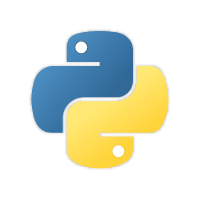
COM3103

Artificial Intelligence

1. Basic Python Programming

Welcome to Python Programming



Before we continue to discuss A.I., let us first introduce a popular AI programming langauge – Python   
Why Python?

* Good in text processing (e.g., producing HTML documents on websites)
* Useful and easy to use built-in types (lists, dictionaries)
* Easy Matrix (arrays) processing
* Support Object Oriented Programming
* Many libraries available for statistical functions: useful for machine learning



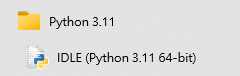
Some points to note about Python, however …

* Syntax is quite different to those of the C-family (C, C++, Java, C#)
* Python programs run slower than Java and C (in particular)
  + - Python is an interpreted language. The source codes are not translated (compiled) until run-time.

Installing & Running Python



* + We will use Python 3.1 in our lab.



* + Two modes:
    - Interactive mode
      * Running one command at a time
    - Script mode
      * Running codes from a file

What Python programs look like

x = 34 - 23 # A comment.

y = "Hello" # Another one.

z = 3.45

if z == 3.45 or y == "Hello":

x = x + 1

y = y + " World" # String concat.

print (x)

print (y)

Hello World in Python (and Java)

**// Hello World in Java.  
public class HelloWorld {  
  public static void main( String[] args ) {  
   System.out.println( "Hello World" );  
 }  
}**

**# Hello World in Python.  
print ("Hello World")**

Note: In Python, execution **starts at the first statement outside of function definitions and class definitions, main() is not required.**

Exercise 1: HelloWorld in Python

1. Run the Hello World Program in Interactive mode of IDLE
   * Start IDLE for Python



1. Open a new file (**hello.py**) in IDLE (File 🡺 New File). Type the program and run it in the script mode.

Grouping of Statements  
- In Python, statements are grouped by writing them after a colon (:) and written in the same indentation level

- The number of spaces used for indentation are not fixed by Python, but the recommended practise is to use four spaces.

**# Python.  
i=1; n=100; sum =0**

**while i <= n :  
    sum = sum +i   
    i = i + 1**print**("The total is ", sum)**

**// Java.  
int i=1, n=100, sum=0;**

**while( i <= n ) {  
  sum = sum + i;**

**i++;  
}  
System.out.println(“The total is “ + sum);**

Reserved words in Python

You cannot use these names for variables and functions.

and assert break class continue

def del elif else except

exec finally for from global

if import in is lambda

not or pass print raise

return try while

Data Types

In Python, we do not need to announce a variable's data type before using it.

**# Python  
year = 2023**

**CEO = "John Lee Ka-chiu"**

**// Java.  
int year = 2023;**

**String CEO = "John Lee Ka-chiu";**

**// Java.  
String month = "Feb";  
month = 3; // ILLEGAL!**

**# Python  
month = "Feb"  
month = 3 # Allowed**

In Python, it is possible to reassign a variable to values of another type

(This is called dynamic typing)

Basic Number Types and Operations

year = 2022 #int

888.88 #float (same as double in Java)

Mathematical Operators

+, -, \*, /, %, \*\*

Example.

x = x + 10  
y += 8

2\*\*3 # 2 to the power 3 (i.e., 8)

Note: there is no x++ in Python, use x+=1 or x=x+1 instead.

Integer division

Unlike Java, dividing two integers in Python keeps the decimal part (as we all expected before learning Java)

**// Java.  
System.out.println(6 / 4);**

**// Result is 1**

**System.out.println(6.0 / 4);**

**// Result is 1.5**

**.5**

**# Python   
print (6 / 4)   
#Result is 1.5**

**print (6.0 / 4)**

**#Result is 1.5**

Integer division is done by the // operator, e.g.,

print (6 // 4) #Result is 1

Non-numbers Data Types

bool (Same as boolean in Java)

Keywords: True, False

cck\_is\_ugly = False # similar to false in Java

cck\_is\_nice = True # similar to true in Java

if (7 < 8): # evaluate to True

print ("smaller")

Note: in Python, any non-zero number or non-empty String or List can also be used to mean "True" in a condition. Zero numbers, empty String, empty list also means False in a condition.

String

The followings are all allowed and has the same meaning

'Norwegian Wood' #Single quote

"Norwegian Wood" #Double quote

'''Norwegian Wood''' #Triple quote

""" Norwegian Wood """ #Triple quote

Example:

quote = "I don't know that you don't know."

quote\_2 = 'What does "Ngor Hmm Dzee" means?'

