**Chapter 2 – Week 4 – Exercises**

Exercises #1 – page 37

**1. List four phases of the software development process and explain what they accomplish.**

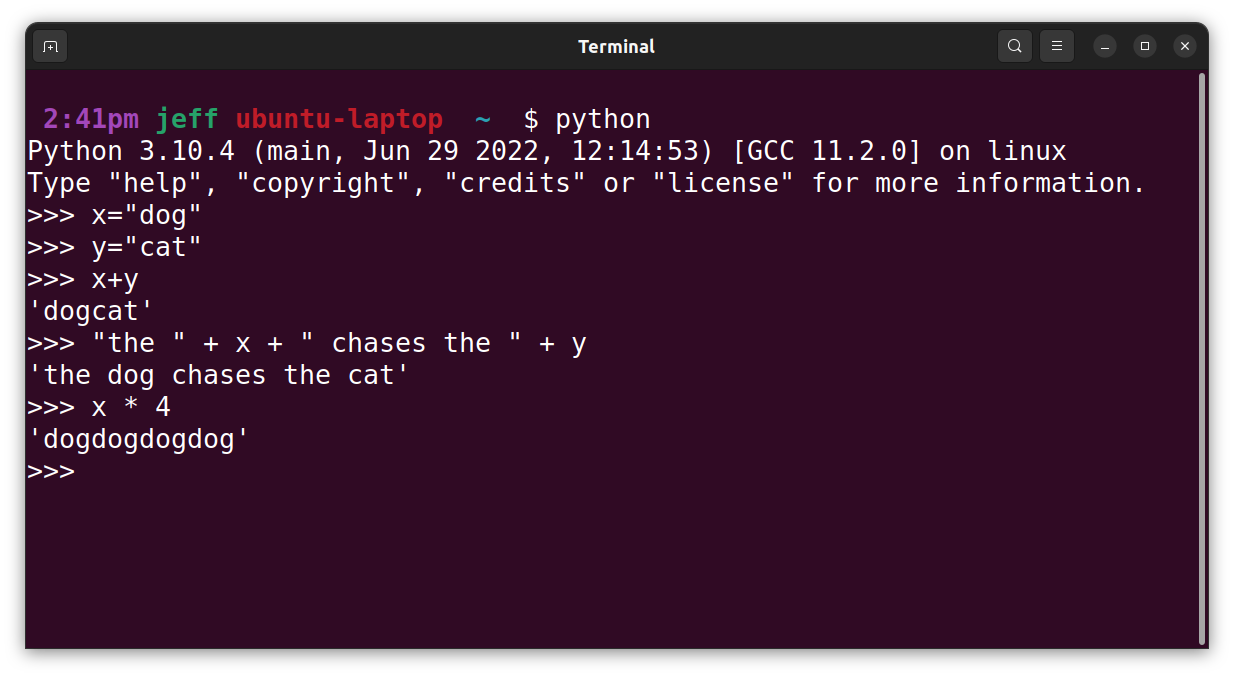
1. **Design** During this phase of development, the development team examines the problem as defined during the Analysis phase (not covered in my list) and begins to plan the coding process. This will include things like what classes, objects, routines, database tables, etc. do we need to accomplish our task.
2. **Implementation** This is the actual process of writing the code, scripts, documents, etc. needed to make the designed software work.
3. **Integration** This is when ensure that all the pieces of our code will work well together. Since many programmers generally work on a single software project and these programmers reside in different parts of the world, we must ensure that everyone wrote their code in the way the other teams expected it to be written.
4. **Maintenance** Over time, bugs will be discovered in the code, changes will be made in the world that require changes in our code, or something else will come up that will require a patch or update to our software. This process is know as maintenance.

**2. Jack says that he will not bother with analysis and design but proceed directly to coding his programs. Why is that not a good idea?**

The text correctly points out that the later in the software development process a mistake is discovered, the more difficult and costly it is to correct. By jumping in without preparation, Jack is ensuring that any mistakes that occur will not be easy to fix as if he had spent some time preparing.

