Introduction to Programming for Computational Linguistics

L555

Dept. of Linguistics, Indiana University
Fall 2010

What is Programming?

Decent definition from wikipedia:

> *Computer programming . . . is the process of designing, writing, testing, debugging / troubleshooting, and maintaining the source code of computer programs. This source code is written in a programming language. . . . The purpose of programming is to create a program that exhibits a certain desired behaviour (customization). . . .*

http://en.wikipedia.org/wiki/Computer_programming (retrieved 8/11/10)

What is a Program?

At an abstract level, a program is a sequence of commands, which produces an output for a given input.

Example 1:

1. Input: a text file containing all of *Ulysses*
2. Program: stuff happens (Input $\mapsto$ Output)
3. Output: every bigram (two-word sequence) with its associated frequency

Example 2:

1. Input: your income information
2. Program: stuff happens (Input $\mapsto$ Output)
3. Output: how much tax you have to pay

Algorithms

A program is basically an **algorithm**, i.e., a sequence of commands

Here's what a sketch of an algorithm for printing out a text's unigrams (i.e., wordlist) might look like:

1. Read in each word from the text
   - 1.1 Store each word
   - 1.2 Add to the count of each word, storing (word,count) pairs in some storage device
2. Read through the storage device
   - 2.1 Print each word with its count

But how do we "read in" something or "store" things?

Programming Languages

Programming languages share a lot in common:

- They often have similar data structures & features (lists, functions, modules, ...)
- They require you to use explicit syntax, e.g.:
  - Only well-defined functions can be used
    - `exec` is a legitimate command in Python
    - `evac` is not a legitimate command
  - The language forces you to follow particular formats
    - In Python, you have to indent within a `for` loop
    - In Perl, you have to enclose the contents of a loop within brackets.

Languages differ in the specifics of the syntax, but good programming practice in one carries over to another

Why Should Linguists Care?

1. Linguists like to work with data
2. Data is often electronically encoded, and there is often huge amounts of it
3. Thus, linguists need some way to manipulate this data

Computational linguists need to learn how to program, not just to analyze data, but also to develop technology

- NB: There are some programming books specifically geared at linguists; see, e.g., http://www.u.arizona.edu/~hammond/

What Will We Learn This Semester?

We'll examine one programming langauge in particular, Python, and you'll learn:

- the basic & not-so-basic capabilities of Python
  - lists, tuples, strings, dictionaries, loops, functions, exceptions, objects, etc.
- how to convert an algorithm into program code
- fundamental concepts for writing good programs
- how to write programs for text processing

Why Python?

Why Python?

- It's quick: It is very good for writing short scripts and for text processing.
- It's powerful: At the same time, Python has much support for turning small programs into much larger projects (such as object-oriented programming)
- It's easy: Function names are (arguably) rather transparent in Python.
- It's free and available across systems (code is generally portable across platforms)
- It's marketable: organizations like Google, Pixar, and the NSA use Python

Resources on Python

Books:

- *Beginning Python: From Novice to Professional* by Magnus Lie Hetlund (our recommended book)
- *Learning Python* by Mark Lutz
- *Think Python* by Allen Downey (freely available online): http://www.greenteapress.com/thinkpython/thinkpython.pdf
- *Dive Into Python* by Mark Pilgrim (also available online; for experienced programmers): http://www.diveintopython.org/

Online resources:

- Guido van Rossum's Python Tutorial: http://www.python.org/doc/current/tut/
- Python-forum.org: http://python-forum.org/pythonforum/index.php
- Or, of course, search for information . . .

Obtaining Python

- The latest python is available for different platforms at: http://www.python.org/download/
- Mac: It should be pre-installed. Type `python` at a terminal to check.

Some notes for Windows users:

- On Windows it may not appear as if Python is installed: it could be installed, but it's only available in the directory where it was downloaded.
- To handle this, you can:
  - work in the directory where Python was installed
  - include the full path of Python when you run your programs, e.g., `C:\Python25\python program.py`
  - change the environment variable `PATH` (check under "Control Panel") to include `C:\Python25`, so the Command Prompt can find python from any directory

NLTK

Natural Language Toolkit (NLTK) is:

> *Open source Python modules, linguistic data and documentation for research and development in natural language processing and text analytics, with distributions for Windows, Mac OSX and Linux.*

http://www.nltk.org/

We will use NLTK towards the end of the semester

Command line interface

Let's step back from Python for just one second and talk about using a command line

Run commands by typing, instead of clicking ...

- Windows: open a Command Prompt
  - Start → Programs → Accessories → Command Prompt
- Mac: open a Terminal
  - Applications → Utilities → Terminal

See the contents of a directory:

- Windows: `dir`
- Mac (Unix): `ls`