Introduction to Python Computer Programming

North Star High School

Unit 5

Conditionals and Recursion

# Reading Material

The text for this unit is Think Python chapter 5 (pages 39 through 49).

It is encouraged to work on your reading with the Python interpreter open on your computer, so that you can type in the examples and experiment as you read.

# Guided Reading

Please complete the following questions using the assigned reading above.

1. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ operator (written as “//”) divides two numbers and rounds down to an integer, for example, 10 // 3 equals 3. (floor division)
2. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ operator (written as “%”), divides two numbers and returns the remainder, for example, 10 % 3 equals 1. (modulus)
3. A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ expression is an expression that is either true or false. (boolean)
4. Not equal (“!=”), greater than (“>”), less than (“<”), greater than or equal to (“>=”), less than or equal to (“<=”), and equal to (“==”) are \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. (relational operators)
5. There are three \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, and, or, and not. (logical operators)
6. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** give us the ability to check conditions and change the behavior of the program accordingly. (conditional statements)
7. In the following code:

|  |
| --- |
| if x > 0:  print('x is positive') |

The boolean expression following the “if” is called the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. (condition)

1. The code below demonstrates \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, (alternative execution)

|  |
| --- |
| if x % 2 == 0:  print('x is even')  else:  print('x is odd') |

and the alternative pathways of code execution are called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. (branches)

1. A function that calls itself is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (recursive), and the process of executing it is called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. (recursion)
2. The bottom of the recursion is called the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. (base case)
3. Recursion that never reaches a base case and goes on making recursive calls forever is know as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. (infinite recursion)

**Be sure that you are familiar with all of the definitions in the glossary (section 5.13)!**

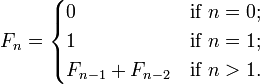
# Operator Practice

Please answer true or false to the following. You can use the Python Interpreter to check your work!

1. 7 != (3+4) (false)
2. (1+10) >= 11 (true)
3. 5 <= 9 (true)

# Assignment

1. Exercise 5.2 in the textbook. (ex\_5.2.py)
2. Exercise 5.3 in the textbook (ex\_5.3.py)
3. Write a program that prompts the user for a number, and then prints the Fibonacci sequence up to that number. Use recursion!! Note that the definition of the Fibonacci sequence is itself recursive:



(fib\_seq.py)

If time allows, please complete exercise 5.1 in the textbook. Don’t forget about the modulo (%) and floor division (//) operators!! Note that GMT (Greenwich Mean Time) is used, and not our local time zone. (ex\_5.1.py)