List

* Like arrays and arrayLists in Java, but more flexible and easier to use.
* List elements do not have to be of the same type.

#Python

days\_in\_months = [31,28,31,30,31,30,31,31,30,31,30,12]

address = [56, "Waterloo Road", "Kowloon"] #OK in Python

//Java

int[] dayInMonths = {31,28,31,30,31,30,31,31,30,31,30,12};

String [] address = {56, "Waterloo Road", "Kowloon"} // ILLEGAL in Java

Note: In Python, it is a common practice to separate words in a variable name using \_. E.g., my\_variable

Functions

Functions in Python in is somewhat similar to *static methods* in Java, except that they are not defined in classes.

Example: Built-in functions for creating numbers from other data types

int (x) Create an int from other data types

float(x) Create a float from other data types

Examples

y = int("12")

y = float ("12.34")

Math functions

Using functions defined in math package

from math import \*

y = sqrt(2) #result is 1.4142135623730951

x = abs(-1) #result is 1

r = ceil (3.5) #result is 4

s = floor (3.5) #result is 3

Standard Input and Output: String

Prompting for user input from command prompt (i.e. keyboard data entry)  
# Python (Method 1)

print ("What is your name? ")  
name = input() #Python will prompt for user input

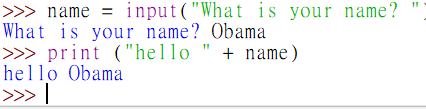
print ("hello " + name)

# Python (Method 2)

name = input("What is your name? ") #Python will prompt for user input

print ("hello " + name)

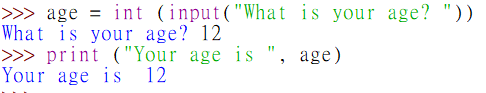
Example Interactive Session:



Standard Input and Output: Numbers

The input( ) function returns data in String. Conversion is needed if other data types are needed.

# Python



age = int (input("What is your age? "))

print ("Your age is ", age)

// Java

Scanner scanner = new Scanner( System.in );  
System.out.println( "What is your age?" );   
int age = scanner.nextInt();

System.out.println( "Your age is " + age);

Output Formatting in Python

The format( ) method of String replaces occurences of { } with its arguments

Example 1:

print("Carrie has {} votes!".format(777))

The output is

Carrie has 777 votes!

Example 2:

animal = "dog"

action = "bite"

print("Does your {} {}?".format(animal, action))

The output is

Does your dog bite?

More about Strings

Converting other data types to String using the str( ) function. E.g.,

year\_str = str(2022)

Multiline Strings can be marked by triple quotes. E.g.,

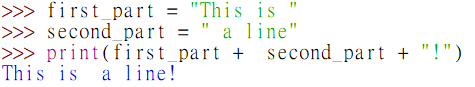
"""This is a string which

can continue onto a new line. When printed, it will appear

exactly as written between the triple quotes.

"""

String Concatenation (joining two or more Strings)

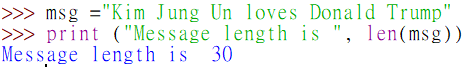


first\_part = "This is "

second\_part = " a line"

print(first\_part + second\_part + "!")

String Length

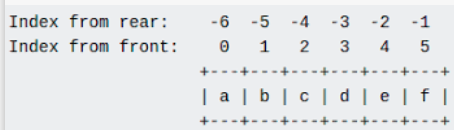


msg ="Kim Jung Un loves Donald Trump"

print ("Message length is ", len(msg))

Character index in Strings

(Similar to charAt() String method of Java)



x = "abcdef"

x[0] 🡺 a

x[1] 🡺 b

x[2] 🡺 c

x[5] 🡺 f

The last character can also be accessed using the index -1

x[-1] 🡺 f

x[-2] 🡺 e

Substrings (String Slicing)

x = "abcdef"

# 012345

*format: string\_name[starting index: ending index plus one]*

x[1:3] 🡺 bc

x[0:4] 🡺 abcd

x[2:] 🡺 cedf (note: position 2 to the end)

x[:3] 🡺 abc (from the beginning, and before position 3)

x[-3:-1] 🡺 ?\_\_\_\_\_\_\_\_\_\_\_\_?

Containing a substring

email = "abc@gmail.com"

if "@gmail.com" in email:

print("gmail accounts not accepted")

if not "@hsu.edu.hk" in email:

print("Please use school email ")

Comparing two Strings

str1 = "AbcDef"  
str2 = "Abc" + "Def"  
  
if str1 == str2 :  
   print ("Equal!!")  
else :  
   print ("Not Equal!!")

