# Python Homework 1—Choose a Python Environment

## Required Reading

If you are familiar with programming, the CyberAces modules speed through the essentials of Python.

CyberAces Introduction to Python Module (part of System Administration), <https://tutorials.cyberaces.org/tutorials/view/3-3-1.html>

## Optional Reading

If the CyberAces modules seem too difficult, *Automate the Boring Stuff with Python* is written to teach Python to people that have never programmed before.

*Automate the Boring Stuff with Python* by Sweigart, Chapters 0 and 1. This book is available online at <https://automatetheboringstuff.com/> (Scroll to the bottom to see the lists of chapters you can download.

If you enjoy cryptography, *Cracking Codes with Python* starts with basic ciphers and teaches you Python as you go.

*Cracking Codes with Python* by Sweigart  
 Introduction, ‘Crypto regarded as a weapon’ and xxv – xxviii, Installing Python  
 Chapter 1, pp. 1 - 11 (or <https://inventwithpython.com/cracking/chapter1.html>) about crypto paper tools  
 Chapter 2, pp. 12 - 19 (or <https://inventwithpython.com/cracking/chapter2.html>) about the Python interactive shell  
 Chapter 3, pp. 30 - 33 about the Idle file editor.  
 Chapter 5, pp. 54 - 67 (or <https://inventwithpython.com/cracking/chapter5.html>) about the Caesar cipher

## Python

We will use Python for several reasons:

1. Python is relatively easy to learn
2. Python is a common language among IT Security professionals and is the language of many ITSec tools.
3. Cryptology involves huge integers, and Python will handle them automatically without having to use special classes, like Java’s BigInteger. We will use Python in our cryptology module.

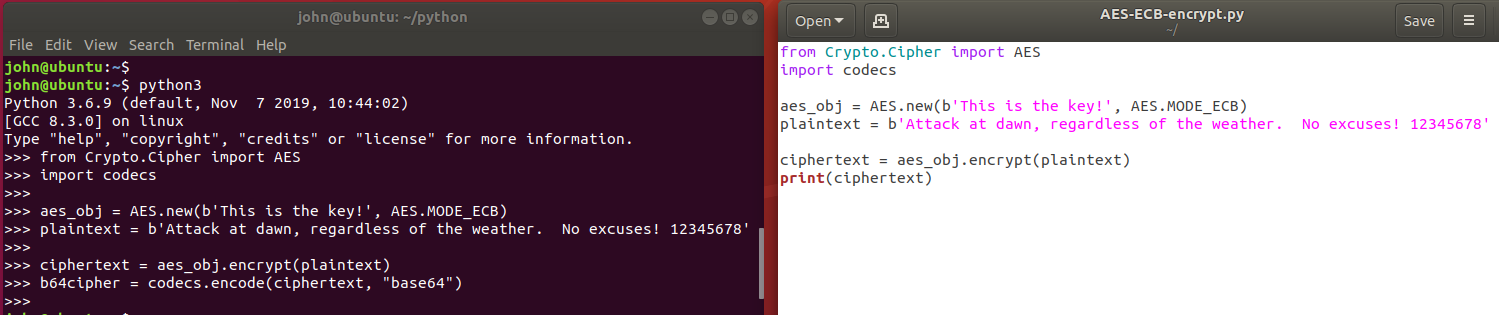
Python is installed by default on most Linux variants (the version is 3.6.x in Ubuntu, 2.7.x in other distros.) In Windows, Python needs to be installed (instructions later.) Make sure to get Python 3, as Python 2 is now deprecated.

## Choosing an Environment

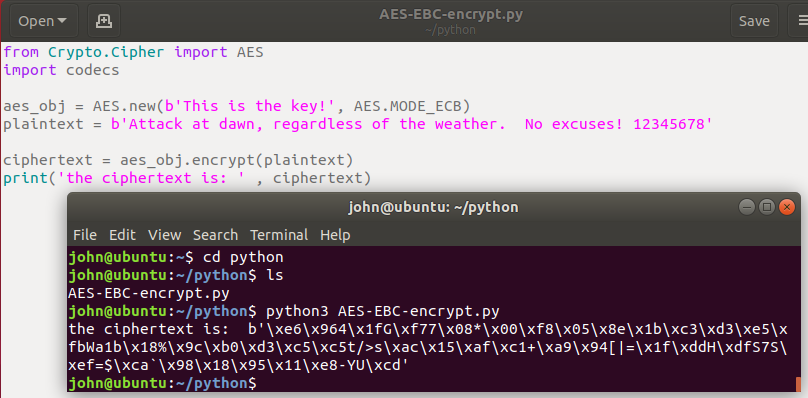
There are many ways to write and use Python.

