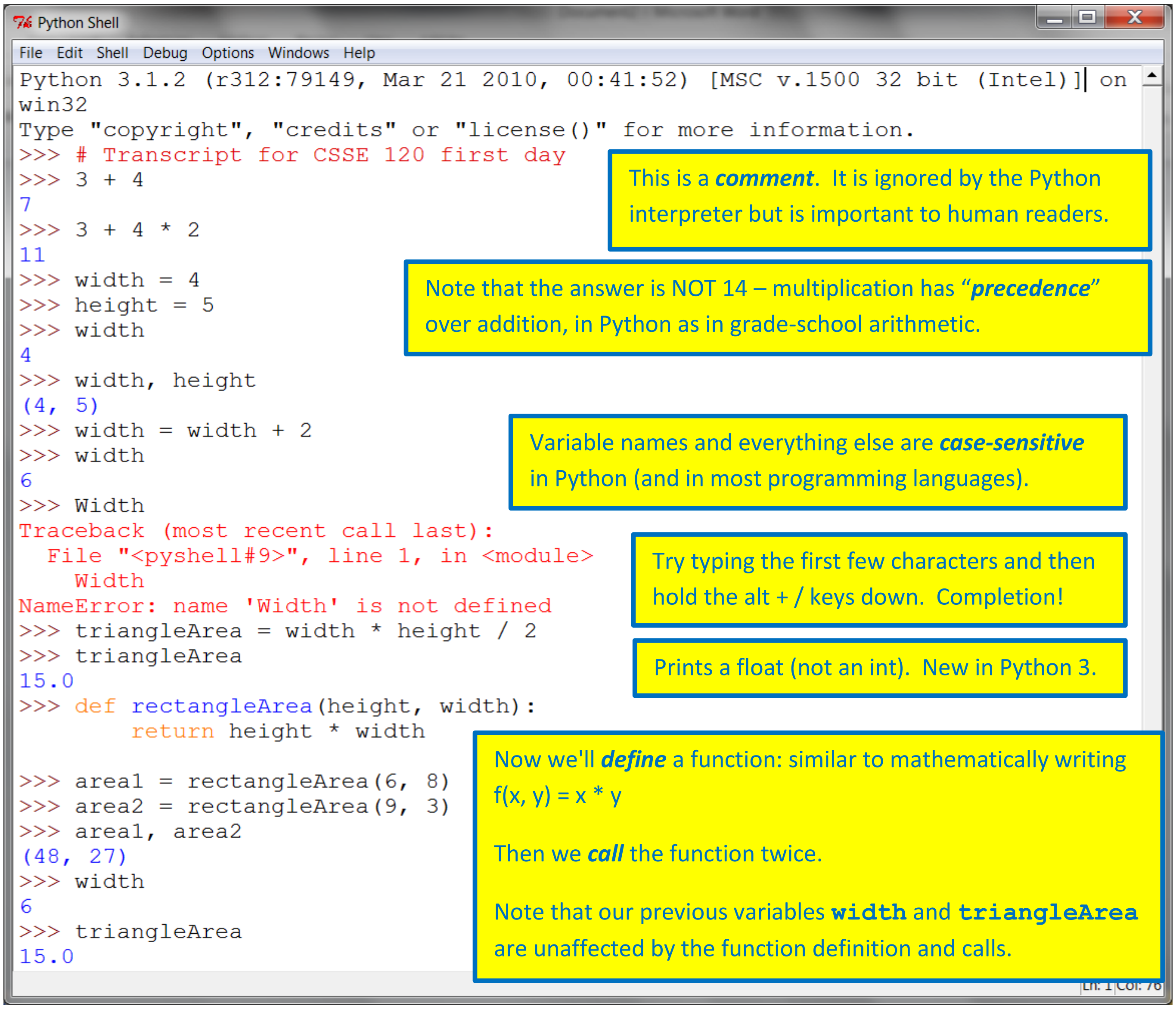
CSSE 120 Session 1 Transcript

This will be a quick intro. We'll come back to this stuff again in more detail starting next time. Instructor: You may want to increase the font size in IDLE at this time (Options → Configure IDLE → Size).

Live code the following with students, working line by line.



```
Python 3.1.2 (r312:79149, Mar 21 2010, 00:41:52) [MSC v.1500 32 bit (Intel)] on
win32
Type "copyright", "credits" or "license()" for more information.
>>> # Transcript for CSSE 120 first day
>>> 3 + 4
7
>>> 3 + 4 * 2
11
>>> width = 4
>>> height = 5
>>> width
4
>>> width, height
(4, 5)
>>> width = width + 2
>>> width
6
>>> Width
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    Width
NameError: name 'Width' is not defined
>>> triangleArea = width * height / 2
>>> triangleArea
15.0
>>> def rectangleArea(height, width):
        return height * width

>>> area1 = rectangleArea(6, 8)
>>> area2 = rectangleArea(9, 3)
>>> area1, area2
(48, 27)
>>> width
6
>>> triangleArea
15.0
```

This is a ***comment***. It is ignored by the Python interpreter but is important to human readers.

Note that the answer is NOT 14 – multiplication has "***precedence***" over addition, in Python as in grade-school arithmetic.

Variable names and everything else are ***case-sensitive*** in Python (and in most programming languages).

Try typing the first few characters and then hold the alt + / keys down. Completion!

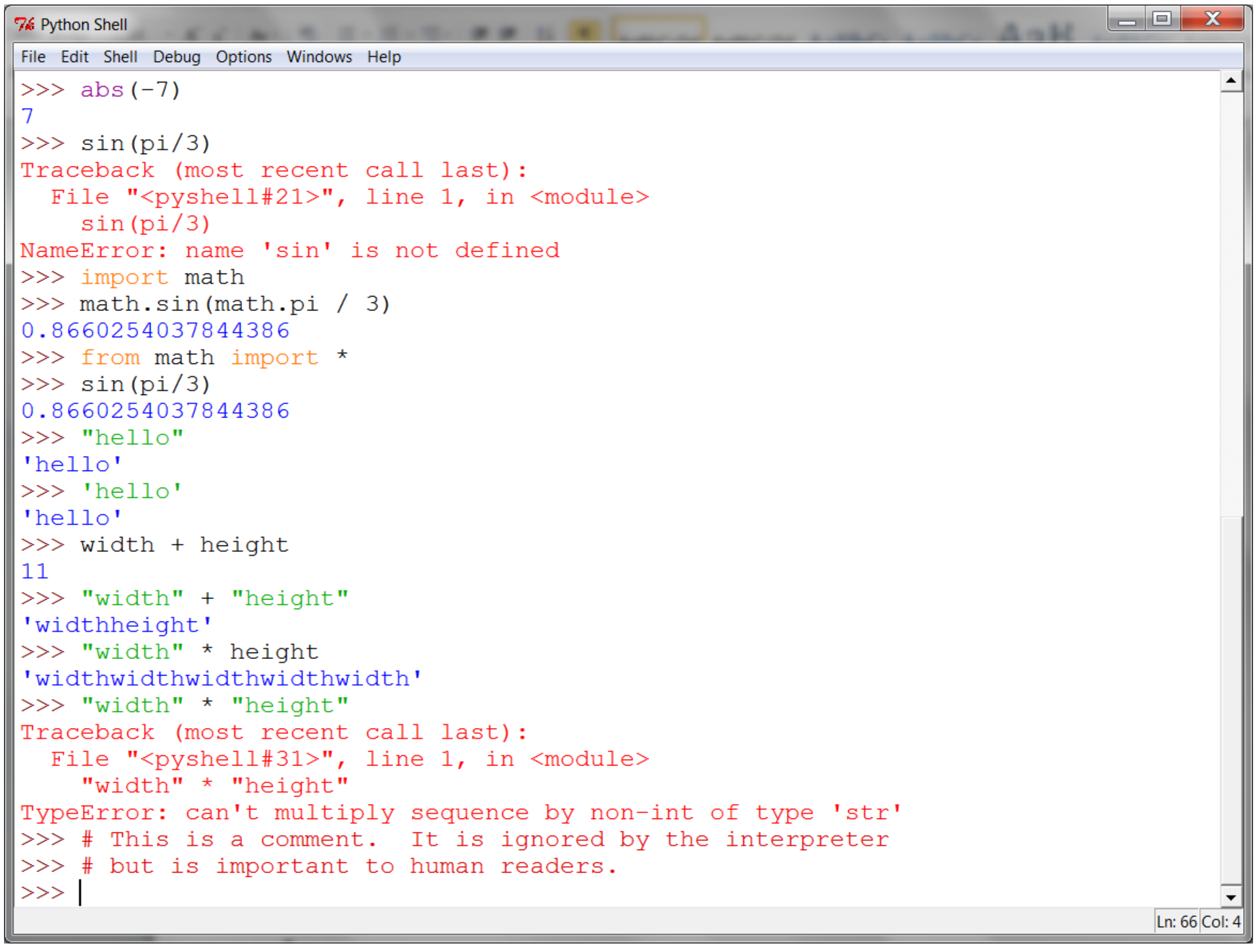
Prints a float (not an int). New in Python 3.