🡺 Equal!!

Note: this is not the same as in Java

Counting number of occurences of a Substring

str = "I love you but you love him more than you love me"

str.count("love")

🡺3

Exercise 2: String

1. Fix the error in the following code.

ford\_quote = 'Whether you think you can, or you think you can't--you're right.'

Answer:

1. What is the output of the following code?

coconut\_count = "34"

mango\_count = "15"

total\_fruit\_count = coconut\_count + mango\_count

print(total\_fruit\_count)

Answer:

1. Fix the above code so that the total fruit count is calculated correctly

Logical Expressions

Relational operators

The following are the same in Python and in Java

< > <= >= == !=

Boolean Operators: and or not

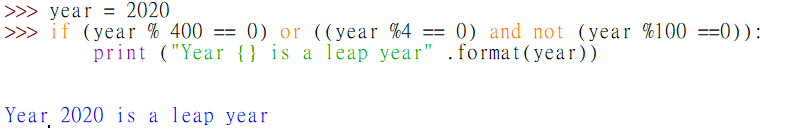
if (x >= 80) and (x <=100):

grade = "A"

if (year % 400 == 0) or ((year %4 == 0) and not (year %100 ==0)):

print ("Year {} is a leap year".format(year))

Sample run:



if … else

num1 = int(input( "Enter value one: " ))  
num2 = int(input( "Enter value two: " ))  
num3 = int(input( "Enter value three: " ))  
  
if num1 < num2 :  
   if num1 < num3 :  
      smallest = num1  
   else:  
      smallest = num3  
else:  
   if num2 < num3 :  
      smallest = num2  
   else:  
      smallest = num3

print ("The smallest value is ", smallest)

elif (else if)

if avgGrade >= 90.0 :  
   letterGrade = "A"  
elif avgGrade >= 80.0 :  
   letterGrade = "B"  
elif avgGrade >= 70.0 :  
   letterGrade = "C"  
elif avgGrade >= 60.0 :  
   letterGrade = "D"  
else:  
   letterGrade = "F"

Exercise 3: Logical Expressions#

Calculate the population densities of Hong Kong and Singapore based on the following facts.

**Save your work as density1.py**

Hong Kong population: 7507000

Singapore population: 5704000

Hong Kong area: 1106 km2

Singapore area: 728 km2

Population density is given by the formula:

density = population / area.

**Compare the two population densities using program codes**, and output the results by listing the **smaller** one first. E.g.,

AAAAAAA has population density XXXXXX

BBBBBBB has population density YYYYYY

where AAAAAAA and BBBBBBB can be either Hong Kong or Singapore, and XXXXXX and YYYYYY are the population densities.

Loop Controls

While loop

end = False

while not end:  
command = input("Please enter a command:")

if command == "end":

end = True  
print ("Good bye")

Note: there is no do…while in Python

For loop

Example 1. Print out 1 to 10

// Java

for( int i = 1; i < 11; i++ )  
  System.out.println( i );

**# Python**

for i in range( 1, 11 ):  
    print (i)

Example 2. Print out 0 to 9

# Python

for i in range(10):  
   print (i)

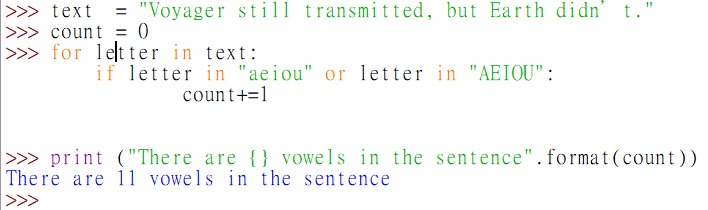
Example: Counting the number of vowels in a String

text = "[Voyager still transmitted, but Earth didn’t.](http://www.sixwordstories.net/2014/02/voyager-still-transmitted-but-earth-didnt/)"

count = 0

for letter in text:  
 if letter in "aeiou" or letter in "AEIOU":  
 count+=1

print ("There are {} vowels in the sentence".format(count))



Exercise 4: Loop Control

1. Write a **for loop** to print out all multiples of 5 between 5 and 50.

Save your work as mult5.py

1. Write a **for loop** to ask the user to enter 10 numbers, then print out the average. #

Save your work as numbers.py

1. Write a **for loop** to ask the user to enter 5 words or phrases, then count the number of words or phrases that contains the letter “x” (non-case-sensitive).

Save your work as countx.py

Functions and Scope of Variables

Define your own function

Functions can be defined using the keyword def.

