CS 101 - Assignment 4 - Email Scraper
Spring 2018
Algorithm Due : **Mar 5th, 2018**
Program Due : **Mar 11th, 2018**
All work submitted must be your own.
Deliverables :

You must use functions to modularize your work in a logical way.
You should use exception handling where necessary as well.
All submitted work must be your own.

Turtle Paint

A new graphics file format called turtle has been created and you are tasked writing a python program to read the data from the file and using the turtle module to draw the shapes from the file. This assignment will have you using your skills with file handling, error handling, string manipulation and using the turtle module. To make this easier you should use your newfound knowledge of functions to break the program down into smaller testable portions. You will be required to use functional decomposition in your assignments from this assignment on.

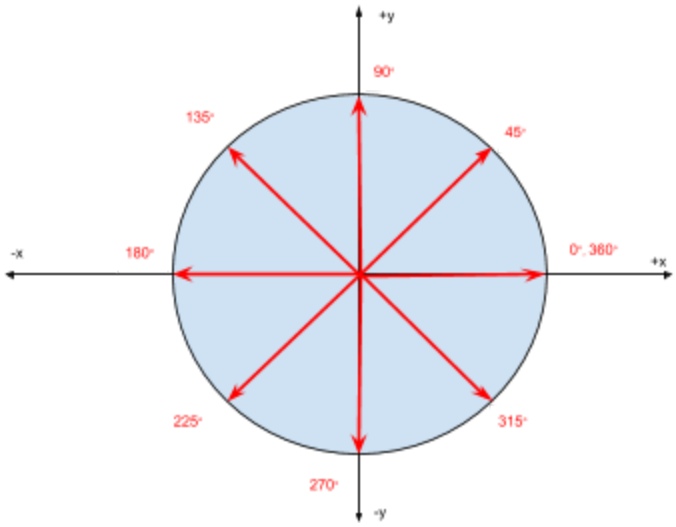
The Turtle Module.

You should spend some time getting familiar with turtle graphics. It can be quite fun. You can view the documentation for the turtle module https://docs.python.org/3/library/turtle.html.

Turtle Coordinate System

https://en.wikipedia.org/wiki/Cartesian_coordinate_system
The turtle module uses a 2d cartesian coordinate system. This means point 0, 0 is at the center of the screen. The y coordinate increases in the up direction and decreases in the down direction. The x coordinate increases going right and decreases going left.

The heading of the turtle defaults to degrees. Zero degree heading is to the right and runs counter clockwise. Try setting the heading to 380 degrees. What happens? What does it do with an angle less than 0? Try -45 degrees for a heading.



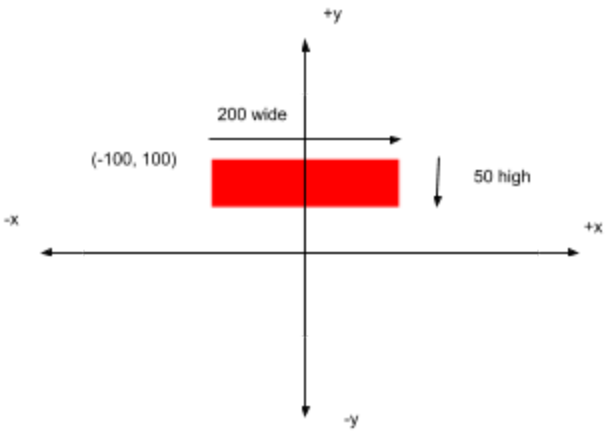
Input file

The input file contains commands to draw with turtle. Each line is a separate action or item to draw. There are three different commands; rect, circle, and line. The beginning of each line will contain the command followed by the arguments for the command. The arguments are separated by a comma.