Exercises #2 – page 46

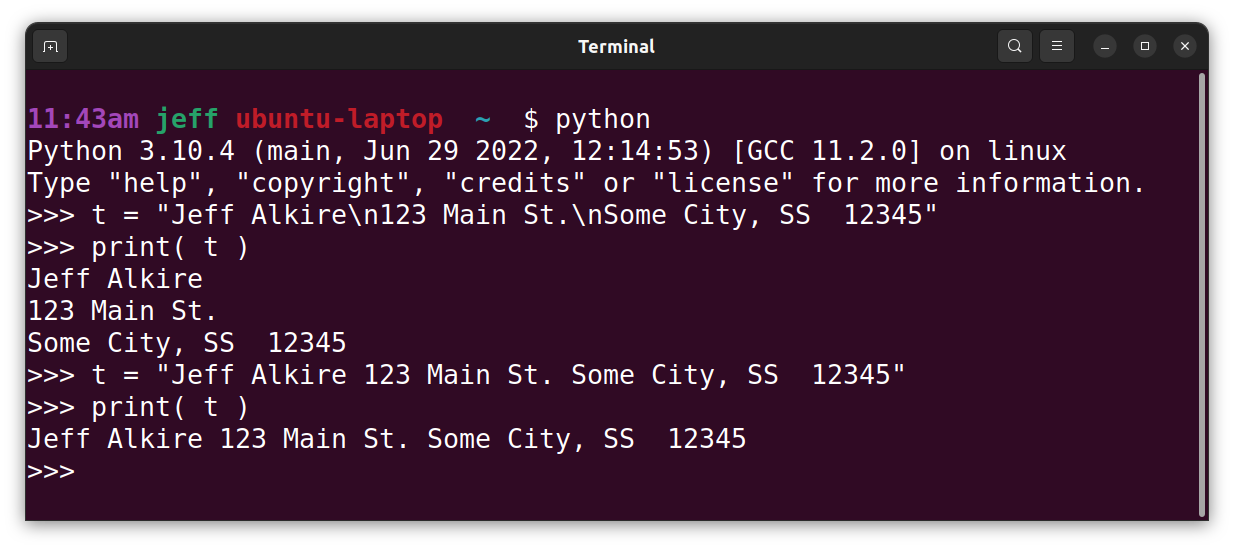
1. **Let the variable x be “dog” and the variable y be “cat. Write the values returned by the following operations:**
2. x+y ‘dogcat’
3. “the “ + x + “ chases the “ + y ‘the dog chases the cat’
4. x \* 4 ‘dogdogdogdog’



1. **Write a string that contains your name and address on separate lines using embedded newline characters. Then write the same string literal without the newline characters.**

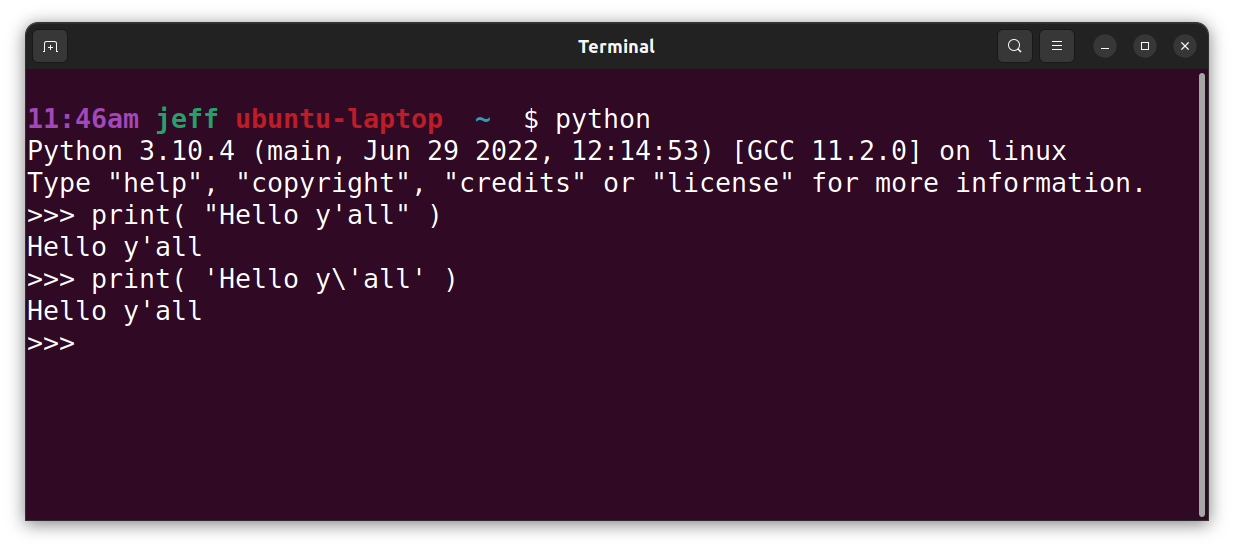
"Jeff Alkire\n123 Main St.\nSome City, SS 12345"

