Emphasize that this will be a quick intro. We'll come back to this stuff again in more detail starting next time.
Walk them through Graphics file installation.
There should be a link on today's details page.

```
IDLE 1.2.1
>>> # Transcript for CSSE 120 first day (point out the comment)
>>> 3 + 4
7
>>> 3 + 4 * 2  # Note that the answer is NOT 14.
```

Note the operator precedence

```
11
>>> width = 4
>>> height = 5
>>> width
4
>>> width, height
(4, 5)
>>> width = width + 2
>>> width
6
```

Mention case-sensitivity variable names and everything else in Python:

```
>>> Width

Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    Width
NameError: name 'Width' is not defined
>>> triangleArea = width * height / 2
>>> triangleArea
15
>>> def rectArea(height, width):
      return height*width
```

defining a function vs calling a function

```
>>> rectArea(6, 8)
48
```

What about built-in functions?

```
>>> abs(-7)
7
>>> sin(pi/3) # not everything is built in

Traceback (most recent call last):
  File "<pyshell#17>", line 1, in <module>
    sin(pi/2)
NameError: name 'sin' is not defined
>>> from math import *
>>> sin(pi/3)
0.8660254037844386
```

Open up a new code window at this point. Save the File as <something>.py.
Place in the window:

5
Run it (F5)
Then do
print 5
run again.
Go back and do both of these things in the interactive window.

Erase the code from 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. Be sure to point out the use of zellegraphics vs. graphics, change p. 126 of text.

```
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)
cir = Circle(Point(500, 100), 70)
cir.draw(win)
```

Now encourage them to add to the program to draw something interesting (for example, a house or a face).

Range and loops (back in the interactive window at first)

```
>>> range(12)

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

Note that the range ends with 11, not 12.

```
>>> range(2, 12)
[2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

Briefly explain the idea of a loop.

```
>>> for i in range(6):
      print i, i*i

0 0
1 1
2 4
3 9
4 16
5 25
```

Ask them to try to make a loop to print

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

Now a loop to do some graphics (back in the program window)

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

```
# animate movement of a rectangle
rectangle = Rectangle(Point(350, 450), Point(400, 500))
rectangle.setFill('green')
rectangle.draw(win)
import time (You may want to put this at the beginning of the file)
for i in range(300):
       rectangle.move(-1, -1)
       time.sleep(0.01)
```

Draw an interesting picture (10 minutes) that includes at least one loop.
Show it off to people near you.
Get help from other students near you, instructor, and lab assistants as needed.