Introduction to Programming in CoCalc and Python

Starting Out in CoCalc

Go to `http://cocalc.com` and enter your username and password. You will next see a screen that is your home directory. To start working with Python, click on "+New", enter a name for your file, and select a new Jupyter notebook. Be sure the right-hand side of the top says "Python 2 (SageMath)." If it does not, click on the "Kernel" menu and select "Python 2 (SageMath)."

Working in Jupyter notebooks

Commands in Jupyter notebooks are listed in cells. These can be executed by pressing "shift-enter" on the keyboard. After a cell is run, either the cursor moves to the next existing cell or a new cell is created.

Cells come in three types: code, markdown, and raw. New cells are assumed to be "code cells" and are used for programming. Markdown cells are used for writing text for documentation and markdown commands may be used (see https://beegit.com/markdown-cheat-sheet).

Programming basics

A program consists of a sequence of lines of commands. The Python language executes each line in the order that it appears. So, the program runs by stepping line-by-line through the code. For example, consider the following program.

```
# first program
print "Hello, World"
name = "Jane Doe"
print "Hello, " + name
```

The first line of this program is called a "comment" and is ignored by Python because the line begins with a #. The second line causes Python to print "Hello, World" on the screen. The third line stores the string "Jane Doe" in the variable named "name." The final line causes Python to print "Hello, Jane Doe" on the screen.

****For your first attempt at programming, start CoCalc, open a new Jupyter notebook, and then type and execute the above program.**

Key programming elements

To learn any new programming language, there are a few key elements of the language for which you should look. Here is a short list.

- Variable types – what kinds of variables are allowed in programs
- Assignments – how do you assign values to variables
- Flow control – what commands can you use to control the flow of a program
- Input/Output (I/O) – what commands are needed to print to the screen and/or get input from a user
- Functions – how do you write small bits of code that you can reference in a larger program

For advanced programmers, there are many other elements of a language for which to look. These five elements, however, will get you writing code quickly and will allow you to write powerful programs.

Variable types

Python contains several variable types, but the basic types are numbers, strings, lists, tuples, and dictionaries. Examples are given in the following table.

Type	Example
numbers	3.14, 2, 3+4j
strings	'hello', "world"
lists	[1,2,3,4], [[1,2],[3,4]]
tuple	(1,'hello',4)
dictionary	{'a':'stuart','b':'kevin','c':'gru'}

Note that for a list, you can access an element of the list by using the following approach.

```
a = [1,2,3,4]
a[0] # returns the value 1
a[1] # returns the value 2
```

Assignments

Python allows for a few basic assignment operations. Here are some examples.

```
a = 3.14159
b = "Good Luck"
```

To assign more than one variable a value in one line, you can use the following formats:

```
a, b  = 3.14159, "Good Luck"

[c, d] = ["hello", 2.71]
```

These commands are called "unpacking assignments" in python and result in the following assignments

```
a = 3.14159
b = "Good Luck"
c = "hello"
d = 2.71
```

Flow Control

Like most other languages, python has several commands for controlling the flow of your program. The critical flow control commands are: for, if, and while. Here are examples of the syntax.

The "for" command requires an index variable and can be indexed by a list or using the "range" command. The "range(a,b)" command produces a sequence of numbers starting with a and ending with b.

```
for i in [‘a‘,‘b‘,‘c‘]:
   print "The letter is:"
   print i
print "The end!"
```

IMPORTANT: Python relies on indentation to tell it which lines are inside the "for" loop. So, the first two print statements are inside the loop, but the last one is not.

The code above will produce the following output:

```
The letter is:
a
The letter is:
b
The letter is:
c
The end!
```

Here is another "for" loop.

```
for i in range(0,3):
    print 2**i
```

Note that the "**" tells python to raise 2 to the power i.