Practice sheet by Percy Teng <3

1. Arithmetic problems:
   1. 12/ 5 = \_\_\_\_\_\_\_
   2. 12 // 5 = \_\_\_\_\_\_\_
   3. 7 % 2 = \_\_\_\_\_\_
   4. 3 \*\* 4 = \_\_\_\_\_\_\_\_
2. Variables and Lists
   1. >>>Str\_1 = ‘hello’

>>>Str\_2 = str\_1[1]

>>>Str\_2

>>> \_\_\_\_\_\_\_\_

>>>Str\_1[1] = ‘joke’

>>>\_\_\_\_\_\_\_\_\_\_\_\_\_

>>>str\_2\*3

>>>str\_1 + str\_2

>>>int\_1 = 1

>>>float\_1 = 1.1

>>>int\_1 + float\_1

>>>\_\_\_\_\_\_\_

>>> bool\_1 = True

>>>bool\_2 = False

>>>bool\_1 and bool\_2

>>>\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>>>bool\_1 or bool\_2

>>>\_\_\_\_\_\_\_\_\_\_\_\_\_\_

>>>bool\_2 or bool\_1

>>>\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. lis\_1 = [‘hello’, ‘world’, ‘coding, ‘with’, ‘percy’]

lis\_2 =lis\_1

lis\_2.append('good stuff’)

lis\_1 = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lis\_1[2] = \_\_\_\_\_\_

lis\_1.remove(‘world’)

lis\_1 = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

lis\_1 + ‘Lie’

lis\_1 + lis\_2\*2 = \_\_\_\_\_\_\_\_\_\_\_

1. Built in function
   1. >>>print(‘hola’)

>>>\_\_\_\_\_\_\_\_\_\_\_\_

>>>max(1,2,3,4,5)

>>>\_\_\_\_\_\_\_\_\_\_

>>>max(1)

>>>\_\_\_\_\_\_\_\_\_\_\_

>>>pow(2,3)

>>>\_\_\_\_\_\_\_

>>>pow(3)

>>>\_\_\_\_\_\_\_\_

>>>pow(5,2,4)

>>>\_\_\_\_\_\_\_\_

>>>abs(-2)

>>>\_\_\_\_\_\_\_

>>>abs(3)

>>>\_\_\_\_\_\_\_

>>>abs(0)

>>>\_\_\_\_\_\_\_

>>>round(3.4999)

>>>\_\_\_\_\_\_\_\_\_\_\_

>>>round(4.5)

>>>\_\_\_\_\_\_\_\_\_\_

1. function and if-else statements

def avg(num1, num2):

“””(int, int) -> int

Return the average value of num1 and num2

>>>avg(3,5)

4

“””

def good\_student(your\_mark, target\_one, target\_two):

“””(int, int) -> bool

return true if your mark is higher than the average of target\_one’s mark and target\_two’s mark.

>>>good\_student(85, 86, 83)

True

>>>good\_student(40, 50, 55)

False

1. loops:

def motivation(grades):

””” (list of integers)->list of integers

Return a new list of IELTS grade in which each element was incremented by 1 from the old grade

>>>motivation([5,6,6.5,7])

[6,7,7.5,8]

”””

def find\_letter\_n\_times(s, letter, n):

””” (str, str, int)->(int)

Precondition: letter occurs at least n times in s

Return the smallest substring of s starting from index 0 that contains n occurrences of letter

>>> find\_letter\_n\_times('Computer Science', 'e', 2)

'Computer Scie’

“””

def sum\_of\_digits(message):

“””(str) -> int

return the sum of all digits in a string

>>>sum\_of\_digits(‘hello123’)

6

>>>sum\_of\_digits(‘brahhhhh’)

0

“””