RECT Command

The rect command is used to draw a rectangle with turtle. It has 5 arguments; x, y, width, height and color. x and y are the top left coordinates of the rectangle. The rectangle drawn should be filled in with the color given when complete.

```
rect,-100,100,200,50,Red
```



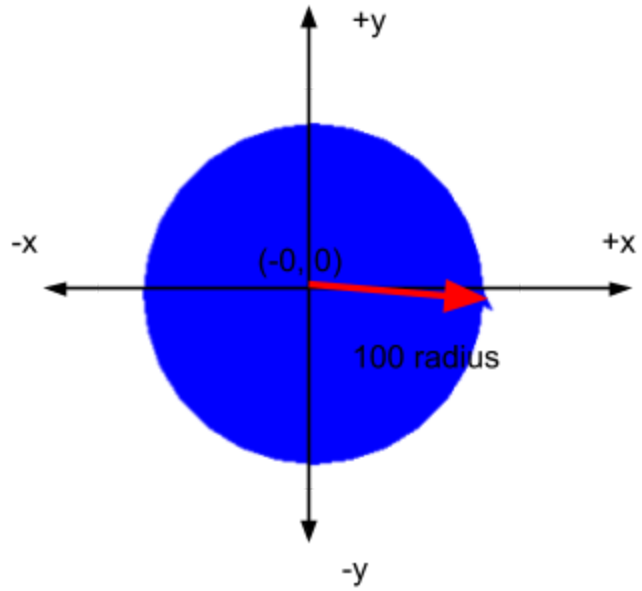
Circle Command

The circle command draws a circle. There are 4 arguments passed to the circle command the x, and y coordinate for the center of the circle and the radius of the circle and the color to fill the circle

Example Line (Color was previously set to blue)

```
circle,0,0,100,blue
```

Result (The grid labels and arrows are added for references only)



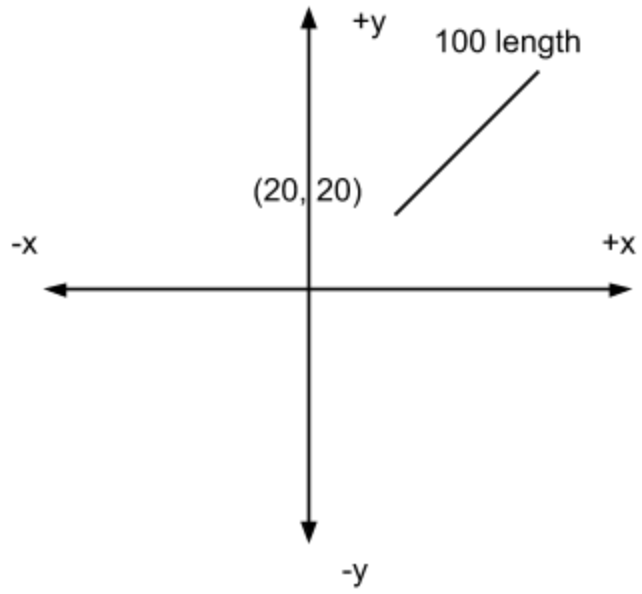
Note : You will want to take note of how turtle draws a circle and what the starting point and direction is. It does not assume the turtle is at the center of the circle. Try a circle with the heading set to 0 degrees. Then try another with heading at 45 degrees. What is the difference? You'll want to account for this when figuring out how to draw it with a given center. You may find setting the initial heading of the turtle aids in drawing consisten circles

Line Command

The line command draws a simple line. It is used to draw a line from a given x, y in a particular direction for the given length. There are 5 arguments; x, y, heading, length and color of line.

```
line,20,20,45,100,green
```

Result (The grid labels and arrows are added for references only)



Functions

You are going to need to create and use functions for this assignment. To help get you started we've created 2 functions definitions for you. You will need to use these in your solution. We've only given you the definitions, so you'll have to write the python code to get them to work. Three functions aren't enough for a program of this size. You will still need to create some of your own.

```
def draw_rectangle(x : int, y : int, width : int, height : int, color : str):
    """ Draws a rectangle of given width and height at position x, y
         Parameters :
              x : int - The x position of the top left corner.
              y : int - The y position of the top right corner.
              width : int - The width of the rectangle to be drawn
              height : int - The height of the rectangle to be drawn
              color : str - The color to draw the circle with and fill
         returns : None
    """

def draw_circle(x : int, y : int, radius : int, color : str) -> None:
    """ Draws a circle with turtle at the point x, y with the given radius
         Parameters :
              x : int - The x position of the center of the circle.
              y : int - The y position of the center of the circle.
              radius : int - The radius of the circle to be drawn
```

```
            color : str - The color of the circle and the fill color
        returns : None
    """
```

Requirements

- The program should ask the user for the file to open and draw with turtle.
    - If the file does not exist, the program should warn the user and ask for a valid filename again.
- Read through the file performing the actions specified with the arguments given. The files will be plain text with the command and arguments separated by commas.
- If the line is an invalid command, warn the user and output the line to the shell. The program should continue execution of the text file.
- If a command has too few arguments, or an integer argument can't be converted, you should warn the user in the console and output the line that caused the problem. The program should continue execution of the text file.
- After the file has been drawn the user can enter another file to be drawn. Before drawing a new image you should clear the current turtle screen.
- If the user enters `quit` for the filename the program ends.

