Introduction to Python/VPython

I. EXERCISES FOR THURSDAY AUGUST 28

A. Getting Started

- Log in.

- `Programs` ⟶ `Programs I-P` ⟶ `IDLE for VPython`

You should initially get a window with a black background and then a window with a white background. You can minimize the window with the black background; it is only there because of some things that have to be set up to make Python and VPython work on the Bucknell computer network. The window with the white background is the IDLE *editor*; you will write your programs in this window.

The first line of your program should be:

```
from visual import *
```

This tells Python that to load the *module* called VPython that will enable you to do 3D animations very easily. The VPython module also includes another module with some mathematical functions. (There are many, many Python modules for a wide variety of computing tasks.)

B. Python as a Calculator

Type the following commands one-by-one into the IDLE editor window. Note that commands are case-sensitive. After each command with a `print` in it, execute the program you have written. You can do this from the pull-down `Run` menu, or you can just hit the `F5` key. The first time you execute the program you must name it and save it; save it in your own filespace. (Anything saved to the desktop will be lost when you log out.) The output of your program will appear in a new window with the title `Python Shell`. It's best to write all of your "input" statements in the program you are writing in the IDLE window — leave the `Python Shell` window for outputs.

- `print 6+5`

- `print 2**3`
- `print a-b` (Erase this command when you are done.)
- `a = 3`
- `b = 5`
- `print a-b`
- `print "Hello World!"`
- `print "a+b"`
- `print (a+b)*(a-b)`
- `c = vector(0,1,0)`
- `d = vector(1,0,0)`
- `print c + d`
- `print dot(c,d)`
- `print cross(c,d)`
- `print mag(c+d)`
- `print sin(0)`
- `print sin(pi/2)`
- `print sin(pi)`
- `print 4.0/3.0`
- `print 4/3`

In the last command, all inputs were integers, so Python assumes you want an integer output. If you don't like the distinction between 4 and 4.0 that you saw in the last two commands, you can get rid of it with the line at the beginning of your program:

```
from __future__ import division
```

You may want to have this command in every program you write.

C. Loops

The first line in a loop is a *conditional statement*. Everything on the following indented lines will be executed repeatedly as long as the condition is true. **The indentation is essential.** IDLE will automatically indent the first line after a conditional statement, but you must "unindent" when you get to the end of the statements that are to be repeated.

-
  ```
  i = 1
  while i<11:
      print i
      i = i+1
  ```

-
  ```
  i = 1
  while 1:
      print i
      i = i+1
  ```

You can break the infinite loop in the last example with a `CTRL-C` (simultaneously pressing the `CTRL` and the `C` keys).

D. 3-D Geometric *Objects*

Start a new program. (Don't forget the `from visual import *` line.)

-
  ```
  ball = sphere(pos=(0,0,0), radius = 0.5, color = color.red)
  ```

-
  ```
  wall1 = box(pos=(-10,0,0),length=.1,height=10,width=5,color=color.blue)
  ```

-
  ```
  wall2 = box(pos=(10,0,0),length =.1,height=10,width =5,color=color.blue)
  ```

Execution of the preceding commands should produce a third window with a 3-D view of a red ball between two blue walls. You can can change the view with your mouse buttons: when the cursor is over the display window the middle mouse button will zoom you in and out when you move the mouse, and the right mouse button will change the viewpoint. **Don't minimize the 3-D graphics window!** There is some bug in either VPython or the Bucknell network that doesn't like this. You can close the window, but don't minimize it.

You change the *attributes* of the *objects* from within the program. For example:

- `ball.pos.x = ball.pos.x + 5`

E. Animation: Combining 3-D Objects and Loops

Add the following lines to your program with the ball and walls.

-
  ```
  v = 1
  dt = .1
  while 1:
        rate(100)
        ball.pos.x = ball.pos.x + v*dt
  ```

II. DOWNLOADING PYTHON FOR YOUR COMPUTER

- Python (download this first): `http://www.python.org/`
- VPython module: `http://vpython.org/`

III. ADDITIONAL DOCUMENTATION

We will be using a very small subset of the Python programming language, but if you're interested in learning more there are many, many books and online tutorials that are available. Here are just a few:

- Documents available on the official Python website: `http://www.python.org/doc/`
- Reference manual for VPython (which is also available as Help once the VPython module is loaded): `http://vpython.org/webdoc/visual/index.html`
- *Learning Python*, Mark Lutz (O'Reilly, Sebastopol, CA, 2008)
- *Python Programming: An Introduction to Computer Science*, John Zelle (Franklin Beedle & Associates, 2003)