Ever More Beautiful Code
For Ever More Beautiful Projects

Passably Proficient Python Programmer
Modestly Realistic Management Material

Brett Paufler
www.paufler.net

I wish to convince you of my ability to Program in Python.

FizzBuzz: how I learned modulo.
Reynold's Boids: but is it right?
Eight Queens: yes, solved.
Conway's Life: haven't we all?
Image Manipulation: up the whazoo.

Python has 200+ modules (the exact number depending upon how one counts them, I suppose) and 10,000+ commands in the base library. Obviously, I do not know them all, but I have a fair understanding of their extent, common usage, and have read all of the base documentation at least once, some of it twice, and others much-much more...

I've conducted a semi-thorough review of many of the common second party libraries, but listing them would be both silly and tedious. Shall I just copy paste a large portion of the Anaconda Package listing into this document? I found nothing of interest in AstroPy or BioPython. Although the NLTK tutorial was immensely interesting, I've never found a use for the library. And in order to keep this document reasonably short and to the point, I shall wrap this section up by saying, I find myself using Numpy, Scipy, Skimage, and Pandas more than anything else, but I can see

switching over to sqlite3 as a replacement to the last for future projects.

Project Highlights
(Functions lead to Classes lead to Modules)
Private Image Manipulation Library
local/remote (system/web/directory) scanners
Genetic Programming Football play generator

No Language is an Island
Python derives from C, which leads to D, with either shining best in the Linux environment... an understanding of which might require some conception of sockets and file handlers; and if you're going to go that direction, one might as well study up on TCP/IP, HTTP, HTML, CSS, and SVG. But do I bore? No? OK. SQL leads to b-trees (seriously, just a word), which led to CouchDB, and map/reduce, which for the most, I already knew; but even so, in those last two, I had fun for a month, maybe two. I have a love (nay, a passion, yes I have a passion, it that a key word search hit) for list comprehensions, but in my mind, closures seem overly complicated for the benefit (why not just use a class or a function with an extra parameter), while generators are just a syntax convenience at my level of code; but hey, (), instead of [] is simple enough to write if that's what you're into. I think I understand the philosophy behind Haskell and functional programming; and at the other extreme, SmallTalk and OOP. Scheme was fun for a week, wasted a year with JavaScript, and someday I'll claim to know PostScript and GhostScript, perhaps as a precursor to assembly, but it seems unlikely my mind will turn that way. Big O notation is not a useful construct in my world, but I've seen folks program in Python in the style of Lisp and Scheme and found it quite compelling

But of what value is the preceding? I don't really know, outside of perhaps getting a match on the keyword search. I simply cannot be

bothered to list every book I've ever read, technology that's captured my eye for a week, or bit of arcane lore that I found intriguing for a day, week, month, or as with Python, for several years, now. For you see, I am a Python Programmer. The rest is very much just a means to that end, towards pushing the limits of my knowledge, seeing what the others are doing, so I can best determine my own personal next step. In short, my experience with Python (and the world of programming in general) goes far beyond 'Hello World' and approaches a nice rudimentary mastery, coupled with some bits of expertise, and heaping helping of ignorance that I do believe will keep me humble to the end of time. I mean, let's be real, Python leads to C, which leads to Linux, which leads to assembler and that's only one vector in the graph. I will forever be ignorant at the edges, looking into the abyss; but as I move ever forward, at least for the moment, Python is my platform of choice, my place of understanding from which to view the rest.

And do you need to know anything else?
I enjoy the journey.
I am a programmer.
I shall program.

Brett Paufler
www.paufler.net

Ever More Beautiful Code
For Ever More Beautiful Projects

Have I succeeding in convincing you of my ability to Program in Python?