**Python Math Concepts**

**By: Rohan Karunaratne**

**Email: rohanhawaii@gmail.com**

**Find me on Linked In @ Rohan Karunaratne**

**© 2016 Rohan Karunaratne All Rights Reserved**

Math concepts in python can be confusing at first, but with a little help from this book you can master the basic concepts needed to perform mathematical functions. Start by opening python (the 2.7 and 3.5 versions work). You will first see the shell. Press File, then New File. Program in this window. To do mathematical functions in python, you will need a library called numpy. To import numpy, you could go online, or just go to the terminal and write import numpy as np. At the beginning of each program, write import numpy as np. You will not need to do this for the first few programs, but as it gets more complex, you will need to start activating and using this library. If you have not programmed at all, you should start doing basic Programs located in this book. The introductory paragraph is located below. For those of you who are experienced in python, skip the paragraph and go right to example 1. Here is the formatting of this book.

First sentence: Introduction to lesson

Italicized paragraph: code

Paragraph after code: code breakdown / explanation

Introductory paragraph:

Lets first begin by understanding what a print is. A print command is issued when the programmer want to convey or print some message to the user. This can be done by using the print command. The print command is stated below:

print ("1")

The command above tells the computer to print the number "1".

Next let’s talk about conditionals. Conditionals give us the ability to check conditions and change the behavior of the program accordingly. This statement gives the computer the power to make decisions based on what we want it to do. Some examples of conditionals are:

IF statements

IF ELSE statements

IF ELIF statements

IF ELIF ELSE statements

FOR statements

and so on......

Most of these statements are covered in this book, but some you will have to search on the internet.

Let’s conclude with a briefing on loops. Loops let us run code for long amounts of time without us having to constantly run the code over and over again. Some examples are:

FOR I IN RANGE statements

IF TRUE statements

WHILE statements

WHILE NOT statements

WHILE TRUE statements

Let’s start with a basic example on how to add two numbers.

Ex 1: Input and adding 2 numbers

*number1 = input ("Give me a number to add:")*

*number2 = input ("Give me a second number to add:")*

*total = number1 + number2*

*print(total)*

So in the first line, you assign the variable "number1" a number. This number comes from the input which we ask "Give me a number to add:"

Let’s say the user puts in 2 for the first number. Next the variable "number2" is assigned to the value from the question "Give me a second number to add:". Let’s say the user puts in 3 for the second question. The third line defined a new variable called "total". The total is the sum of number1 plus number2. So the computer would do the calculation 2 plus 3 which is 5. Finally, we ask the computer to print the "total". If you want the input to be only numbers, include *int* before the variable input. Notice how we don’t use quotation marks to print variables. This is because we don’t want to ask or print any variable, we simply want to print the variable.

You can repeat these steps for subtraction, multiplication, and division just by replacing the "+" in example one with the desired operator.

The next concept we will work is comparing numbers. You can compare numbers by using the "<" and ">" operators. This example will also show how to use if statements.

Ex 2: Comparing 2 numbers

*number1 = input("Give me a number:")*

*number2 = input("Give me another number:")*

*if number1 > number2:*

 *print(number1, " is greater than “, number2)*

*if number1 < number2:*

 *print(number2, " is greater than ", number2)*

*if number1 == number2:*

 *print(number1 " is equal to " number2)*

In the first two lines of code, we again identify what number1 and number2 are. Then, in the first "if" statement, we say that if the first inputted number is greater than the second inputted number, we basically print say that in words. In the second "if" statement, we say that if the first inputted number is less than the second inputted number, print it we print it in words. Then in the last "if" statement, we say that if number1 is equal to number2, print it in words. Remember to put double == signs. This means "equal to". To create a "not equal to" statement, we use !=.

You can make multiple "if" statements if you want to sort a group of numbers. Also note that there is a semicolon after the "if" statements. NEVER forget it in python or you will get many syntax error and they will drive you nuts. Usually in python, during a syntax error there will be a red highlighted area. This is usually where the syntax error is located.

Next I will show you how to compute the average of 5 numbers.

Ex 3: Computing average of 5 numbers

*number1 = input ("Give me a number:")*

*number2 = ("Give me a second number:")*

*number 3 = ("Give me a third number:")*

*number4 = ("Give me a fourth number:")*

*number 5 = ("Give me a fifth number:")*

*total1 = number1 + number2 + number3 + number4 + number5*

*total2 = total1 / 5*

*print("The average of the 5 numbers you inputted is ", total2)*

The first five lines of code are asking for the 5 input numbers. You should be pretty familiar with these commands. For more clarification, look back at examples 1 and 2. For the variable total1, you add the 5 input numbers. Then total2 is total1 divided by 5. If you know in math, when averaging, you add the numbers before dividing them. This is why we did the adding command before the dividing command. Then we printed total2 or the average.