Unlike methods in Java, Python functions are not part of any classes and objects.

//Java Static method

public class MyClass {

**public static int addNumber(int a, int b) {  
 int result = a+b;**

**return result;**

**}**

public static void main (String[] args) {  
 System.out.println(addNumber(2,2));

}

}

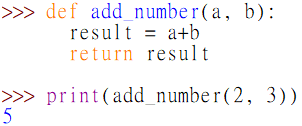
**#Python function**

**def add\_number(a, b):**

**result = a+b**

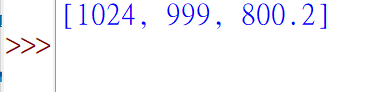
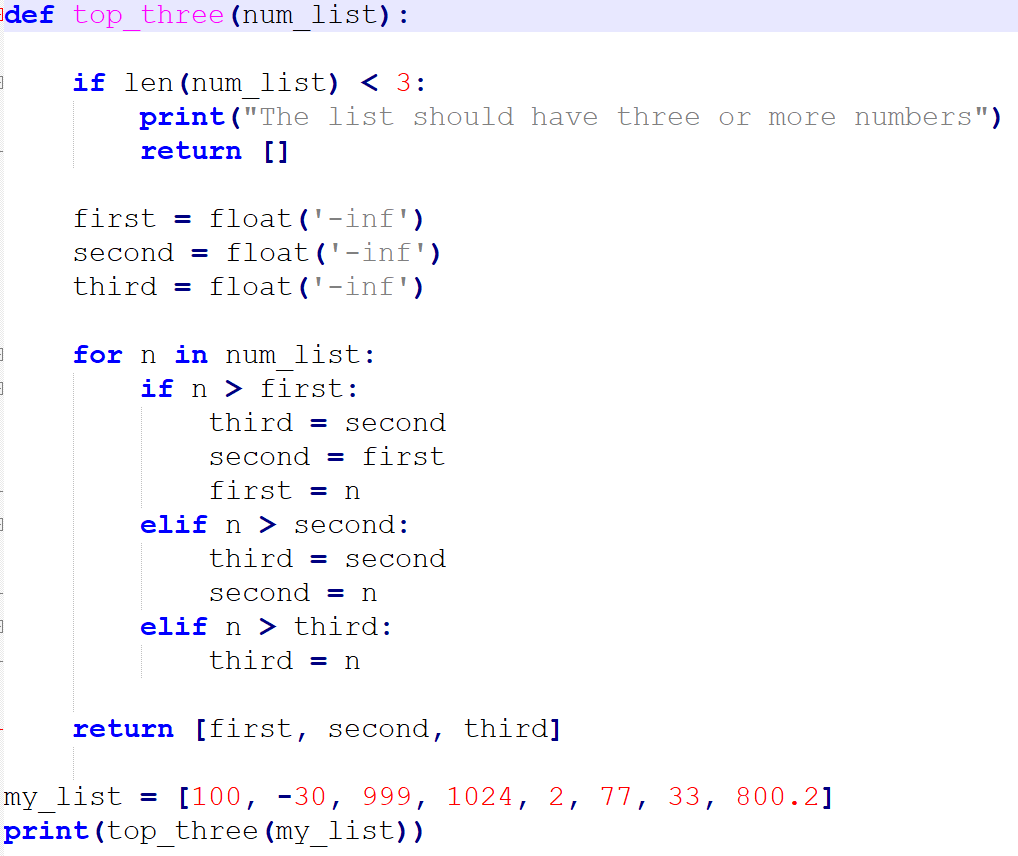
**return result**

print(add\_number(2, 3))



Note: you do not need to specify the return type in Python

**Example:** A function that returns the three largest number in a list.



Variable Scope: Global and Local Variables

*Global variables* are defined outside of any functions (or classes)

*Local variables* are defined inside a function.

Example:

a = 10

def my\_function():

b = 777

c = a + b

Here, a is a global variable. It can be accessed anywhere.

b and c are local variables. They can be accessed inside the function only.

Changing a global variable inside a function

To change the value of a global variable inside a function, you need to use the global keyword.

Example:



count = 10

def reset\_count():

global count

count = 0

reset\_count()

In this example, the global keyword is needed as count

is a global variable defined earlier.

Example: Last Stick Win

Write a program that allows a single user to play the following game with a computer.

1. The computer asks for the user to enter two numbers, namely, *n* and c.
2. The computer shows *n* sticks on the screen.
3. The computer and the user take turn to remove sticks from the screen. Each time, the highest number of sticks that can be removed is *c*
4. The side who removes that last stick win the game.