"Jeff Alkire 123 Main St. Some City, SS 12345"



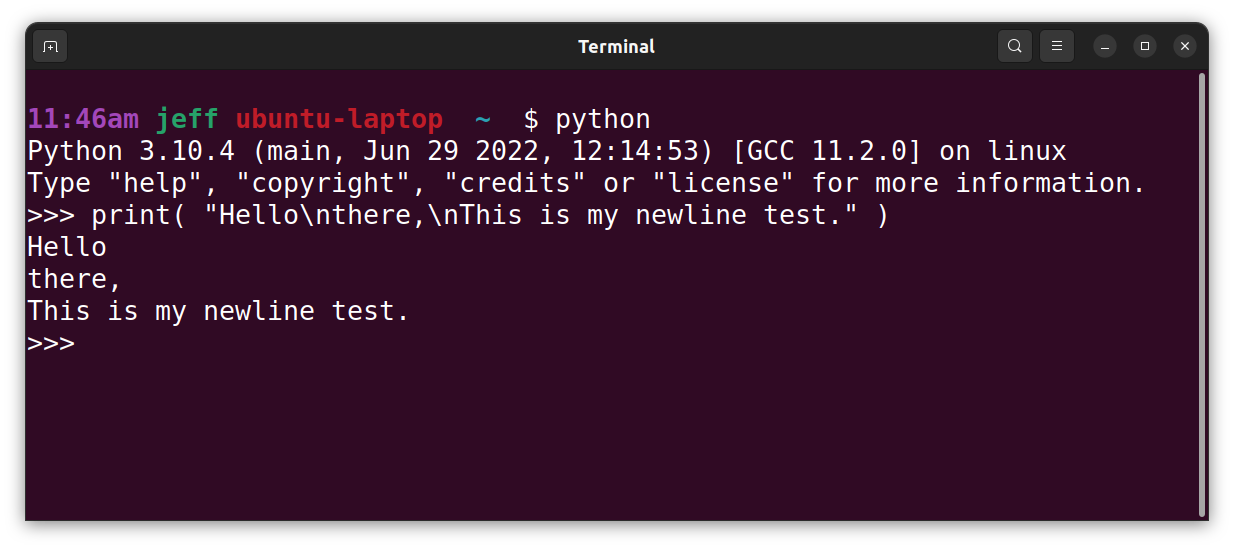
1. **How does one include an apostrophe as a character within a string literal?**

By either placing it within a double quoted (“) string literal or by escaping it with a \.

