



Temasek Junior College
2022 JC1 H2 Computing
Practical 7 – Iteration Control Structures in Python I

Session Objectives

By the end of this session, you will learn:

- (i) the use of the `for` loop in Python.
- (ii) the use of the `while` loop in Python.

For this practical session, we shall make use of a Jupyter Notebook to document the exercises.

- Download the Jupyter Notebook **Practical_7.ipynb** from Google Classroom.
- Submit your file through the Practical 7 Google Classroom link at the end of the session.

§7.1 Performing Iteration in Python

In Python, loops are used to repeatedly execute a block of statements so long as some conditions remain true.

There are two types of loops in Python:

- `for` loop;
- `while` loop.

The use of loops is important to ensure that statements that need to be repeatedly executed are coded once only and not repeatedly. This reduces unnecessary code length.

§7.2 The `for` loop

A `for` loop provides a way to perform actions for all items in an **iterable**.

An iterable can be a string, a list, a range of values, a tuple, a dictionary etc.

The general syntax structure of a `for` loop is as follows:

```
for <item> in iterable:
    statement
```

Exercise 1

Type the following code in a new cell of your Jupyter Notebook.

```
# for loop (range with upper limit only, default lower limit is 0).
for i in range(10): # loop condition where i < 10.
    print("Computing is fun")
```

Exercise 2

Type the following code in a new cell of your Jupyter Notebook.

```
# for loop (printing 0 to 4).
for i in range(0, 5): # loop condition where 0<=i<5.
    print(i)
```

Exercise 3

Type the following code in a new cell of your Jupyter Notebook.

```
# for loop (to understand how range works with step size)
# loop condition where 0 <= i < 30 and increases by 3 each iteration
for i in range(0, 30, 3):
    print(i)
```

Exercise 4

Can you implement the code in **Exercise 3** using the modulo (%) operator instead of a step size?

Type your code in a new cell of your Jupyter Notebook.

Exercise 5

Can you implement a code that counts down from 10 to 0 using the concept of step size in a range?

Type your code in a new cell of your Jupyter Notebook.

Exercise 6

Type the following code in a new cell of your Jupyter Notebook.

```
# for loop (summation of values from 1 to n)
total = 0
n = 100

for i in range(1, n+1): # use n+1 since upper limit is not included.
    # add the value of i in current iteration to the existing total
    total = total + i
print(total) # print total after final iteration
```

Exercise 7

Can you implement a code that finds the product of all integers from 1 to 100?

Type your code in a new cell of your Jupyter Notebook.

Exercise 8

Type the following code in a new cell of your Jupyter Notebook.

```
# for loop iteration of a string
for letter in 'Python':
    print(letter)
```

§7.3 The while loop

The Python while loop operates on the following principle:

As long as the condition remains True, the statement will be executed repeatedly until the condition is no longer True.

The general syntax structure of a while loop is as follows:

```
while <condition>:
    statement
```

Exercise 9

In **Exercise 2**, we used the for loop to print the integer values 0, 1, 2, 3 and 4.

We shall now print the same values by implementing a while loop.

Type the following code in a new cell of your Jupyter Notebook.

```
# while loop (printing 0 to 4)
i = 0 # initialize
n = 5
while i < n:
    print(i)
    # i is set to increase by 1 at the end of each iteration.
    i = i + 1
```

Exercise 10

Type your code in a new cell of your Jupyter Notebook. Use a new cell for each part of the question.

Can you implement a while loop that finds the

- (i) sum of all integers from 1 to 100?
- (ii) product of all integers from 1 to 100?

Mini Hackathon – 30 minutes

In the same Jupyter Notebook, complete the questions in this mini hackathon.

**You are reminded to use meaningful identifiers and identifier conventions adhering to PEP 8.*

**You are reminded to also include the use proper commenting in your program*

Problem 1

Write a program to prompt the user for x positive integers, one at a time. Compute and output the sum of all x integers entered.

Problem 2

Write an enhanced version of the program in **Problem 1** to also compute and output the average of x positive integers.

Problem 3

Write a program to prompt the user for x positive integers, one at a time, until the user terminates with -1. Compute and output the sum of all x integers entered.

Problem 4a (Tutorial 2.4 Q6 of Problem Solving and Algorithm Design 2)

Write a program to prompt the user for a positive integer n, compute and output n factorial, using for loop.

Problem 4b (Tutorial 2.4 Q6 of Problem Solving and Algorithm Design 2)

Write a program to prompt the user for a positive integer n, compute and output n factorial, using while loop.

Problem 5

Create a number guessing game that will perform the following

- Generate at random a secret integer value from 1 to 100
- Allow user to make a guess
 - If the guess is correct, congratulate the user
 - If the guess is too small, prompt the user to guess a larger value.
 - If the guess is too large, prompt the user to guess a smaller value.

To generate a random value, think of how you can use the follow statements:

```
import random
# x is the lower limit, y is the upper limit
random.randint(x, y)
```

You may search the internet on the use of the above two statements.