6.001 Tutor system.
- Next recitation Seth Tieler—I’ll be back on Sept 17.
- Problem Set 1: Basic Scheme
  - To Be Completed By: September 14.
- Project 0 – play with Edwin
  - To Be Completed By: September 15.