

## Class 03: Python Lab

**Held:** Tuesday, February 2, 2010

**Summary:** You ground your learning of the Python programming language with some hands-on exercises.

### Related Pages:

- EBoard.

### Notes:

- *Beautiful Code* reading for Thursday: Chapter 19: Multidimensional Iterators in NumPy.
- Other reading for Thursday: Find something interesting on UML and then send me a question.
- Summer MAP discussion, today at 4:15 in JRC 101.
- EC: Thursday's CS Extra: Summer opportunities in CS.
- EC: Friday's CS Table: Pair Programming.

### Overview:

- Beautiful Code 27.
- About programming exercises.
- Programming Exercise 1: Some Basics.
- Programming Exercise 2: Binary Search.
- Programming Exercise 3: A Fraction Class.

## Beautiful Code 27: RESTful Programming

- Don't forget to check out the questions/comments on this reading.
- What is the application that Patzer describes?
- What is Patzer's primary thesis?
- What lessons does Patzer expect you to take from this example?
- What makes the code beautiful?
- What makes the code ugly?

## Programming Exercises

- We'll do a few rounds of programming exercises.
- At some point, I'll stop you and we'll go over the code you've written for one round.
  - It's okay if you're not finished by the time I stop you.
  - If you finish one round before the discussion, just go on to the next round.
- Be prepared to discuss what you had difficulty with in each round.

## Round 1: Some Basics

- Write some simple program segments to make sure that you understand the core aspects of Python: Variables, Strings, Procedures, Conditionals, Lists, Loops, Classes, and Dictionaries
- Create a small inheritance heirarchy and figure out what happens when multiple ancestors provide methods with the same name.
- Write a simple grep procedure that takes as parameters a pattern string and a list of strings, and returns a list of all strings in the list that contain the pattern. (The pattern should be taken verbatim.)

## Round 2: Binary Search

- Simplified problem:
  - Inputs:
    - i, an integer
    - vals, a sorted array (list) of integers. (Sorted in increasing order.)
  - Output:
    - If i is in vals, the index of i
    - If i is not in vals, -1

## Round 3: A Fraction Class

- Implement a simple fraction class to represent rational numbers. Make sure to support traditional operations on rationals.
- 

Copyright © 2010 Samuel A. Rebelsky. 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.