* Use a terminal Command Line Interface (CLI) and a text editor
* Use Python’s built-in development environment, called Idle
* Use a full-feature development environment

### Terminal and Text Editor

This is the simplest method, and the easiest to use when you move between environments. In this method, I have the terminal with the Python interactive prompt and the text editor side by side. I will often test a line or two of code in the terminal and then paste it into the text editor for safe keeping. Other times I will type a few lines of code into the text editor and then test them in the terminal.  


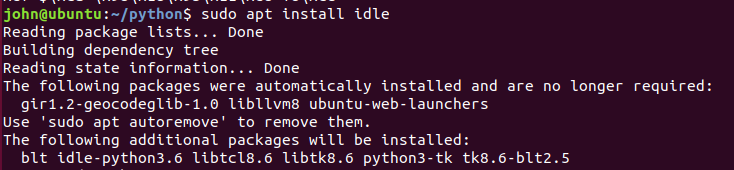
Once I have enough of the code written to have a testable script, I run the entire script in the terminal and begin debugging the errors.



It is important that the working directory of the terminal is the same as the directory where the file is stored; that way you don’t have to specify a path.

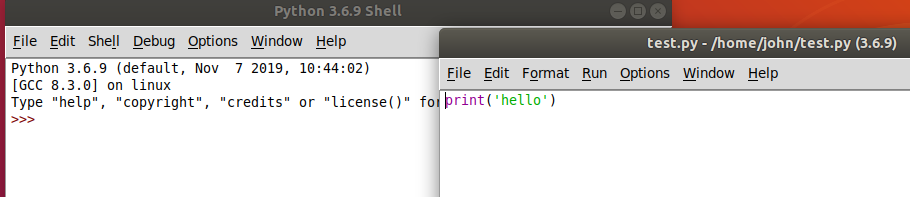
In Linux, the gedit text editor understands Python and will color code your scripts. This handy for spotting errors like forgetting to close quotes. The default editor in Windows, notepad, does not do this.

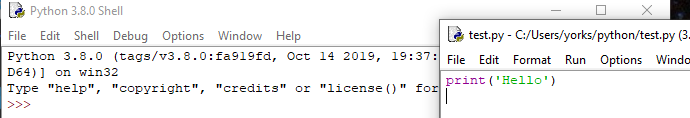
### Idle

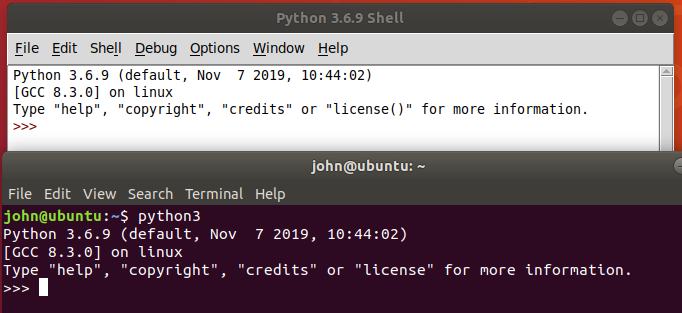
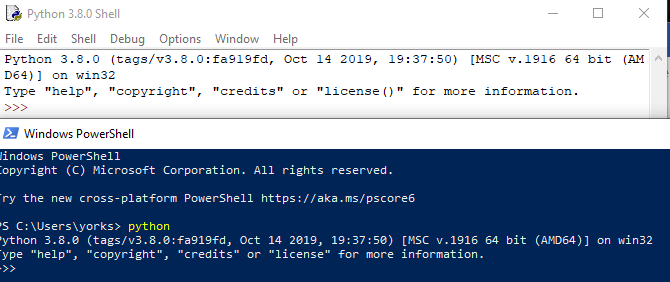
Idle is a simple development interface that comes with Python. In Windows it is installed by default when Python is installed. In Linux, you can install Idle with apt. <https://www.techrepublic.com/article/how-to-install-the-idle-python-ide-on-ubuntu-desktop-19-10/>   
sudo apt install idle  
  
<snip>

In Linux, make an Idle desktop shortcut by clicking on Activities in the upper-left corner. Type idle, right-click on the IDLE (using Pyth..) icon, and select Add to Favorites.  


Idle works the same way in both Windows and Linux, so I will use it in all the examples and exercises.

Idle in Linux (Ubuntu 18.04).  


Idle in Windows 10.  


When you open Idle directly (not by opening a Python script in Idle first) you get the Python Shell window.  
Linux  
  
Windows  


The Idle Shell window is the same as the interactive Python prompt in a terminal, except that it has options along the top.