Now we'll ***define*** a function: similar to mathematically writing f(x, y) = x * y

Then we ***call*** the function twice.

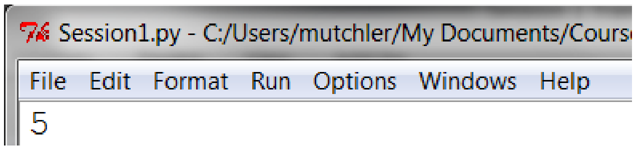
Note that our previous variables **`width`** and **`triangleArea`** are unaffected by the function definition and calls.

Continue to live code the following with students, working line by line



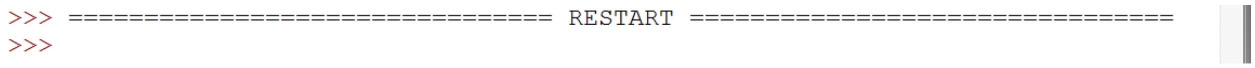
```
>>> abs(-7)
7
>>> sin(pi/3)
Traceback (most recent call last):
  File "<pyshell#21>", line 1, in <module>
    sin(pi/3)
NameError: name 'sin' is not defined
>>> import math
>>> math.sin(math.pi / 3)
0.8660254037844386
>>> from math import *
>>> sin(pi/3)
0.8660254037844386
>>> "hello"
'hello'
>>> 'hello'
'hello'
>>> width + height
11
>>> "width" + "height"
'widthheight'
>>> "width" * height
'widthwidthwidthwidthwidth'
>>> "width" * "height"
Traceback (most recent call last):
  File "<pyshell#31>", line 1, in <module>
    "width" * "height"
TypeError: can't multiply sequence by non-int of type 'str'
>>> # This is a comment.  It is ignored by the interpreter
>>> # but is important to human readers.
>>> 
```

Open up a new code window at this point. Save the File as <something>.py. (e.g. *Session1.py*). Place in the window:


Session1.py - C:/Users/mutchler/My Documents/Cours
File Edit Format Run Options Windows Help

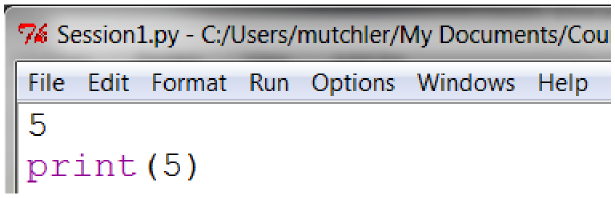
```
5
```

Run it (Run menu or F5). This is what you get back in the Python Shell (nothing shows up).



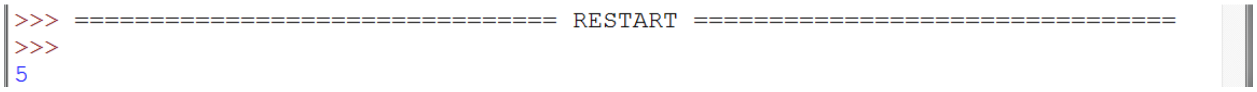
```
>>> ================================ RESTART ================================
>>>
```

Continuing in the new code window:


Session1.py - C:/Users/mutchler/My Documents/Cou
File Edit Format Run Options Windows Help

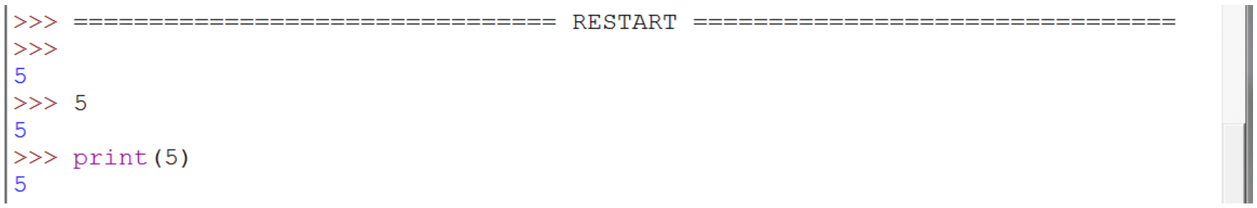
```
5
print(5)
```

Now the 5 shows up.



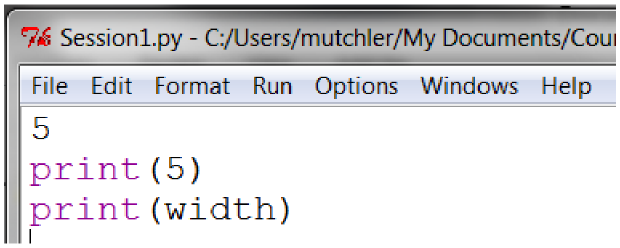
```
>>> ================================ RESTART ================================
>>>
5
```

Also try both lines in the interactive Python Shell, like this:



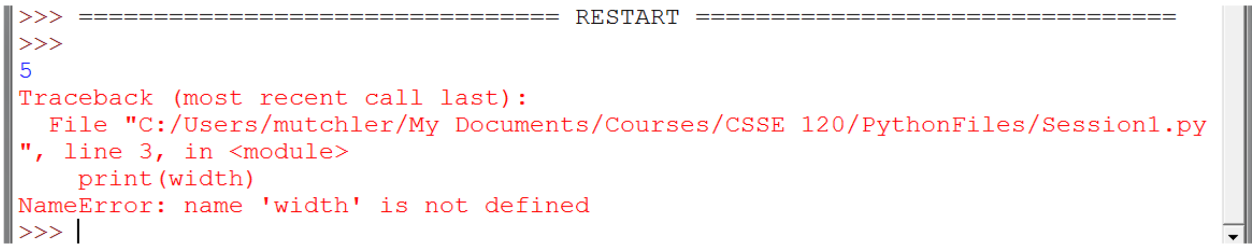
```
>>> ================================ RESTART ================================
>>>
5
>>> 5
5
>>> print(5)
5
```

Back in the new code window, add a line:


Session1.py - C:/Users/mutchler/My Documents/Cou
File Edit Format Run Options Windows Help

```
5
print(5)
print(width)
```

This results in an error message, back in the interactive Python Shell:



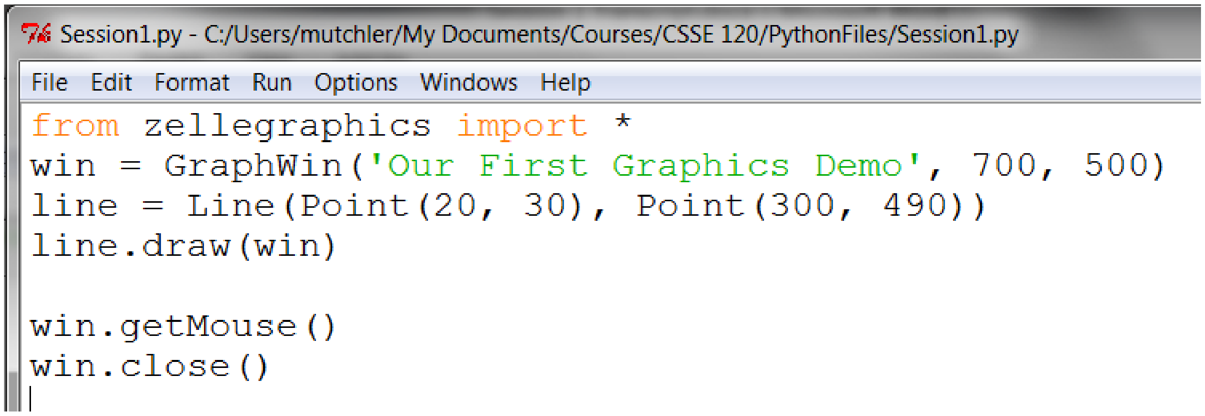
```
>>> ================================ RESTART ================================
>>>
5
Traceback (most recent call last):
  File "C:/Users/mutchler/My Documents/Courses/CSSE 120/PythonFiles/Session1.py
", line 3, in <module>
    print(width)
NameError: name 'width' is not defined
>>> 
```

Erase the code from the new code window (the script file).

