Python: Getting Started

Installing Python

This section describes how to install the Python environment onto a Windows PC and a Macintosh. For advanced Macintosh and Linux users we show how to invoke python from a terminal session. Most students will only be interested in using Python via the *IDLE* application and can safely ignore the Advanced Macintosh and Linux section below. For this course we will assume that the student is using *IDLE*.

Beginner Windows and Macintosh

Python is available for download free of charge from the Python website at https://www.python.org/downloads/. From this page you can download the appropriate Python 3.4.2 installer package for your system (probably either the Windows installer or the Macintosh OS X universal installer). Make sure **not** to download the Python 2.7.9 installer as Python 3.3 will be used this quarter, **not** Python 2.7. Once the download is completed simply open the installer and follow the directions given.

To invoke a Python session in Windows simply click on the Start button, expand the Python 3.4 folder and click on *IDLE (Python GUI)*. In Mac OS X look in the /Applications/ directory for the Python 3.4 folder. In this folder double click on *IDLE*.

Advanced Macintosh and Linux

Macintosh OS X and most Linux distributions have a Python run-time environment pre-installed. To invoke the run-time environment simply type **python** in a terminal session. To open a terminal in Macintosh OS X, double click on *Terminal* located in /Applications/Utilities/. In most Linux distributions you can launch a terminal session from somewhere in the tool bar.

If you choose to use Python in this fashion you must write your programs using a plain text editor, examples of which include *emacs* and *vi*. To execute your program from the command line type **python myprog.py** where myprog.py is the name of your program's source code file. Alternatively you can include the line

```
#! /usr/bin/env python
```

at the top of the source file and give the file "execution" permissions (from the command line type **chmod +x myprog.py**) and then you can execute your program by typing **/path/to/myprog.py** in the terminal (where /path/to/ is the path to myprog.py in the filesystem).

Using IDLE

In this section we will introduce the major components of *IDLE* and interact with the top-level. Then we will focus on how to write commands in a source file and then run that file.

The Top-level

The first screen that comes up once you launch IDLE is the top-level. In the top-level you can interact with Python directly. (Sometimes this is referred to as "interactive mode" rather than the "top-level".) You will notice a prompt that looks like

```
>>>
```

at which you can type commands for Python to evaluate immediately. If the command involves a **print** statement then Python will print the results on the following line and then another prompt.. If the command is an assignment or something that does not produce output then the following line will simply be another prompt.

Consider the following simple example:

```
>>> number_of_cats=22
>>> owners_name="Peggy"
>>> print owners_name, "has", number_of_cats, "cats."
Peggy has 22 cats.
>>>
```

In this example we first define two variables, **number_of_cats** and **owners_name**, and assign values to them with the assignment operator, =. The third line uses the **print** function then prints out the message "Peggy has 22 cats.". Notice that the print function printed the contents of the variables **number_of_cats** and **owners_name** rather than the names of the variables. Instead, the variables were first evaluated and then their value was printed. Variables will be discussed in more detail below, but first we will turn our attention to creating and editing source files in *IDLE*.

Editing a source file

You will rarely be writing programs directly in the top-level because it does not provide a mechanism with which to save the program. Instead you will be using a plain-text editor (or programmer's text editor). Such an editor is available in *IDLE*. To get to it and create a new file select **File → New File** from the menu bar. This will open a blank window in which you can write your Python source code. To save the source code you have written simply go to **File → Save As...** just as you would save a document in a word processor or photo in a image editing program. To open an existing file, select **File → Open** from the menu bar as you would if you were opening a document in a word processor. By convention, Python source code files end with .py extensions and *IDLE* does not add this extension automatically. So you will want to save your programs with names like myprogram.py (with the .py extension). To run the program that you are writing you use the **Run → Run Module** option from the menu bar. Alternatively, you can press F5 on the keyboard.

Try writing a simple program.

```
name = raw_input("Enter your name: ")
age = raw_input("Enter your age: ")
print "Hi", name, ". You claim to be", age, "years old."
```

When running this program it will pause at each line that calls the function **raw_input()** where it waits for the user to type something. It will proceed once the user presses the enter key on the keyboard. A typical run of this program will look something like this:

```
>>> ========================== RESTART ==========================
>>>
Enter your name: Rob
Enter your age: 45
Hi Rob . You claim to be 45 years old.
>>>
```

For the most part this quarter you will not need to be writing your Python programs from scratch, but will be modifying simulators provided to you.