When you use File > New File in the Idle Shell window (or open an existing file) you see Idle’s Text editor.  
A screenshot of a social media post

Description automatically generated

You can jump back and forth between the shell and editor just like you would between a terminal and text editor. When you run a script from the editor, it will run in the shell. For more information, see Chapter 3 of *Cracking Codes with Python*.

Idle is a minimal development environment, but it is useable. The most annoying thing for me is that you scroll through previous commands in the Shell window using Alt-P (previous) and Alt-N (next). I would much rather use the up and down arrows.

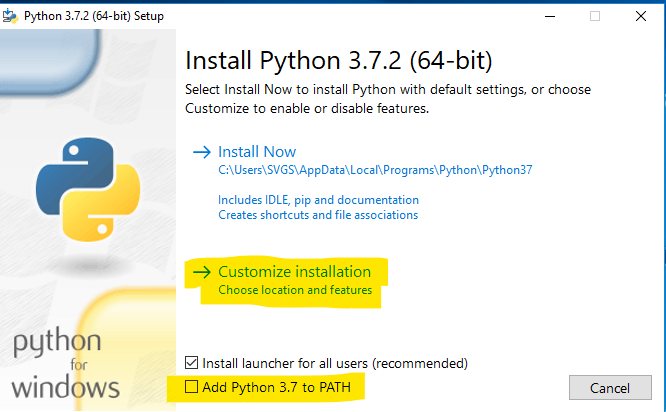
### Full-featured Development Environment

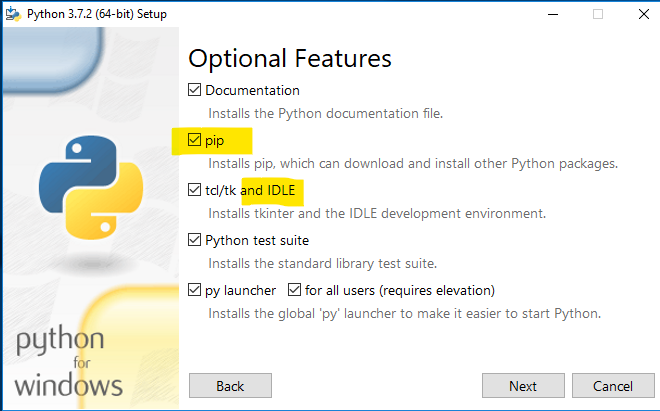
If you do a lot of work in Python, you will undoubtedly select a better Integrated Development Environment (IDE). You can find several by searching the Internet for “python best IDE”. It takes a while to learn to use a complicated IDE; we don’t have much time to spare so we won’t use a full IDE in this class.

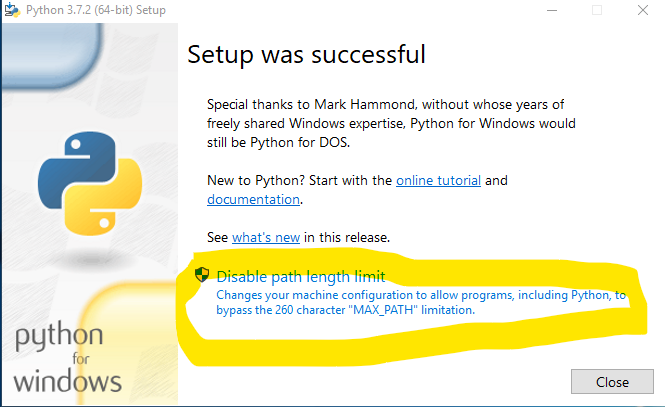
Many of the security people I know use Visual Studio Code (<https://code.visualstudio.com/docs/languages/python>), which is a free download. They like it because it supports the other languages they use, so they only need one IDE. Another popular IDE is Spyder, which comes with the Anaconda addition to Python (<https://docs.anaconda.com/>).

## Installing Python on Windows

Windows users can download Python from <https://www.python.org/downloads/windows/>. This will also install the Idle editor for Python. When you install Python on Windows be sure to check the box to add Python to your path!! It will make life much easier. It is very handy to just type “python” in a terminal and have it work, rather than typing the entire path.



In the custom installation make sure that pip is selected. 

Window has a limit of 260 characters in a file path. Since Python installs itself on a long file path (like C:\Users\SVGS\AppData\Local\Programs\Python\Python37\) the limit can cause problems.  


## For Turn In

Please answer the following questions.

1. What operating system did you use?
2. Did you create your python scripts in a text editor or an IDE? Which one?
3. Can you run this simple script?  
     
   print('Hello World')