Graphics:

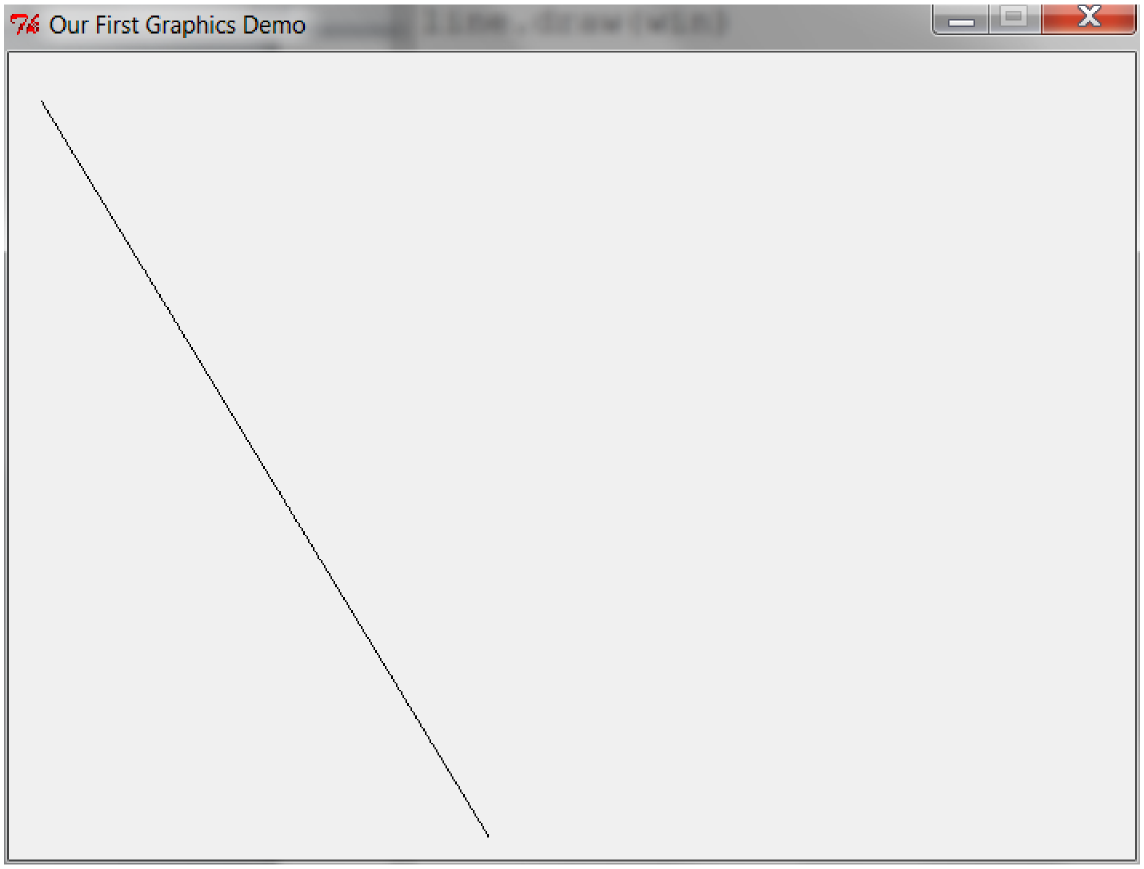
Now, do something like the following. After each new line of code (or enough to get new stuff in the graphics window) run it. *But before running the code, make sure the last 2 lines are entered, else the window may hang.*

Note the use of *zellegraphics* vs. *graphics* – this is a change from page 82 of the text.

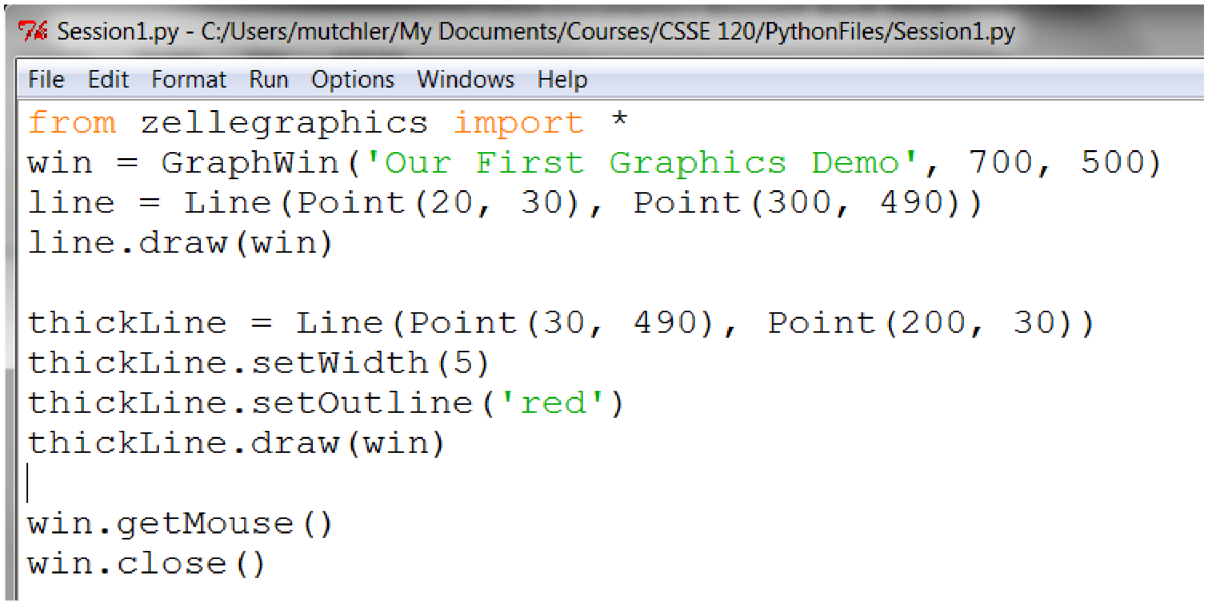

Session1.py - C:/Users/mutchler/My Documents/Courses/CSSE 120/PythonFiles/Session1.py
File Edit Format Run Options Windows Help

```
from zellegraphics import *
win = GraphWin('Our First Graphics Demo', 700, 500)
line = Line(Point(20, 30), Point(300, 490))
line.draw(win)

win.getMouse()
win.close()
```

