Intro to Scripting Python

Intro

ikalb@ucsc.edu, ikalb@cogswell.edu, ikalb@edmc.edu,

In this class, I will give you an introduction to the very basics of programming – also known as software development. To do this, we will be using the Python computer programming language.

My background: BS, MS in Computer Science, over 30 years software development & management of software developers. Have been teaching software development for over 5 years.

I am not an artist. I can't draw, so please excuse that from the beginning.

I also have poor handwriting, so if anything that I write is unclear, please let me know, and I will try to do better.

I have a disability. It is difficult for me to stand for long periods of time, so please bear with me when I need to sit or kneel on a chair.

Please ask questions if information is not clear. If I don't hear from you, I will assume that the information is understood. Tangents can be good.

[Ask about Software students vs Art students]

[Ask how many have written some type of software – not HTML]

++++Syllabus

Rules of the class:

Mutual respect. I will respect you by working hard at giving you the best of my knowledge, and in return, I expect you to listen when I am talking.

There will be homework – small programming assignments along the way. Homework assignments are due at the beginning of each week. I will not accept late homework. Even if you can't come to class, the homework is still due. Homework should be uploaded to the class site before each class.

Intro Notes Cogswell

or

Intro Notes UCSC

Vocabulary – very important, there may be a quiz on it. Take out a piece of paper or open a file, and I will give you definitions of words as the course progresses.

A little more about my background:
Started working in software at 16.
First computer IBM 1130 computer with 8K of “core” memory.

Rutgers University, NJ BS in Computer Science
Moved to CA to take first job.

Worked in compiler design – (ask if anyone knows what that is)
Got Masters degree at night while working during the day.

I have used many different languages
Fortran, APL, Assembly, PL/1, Cobol, WatFor, RPG, PL/M.
Worked for Apple Computer for a few years in UI software.
Learned object oriented programming using Object Pascal
Worked for Acuson as manager of UI group –
Most of my work was in Adobe Director with its programming language Lingo. (including an entire eLearning System)
Flash and ActionScript.
Over the past few years, I’ve gotten into Python.

The point is, I’ve been doing programming for many years. And the same concepts I learned in my first programming language are still usable today. In this course, we will be using the Python language. Python was designed to be very clear and easy to read. The syntax – the way that you write Python code was specifically designed to allow programmers to build programs quickly without the need for a lot of extraneous words and symbols. I will talk about the ‘readability’ of

program very often, and it is a key concept in writing code in what has become known as a 'Pythonic' way.

Once you learn the basic concepts of a programming language like Python, you will find that you can pick up a new computer language very quickly. No matter what the language – and there are many – the underlying concepts are very similar. The key things that we will learn about: variables, assignment statements, if statements, while loops, function calls – are all concepts that are easily transferable to any other language.

Python was created in the 90's by Guido van Rossum. (Worked for Google, now at DropBox in SF.) It has two current versions, 2.7 and 3.4. We will be using the version 2.7. This version is widely used in both schools and industry it is very mature – meaning that it is known to work very well, and there are many add on packages that have been written to be used with it.

The 3.4 version is obviously newer, but is not nearly as mature. Not nearly as many people are using 3.4. Further, there have been some changes to the language that make code written for version 2.7 not completely compatible with version 3.4.

The bottom line is this: we will be using version 2.7 which is considered state-of-the-art, and you will not run into any problems with this version because it is battle tested.

Version 2.7 has already been installed on all the computers in the classroom. Here's how to get to it:

Mac: Open Applications, open Python 2.7, double click on IDLE

Windows: Click on the Start button, type in IDLE