**IT FDN 100A: Foundations of Programming: Python**

**Instructor: Randal Root**

**Student: Morgan Lang**

**June 23, 2016**

# Assignment 1

## Assignment objectives:

1. Write a simple Python program asking the user for input.
2. Write a batch script that opens the program in the Python interpreter.
3. Test the batch file and program.
4. Document the learning experience.
5. Zip the assignment deliverables and submit them via canvas.

## Introduction

Prior to this class, my experience with Python had been through my class in Linux system administration. I knew how to import modules and how to indent and comment Python, but not much more than that. I definitely did not get a firm grounding in the basics of the language, so this class is perfect for me.

I also had some experience using batch scripts to start programs and pass paths to them. Lately, I have found myself almost exclusively using batches to start VMs by passing their paths to a hypervisor, thus avoiding having to interact with the hypervisor directly.

Therefore, considering my familiarity with both Python and batching, I expected this to be straightforward. However, I found that I still had – and have – a lot to learn.

## The batch file

The main difficulty I encountered in writing the batch file stemmed from my confusion regarding Python’s installation location. When downloading and installing Python, I did not change the default installation method or location, and therefore Python was added as a local program here:

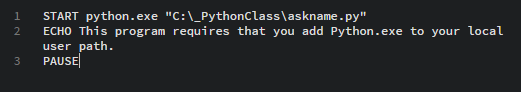
**C:\Users\morgan\AppData\Local\Programs\Python\Python35-32\Python.exe**

From my perspective, this is acceptable, since Python.exe has been added to my path and I can invoke it without having to explicitly invoke its absolute path.

However, I was concerned that because this first assignment required creating a batch file that pointed to an absolute path, other users who attempted to run my program would encounter difficulty, since I can’t predict their installation path or Python version.

My attempt to “solve” this issue was to insert a warning in the batch script indicating that users would have to add Python.exe to their path before running my program. This is not the best solution. Instead, there should be a way to programmatically test a user’s computer to determine the path to Python.exe. However, that’s beyond my understanding at this point.

**Startpython-askname.bat:**



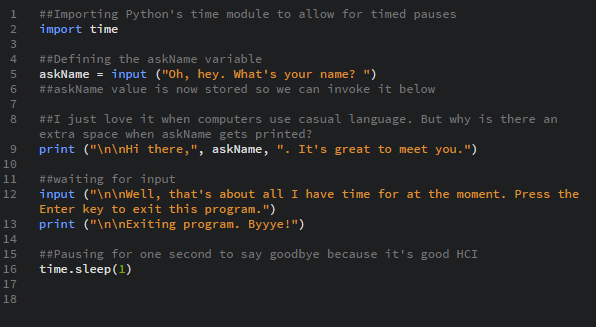
## The Python script

You will notice that I strayed from the instructions, both in the content of the batch and Python files and in the file names used.

I chose to respect the purpose of the assignment: to demonstrate a basic understanding of batching and of user interactivity in Python. However, I tried to add in as much of the content from the videos and book as possible. I also imported and used the time module from the Python library, something I had learned in my Linux classes.

I did that in order to challenge myself and to provide a better interactive experience.

**Askname.py:**



You will notice that I also changed the names of the batch and Python files in order to better reflect their actual function. I have found that it is hard to stay organized if file names are not explicitly related to their contents. Giving all my scripts unique names will help avoid confusion later in the quarter when we have written dozens of programs.

## Miscellany:

I chose to use Brackets rather than IDLE for this assignment simply because it’s what I’m accustomed to using. I certainly can use IDLE if it is required, but I feel that a more general-purpose editor is probably closer to a real-world experience.