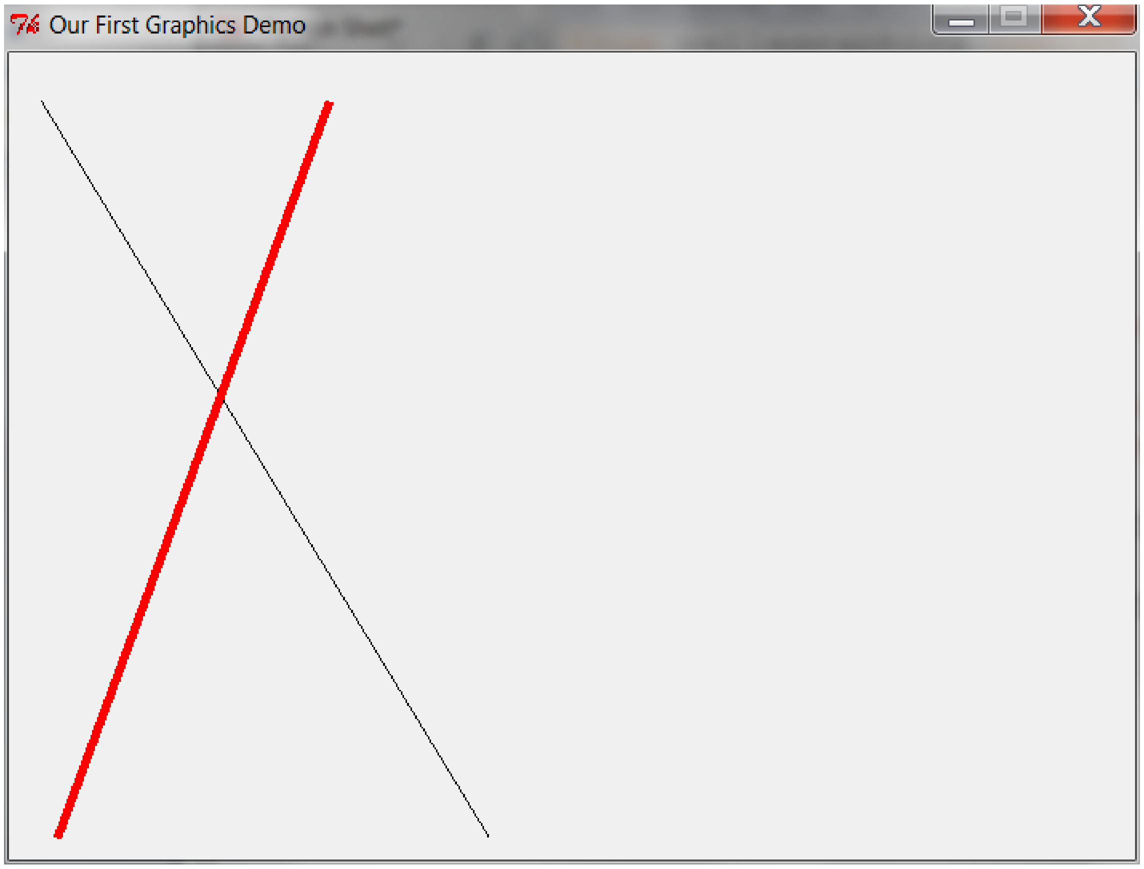
The above causes a new window to put up, with a line on it like this: (Clicking in the window closes it – students, do you see why?)



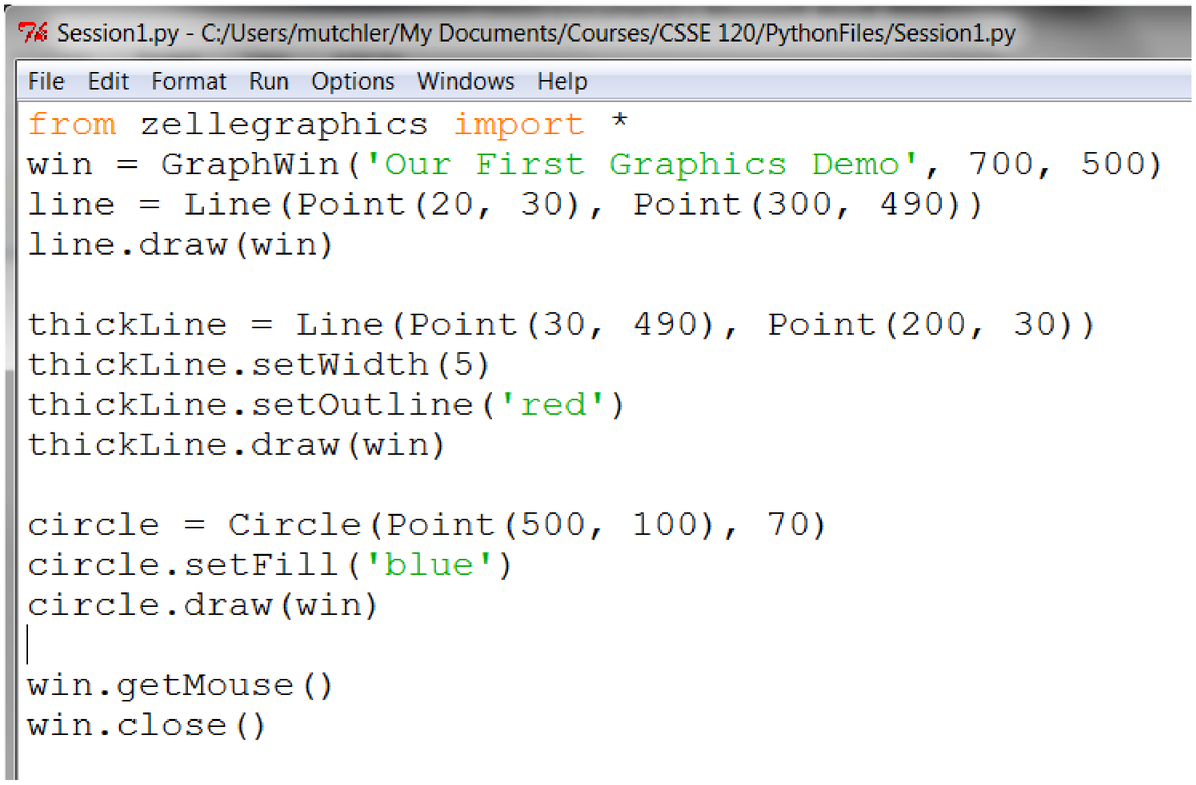
Continue entering lines in the program window and running them, like this:


Session1.py - C:/Users/mutchler/My Documents/Courses/CSSE 120/PythonFiles/Session1.py
File Edit Format Run Options Windows Help

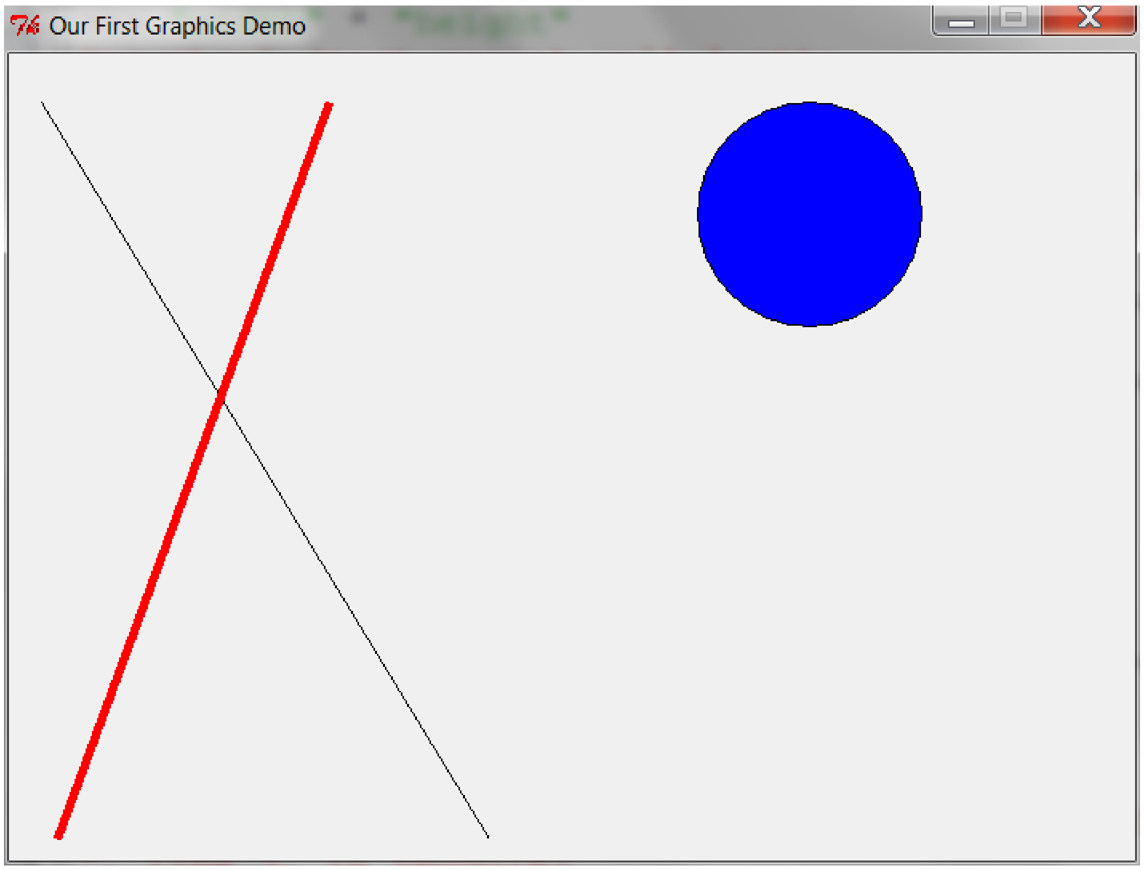
```
from zellegraphics import *
win = GraphWin('Our First Graphics Demo', 700, 500)
line = Line(Point(20, 30), Point(300, 490))
line.draw(win)

thickLine = Line(Point(30, 490), Point(200, 30))
thickLine.setWidth(5)
thickLine.setOutline('red')
thickLine.draw(win)

win.getMouse()
win.close()
```



