Triple Quotes in Python.

Figure 1: Triple quotes are a peculiarity of Python's.

Triple quotes are peculiar to the syntax of the Python programming language.

Multi-line Comments:
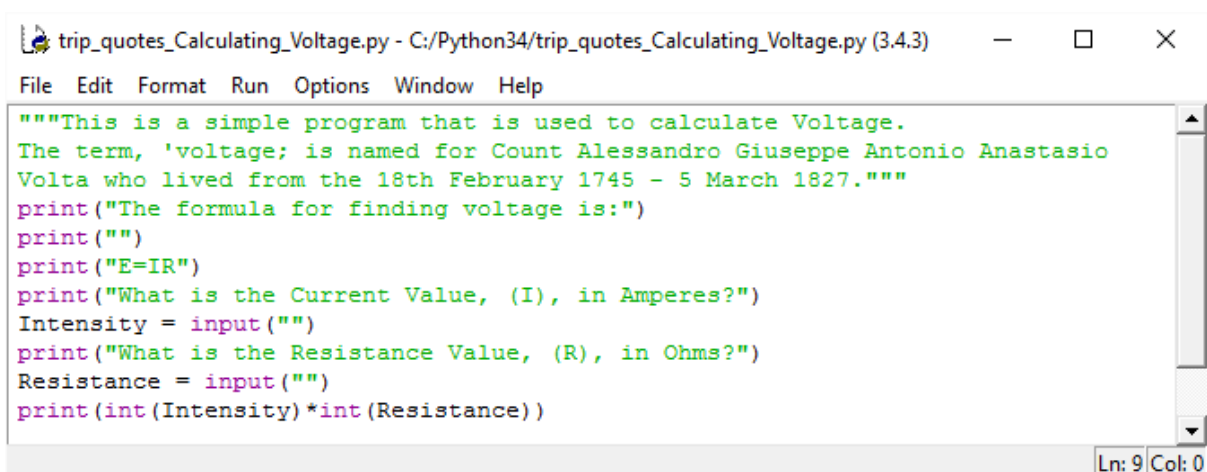
One use for them is to enable multi-line comments:


```
"""This is a simple program that is used to calculate Voltage.
The term, 'voltage; is named for Count Alessandro Giuseppe Antonio Anastasio
Volta who lived from the 18th February 1745 - 5 March 1827."""
print("The formula for finding voltage is:")
print("")
print("E=IR")
print("What is the Current Value, (I), in Amperes?")
Intensity = input("")
print("What is the Resistance Value, (R), in Ohms?")
Resistance = input("")
print(int(Intensity)*int(Resistance))
```

Ln: 9 Col: 0

Figure 2: In this short program that calculates Voltage, I use triple quotes so as to make a multi-line biographical comment concerning Count Alessandro Volta.

In many programming languages, such as C, we can make similar multi-line comments by enclosing these comments within:

$$/* */$$

as in:

$$/* \text{ Your multi} - \text{line comment goes here. } */$$

Multi-line Strings:

Triple quotes can also be employed so as to print multi-line strings.

Figure 3: In the following example, we are going to use triple quotes so as to print the nursery rhyme, *Three Blind Mice*, as a multi-line String.