Exercise 5: Function

1. Write a function named population\_density that takes two arguments, population, and land\_area, and returns a population density calculated from those values. Part of the codes are given to you below.

**Save you work as density2.py**

#Density2.py  
#write your function here

# test cases for your function

test1 = population\_density(10, 1)

expected\_result1 = 10

print("expected result: {}, actual result: {}".format(expected\_result1, test1))

test2 = population\_density(864816, 171.4)

expected\_result2 = 5045.600933488915

print("expected result: {}, actual result: {}".format(expected\_result2, test2))

1. HSUHK amusement park has the following pricing schemes for their tickets. #

|  |  |
| --- | --- |
| **Number of Tickets** | **Pricing Scheme** |
| 1 – 5 | $100 per ticket |
| 6 – 10 | $ 95 per ticket |
| 11 or above | $ 90 per ticket |
|  |  |

Write a program **tickets.py** to calculate the total ticket fees, as entered by the user

Requirements:

* The number of tickets is entered by the user at the command prompt.
* The program should contain a function calc\_fee( ) that accepts an integer *n* as argument, where *n* is the number of tickets to be bought. The function then calculates and returns the total ticket price.

For example, calc\_fee(5) will return 500.

* After asking the users to enter the number of tickets, the program should call the function and print out the returned price accordingly.

1. A bank has the following plans for paying interests to their time-deposits accounts holders. \*\*

|  |  |  |
| --- | --- | --- |
| **Deposit amount** | **Length of Deposit** | **Interest Rate** |
| Less than $500,000 | 1 year | 2.0%. |
| 2 years | 3.0%. |
| $500,000 or more | 1 year | 4.0%. |
| 2 years | 5.0%. |

Write a program **bank.py** to calculate the interest amount according to the deposit amount and length of deposit, as entered by the user.

Requirements:

1. Ask the user to enter a deposit amount and the length of deposit (1 year or 2 years) at the command

prompt.

1. The program should contain a function calc\_interest( ) function that accepts an amount and a

deposit length as arguments. The function then calculates and returns the interest using the formula

*Interest = amount \* interest rate \* length\_of\_deposit.*

For example, calling calc\_interest(1000000, 2) will return 1000000 \* 5% \* 2 = 100000.

1. Use the function in Step 2 to compute the interest payable to the customer and print it out.
2. Display an appropriate message to ask whether the user would like to continue (Y/N). If the user chooses “Yes” (Y), proceed to step 1 and repeat the process. Otherwise, end the program.
3. Write a program **sid.py** that asks the user to enter a HSUHK student number, and print “OK” if the format is valid, and “NOT OK” if it is not. Repeat until the user choose to exit from the program

A valid HSUHK student number must begin with the lower case letter “s” and followed by 6 digits.

Requirements:

1. Ask the user to enter a Student ID.
2. The program should contain a is\_hsu\_sid( ) function that accepts an ID as a String argument. The function then returns either True or False. For example, calling is\_hsu\_sid("s200001") will return True, whereas is\_hsu\_sid("3190001") and is\_hsu\_sid("s 200001") will return False.
3. Use the function in step 2 to validate the ID entered by the user in step 1, and print out the result (i.e., “OK” or “NOT OK”)
4. Display an appropriate message to ask whether the user would like to continue (Y/N). If the user chooses “Yes” (Y), proceed to step 1 and repeat the process. Otherwise, end the program.

Tips:

The String method isdigit() could be useful. For example, given the codes

a = "abc123"