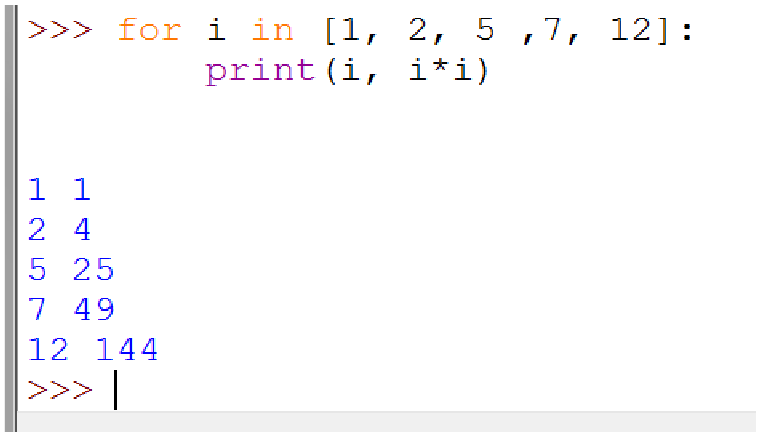
Continue entering lines in the program window and running them, like this:


Session1.py - C:/Users/mutchler/My Documents/Courses/CSSE 120/PythonFiles/Session1.py
File Edit Format Run Options Windows Help

```
from zellegraphics import *
win = GraphWin('Our First Graphics Demo', 700, 500)
line = Line(Point(20, 30), Point(300, 490))
line.draw(win)

thickLine = Line(Point(30, 490), Point(200, 30))
thickLine.setWidth(5)
thickLine.setOutline('red')
thickLine.draw(win)

circle = Circle(Point(500, 100), 70)
circle.setFill('blue')
circle.draw(win)

win.getMouse()
win.close()
```

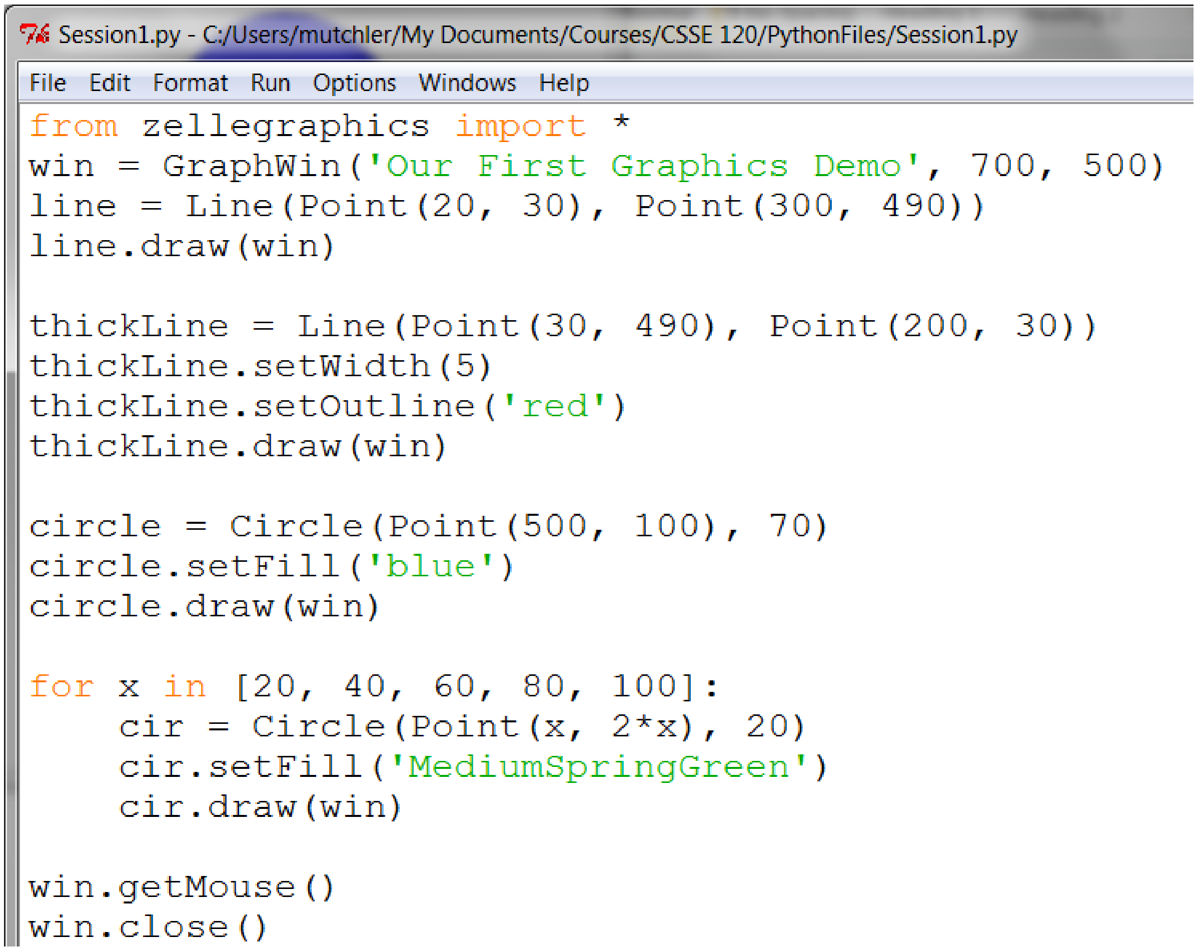


Loops: **Try this in the interactive Python Shell:**

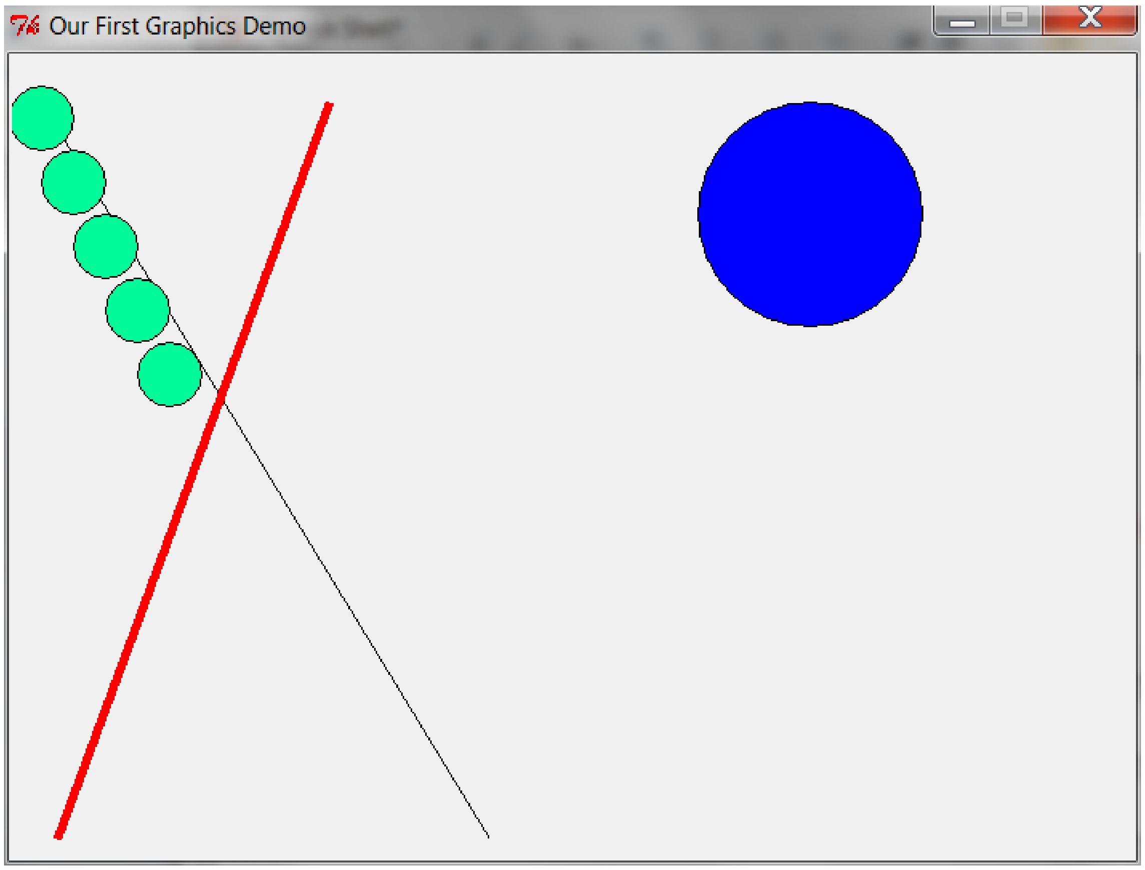


```
>>> for i in [1, 2, 5 ,7, 12]:
	print(i, i*i)


1 1
2 4
5 25
7 49
12 144
>>> 
```

Then back in the program window to get a bunch of circles.


