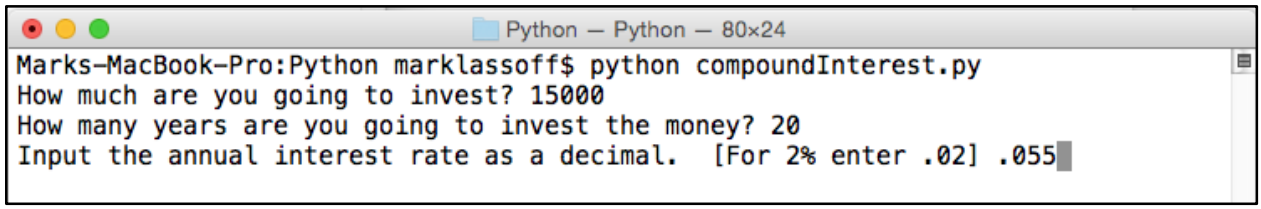
Activity 12.5 Getting Rich with Compound Interest

Compound interest calculations are extremely difficult to do without a computer, yet, this is a fairly common task. You're going to be the next in a long line of programmers to write a program that compounds interest. We'll take several inputs from the user and then show how interest compounds over the term.

1) Create a new Python file in your IDLE environment and save that blank file as compoundInterest.py.

2) Create three "input" statements that query the user as illustrated in the screenshot below. Save the responses in variables called *initialInvestment*, *term*, and *interestRate*.



3) To calculate interest use the following formula each month:

```
interestEarned= initialInvestment * (interestRate/12)
initialInvestment = interestEarned + initialInvestment
```

Remember that the interest rate was entered as an annual interest rate, so we have to divide it by 12 to compound it monthly.

4) Use a while loop and the formula above to print out a result in the command line that looks similar to the following screenshot.

HINT: Use the string "\t" to insert a tab and layout the results in nice, neat columns.

```
Python — bash — 80×50
Marks-MacBook-Pro:Python marklassoff$ python compoundInterest.py
How much are you going to invest? 25000
How many years are you going to invest the money? 2
Input the annual interest rate as a decimal.  [For 2% enter .02] .0295
Month   Interest Earned Total
1       61.4583333333   25061.4583333
2       61.6094184028   25123.0677517
3       61.7608748897   25184.8286266
4       61.9127037071   25246.7413303
5       62.0649057704   25308.8062361
6       62.2174819971   25371.0237181
7       62.370433307    25433.3941514
8       62.5237606222   25495.917912
9       62.6774648671   25558.5953769
10      62.8315469682   25621.4269239
11      62.9860078545   25684.4129317
12      63.1408484571   25747.5537802
13      63.2960697096   25810.8498499
14      63.4516725476   25874.3015224
15      63.6076579093   25937.9091803
16      63.764026735    26001.6732071
17      63.9207799674   26065.593987
18      64.0779185515   26129.6719056
19      64.2354434346   26193.907349
20      64.3933555664   26258.3007046
21      64.5516558988   26322.8523605
22      64.7103453862   26387.5627059
23      64.8694249853   26452.4321309
24      65.0288956551   26517.4610265
Marks-MacBook-Pro:Python marklassoff$
```

5) While the results are mathematically sound, these values don't look much like currency. Let's use a formatting function that will correctly format the values. Just as **\t** was used to add tab stops to the string, you can use the formatting code **{:.2f}** to display a floating point number with only two decimal places. The formatting code must be used in conjunction with the format() function for example:

Let's assume x = 1.0222224.

print "{:.2f}".format(x) would result in 1.02 being printed.

With this in mind see if you can manipulate the results to get something closer to the result in the subsequent screenshot.

```
Python — bash — 80×50
Marks-MacBook-Pro:Python marklassoff$ python compoundInterest.py
How much are you going to invest? 25000
How many years are you going to invest the money? 3
Input the annual interest rate as a decimal.  [For 2% enter .02] .031
Month   Interest Earned Total
1       $ 64.58         $ 25064.58
2       $ 64.75         $ 25129.33
3       $ 64.92         $ 25194.25
4       $ 65.09         $ 25259.34
5       $ 65.25         $ 25324.59
6       $ 65.42         $ 25390.01
7       $ 65.59         $ 25455.60
8       $ 65.76         $ 25521.36
9       $ 65.93         $ 25587.29
10      $ 66.10         $ 25653.39
11      $ 66.27         $ 25719.66
12      $ 66.44         $ 25786.11
13      $ 66.61         $ 25852.72
14      $ 66.79         $ 25919.51
15      $ 66.96         $ 25986.47
16      $ 67.13         $ 26053.60
17      $ 67.31         $ 26120.90
18      $ 67.48         $ 26188.38
19      $ 67.65         $ 26256.04
20      $ 67.83         $ 26323.86
21      $ 68.00         $ 26391.87
22      $ 68.18         $ 26460.05
23      $ 68.36         $ 26528.40
24      $ 68.53         $ 26596.93
25      $ 68.71         $ 26665.64
26      $ 68.89         $ 26734.53
27      $ 69.06         $ 26803.59
28      $ 69.24         $ 26872.83
29      $ 69.42         $ 26942.26
30      $ 69.60         $ 27011.86
31      $ 69.78         $ 27081.64
32      $ 69.96         $ 27151.60
33      $ 70.14         $ 27221.74
34      $ 70.32         $ 27292.06
35      $ 70.50         $ 27362.57
36      $ 70.69         $ 27433.25
Marks-MacBook-Pro:Python marklassoff$
```

6) If you need it, here is the line used to output the formatted values:

```
print x , "\t" , "$" , "{:.2f}".format(interestEarned) , "\t" , "$",
"{:.2f}".format(initialInvestment)
```