CHAPTER 3

EXERCISES

[1](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1215). Why are functions advantageous to have in your programs?

For using a particular block of code for multiple times

Functions are used, with functions we can avoid

* Duplication of code.
* Copy pasting of code

[2](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1216). When does the code in a function execute: when the function is defined or when the function is called?

* The code in a function gets executed when the function is called.

[3](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1217). What statement creates a function?

* def statement.

[4](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1218). What is the difference between a function and a function call?

* A function is a block of code that’s defined using def statement and it can’t be executed automatically. For example,

def hello():

print('Hello world')

* A function call statement is used for calling the block of code to execute. It can be done after defining a particular function. We can call a function by using it’s name followed by a paranthesis. For example.

hello()

[5](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1219). How many global scopes are there in a Python program? How many local scopes?

* There is only one global scope, and it is created when the program begins. The global scope is destroyed when the program terminates
* A local scope is created whenever a function is called. Any variables assigned in the function exist within the function’s local scope. When the function returns, the local scope is destroyed. There can be more than one local scope.

[6](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1220). What happens to variables in a local scope when the function call returns?

* All the variables in a local scope will be destroyed and forgotten.

[7](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1221). What is a return value? Can a return value be part of an expression?

* the value that a function call evaluates to is called the return value of the function
* Yes, a return value can be a part of an expression.

[8](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1222). If a function does not have a return statement, what is the return value of a call to that function?

* It returns a value called None. , which represents the absence of a value.

[9](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1223). How can you force a variable in a function to refer to the global variable?

* By using the global keyword, we can refer a global variable inside a function

[10](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1224). What is the data type of None?

* NoneType Datatype

[11](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1225). What does the import areallyourpetsnamederic statement do?

It imports the areallyourpetsnamederic.py module to the current python file. And we can call the functions in that module from our current python file.

[12](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1226). If you had a function named bacon() in a module named spam, how would you call it after importing spam?

* spam.bacon( )

[13](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1227). How can you prevent a program from crashing when it gets an error?

* Exception handling statements (try and except) are used for avoiding the program from getting crashed.

[14](https://automatetheboringstuff.com/2e/chapter3/#calibre_link-1228). What goes in the try clause? What goes in the except clause?

* The code that could potentially have an error is put in a try clause.
* The program execution moves to the start of a following except clause if an error happens.

***The Collatz Sequence***

Write a function named collatz() that has one parameter named number. If number is even, then collatz() should print number // 2 and return this value. If number is odd, then collatz() should print and return 3 \* number + 1.

Then write a program that lets the user type in an integer and that keeps calling collatz() on that number until the function returns the value 1. (Amazingly enough, this sequence actually works for any integer—sooner or later, using this sequence, you’ll arrive at 1! Even mathematicians aren’t sure why. Your program is exploring what’s called the *Collatz sequence*, sometimes called “the simplest impossible math problem.”)

Remember to convert the return value from input() to an integer with the int() function; otherwise, it will be a string value.

Hint: An integer number is even if number % 2 == 0, and it’s odd if number % 2 == 1.

The output of this program could look something like this:

Enter number:  
3  
10  
5  
16  
8  
4  
2  
1

#### Input Validation

Add try and except statements to the previous project to detect whether the user types in a noninteger string. Normally, the int() function will raise a ValueError error if it is passed a noninteger string, as in int('puppy'). In the except clause, print a message to the user saying they must enter an integer.

Solution:

def Get\_num():

while True:

try:

print('Enter the number: ')

n = int(input())

return n

except:

print('Number must be an integer, Type again')

continue

def collatz(number):

if number%2==0:

return number//2

else:

return 3\*number + 1

num = Get\_num()

ans = collatz(num)

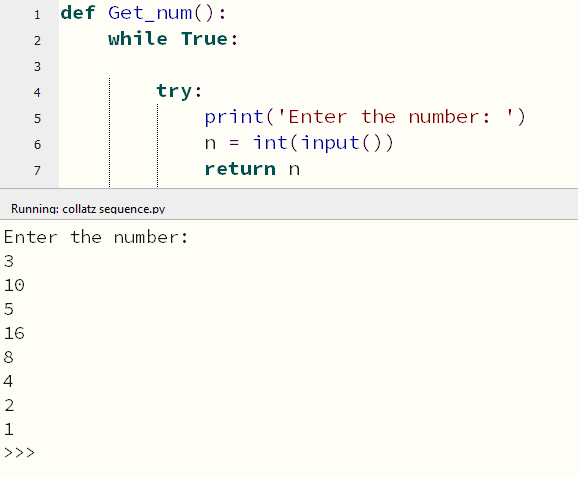
print(ans)

while ans!=1:

ans = collatz(ans)

print(ans)

OUTPUT:



Exception Handling:

Graphical user interface, text, application, email

Description automatically generated