Session1.py - C:/Users/mutchler/My Documents/Courses/CSSE 120/PythonFiles/Session1.py
File Edit Format Run Options Windows Help

```python
from zellegraphics import *
win = GraphWin('Our First Graphics Demo', 700, 500)
line = Line(Point(20, 30), Point(300, 490))
line.draw(win)

thickLine = Line(Point(30, 490), Point(200, 30))
thickLine.setWidth(5)
thickLine.setOutline('red')
thickLine.draw(win)

circle = Circle(Point(500, 100), 70)
circle.setFill('blue')
circle.draw(win)

for x in [20, 40, 60, 80, 100]:
    cir = Circle(Point(x, 2*x), 20)
    cir.setFill('MediumSpringGreen')
    cir.draw(win)

win.getMouse()
win.close()
```

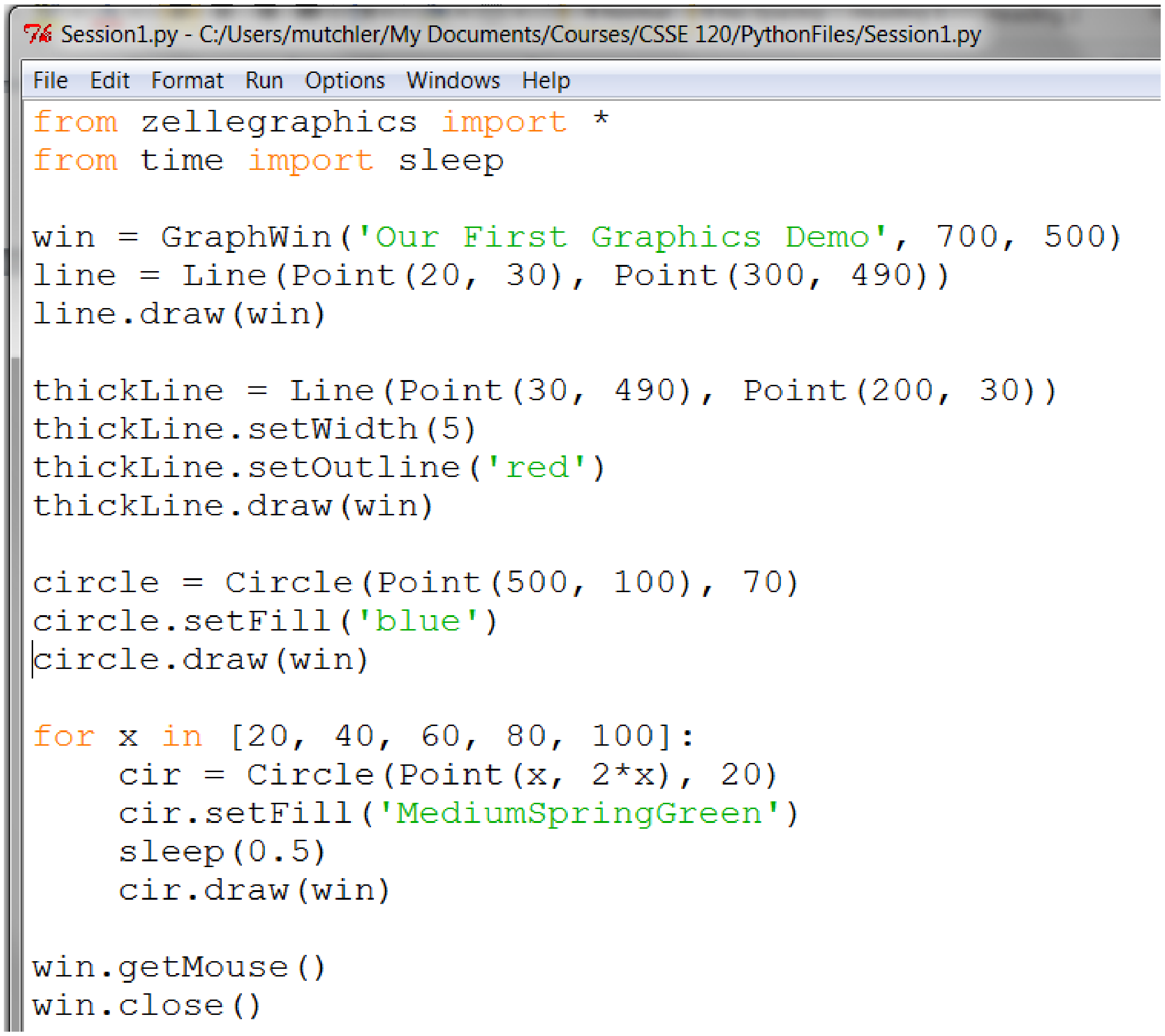


Add the lines:

```
from time import sleep
```

And later in the file:

```
sleep(0.5)
```

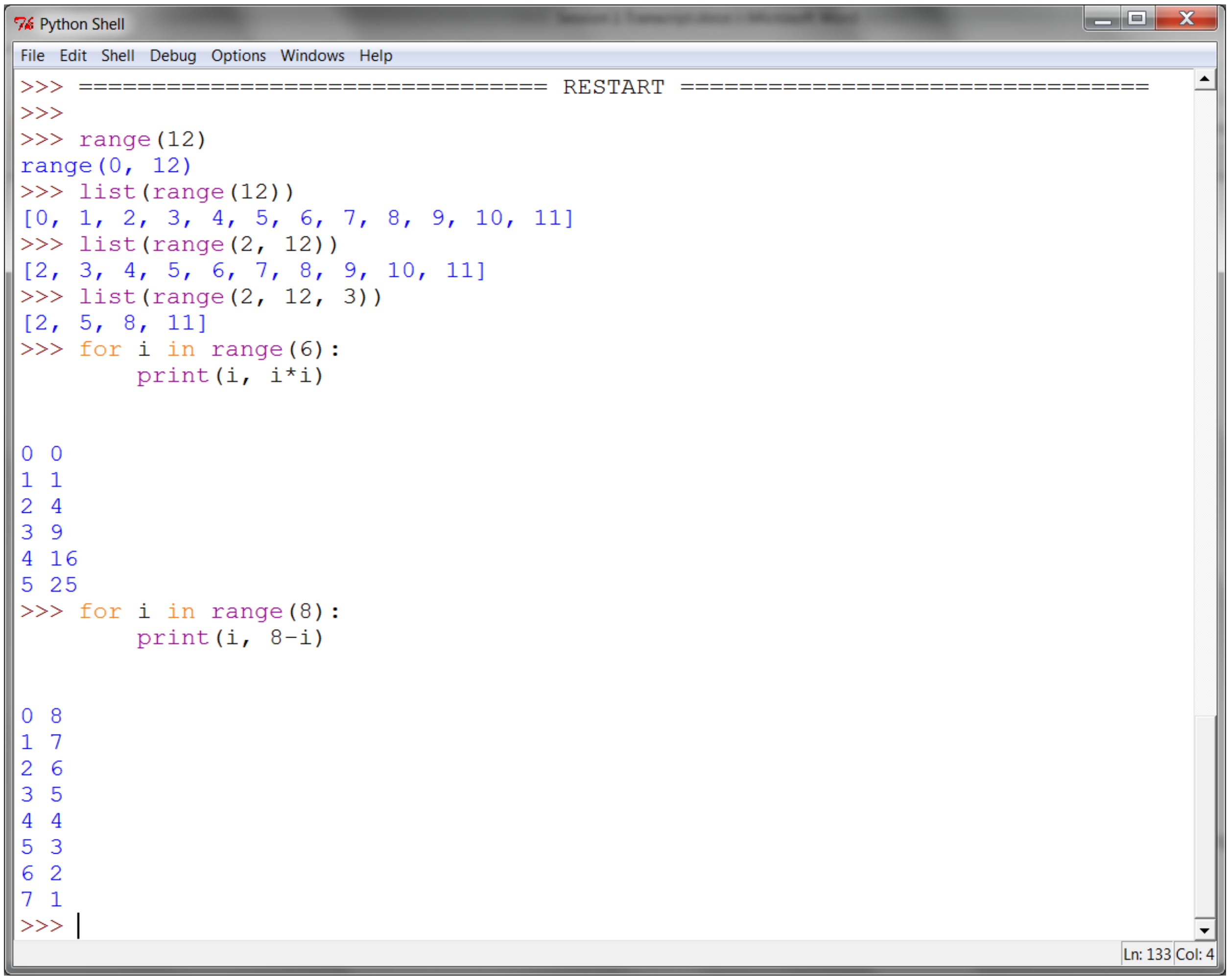

Session1.py - C:/Users/mutchler/My Documents/Courses/CSSE 120/PythonFiles/Session1.py
File Edit Format Run Options Windows Help

```
from zellegraphics import *
from time import sleep

win = GraphWin('Our First Graphics Demo', 700, 500)
line = Line(Point(20, 30), Point(300, 490))
line.draw(win)

thickLine = Line(Point(30, 490), Point(200, 30))
thickLine.setWidth(5)
thickLine.setOutline('red')