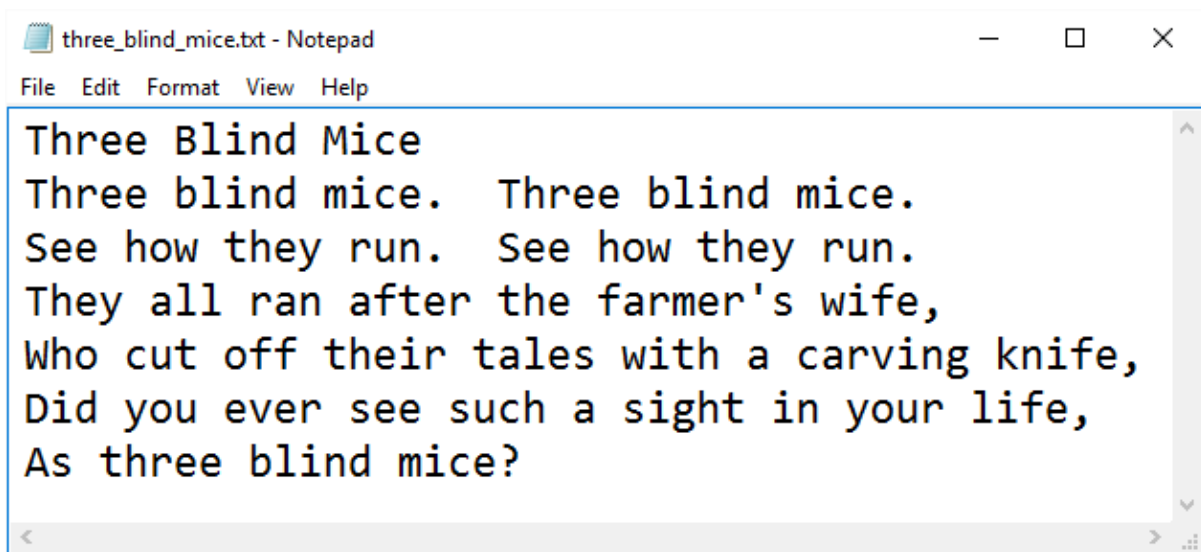Let us say that we have a piece of text that we wish to print to screen:

Figure 4: This is the piece of text that we wish to print to screen.

One way, that we might imagine of doing this, would be to simply copy the text and then to paste it inside a:
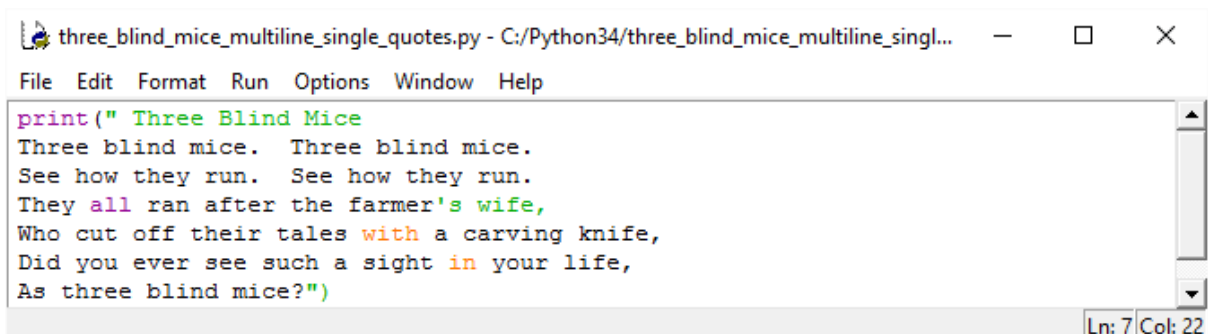
$$print("")$$

statement:

Figure 5: We do not even need to run this program to see that it will not work, but we shall:
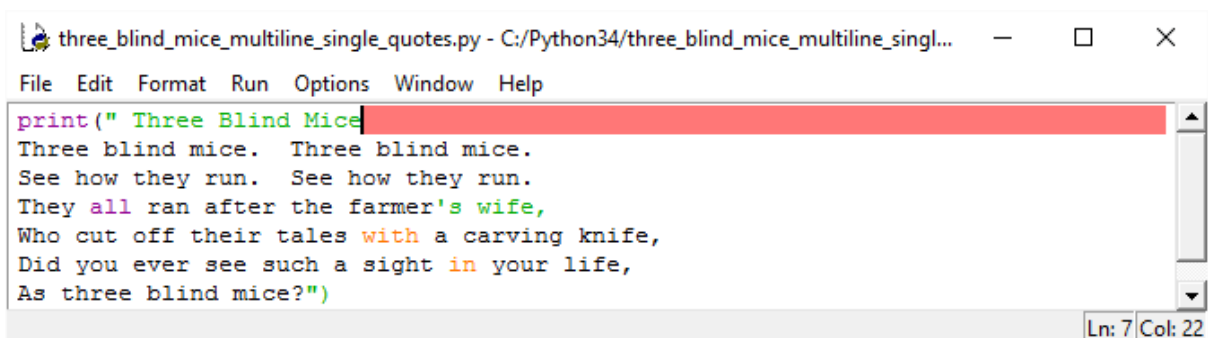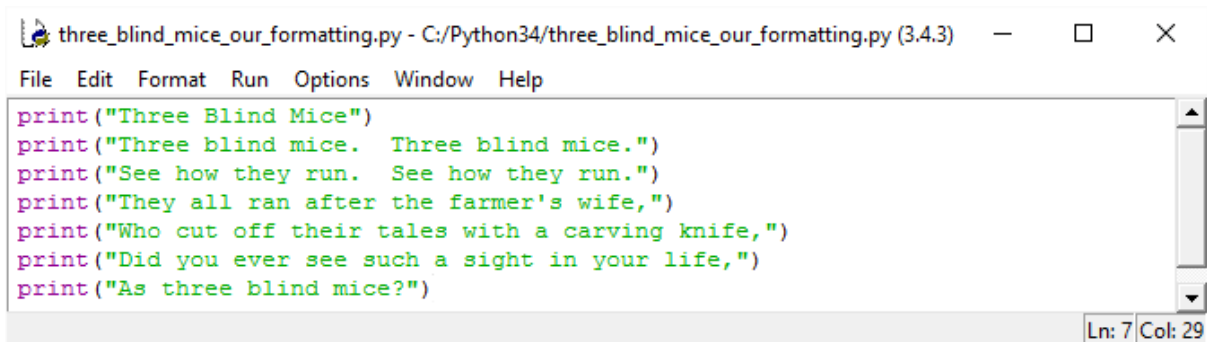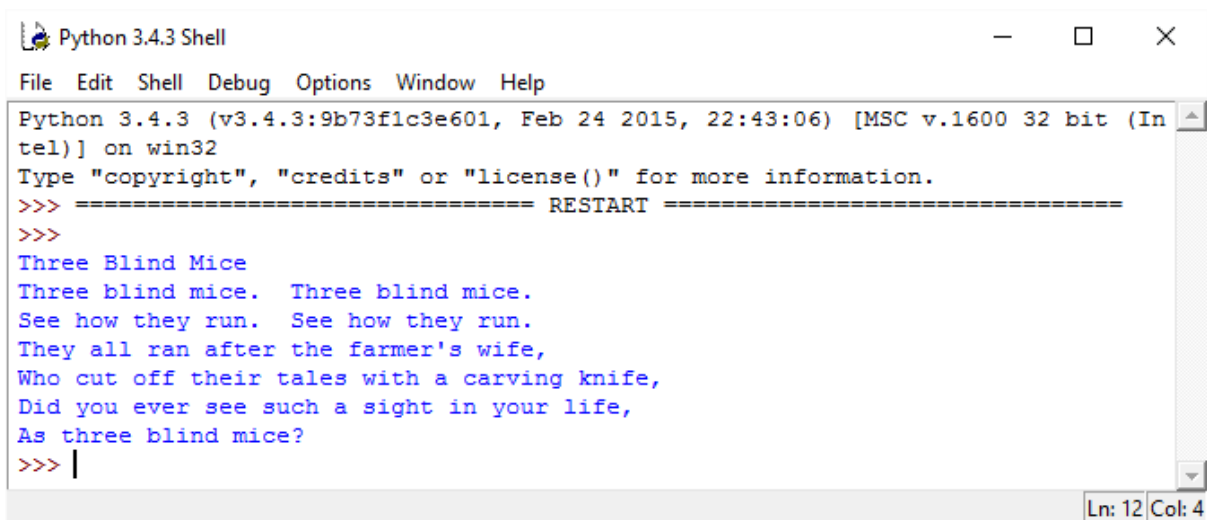

Figure 6: This is what occurs when we try to run this defective program.

