CS1010E Programming Methodology

Semester 2 2019/2020

Tutorial Week 02

Date: 22/01/2020 - 24/01/2020

Overview

1. Arithmetic evaluations
2. Logical evaluations
3. String evaluations
4. Turtle graphics

Python

Python supports a large number of different binary operators. Some are used for both number and string with different meaning. We can summarize the operations for integer, floating point, and string as a table below.

Operator	Integer	Floating Point	String
+	*Addition*	*Addition*	*Concatenation*
-	*Subtraction*	*Subtraction*	-
*	*Multiplication*	*Multiplication*	*String * Integer or Integer * String is the string repetition*
/	*Division*	*Division*	-
**	*Exponentiaion*	*Exponentiaion*	-
//	*Floor division*	*Floor division*	-
%	*Modulo*	*Modulo*	-
<	*Less than*	*Less than*	*Same as number but uses lexicographical ordering based on ASCII value*
>	*Greater than*	*Greater than*	
<=	*Less than equal to*	*Less than equal to*	
>=	*Greater than equal to*	*Greater than equal to*	
==	*Equality*	*Equality*	*Equality*
!=	*Non equality*	*Non equality*	*Non equality*

For each of the parts below, please try to come out with the answer without running the code first. Afterwards, you can verify your answer by running the code. The objective is for you to understand how they work.

Part 1: Arithmetic Evaluation

Expression	Output
`3 * 4 + 5`	
`3 + 4 * 5`	
`5 ** 3 % 4`	
`97 / 4`	
`97 // 4`	

Part 2: Logical Evaluation

Expression	Output
`3 + 2 == 1 + 4`	
`4 > 4`	
`True or False`	
`not False`	
`not not True`	
`not 0`	
`not 9999`	

Part 3: String Evaluation

Expression	Output
`'abc' + 'def'`	
`'gala' * 3`	
`'mu' + 'ha' * 4`	
`('ba ' * 2 + 'bidu' * 2 + 'bi' + 'jam ' * 2)*3`	
`'banana'[3]`	
`'banana'[2:4]`	
`'aaa' > 'aab'`	
`'aaa' > 'Zaa'`	

Part 4: Operator Precedence

Expression	Output
`1 + 2 * 3`	
`1 + 2 * 3 ** 4`	
`1 + 2 * 3 ** 4 - 5`	
`not 0 + 1`	

Part 5: Turtle Graphics

You can draw pictures with the turtle package in Python. Try out the following commands and guess what they will do.

Code Snippet	Output

`from turtle import *`
 	
`lt(45)`
`fd(100)`
`rt(45)`
`fd(100)`
`rt(45)`
`fd(100)`
`rt(45)`
`fd(100)`
`rt(45)`
`fd(300)`
`rt(90)`
`fd(300)`
`rt(45)`
`fd(100)`
`rt(45)`
`fd(100)`
`rt(45)`
`fd(100)`
`rt(45)`
`fd(100)`	