Development notes

- When moving the turtle to a new location it will continue to draw. You need to find a way to suppress that action. If it was a pen on paper you'd pick the pen up.
- Circle and rectangle should fill when complete.
- Set the turtle speed to the fastest speed before submitting your program. While debugging you may want to slow it down if you are having trouble and want to see what the turtle is doing better.
- Break this down into small parts. Debug each part thoroughly before moving on.
    - Create a function to draw a filled circle with the center at a given point and given radius.
    - Create a function to draw a filled rectangle
    - et.

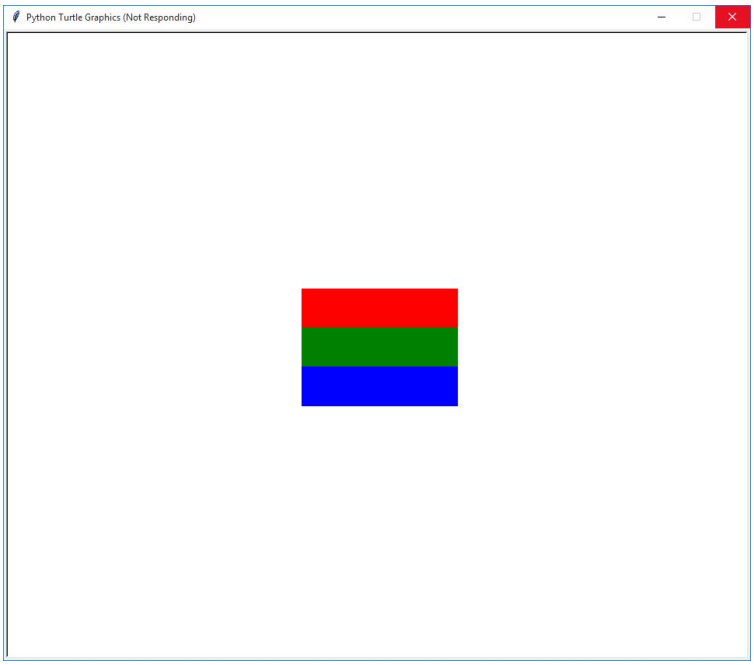
Example

```
>>> ================================ RESTART ================================
>>>
Enter File to open and draw ==> notexists.turtle
Could not open file : notexists.turtle
Enter File to open and draw ==> badcommand.tff
Bad command given could not interpret
```

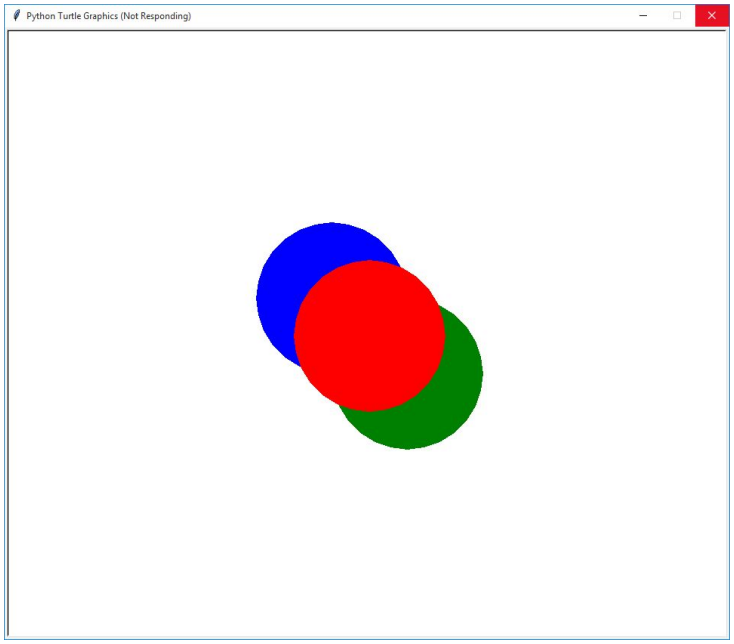
```
 bigstar 200 200 500 100
Enter File to open and draw ==> rectangle.tff
Enter File to open and draw ==> rectangle_index_error.tff

The line did not have proper number of arguments.
 rect 50
Enter File to open and draw ==> rectangle_value_error.tff

Could not convert an argument to int
 rect not_int 75 200 50

Enter File to open and draw ==> circle.tff
Enter File to open and draw ==> star.tff
Enter File to open and draw ==> pythonlogo.tff
Enter File to open and draw ==> quit
>>>
```

