Notes for Assignment 0.

The Python installation comes with an 'integrated development environment' (IDE) called IDLE. You don't really need to use IDLE---you can create Python programs in any text editor and run them from the Terminal utility on Mac or the Command Prompt in Windows. IDLE is just much more convenient because, among other things, it combines the editing and running in a single application (that's the 'integrated' part). Python also comes with a 'graphical user interface' (GUI) package called tkinter. Most of these notes are devoted to an explanation of some of the drawing functions.

After you get Python installed and running, you should test the two programs provided. One simply prints some text to the window called 'Python Shell' in IDLE, the other draws a picture.

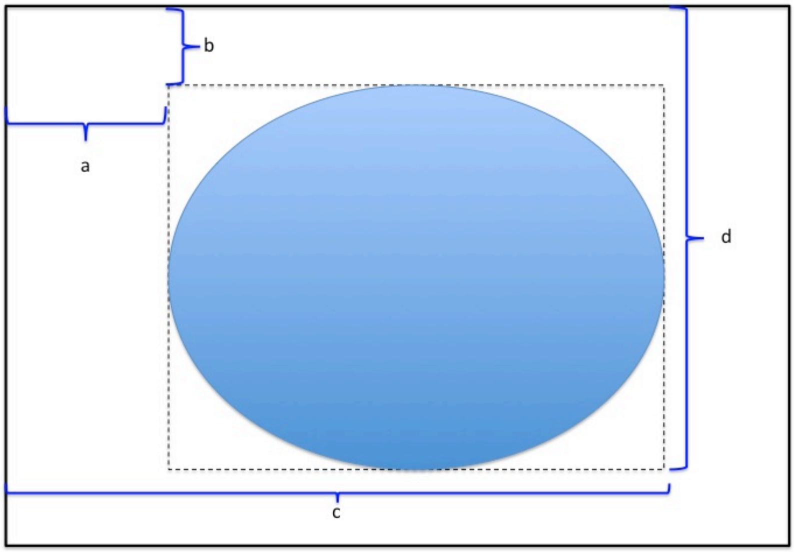
To complete Assignment 0, you will have to modify both these programs. There's not a lot to do to modify the text program---you just have to add a line or two directing the program to print some additional text. You can modify the graphics program without understanding anything about the statements in the program, but if you want to do anything interesting here, you should know about how the graphics co-ordinate system works, and what the four drawing statements in the program are doing..

The 'canvas' on which this program draws is 500 units wide and 500 units high. The coordinates of the top-left corner of the canvas are (0,0), and of the bottom right (300,300). Note that unlike the standard co-ordinate geometry you use in math class, the y-coordinates of a point increase *downward*. Also note that the coordinates of a point are *integers*.

The instruction

```
the_canvas.create_oval(a,b,c,d,....)
```

inscribes an *ellipse* inside a *rectangle* (which is not drawn) whose top left corner has coordinates (a,b) and whose bottom right corner has coordinates (c,d). If the bounding rectangle is a square, then the ellipse is a circle.



If you don't specify a fill color, only the black outline of the ellipse will be drawn. The posted program draws one large yellow-filled oval for the head, and two tiny black-filled ovals for the eyes.

The instruction

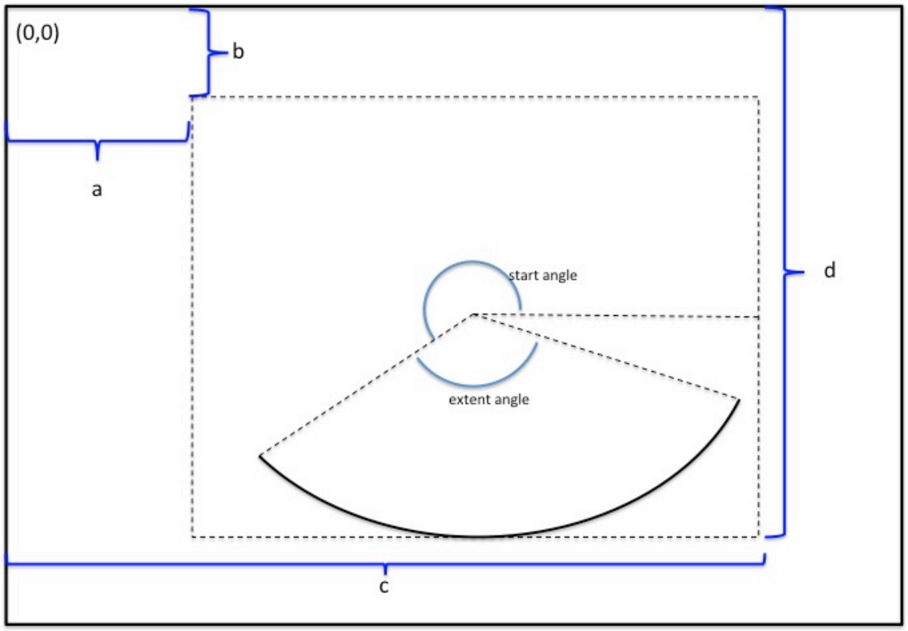
```
the_canvas.create_arc(125,125,175,175,start=225,extent=90,style=ARC)
```

which creates the smile, is more complicated: The first four arguments again specify the corners of a bounding rectangle of an ellipse; the next two specify what portion of the ellipse to draw. In our program, the arc starts at an angle 225 degrees counterclockwise from the positive x-direction, and extends counterclockwise for 90 degrees.

To complete Assignment 0, you can change the color or the dimensions of the face, change the smile to a frown, the circle to a rectangle, whatever you like---you have carte blanche. The command for drawing a rectangle is `the_canvas.create_rectangle`, with arguments that have the same meaning as those of the_canvas.create_oval.   You can also try

```
the_canvas.create_line(a,b,c,d)
```

which draws a line segment between the points (a,b) and (c,d).


(0,0)
b
a
start angle
d
extent angle
c