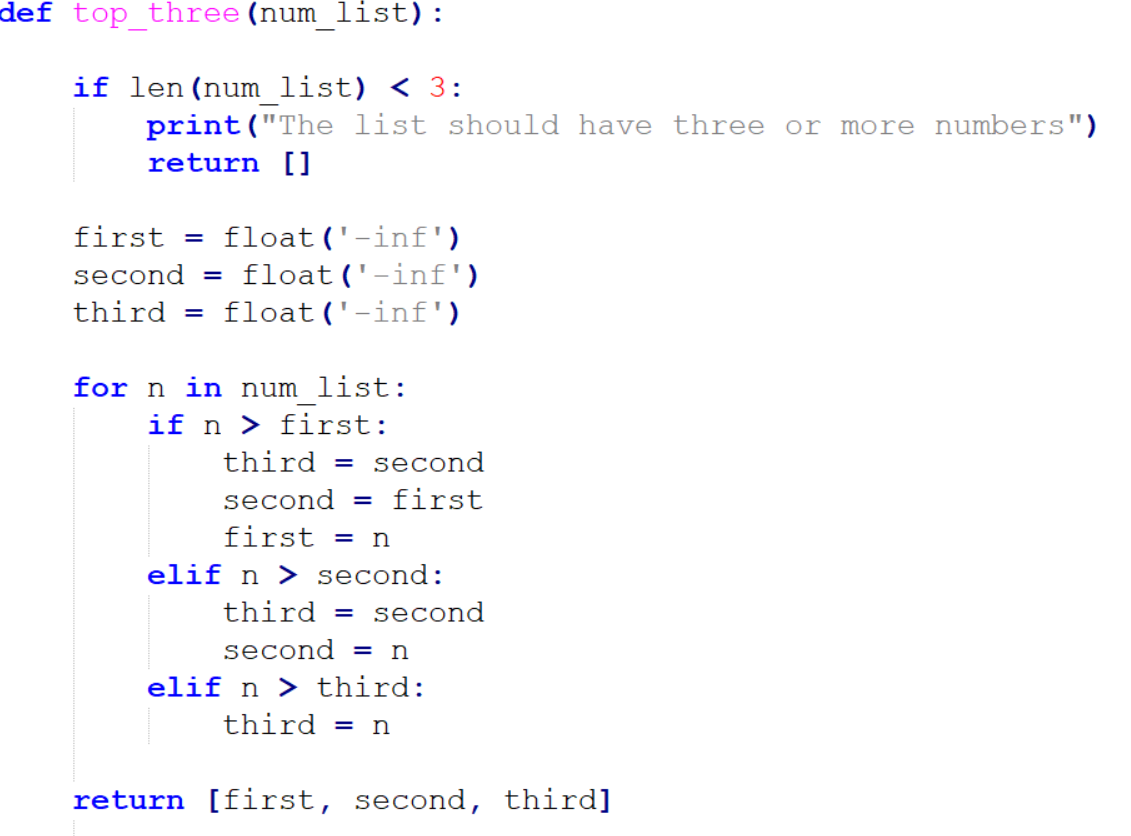
b = "456"

a.isdigit() returns False, whereas b.isdigit()returns True

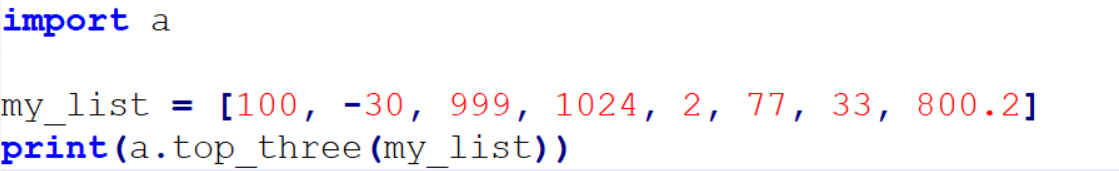
Importing functions from other modules.

Suppose that you want to use a function that is written in another python program file (i.e., another module). For example, given the previous example in a module named **a.py**:

**a.py**



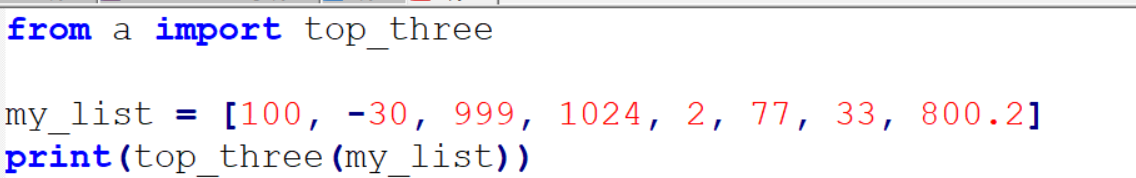
Suppose that we wish to use the top\_three function in another program **b.py**, we can import it as follows:



**b.py**

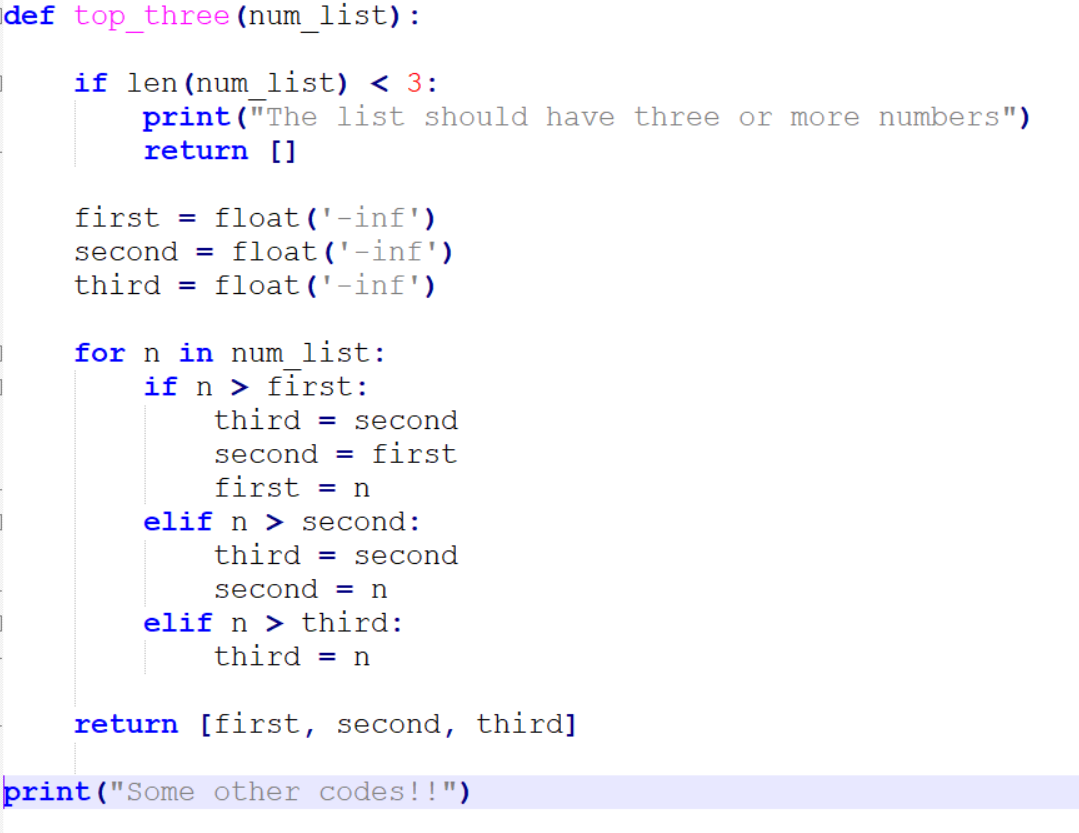
Alternatively, we can import a specific function by writing:

**b.py**



Note that, however, when you import another module, if there are any executable code (outside the functions) in the imported module, they will be run as well!