Artwork generated for the files given above.

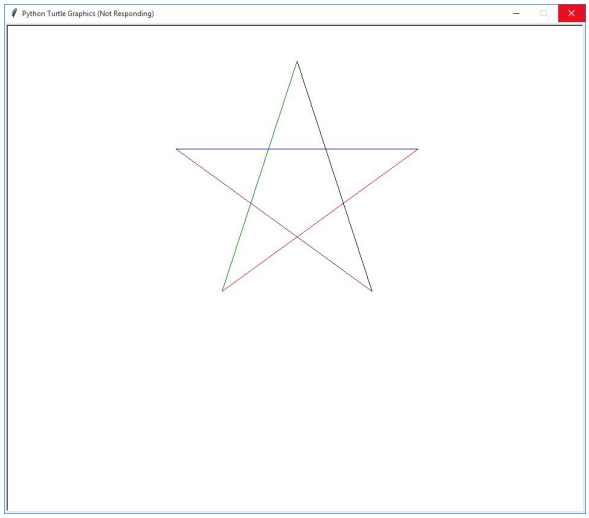
Rectangle.tff



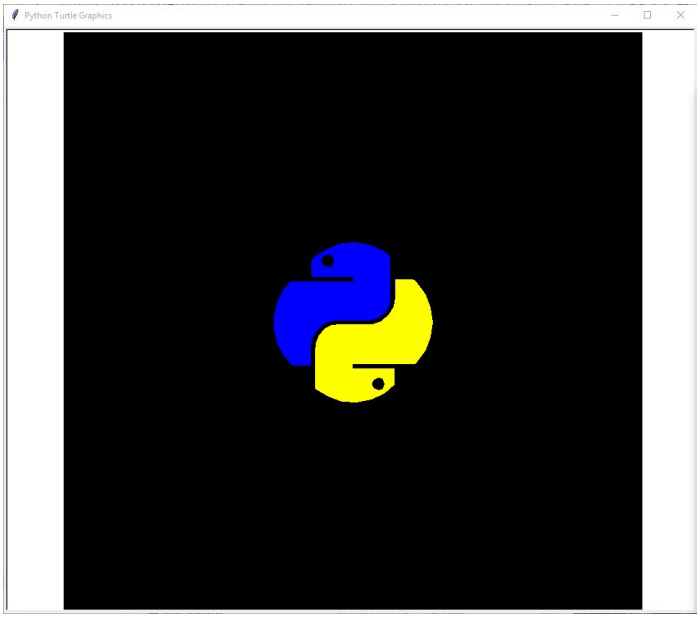
Circle.tff


Python Turtle Graphics (Not Responding)

Star.tff


Python Turtle Graphics (Not Responding)

pythonlogo.tff



Point Breakdown - May be modified as needed

Points	Requirement
5	Header
25	Readability, variable naming, comments
15	Creation of meaningful functions to help solve the problem. (you'll need more functions than you've been assigned)
10	Ask for an input file, correctly respond if the file does not exist or can be opened.
10	Draw a rectangle correctly when given in the file.
10	Draw a circle correctly when given in the file.
10	Draw a line correctly when given in the file.
5	Warn user about incorrect command given in a line in the file
5	Warn the user about incorrect number of arguments given in a command
5	Warn the user about invalid argument (cannot convert to an integer when an integer argument is required)

50 points off for programs that crash on expected input.

References

1 - Turtle Graphics Documents https://docs.python.org/3/library/turtle.html
2 - Turtle Graphics Examples http://michael0x2a.com/blog/turtle-examples