SOFTWARE LAB FOR BASIC SCIENTIFIC PROBLEM SOLVING: ITCS159

Lab1: Basic Python Programming

SNIT SANGHLAO*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

All the laboratories each week will take 2 hours of working through a worksheet of example problems. To gain credit for the labs, you will be asked to explain and demonstrate your results with the lab assistant.

Note that:

- Each lab consists of a number of practical exercises, all of which are required for marks to be attained. In particular, you should try to accomplish the 3 labelled milestones.
- Attendance is compulsory and will be monitored on Mycourse. If you are unable to attend a session you must inform a tutor in advance.

During the assessments, your understandings will be tested, and the tutor will not give the marks if you do not display an understanding of the problem and its solution. The tutors are there to help you think through any issues — but they won't write the programs for you!

The normal rules of assessment apply: if you can't make a laboratory or miss one, you must have a good reason. If the problem is medical then a doctor's note must be presented as evidence. You should attempt to make up for lab absences in your own time.

Finally, you will be allowed to submit work in the lab a week after it was given. The work for this lab (lab1) can be submitted at the start if the lab next week (lab2). However, the marks will not be awarded beyond the next lab, meaning lab1 will not be marked during lab 3, 4 and so on.

Get start with Python

Python is a popular interpreted language that easy to learn and practice. To get start with Python, your computer need to install Python interpreter. In this class, Python on Powershell as show on "How to Install and Setup file" provides all basic environment for running Python including stand library to learn Python programming. In the first week, we will look at the basic python programming which will be used as a tool to solve scientific problem as follow.

LABORATORY 1

As we mentioned earlier that Python is an interpreted language. There are two basic mode for using Python. First is the normal mode (commonly used), which is about creating script file ".py" where Python statements are written for a specific task and run this file in Python interpreter. Second is the Python interactive mode, which is a command line shell where each of input statement will be executed and gives immediate feedback after submission. To introduce the basic statement of Python language, Python Interactive Shell will be discussed first.

PYTHON INTERACTIVE SHELL

With Python Interactive Shell, it is easy to learn and check Python commands. If you have setup Python PATH to environment of your operating system, you can invoke Python Interactive Shell by launching the Command Prompt by press `Windows + R` keys, then type `cmd` and press "return". Next, typing `python` without any parameter and followed by the "return" key at the shell prompt. If everything correct, Python will come back with the following information.

```
Python 3.9
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Follow the steps to install and test Python in "How to install Python, you should enter to the Python Interactive shell.

Once the Python interactive shell is started, you can issue any command at the command prompt »>. To quit the python shell, you can type `exit()` or Press (Ctrl+Z). Let's get started!!!

PRINTING

The very first basic thing to start a new programming language is to display some text on the screen. Let's go through the following exercise.

EXERCISE 1) You can now enter some code for python to run:

```
>>> print("Hello world")
```

What does the `print()` do?. What's it in python?

EXERCISE 2) Now, try to enter the following code in console to print Hello World and *Your name*.

```
>>> print(
... "Test123"
... );
```

What is happened and why?.

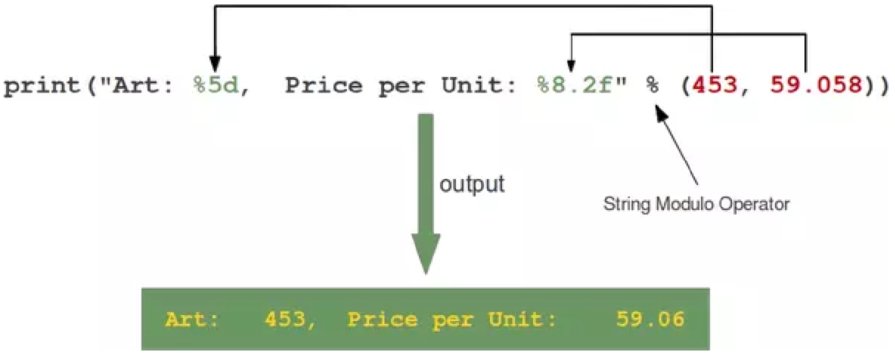
Let's try another example below:

```
>>> print("Hello world"); \
... print("My name is <YourName>");
```

What does the \ do?.

The general syntax for a format placeholder is

```
%[flags][width][.precision]type
```



```
print("Art: %5d,  Price per Unit: %8.2f" % (453, 59.058))
```

```
Art:   453,  Price per Unit:    59.06
```

Figure 1: Process image (from https://python-course.eu/python-tutorial/formatted-output.php).

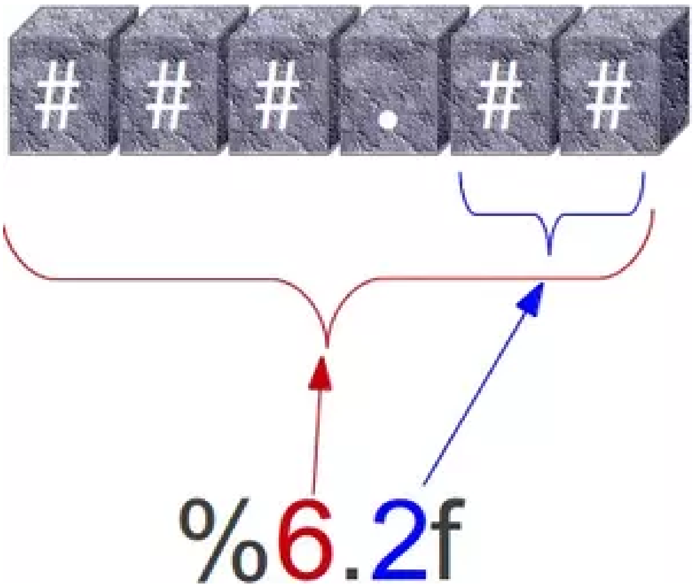


Figure 2: Process image (from https://python-course.eu/python-tutorial/formatted-output.php).

STRING

String in Python can be created by putting any text inside quotes. The previous section illustrates the use of `print()` to display **strings** on the console.

EXERCISE 3) **Strings** can be combined together as follow.

```
>>> print("Hello world" + "This is python class")
```

Python does not add any whitespace between strings. What should we do?

EXERCISE 4) Next, try the following command.

```
>>> print(len("Hello world" +" "+"This is python class")); \
... print(len("Helloworld"))
```

What are the outputs and what does the `len()` do?

EXERCISE 5) Now, write the code to calculate summation of lengths of two String outputs from Exercise 4 together. What is the result?

MILESTONE 1) You have now completed the Milestone1. You should get sign off by your lab assistant.

VARIABLES

Variable is used to store **value** in a given name. The purpose is for reusing the name later to refer to the value.

EXERCISE 6) In python, variable can be initialized and assigned the value by:

```
 >>> fruit = "Mango";
 >>> print(fruit);
```

Once the variable is assigned the value, it can be updated by setting it to a new value

```
 >>> fruit = "Orange";
```

What is the output?

EXERCISE 7) Can we do the following commands? Why or Why not?

```
>>> print(animal);
>>> animal = "Cat";
```

NUMBER

In python, it is easily to manage Numbers. Let's try the following Exercise

EXERCISE 8) Try printing out the following numbers

```
>>> print(5);\
>>> print(1.55);\
>>> print(True);
```

Every element in Python has it own **type** that determine the operation supported by the element as well as it range of possible value. The following list show basic data types in python:

- `bool` True or False.
- `int` Whole number value such as 3.4141 or -2.234.
- `float` Numeric value with the decimal point such as 587 or -45.
- `str` String of characters inside the quote (usually double quote is used)

EXERCISE 9) Try the following code.

```
>>> x = "100";\
>>> y = "250";\
>>> print(x+y);
```

What is the result? Is it correct? Many times number are kept inform of **String** values. The conversion is needed before use mathematical operation.

- `int(x)` Convert x to an **integer** number (discard all fragment parts).
- `float(x)` Convert x to a **floating-point** number.
- `str(x)` Convert x to a **String**.

Can you edit code in Exercise 9 and make it display the correct output?

EXERCISE 10) Next, Let's try the following command.

