

LAMC Intermediate Group

October 4, 2015

Oleg Gleizer

oleg1140@gmail.com

Intro to Python

On the positive side:

- Python – an industrial-grade yet easy to learn programming language. If the only thing you know is coding in Python, you already can get a job.
- Takes much less code than C, C++, and Java to program the same task.
- Runs without changes on all major platforms. Moving a Python program from Linux to Windows most often takes no more than copying the file from one computer to the other.
- Python programs can use and can be used by programs written in C/C++.

On the negative side:

- For large computations, Python can be much slower than C/C++.

The book: *Hacking Math Class with Python* by Peter Farrell,
<http://www.farrellpolymath.com>

IDLE – Integrated DeveLopment Environment, a Python shell

The Turtle Module

Problem 1 *Type in the following commands. Hit ENTER after entering each command. See what happens.*

```
>>> from turtle import *
>>> fd(100)
>>> rt(90)
>>> fd(200)
```

You can clear the above picture using the following prompt.

```
>>> clear()
```

Note that the *clear()* command clears the picture, but does not revert the turtle to the original position. The *reset()* command does just that.

A loop: to draw a square, do the following.

```
>>> for i in range(4):
    fd(100)
    rt(90)
```

Then press ENTER twice.

Problem 2 *In the Turtle module, draw an equilateral triangle.*

Problem 3 *In the Turtle module, draw a beautiful picture of your own.*

A Function

Let us define a function *square*.

```
>>> def square() :  
>>>     for i in range(4):  
           fd(100)  
           rt(90)
```

Problem 4 Enter the prompt `>>> square()` and see what happens.

Problem 5 Type the following few lines of code. What do you think is going to happen when you hit ENTER twice? Discuss your idea with the class before drawing the picture.

```
>>> for i in range (36):  
           square()  
           rt(10)
```

Question 1 What if we want the turtle to draw a square with a side length different from 100 units?

A Variable

Let us use the variable *side* for the purpose. Please type in the following code

```
>>> def square(side) :
```

The code continues to the next page.

```
for i in range(4):  
    fd(side)  
    rt(90)
```

and hit ENTER twice. Now you can draw squares with various side lengths.

Problem 6 *Run the following prompts.*

```
>>> square(80)  
>>> square(100)  
>>> square(120)
```

We can also change the value of a variable inside a loop.

Problem 7 *Type in the following lines of code.*

```
>>> side=20  
>>> for i in range(30):  
    square(side)  
    rt(5)  
    side=side+10
```

Then hit ENTER two times and see what happens.

Problem 8 *Now try this line. >>> square()*

*What's wrong? How can we fix it?
(The answer is on the next page.)*

The command `def square(side=100)` saves the day. Now if you set the value of the variable `side`, the program will use that value. It will use the value `side = 100` otherwise.

Problem 9 *Let n be the number of sides of a regular n -gon with a side length s . Define `plygon(s, n)` as a function of the variables s and n . Use the function to draw a regular*

- *pentagon,*
- *hexagon.*

In the Turtle mode, there exists a command `circle(r)` that draws a circle of radius r .

Problem 10 *Assume that the command `circle(r)` does not exist. Define a function `circle(r)` that draws a circle of radius r yourself.*

The Math Module

Problem 11 *Type in the following commands. Hit ENTER after entering each prompt. See what happens.*

```
>>> from math import *
>>> sqrt(81)
>>> log(8,2)
>>> cos(pi/3)
```

The problem continues to the next page.

```
>>> degrees(pi/2)
>>> floor(3.62)
>>> ceil(4.12)
```

Problem 12 Use the *Math* module to solve the following quadratic equation.

$$3.84x^2 + 8.26x - 11.76 = 0$$

Problem 13 Use the *Turtle* module to draw the first six shapes from page 19 of the course book. If you do not finish the task in class, this becomes your homework.