Allan Hollander
4 October 1999

An introduction to the Python programming language

1. **What is Python?**
— "An interpreted, interactive object-oriented programming language"
— Seen as one of the 3 big scripting languages these days, the others being Tcl and Perl
— Is almost 10 years old

2. **The rise of scripting languages.**
— Prominent development in programming over past decade is use of scripting languages esp. Perl, Tcl, Python
— Ancestors are Unix shell scripts, awk, sed etc.
— These are high level languages: dynamically typed, don't have to declare variables, automatically allocate memory.
— Cost to all of these — execution speed. These interpreted languages are much slower than C or C++.
— Different sorts of efficiencies (c.f. Larry Wall): time, space, programming, maintainence, user, porting
— Not all tasks require execution efficiency. System administration tasks, run-once-only programs, application front ends.

3. **Python is not just a scripting language.**
— Its readability and use of high-level constructs (modules, classes) means it scales well
— Useful for medium-to-large scale projects, involving teams of programmers
— (Aside on Python vs. Perl)

4. **History.**
— Author is Guido van Rossum, a Dutch researcher now at CNRI in Reston, Va.
— First conceived over the winter holidays of 1989–90, first public release 1991. Current version 1.5.2.
— Named for Monty Python, not the snake.
— Usage of language steadily increasing ($\sim$40%/year since 1995 based on newsgroup postings and downloads.) 7 books on Python to be published this fall!

5. **High-level properties.**
— Free and open-source (BSD-style license).
— Extremely portable (Unix, Windows, Mac, BeOS, VMS ...)
— Compiles to interpreted byte-code (only need to port interpreter)
— Can run interactively.
— Easily extended and embedded.

6. **Language-level properties.**
— Object-oriented to the core.
— Emphasis on readability and clarity of constructs.
— Modularity.
  - much functionality in imported library modules
  - .py files $\Longleftrightarrow$ modules

— Indentation marks block structure. Unusual, but works well.
— Clean exception handling structures.

7. **Datatypes.**
— Numbers — int, long, float, complex
— strings
— lists and dictionaries.
— New datatypes through class definitions in extension modules.

8. **Python examples.**

A tiny Python program.

```
# Converts whr tables from/to polygon-row x species-column format to
# three-column format (polyid,species code, value)
# The original tables are in a format like the following:
# "Poly-id", "A001", "A002", "A007", "A023"
# 244,1,3,0,0
# 245,1,2,0,1
# I want these to be formatted like:  244,"A001",1
#                                     244,"A002",3 etc.

import sys, string
from string import split

infile = sys.argv[1]
outfile = sys.argv[2]
f = open(infile, "r")
of = open(outfile, "w")

line = string.rstrip(f.readline())
colnames = split(line, ",")

lines = f.readlines()
of.write("%s,%s,%s\n" % ("poly-id","speccode","value"))

for line in lines:
    ll = split(string.rstrip(line), ",")
    for i in range(1,len(colnames)):
        if int(ll[i]) > 0:
            speccode = string.replace(colnames[i], '"','')
            of.write("%s,%s,%d\n" % (ll[0], speccode, int(ll[i])))

f.close
of.close
```

An example of a class.

```
class Stack:
    def __init__(self):    # constructor
        self.items=[]
    def push(self,x):
        self.items.append(x)
    def pop(self):
        x = self.items[-1]
        del self.items[-1]
        return x
    def empty(self):
        return len(self.items) == 0
```

Object-oriented programming for the faint of heart. No need to get into complexities of C++ or Java, use Python!

9. **Computer Programming for Everybody.**
— GvR's initiative (currently with 2 years of funding from DARPA) to develop materials and environments for teaching programming to non-programmers.
— Python may be ideal first programming language: simple and clear, yet powerful and used in the real world.

10. **Interfaces.**
— GUIs. Tkinter is standard, many others with more limited use.
— All major database programs.
— COM objects. (O'Reilly has Python programming under Windows book coming out soon.)
— Java. JPython ~ implementation of Python in Java.

11. **Applications and useful packages.**
— Common architecture to these: Python front end, computationally intensive bits in C or C++.
— Gadfly (small SQL database program).
— Python Imaging Library.
— Numeric Python (array language extension).
— XML processing.
— HTMLGen (class methods for automatically generating HTML).
— Zope (open source application server for managing dynamic websites).
— Users include LLNL, Industrial Light & Magic, Red Hat, . . .

12. **Resources.**
— Web sites `http://www.python.org`, `http://starship.python.net`.
— Usenet `comp.lang.python`
— Books *Learning Python* (Lutz & Ascher), *Programming Python* (Lutz).