Another option would be to enter in the text manually, and to apply the correct formatting ourselves:

```
three_blind_mice_our_formatting.py - C:/Python34/three_blind_mice_our_formatting.py (3.4.3)     —   □   ×
File  Edit  Format  Run  Options  Window  Help
print("Three Blind Mice")
print("Three blind mice.  Three blind mice.")
print("See how they run.  See how they run.")
print("They all ran after the farmer's wife,")
print("Who cut off their tales with a carving knife,")
print("Did you ever see such a sight in your life,")
print("As three blind mice?")
                                                              Ln: 7 Col: 29
```

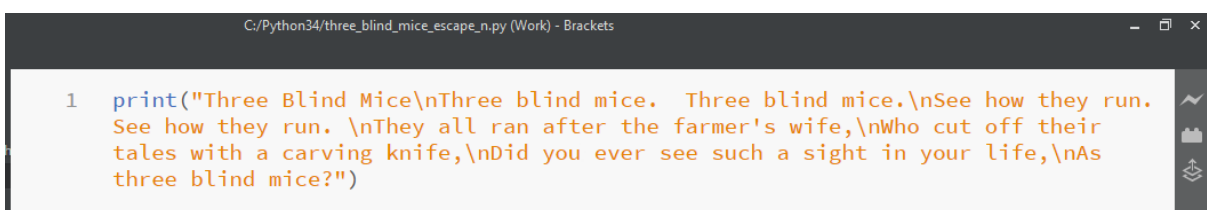**Figure 7:** This works. It took a lot of effort on our part, though.

```
Python 3.4.3 Shell                                       —   □   ×
File  Edit  Shell  Debug  Options  Window  Help
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ============================ RESTART ============================
>>>
Three Blind Mice
Three blind mice.  Three blind mice.
See how they run.  See how they run.
They all ran after the farmer's wife,
Who cut off their tales with a carving knife,
Did you ever see such a sight in your life,
As three blind mice?
>>> |
                                                              Ln: 12 Col: 4
```

Figure 8: The result of the program depicted in **Figure 7**'s being run.

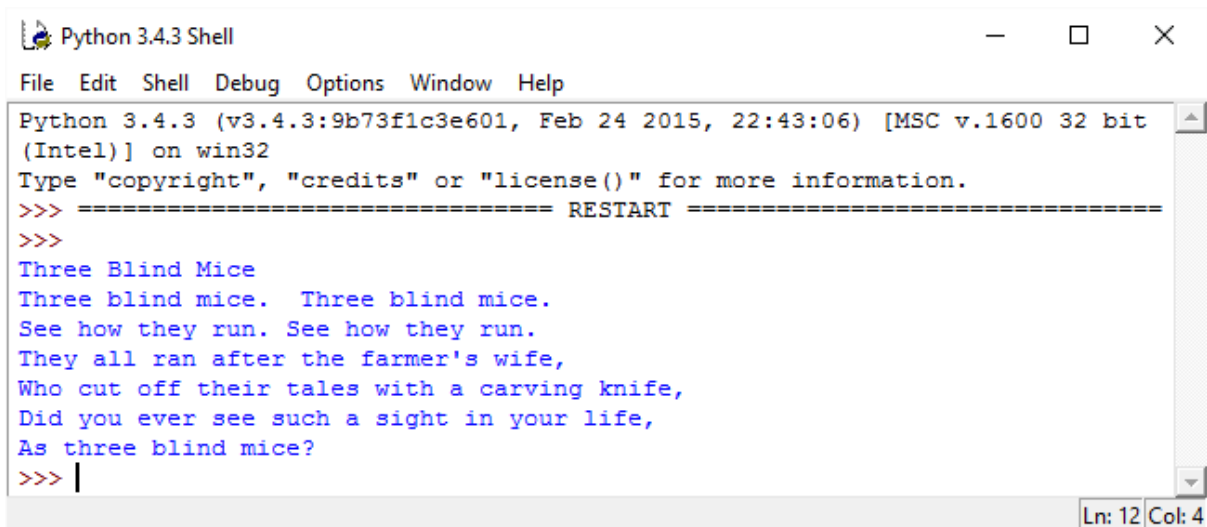We can actually print the above String within single quotes by using the:

\n

sequence:

```
C:/Python34/three_blind_mice_escape_n.py (Work) - Brackets                     _  □  ×
1    print("Three Blind Mice\nThree blind mice.  Three blind mice.\nSee how they run.
     See how they run. \nThey all ran after the farmer's wife,\nWho cut off their
     tales with a carving knife,\nDid you ever see such a sight in your life,\nAs
     three blind mice?")
```
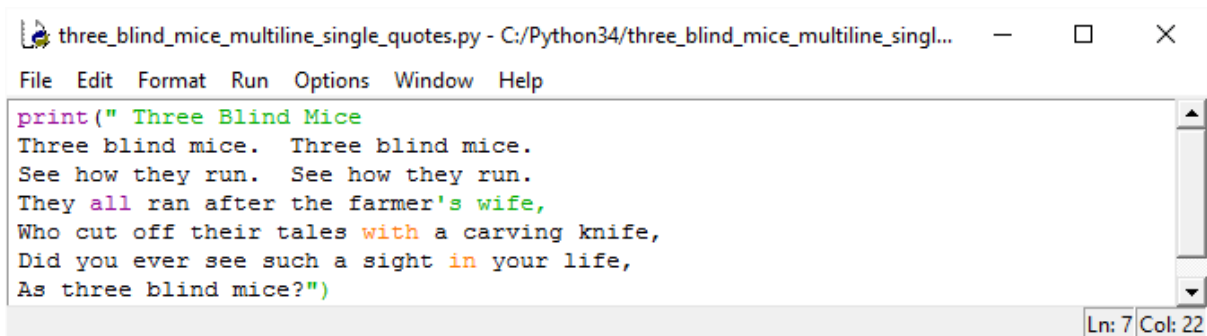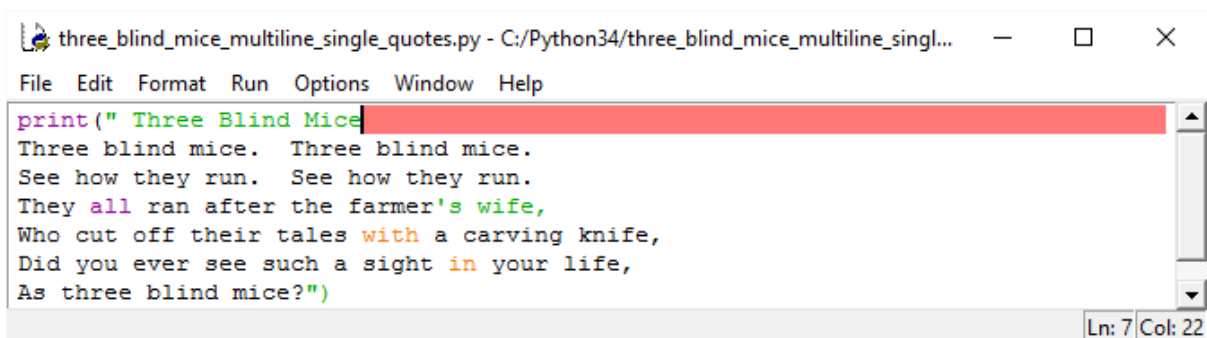
Figure 9: We can print *Three Blind Mice* as a String within a single print() statement by using the \n sequence.

Figure 10: The result of the program depicted in **Figure 9**'s being run.