Now, I will be showing you how to count numbers in python.

Ex 4: How to count by 1's, 5's and 10's in python and introduction to the for i in range loop

*print("Counting by 1's:")*

*for i in range(0,11,1):*

 *print(i, end=" ")*

*print("Counting by 5's:")*

*for i in range(0, 100, 5):*

 *print(i, end=" ")*

*print("Counting by 10's:")*

*for i in range(0, 100, 10):*

 *print(i, end=" ")*

So, if I explain the first loop you will pretty much get the rest. So we say that we are counting by 1's in the print command. Now it gets a little tricky. For i in range means for every integer in the range...... There are three numbers after that. The first number is where the range starts, the second is where the range ends and the last is what increments are you counting at. So in loop two, you start at zero, end at 100, and count by 5's. The third line is where we print the counting and at the end, we add a space. So just by fiddling around with three numbers, you can create different counting sequences. I would like to point out something in the first loop that also applies to the other loops. If you try to run the first loop, the notice how it stops at 10 but the loop says 11. This is because the for i in range loop is not a greater than or equal to loop. I mean that the loop is looking through the numbers like 9 is in the loop 10 is in the loop, but it doesn’t count 11 because it is not less than 11. Even in the second loop, It counts to 95, but not 100, and even in the third loop, it counts till 90, then stops. So if you want to count in increments of 3 and end at 99, you need to go to 102 to get the number 99.

I will now explain how to add values to a variable.

Ex 5: How to add values

*Count = 1*

*Count +=1*

So in line 1, we define what value count has (1). In python, we don't say Count = +1, we say Count +=1. If you wanted to add 5 to Count, you would write Count+=5.

Ex 6: How to subtract values

*Count = 5*

*Count -= 3*

So same with example five, you write count -=3, not count = -3

Now let’s get into some trigonometry with Sin., Cos., and Tan. The formulas for Sin, Cos, and Tan are below.

Sin = opposite/hypotenuse

Cos = adjacent/hypotenuse

Tan = opposite/adjacent

*y = math.sin(theta)*

*y = math.cos(theta)*

*y = math.tan(theta)*

Theta is another word for angle and by manipulating theta, you can create different types of waves and designs. These designs are called Lissajou patterns. Let’s go to this next example.

Ex 7: Manipulating theta

*for i in range(-400 , 400, 1):*

 *theta = (i \* 0.01)*

 *y = math.sin(theta \* 15.7)*

 *# theta \* 3*

 *#theta \* 2*

 *#theta + 3.14*

 *#theta + 1.57*

 *#theta \* 6*

 *#theta \* 12*

 *#theta (3.14 \* 5)*

 *#((theta \* 9) + 0.785)*

 *#theta \* 2, theta \* 3*

 *#(theta \* 3) + (3.14/4)*

 *#theta \* 15.7*

 *#theta \* 5*

Again we use a for i in range loop and we start at -400 and count by increments of 1 to get to 400. Then we assign theta to and integer \* 0.01. Then you have the math.sin(x) function with theta in replace for "x". After that, you see 12 different manipulation of theta that begin with #. Open python and use the code above. First try sin(theta), then try cos(theta). Later, use the different examples and change theta and you will see the different Lissajou patterns. Try creating your own theta manipulations.

Now I will explain about radians in python. A radian is a different way of measuring degrees. There are 2π radians in a circle. The next code I will show you can help you use radians to create sin and cos waves, just by manipulating radians.

Ex 8: Radians

*for angle in range(360):*

 *y = math.sin(math.radians(angle))*

 *turtle.goto(angle, y \* 80)*

 *turtle.goto(0,0)*

So we begin with a for angle in range(360) loop where we basically find the integer in 360 degrees. Then we set the math.sin function to a math.radians function. We then tell the turtle to go to the angle of y times 80 and once that is done, go back to 0,0.

Thank You for reading this book and I hope that now you understand math concepts in Python. Be sure to check out my website (programinator.weebly.com) for more coding activities. I would like to thank my friend Martin Liu for helping edit this book. And Keep Calm and Code on!

**© 2016 Rohan Karunaratne All Rights Reserved**