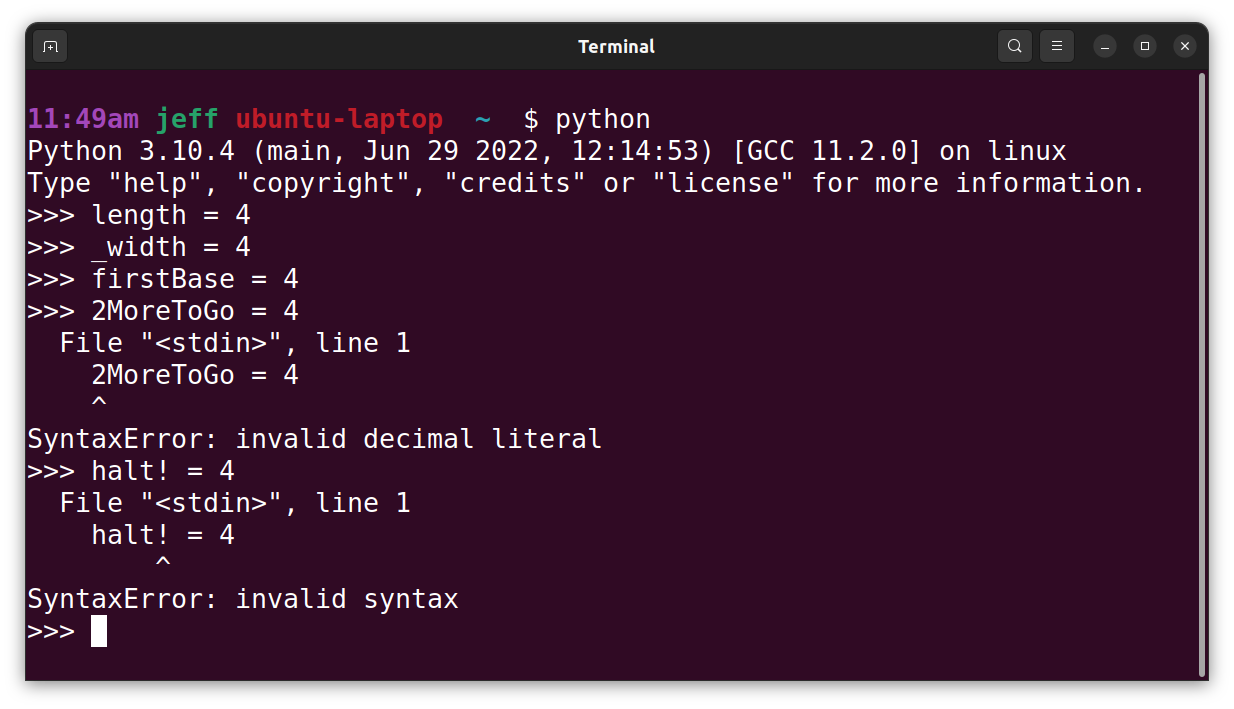


1. **What happens when the print function prints a string literal with embedded newline characters?**

The print function shows the string literal on different lines with the text broken at the embedded newline character.



1. **Which of the following are valid variable names?**
   1. length valid
   2. \_width valid
   3. firstBase valid
   4. 2MoreToGo INVALID (must start with \_ or letter)
   5. halt! INVALID (can only contain letters, numbers, or \_)



1. **List two of the purposes of program documentation.**
   1. Include big picture information such as who wrote the code, when and why. Who is maintaining the code. This can include copyright information as well as many other things sometimes dictated by the employer.
   2. To document important logic decisions, purposes of variables, what function parameters are for, and other useful information for finding and fixing bugs in code.

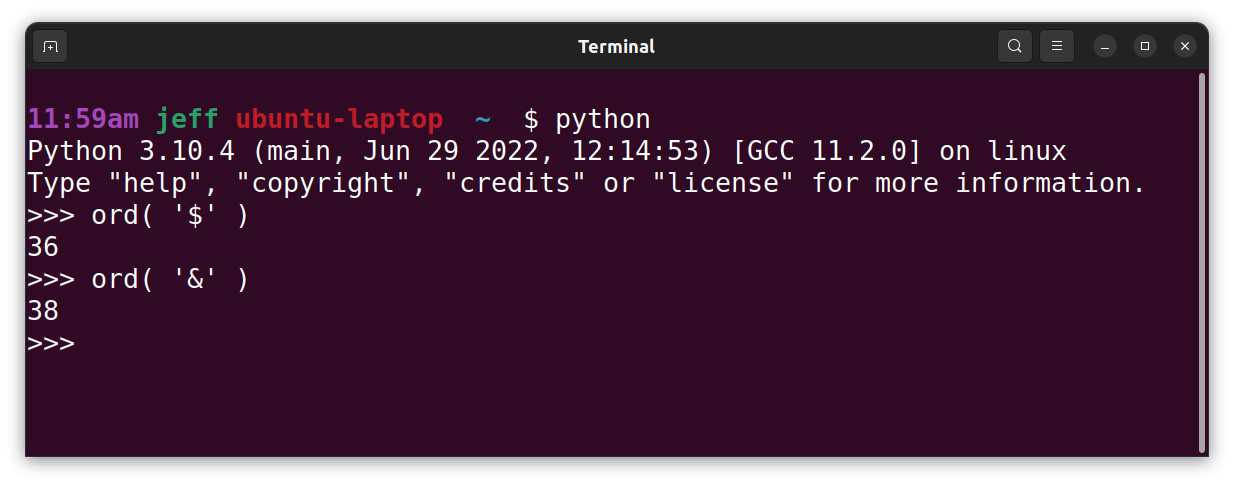
Exercises #3 – page 49

1. **Which data type would most appropriately be used to represent the following data values?**
2. The number of months in a year int
3. The area of a circle float
4. The current minimum wage float
5. The approximate age of the univers (12 billion years) int
6. Your name string
7. **Explain the differences between the data types int and float.**