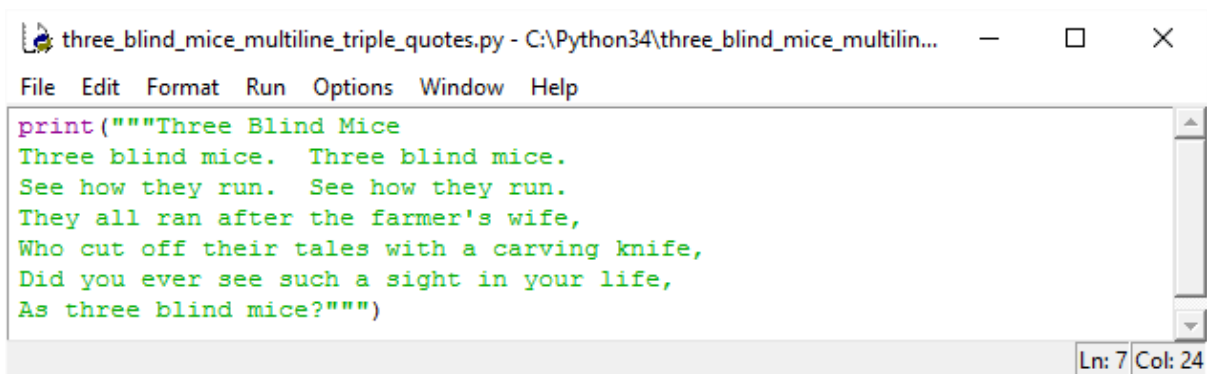But that program that the computer refused to interpret:

Figure 11: We tried to print out a multi-line string by only using single quotes. As we can observe from the syntax-highlighting, the program is defective and cannot be interpreted.

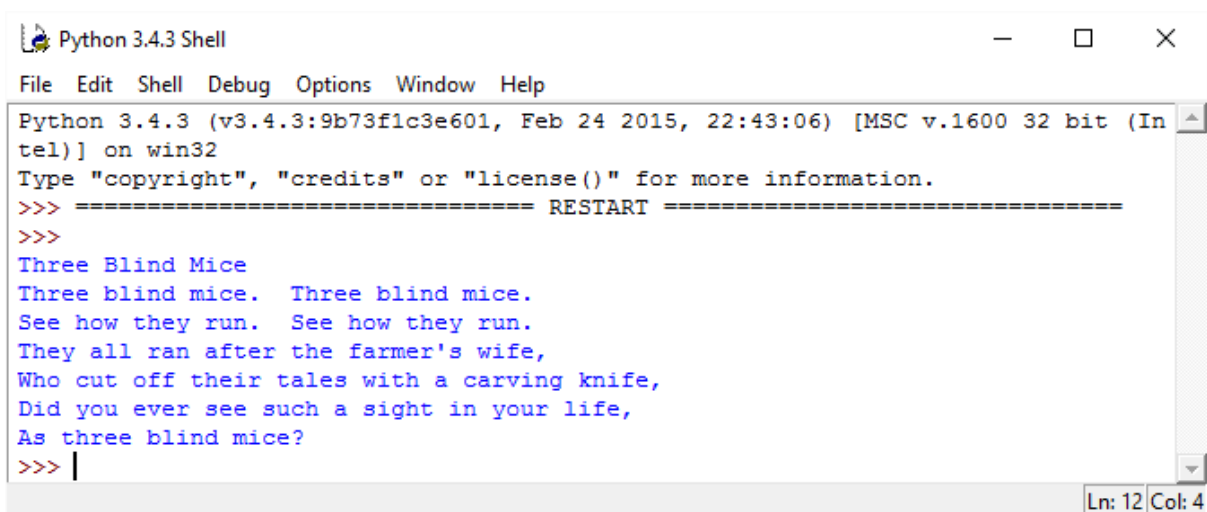**Figure 12:** To restate: the computer refuses to interpret this program.

There is a way of fixing the program that will not run in **Figures 11** and **12** without too much ado: by simply adding two more quotation marks at each end:

Figure 13: Now our print() statement contains triple quotation marks at each end.

Having added the triple quotes, our program will now be interpreted without any difficulty:

Figure 14: What the previous debugged program outputs. We debugged the program that would not be interpreted by replacing single quotes with double quotes.