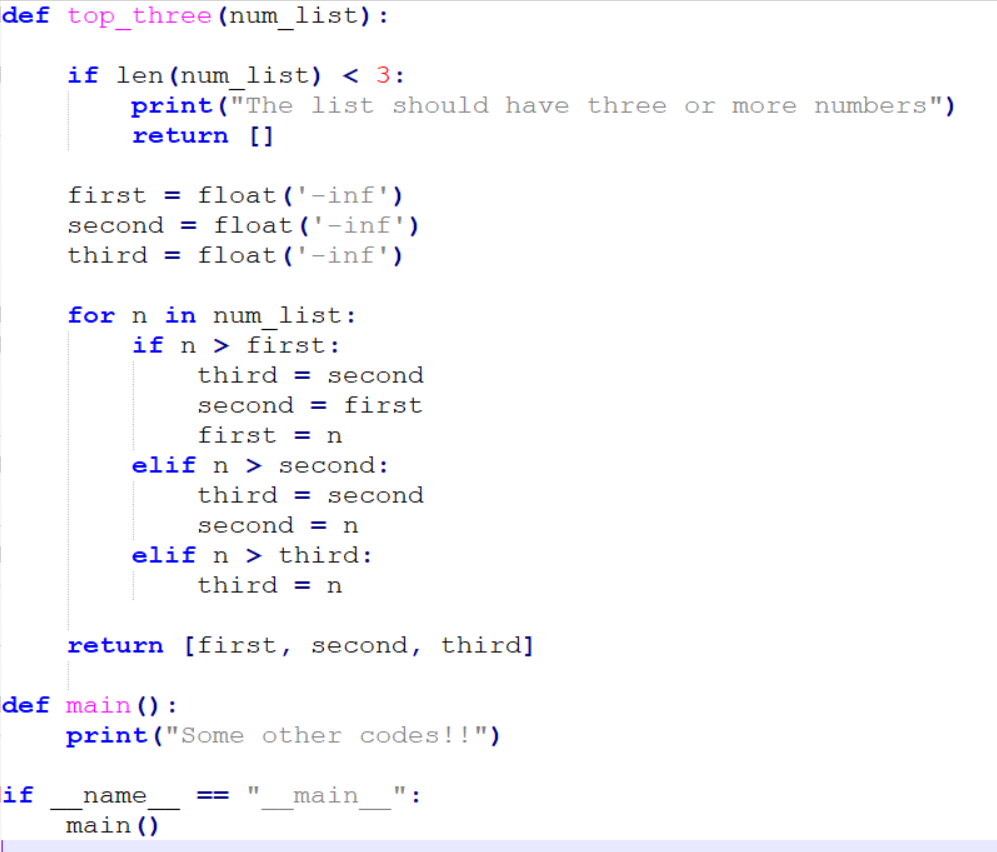
**a.py**



This line will also be run if another module imports the module **a**!

To avoid the code being run when imported into another module, we usually group those code into a "main" function, as follows:

**a.py**



This time, the red part is only run if the module **a** is run directly, but not when it is imported into another module.

Exercise 6: importing function

Repeat exercise 5d, but this time, write **is\_hsu\_sid( )** function in another module, and import it into your main program. I.e.,

Write a program **sid2.py** that asks the user to enter a HSUHK student number, and print “OK” if the format is valid, and “NOT OK” if it is not.

A valid HSUHK student number must begin with the lower case letter “s” and followed by 6 digits.

Requirements:

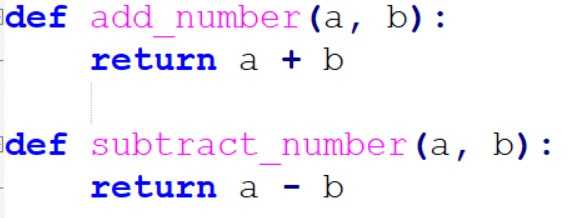
1. In the **sid2** module, ask the user to enter a Student ID.
2. In another module (choose a suitable program file name for it), write your is\_hsu\_sid( ) function that accepts an ID as a String argument. The function then returns either True or False. For example, calling is\_hsu\_sid("s200001") will return True, whereas is\_hsu\_sid("3190001") and is\_hsu\_sid("s 200001") will return False.
3. Import your new module (that contains is\_hsu\_sid( ) function) into the **sid2** module.
4. In sid2 module, use the imported function to validate the ID entered by the user in step 1, and print out the result (i.e., “OK” or “NOT OK”).

Note: you do not need to repeat the processing in this question.

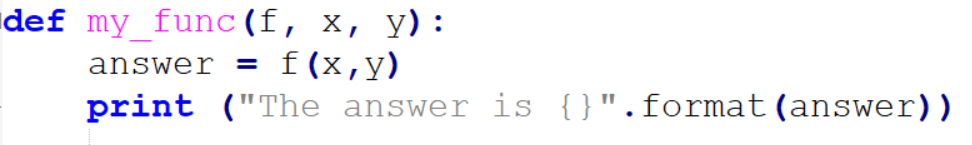
Passing functions as arguments

In python, it is possible to pass a function as arguments for another function.

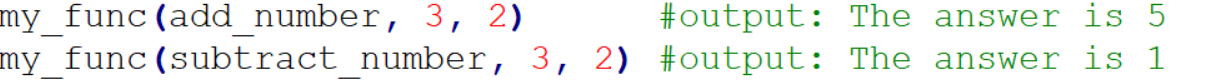
For example, given the following two functions:



And a third one below. Note that the argument **f** here is a function that can be called.



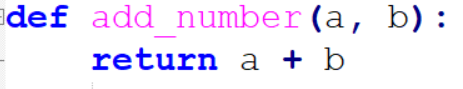
And we can call **my\_func** as follows:



Lambda function

A *lambda function* is a temporary one-line function.

Instead of written a normal function such as



We can also write it as a lambda function:

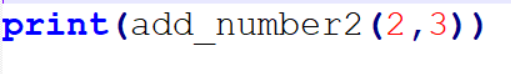


Function name (optional)

What it returns

Arguments

And we can call it as in a normal function:



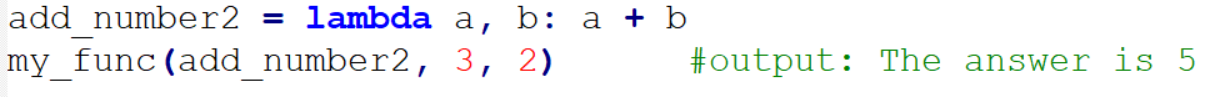
Or we can pass it as an argument, just like before:



Using Lambda function as argument

We can also use a lambda function directly without naming it first.

Instead of:



We can also write:

