

Introduction to IDLE and Python

Summary of what you will do:

This lab is to get you started on programming. Programming is a practical endeavor that requires practice and experimentation to learn. You will start up IDLE and run some simple programs. You will record some simple observations on this handout as you go through the instructions below. You will turn in this handout at the end of lab.

If you have extensive programming experience, I encourage you to go beyond the examples here, trying more complicated exercises. (See suggestions at the end and try out some of the Visual library.)

PYTHON:

Python is a programming language that is powerful and easy to learn. There are a large number of libraries or prewritten programs that you can call upon. In particular, we will use the **visual** library, which allows us to simulate 3D objects and call upon mathematical methods, without having to do all of that work ourselves.

A program in Python is a text file (or more than one file or sometimes not text, but we will not worry about that.) This text file is just a collection of lines of text that are read through and interpreted by the computer as commands (which have meanings such as “print this” or “draw that” or “multiply two numbers and put the result there” or “do the following commands 1000 times”.)

(For more about the choice of Python for this course, see the PHY307 web page, physics/courses/PHY307, and click on “Languages” near the bottom of the home page.)

IDLE:

IDLE stands for Integrated DeveLopment Environment. It is the interface that you will use as the base for writing and executing Python programs. It contains (among other things):

- An interactive “shell” – this executes your Python commands immediately.
- A program editor – this allows you to put a set of commands together into a program file.
- The capability to execute (“run”) your program files.

STARTING IDLE AND AN INTERACTIVE “SHELL”:

OK, here we go.

1. You need to log on to your workstation. Enter your CMS/SUnix ID and password at your PC. The work that you create can be saved to your CMS/SUnix account.

- To call up IDLE, look under the “Start” menu on your PC station. Look under the “Programming” item, then “Python”. There will be an icon for “VPython (IDLE)”. Select that item.
- IDLE will open up a single window. The two types of windows we will be using are *shell* windows and *file* windows. Shell windows have some words in them and the Python prompt. The Python prompt looks like “>>>”. A file window will look empty.

IS YOUR FIRST WINDOW A SHELL WINDOW (yes/no)? _____

If your first window is a shell window, call up a file window for fun. Do this by going to the “File” menu at the top of your shell window and select “New Window” (note you can get this action by holding down the control key while striking the “N” key – this is written as ‘ctrl-N’.) Once you open the file window, close it (by clicking on the “X” in the upper right.)

If your first window is not a shell window, call up a shell window. Do this by going to the “Run” menu at the top of your file window and selecting “Python shell”. Then close your file window (by clicking on the “X” in the upper right.)

- Now that you have a shell window, give the interpreter some commands to execute. Enter a command after the Python prompt “>>>” and then hit the “Enter” key to have the command executed. Remember, the interpreter just does what you tell it to, right away.

Try the following commands and record what happens:

COMMAND	WHAT HAPPENED, COMMENTS
print “Hello, out there!”	
print 3	
print 2+7	
print 2*7	
print “2 times 7 is“, 2*7	
print “2 times 7 is“, 15	
print “a”	
print a	
a=3	
print “a”	
print a	

The last few examples above demonstrated two things: ‘string literals’ and variables:

* A string literal is just a set of characters (letters, numbers, punctuation, etc.) that is taken literally and treated just as a set of characters.

* A variable is a place for holding **objects**. Above, the command **a = 3** created a variable called **a** which holds the number object **3**. When you executed **print a** the first time, the interpreter did not know what to do, as the variable **a** did not exist. After **a** was created as a variable and given a value, you could print out the value of **a**.

5. Try the interactive shell a bit more, using the same procedure with the following commands:

a=3	
b=5	
print a+b	
print a*b	
c=range(1,5)	
d=range(8,12)	
print c	
print d	
print c+d	

These examples start to show you that not all variables hold numbers. The **range** function here makes objects that are **lists**, which are put into **c** and **d** using “=”.

Now *experiment!* Enter 4 commands and record and discuss what you did below:

WRITING AND RUNNING PROGRAM FILES:

Interactive mode is useful for trying ideas out, but it is tedious to write a real program this way. You want to be able to build up sequences of commands and store them in a file, so you can use the same commands repeatedly. This sequence of commands is the program file. So you will now make a program:

6. Open up a program file window. Do this from any IDLE window by going to “File” and selecting “New”.
7. *The very first thing you should do is SAVE this empty file to a file with a new name. This is important for not erasing old work and for backing up your new work.* Save your (currently empty) program by going to the “File” menu and selecting “Save as ...”. Give the program a name (e.g., **Aug28_Test.py**) and save it to your CMS/SUnix directory. [Desktop or drivename associated with your CMS user name; the I: or K: drives.]
8. Create a short program. Enter the following lines, hitting the Enter key at the end of each line. (The file window has editing capabilities, allowing you to use up/down & left/right arrows or to click anywhere in your file and also colorizes your program):

```
a = 3
b = 9
c = range(1, 10)
print a, “+”, b, “is”, a+b
print “A list of the numbers from 1 to 10 is”, c
d = range(a, b)
print “A list of the numbers from”, a, “to”, b, “is”, d
```

9. *SAVE YOUR PROGRAM* by selecting on “Save” under the “File” menu.
10. Run your program. Look under the “Run” menu and select “Run program” *OR* just hit the “F5” key at the top of your keyboard.
11. What happens? Note any errors and try to fix them by editing the program and trying to execute it again, recording what happens below in this box:

12. Modify your program. *Make it more interesting. Ask around for other commands to try.* Print it out. To do this, you will need to
 - a. Select the whole program by finding “Select All” under the “Edit” menu (or by holding the “Alt” key and hitting “A”; “Alt-A”.)
 - b. Copy your selection using “Copy” under “Edit” (Ctrl-C).
 - c. Opening a text editor (say Microsoft Word.)
 - d. Pasting your selection into the editor (“Paste” under the “Edit” menu or Ctrl-V, usually.)
 - e. Selecting “Print” under file or clicking on a printer Icon (within the text editor or word processor.)
13. Restart IDLE (exit, then go to the “Start” menu under Windows, again, finding the “IDLE” selection.)
14. Run a demo program: go to “Open” under the “File” menu (or “Ctrl-O”). Double-click on the “bounce” program. Run this program by hitting the F5 key. Record what happens in this box:

15. Finally, try your own VPython program, using the visual library. Open up a shell and enter the following commands to call upon the visual library and to execute a command from that library:

```
from visual import *  
box(color=color.green)
```

What happens? Can you modify the scene using the mouse while holding down the left or right buttons? Try dragging the mouse while holding both buttons. [WHEN DONE, turn this in with your program printout.]