thickLine.draw(win)

circle = Circle(Point(500, 100), 70)
circle.setFill('blue')
circle.draw(win)

for x in [20, 40, 60, 80, 100]:
    cir = Circle(Point(x, 2*x), 20)
    cir.setFill('MediumSpringGreen')
    sleep(0.5)
    cir.draw(win)

win.getMouse()
win.close()
```

The result is the same picture as before, but the circles show up one by one (with half a second pauses in between each one showing up).

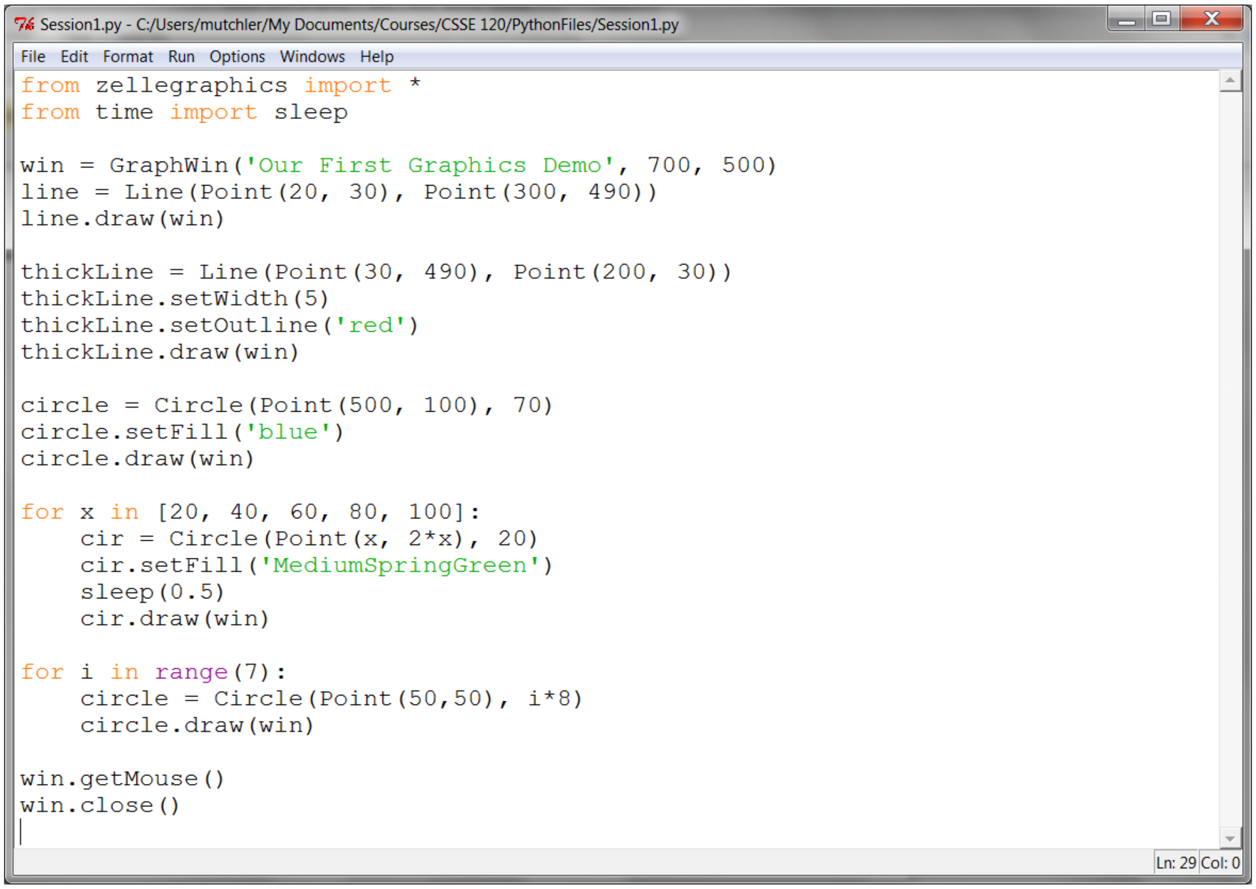
Range expressions and loops: **Back in the interactive Python Shell, try this, line by line. (Ask students to come up the second loop, that is, to come up with a loop whose output isL**

0	**8**
1	**7**
2	**6**
3	**5**
4	**4**
5	**3**
6	**2**
7	**1**



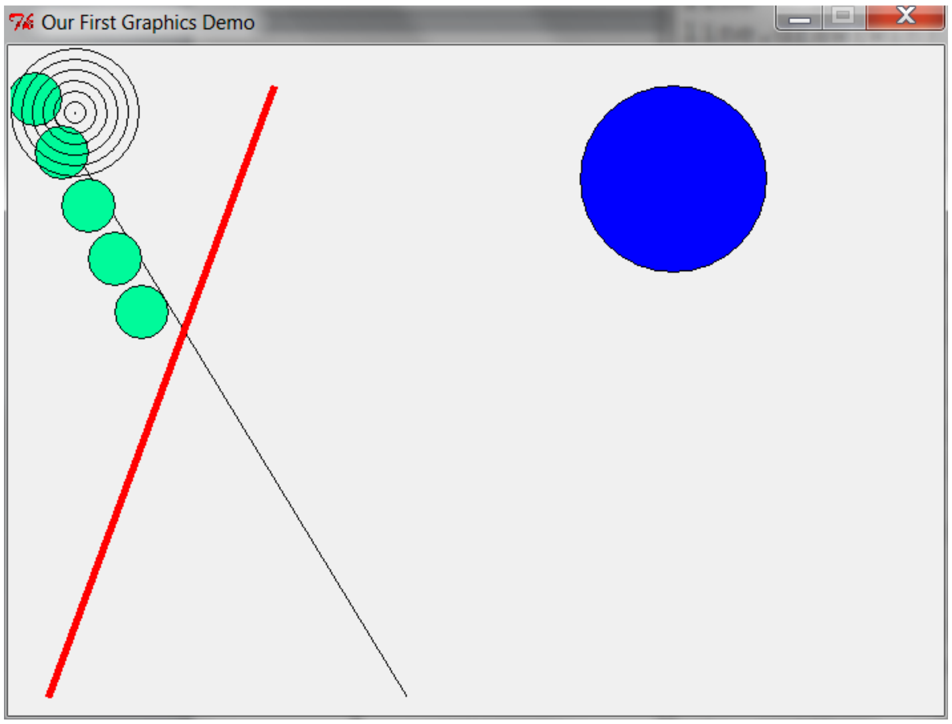
```
>>> ================================ RESTART ================================
>>> 
>>> range(12)
range(0, 12)
>>> list(range(12))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
>>> list(range(2, 12))
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
>>> list(range(2, 12, 3))
[2, 5, 8, 11]
>>> for i in range(6):
        print(i, i*i)


0 0
1 1
2 4
3 9
4 16
5 25
>>> for i in range(8):
        print(i, 8-i)


0 8
1 7
2 6
3 5
4 4
5 3
6 2
7 1
>>> |
```

