Discovery Piscine

Module3 - Python

Summary: *In this Module3 we will see how to use loops.*

Version: 1.00

Contents

Chapter I

A word about this Discovery Pool

Welcome !
You will begin a Module of this Discovery Piscine of computer programming. Our goal is to introduce you to the code behind the software you use daily and immerse you in peer learning, the educational model of 42.

Programming is about logic, not mathematics. It gives you basic building blocks that you can assemble in countless ways. There is no single "correct" solution to a problem—your solution will be unique, just as each of your peers' solutions will be.

Fast or slow, elegant or messy, as long as it works, that's what matters! These building blocks will form a sequence of instructions (for calculations, displays, etc.) that the computer will execute in the order you design.

Instead of providing you with a course where each problem has only one solution, we place you in a peer-learning environment. You'll search for elements that could help you tackle your challenge, refine them through testing and experimentation, and ultimately create your own program. Discuss with others, share your perspectives, come up with new ideas together, and test everything yourself to ensure it works.

Peer evaluation is a key opportunity to discover alternative approaches and spot potential issues in your program that you may have missed (consider how frustrating a program crash can be). Each reviewer will approach your work differently—like clients with varying expectations—giving you fresh perspectives. You may even form connections for future collaborations.

By the end of this Piscine, your journey will be unique. You will have tackled different challenges, validated different projects, and chosen different paths than others—and that's perfectly fine! This is both a collective and individual experience, and everyone will gain something from it.

Good luck to all; we hope you enjoy this journey of discovery.

Chapter II

Introduction

What this Module will show you:

- You will learn how to do some loops.

Chapter III

General instructions

Unless otherwise specified, the following rules apply every day of this Piscine.

- This document is the only trusted source. Do not rely on rumors.

- This document may be updated up to one hour before the submission deadline.

- Assignments must be completed in the specified order. Later assignments will not be evaluated unless all previous ones are completed correctly.

- Pay close attention to the access rights of your files and folders.

- Your assignments will be evaluated by your fellow Piscine peers.

- All shell assignments must run using `/bin/bash`.

- You <u>must not</u> leave any file in your submission workspace other than those explicitly requested by the assignments.

- Have a question? Ask your neighbor on your left. If not, try your neighbor on your right.

- Every technical answer you need can be found in the `man` pages or online.

- Remember to use the Piscine forum of your intranet and Slack!

- Read the examples thoroughly, as they may reveal requirements that aren't immediately obvious in the assignment description.

- By Thor, by Odin! Use your brain!!!

Chapter IV

Exercise 00: Up to 25

	Exercise 00
	Let's go up to 25!
Turn-in directory: *ex00/*	
Files to turn in: `to25.py`	
Allowed functions: `All`	

- Create a program called `to25.py`.

- Ensure the program is executable.

- the program should:

  - Accepts user input, which will be stored in a numeric variable.
  - Use a loop to display all numbers from en entered number up to 25.
  - If the input number is greater than 25, display "Error" followed by a new line.

```
?> ./to25.py
Enter a number less than 25
45
Error
?> ./to25.py
Enter a number less than 25
20
Inside the loop, my variable is 20
Inside the loop, my variable is 21
Inside the loop, my variable is 22
Inside the loop, my variable is 23
Inside the loop, my variable is 24
Inside the loop, my variable is 25
?>
```

Use a while loop.

Chapter V

Exercise 01: multiplication_table

![icon]	Exercise 01

The return of multiplication tables
Turn-in directory: *ex01/*
Files to turn in: `multiplication_table.py`
Allowed functions: `All`

- Create a program called `multiplication_table.py`.

- Ensure the program is executable.

- The program should:

  ○ Accepts user input, which will be stored in a numeric variable.

  ○ Display the multiplication table for that number (e.g., if the input is 2, display the multiplication table for 2).

```
?> ./multiplication\_table.py
Enter a number
8
0 x 8 = 0
1 x 8 = 8
2 x 8 = 16
3 x 8 = 24
4 x 8 = 32
5 x 8 = 40
6 x 8 = 48
7 x 8 = 56
8 x 8 = 64
9 x 8 = 72
?>
```

Chapter VI

Exercise 02: i_got_that

	Exercise 02
	Do you got it?
Turn-in directory: *ex02/*	
Files to turn in: `i_got_that.py`	
Allowed functions: `All`	

- Create a program called `i_got_that.py`.

- Ensure the program is executable.

- The program should:

  ○ Use a `while` loop that continuously accepts user input and responds with "I got that! Anything else?" after each input.

  ○ The loop should only stop when the user enters "STOP".

```
?> ./i_got_that.py
What you gotta say? : Hello
I got that! Anything else? : I like ponies
I got that! Anything else? : stop...
I got that! Anything else? : STOP
?>
```

 While, break.

Chapter VII

Exercise 03: advanced_mult

	Exercise 03
	The Return of the Return of Multiplication Tables

Turn-in directory: *ex03/*
Files to turn in: `advanced_mult.py`
Allowed functions: `All`

- Create a program called `advanced_mult.py`.

- Ensure the program is executable.

- The program should:

  ○ Display all multiplication tables from 0 to 10 in the following format:

```
?> ./advanced_mult.py
Table of 0: 0 0 0 0 0 0 0 0 0 0 0
Table of 1: 0 1 2 3 4 5 6 7 8 9 10
Table of 2: 0 2 4 6 8 10 12 14 16 18 20
Table of 3: 0 3 6 9 12 15 18 21 24 27 30
Table of 4: 0 4 8 12 16 20 24 28 32 36 40
Table of 5: 0 5 10 15 20 25 30 35 40 45 50
Table of 6: 0 6 12 18 24 30 36 42 48 54 60
Table of 7: 0 7 14 21 28 35 42 49 56 63 70
Table of 8: 0 8 16 24 32 40 48 56 64 72 80
Table of 9: 0 9 18 27 36 45 54 63 72 81 90
Table of 10: 0 10 20 30 40 50 60 70 80 90 100
?>
```

- You are only allowed to use two while loops.

Chapter VIII

Submission and peer-evaluation

- You must have `discovery_piscine` folder at the root of your home directory.

- Inside the `discovery_piscine` folder, you must have a folder named `module3`.

- Inside the `module3` folder, you must have a folder for each exercise.

- Exercise 00 must be in the `ex00` folder, Exercise 01 in the `ex01` folder, etc.

- Each exercise folder must contain the files requested in the assignment.

⚠️ Please note, during your defense anything that is not present in the folder for the module will not be checked.