Int is for integers. These are the whole numbers including both positive and negative

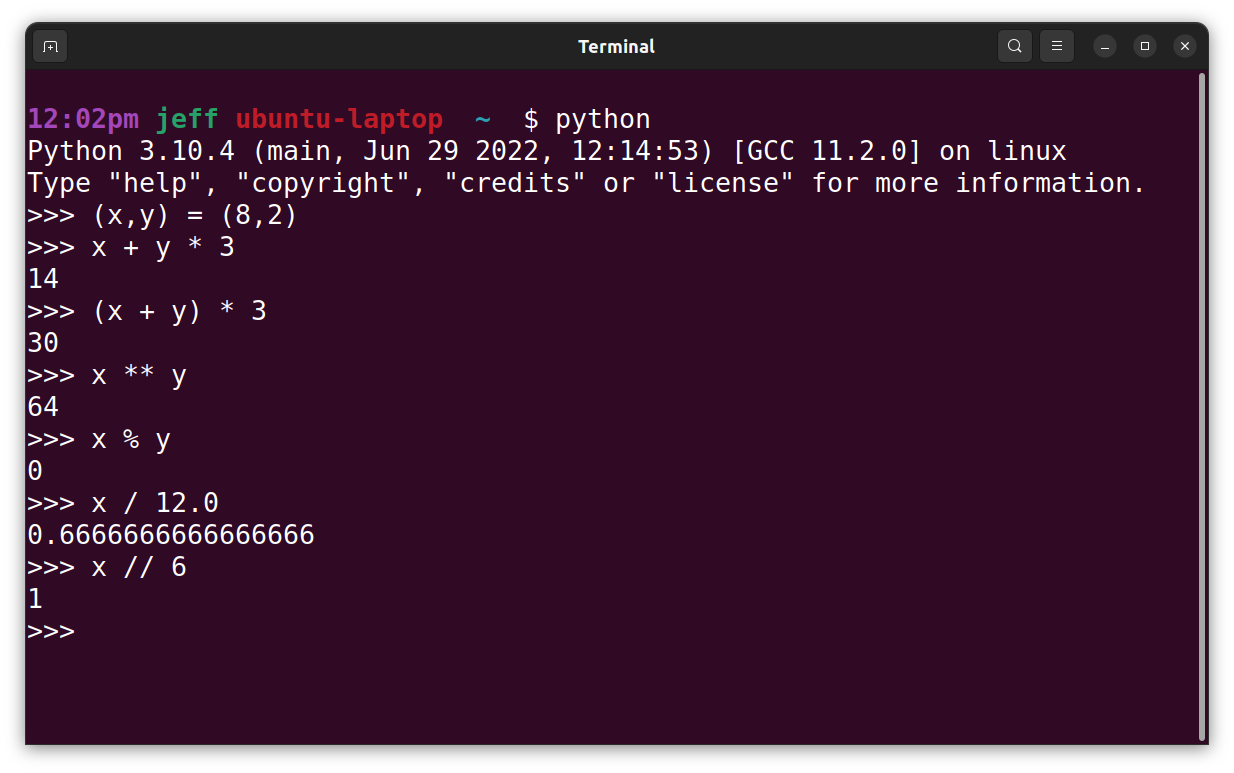
1. **Write the values of the following floating-point numbers in Python’s scientific ntoation:**
2. 355.76 3.5576e2
3. 0.007832 7.832e-3
4. 4.3212 4.3212e0
5. **Consult Table 2-5 to write the ASCII values of the characters ‘$’ and ‘&’**

