Object-Oriented Programming

The Big Ideas:	• Primitives lead to combinations via composition. • If a particular kind of combination is seen frequently, it can be identified as a pattern. • If we want to interact with the pattern, and not the primitives or combination, we create an abstraction. A procedure is a good example of such an abstraction. • Environments: Look in the most local scope for a binding of a given variable. If found, return the binding. If not, check the parent scope. Repeat until you find an assignment, or reach the global scope. • Classes and Instances are Environments • `pat.salutation()` is exactly equivalent to `Staff601.salutation(pat)` (see Chapter 2 of the Course Notes)

Introduction

This week, we introduce you to some of the core concepts of 6.01. Our four units are Programming and State Machines, Signals and Systems, Circuits, and Probability and Planning.

This week, we will also be focusing on programming, in particular the Object Oriented Programming paradigm, in Python. This week is a special week; people will be coming into 6.01 with many different backgrounds.

- If you have never programmed before, we recommend one of the introductory books listed in the References section of this document, as well as reading Chapter 2 in detail. You should also complete the Python Tutor.
- If you have programmed before but not in Python, we recommend using one of the introductory books below as a reference, as well as being familiar with the information in Chapter 2. You should also complete the Python Tutor.
- If you have programmed before in Python, we recommend using one of the introductory books as a reference. You are expected to be capable of the problems in the Python Tutor, so take a look and make sure you understand them.

6.01 this week is all about programming: primitives, expressions, assignments, functions, environments, OOP, and inheritance.

Vocabulary

In order to engage the material, be able to communicate about the topic with others, and in particular ask questions, we encourage familiarity with the following terms:

Theory

- PCAP

- Interpreter (versus Compiler)
- Expression
- Variable
- Structured Data
- Mutation
- Aliasing
- Procedures and Procedure Calls
- Non-Local Reference
- Environments
- Functions as First-Class Objects
- Object Oriented Programming
- Class
- Instance
- Method
- Inheritance
- Recursion

Practice

- Python
- Datatypes
    - Numbers: ints, bools, float
    - Strings: chars, strings
    - Sequences: strings, lists, tuples
    - Dictionaries: dictionaries
- Multi-Assignment
- Error Messages
- List Comprehensions
- Lambda

Check Yourself

After this week in 6.01, you should be familiar with the following:

Theory: you should understand:

- Basic programming concepts
- OOP paradigm concepts
- PCAP: modularity, abstraction, and why it's important and useful

Practice: you should be able to:

- Navigate the 6.01 Software Documentation
- Complete the Python Tutor
- Use a Python error message
- Know Python datatypes and when to use them

This week could be a really heavy week, if you are a new programmer. You should focus on gaining a strong grounding in the fundamentals. If you are not new to programming, but new to Python, you should focus on improving your programming deficiencies, Python deficiencies, and familiarity with the course basics.

Resources

In addition to Chapter 1 of the 6.01 Course Notes giving a good general overview of the course's content and objectives, and Chapter 2 giving good background for those unfamiliar with Python, Chapter 3 covers the content you will be expected to know for the first week of 6.01. The Course Notes not only contain a solid background on the content of this course; it also contains helpful walkthroughs of problems (some very similar to those in the tutors). Use them!

Think Python is an excellent free resource for learning Python.

The Python Tutorial (note this is *not* the 6.01 Python Tutor) is also an excellent free resource, and segues into the Python documentation well. Bookmark and visit early and often.

Learning Python, the O'Reilly book on Python, assumes little programming experience and covers programming topics in great detail. You must have MIT Certificates to view it for free.