```
>>> year = 2018 ; faculty_name = 'ICT';
>>> print("The faculty of {faculty_name} year {year}");
>>> print(f"The faculty of {faculty_name} year {year}");
```

What are the outputs and what does the `f` do?

EXERCISE 11) Determine the output of the following code:

```
>>> x = 10.25
>>> y = "15"
>>> print(int(x), int(y))
>>> y = int(x)
>>> x = int(y)
>>> print(x, y)
>>> x = 5
>>> print(x, y)
```

MILESTONE 2) You have now completed the Milestone2. You should get sign off by your lab assistant.

NOTE: On following file manipulation there is difference PATH notation for where file is place or refer for difference Operation system.

- Windows: "Drive:\folder\ subfolder" such as "C:\Users\Bob" in Python argument "ïs special character so add on more such as in Python to refer to "C:\\User\\Bob\\itcs155 "
- MacOS: "/user/bob/itcs159" can be apply in Python file operation without modification

READ AND WRITE FILES

Python also provides feature to deal with files as other programming languages. The basic concept of file writing is dealing with file object by using the `open`, `read`, `write`, and `close` methods. Let's try the following example, to learn the basic command for read and write files.

EXERCISE 12) At the Python shell, you can now enter some code for python to run:

```
>>> import os
>>> os.system('mkdir c:\\itcs159');
>>> file = open("c:\\itcs159\\testfile.txt", "w");
>>> file.write("Hello, This is my file");
>>> file.write("line 2: test 2");
>>> file.write("line 3: test 333");
>>> file.write("line 4: test 4444");
>>> file.write("\n--- nextline ---");
>>> file.close();
```

What are the outputs and what does the `open` and `write` command do? What is the number after typing `file.write` command mean?

EXERCISE 13) Next, try another following command

```
>>> import os
>>> file = open("c:\\itcs159\\testfile.txt", "r");
>>> read_1 = file.read();
>>> read_2 = file.read(5);
>>> read_3 = file.readline();
>>> read_4 = file.readline();
>>> file.close();
```

Then, use `print()` command to display all variable named `read_1`, `read_2`, `read_3`, and `read_4`. What are the output? How each read command are different?

EXERCISE 14) Next, try another following command

```
>>> file = open("c:\\itcs159\\testfile.txt", "a");
>>> file.write("\n -- append --");
>>> file.close();
```

Then, use `print()` command to display the file information. What are the different between open file with 'w', 'r', and 'a' parameter?

PYTHON SCRIPT

As we mentioned at the beginning, there is a common way of writing python code. The python script is about writing a set of Python statement into a file with extension ".py". Then, execute it to get the results.

To start the Python script, you can open any text editor, for instant, Notepad (Later, next week we will introduce a proper Python editor the Jupyter notebook). Then go through the following Exercises.

EXERCISE 15) Write down the following code in Python editor. Then save as 'firstProg.py' on 'C:\itcs159' (Note that: at the bottom of the save as page, don't forget to select all file instead of txt file).

```
print("This is my first Python program");
num1 = 2; num2 =3;
sum=num1+num2;
print(f"The result of {num1}+{num2}={sum}");
```

Next, launching the Command Prompt by press `Windows + R` keys, then type Powershell. Next, typing `'cd c:\itcs159'` without any parameter and followed by the "return" key at the shell prompt.

START EXECUTING PYTHON SCRIPT At the command prompt, you should type `python c:`
`itcs159`
`firstProg.py`. The output should be:

```
This is my first Python program
The result of 2+3=5
```

EXERCISE 16) Create a new Python script file named 'lab01.py'. Write a program to convert Fahrenheit to Kelvin and Kelvin to Fahrenheit. Ask user to 'Enter temperature in Fahrenheit :' then convert to Kelvin. Then verify your code by convert Kelvin back to Fahrenheit:-

```
    Fahrenheit to Kelvin  = (T(°F) + 459.67)× 5/9
```

Test input -459.67 °F = 0 Kelvin.

MILESTONE 3) You have now completed the Milestone3. You should get sign off by your lab assistant.