36 and 38

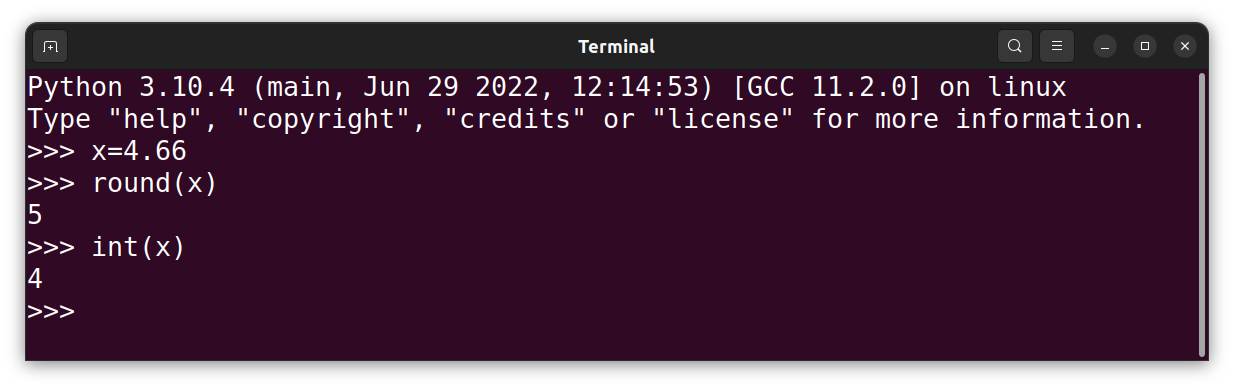


Exercises #4 – page 53-54

1. **Let x=8 and y=2. Write the values of the following expressions:**
2. x + y \* 3 14
3. (x + y) \* 3 30
4. x \*\* y 64
5. x % y 0
6. x / 12.0 0.666666
7. x // 6 1



1. **Let x = 4.66. Write the values of the following expressions:**
2. round(x) 5
3. int(x) 4



1. **How does a Python programmer round a float value to the nearest int value?**

Using the round function

1. **How does a Python programmer concatenate a numeric value to a string value?**

Using string concatenation + between the string and the string representation of the numeric value received by using the str function.

Example: “Hello Johnny “ + str(5)

1. **Assume that the variable x has the value 55. Use an assignment statement to increment the value of x by 1.**

Use either **x+=1** or **x=x+1**

Exercises #5 – page 59

1. **Explain the relationship between a function and its arguments.**

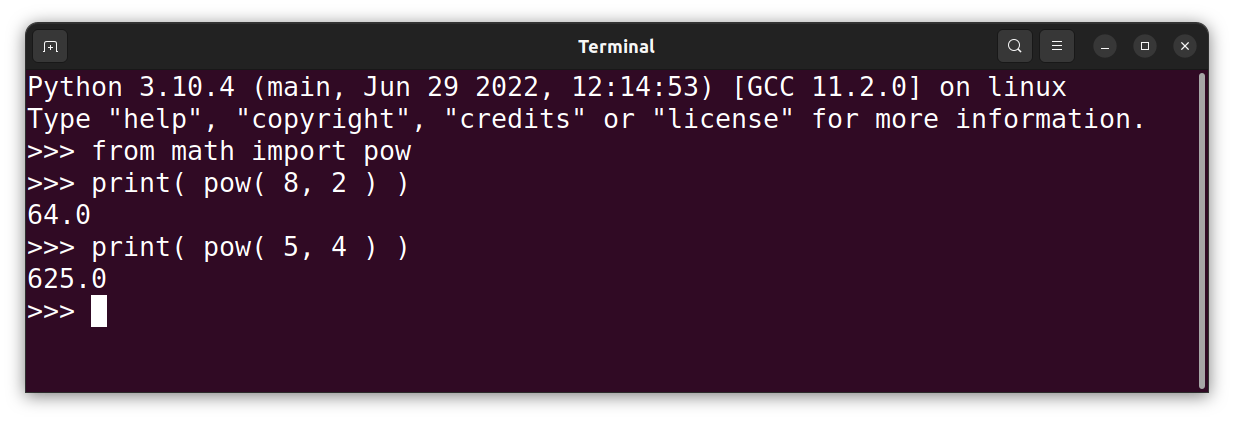
A function is a block of code. The arguments are supplied to the function to allow it to produce a result that is specific to the arguments supplied. By way of example. If the function is add, then add( 3, 5 ) should return 8. However add( 20, 100 ) should return 120. As the example shows, the function executes the same logic using different values each time it is called. The function is the logic. The arguments are the values supplied to this particular invocation of the function.

1. **The math module includes a pow function that raises a number to a given power. The first argument is the number and the second argument is the exponent. Write a code segment that imports this function and calls it to print the values 82 and 54.**

from math import pow

print( pow(8,2) )

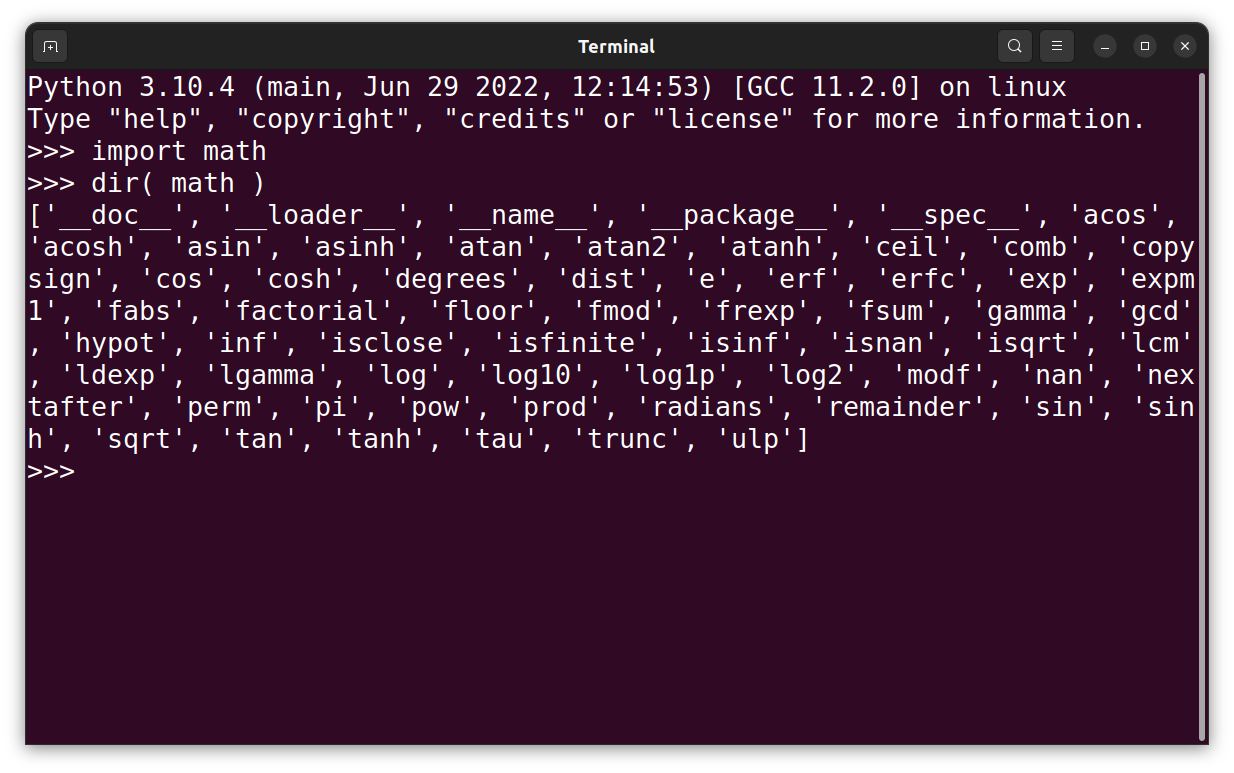
print( pow(5,4) )



1. **Explain how to display a directory of all the functions in a given module.**

A “directory” of a module list in available functions can be obtained by:

* + - 1. Opening the Python REPL (or shell)
      2. Importing the module in question, and
      3. Entering the command dir( <module name> )



1. **Explain how to display help information on a particular function in a given module.**

To get help information on a given function within a given module, one should:

* 1. Open the Python REPL (or shell)
  2. import the module in question
  3. enter the command help( <module name>.<function name>)