Then back in the program window, add the range statement loop shown below:

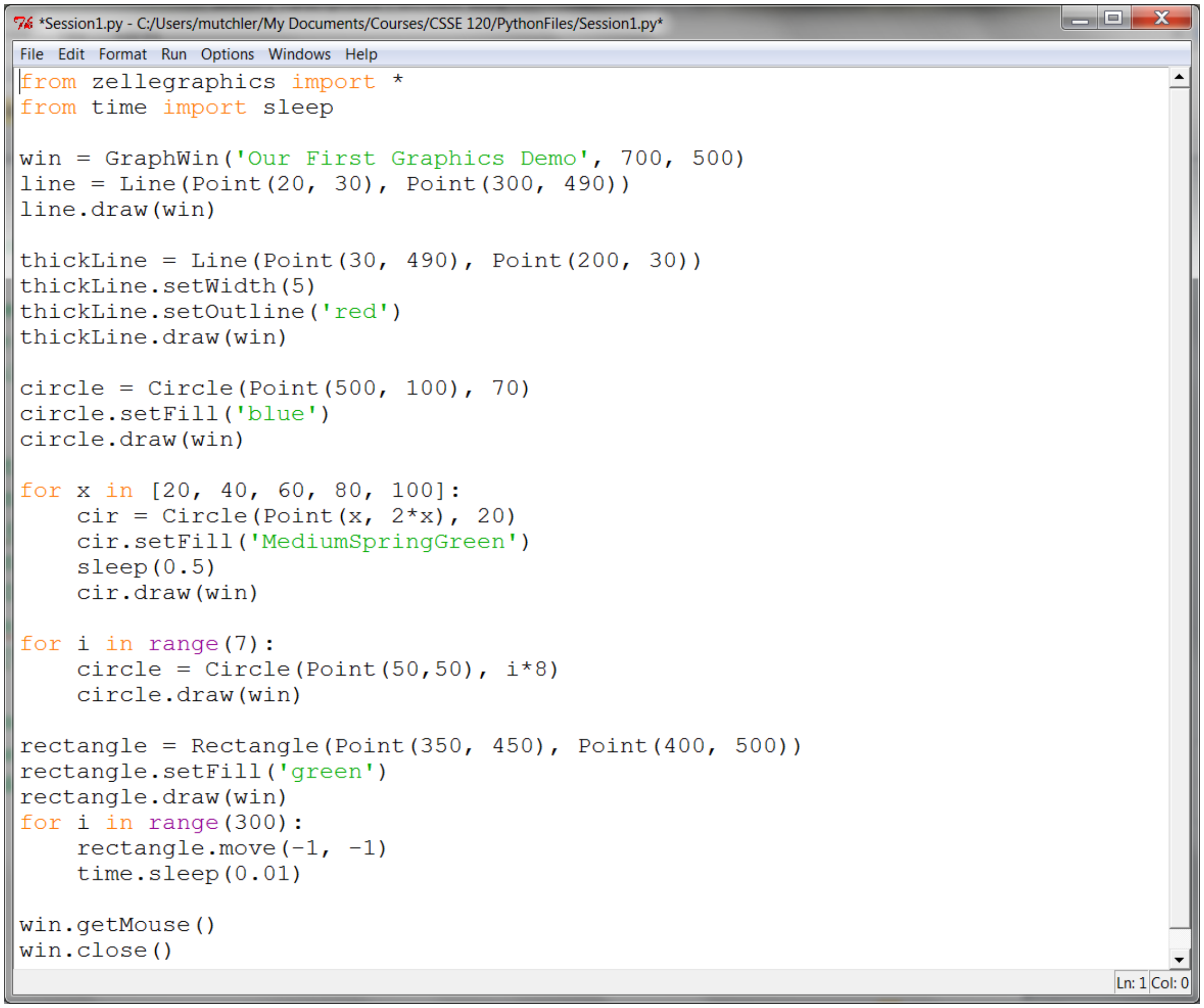

Session1.py - C:/Users/mutchler/My Documents/Courses/CSSE 120/PythonFiles/Session1.py
File Edit Format Run Options Windows Help
Ln: 29 Col: 0

```python
from zellegraphics import *
from time import sleep

win = GraphWin('Our First Graphics Demo', 700, 500)
line = Line(Point(20, 30), Point(300, 490))
line.draw(win)

thickLine = Line(Point(30, 490), Point(200, 30))
thickLine.setWidth(5)
thickLine.setOutline('red')
thickLine.draw(win)

circle = Circle(Point(500, 100), 70)
circle.setFill('blue')
circle.draw(win)

for x in [20, 40, 60, 80, 100]:
    cir = Circle(Point(x, 2*x), 20)
    cir.setFill('MediumSpringGreen')
    sleep(0.5)
    cir.draw(win)

for i in range(7):
    circle = Circle(Point(50,50), i*8)
    circle.draw(win)

win.getMouse()
win.close()
```

When it is run, you get this window (note the concentric circles in the upper left):



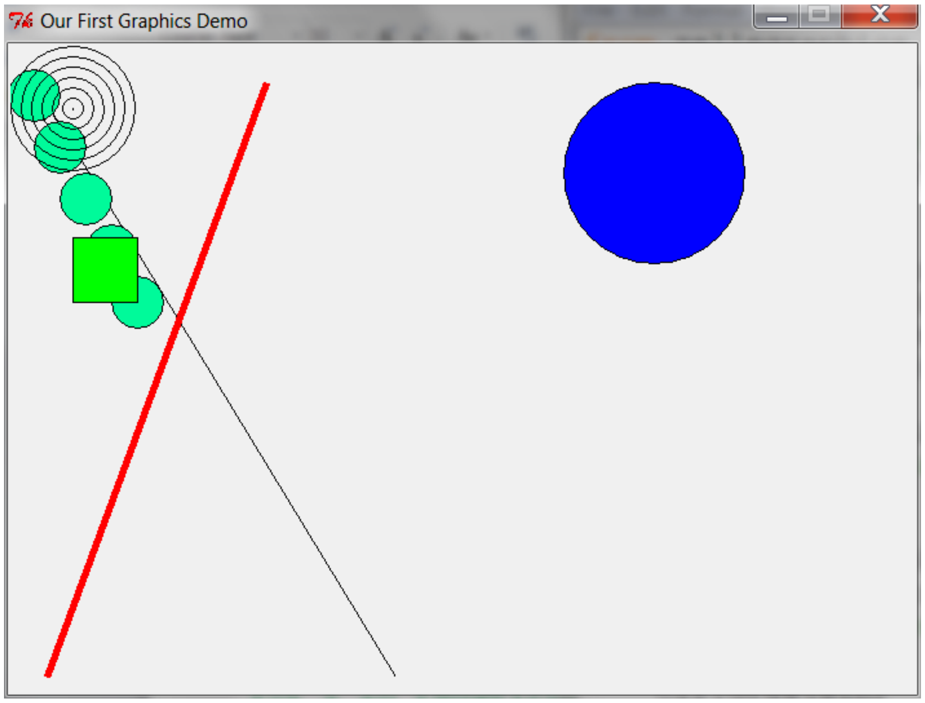
Finally, let's animate a Rectangle, like this:



Session1.py - C:/Users/mutchler/My Documents/Courses/CSSE 120/PythonFiles/Session1.py
File Edit Format Run Options Windows Help
Ln: 1 Col: 0

```
from zellegraphics import *
from time import sleep

win = GraphWin('Our First Graphics Demo', 700, 500)
line = Line(Point(20, 30), Point(300, 490))
line.draw(win)

thickLine = Line(Point(30, 490), Point(200, 30))
thickLine.setWidth(5)
thickLine.setOutline('red')
thickLine.draw(win)

circle = Circle(Point(500, 100), 70)
circle.setFill('blue')
circle.draw(win)

for x in [20, 40, 60, 80, 100]:
    cir = Circle(Point(x, 2*x), 20)
    cir.setFill('MediumSpringGreen')
    sleep(0.5)
    cir.draw(win)

for i in range(7):
    circle = Circle(Point(50,50), i*8)
    circle.draw(win)

rectangle = Rectangle(Point(350, 450), Point(400, 500))
rectangle.setFill('green')
rectangle.draw(win)
for i in range(300):
    rectangle.move(-1, -1)
    time.sleep(0.01)

win.getMouse()
win.close()
```

That results in this (but the square moves).



Our First Graphics Demo