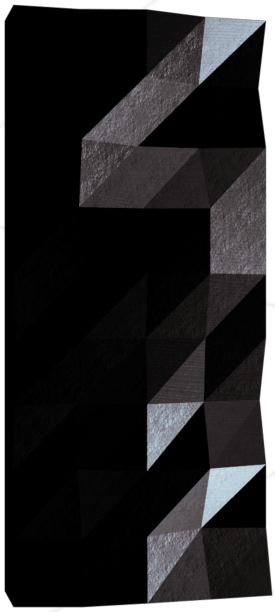

Discovery Piscine

Module4 - Python

Summary: In this Module4, you will learn how to manipulate simple objects.

Version: 2.0

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 handle simple objects.

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 must not 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: UPCASE_IT

	Exercise 00
See it in Capital	
Turn-in directory: *ex*00/	
Files to turn in: `upcase_it.py`	
Allowed functions: `All`	

- Create a program called `upcase_it.py`.
- Ensure this program is executable (pay attention to permissions).
- This program should:
  - Prompt the user to enter a word.
  - Display the word in uppercase.
- Example:

```
?> ./upcase_it.py
Give me a word: banana
BANANA
?>
```

`Search for "upcase" in Google.`

Chapter V

Exercise 01: Age

	Exercise 01
Retrieve and Modify a Number	
Turn-in directory: *ex*01/	
Files to turn in: `age.py`	
Allowed functions: `All`	

- Create a program called `age.py`.
- Ensure this program is executable.
- This program should:
  - Prompt the user to enter their age.
  - Display the user's current age and their age in 10, 20, and 30 years.

```
?> ./age.py
Please tell me your age: 15
You are currently 15 years old.
In 10 years, you'll be 25 years old.
In 20 years, you'll be 35 years old.
In 30 years, you'll be 45 years old.
?>
```

`Search for "string to integer conversion" in Python.`

Chapter VI

Exercise 02: Calculator

	Exercise 02
Performing Basic Operations	
Turn-in directory: $ex02/$	
Files to turn in: `calculator.py`	
Allowed functions: `All`	

- Create a program called `calculator.py`.
- Ensure this program is executable.
- The program should:
  - Prompt the user to enter two numbers.
  - Store these numbers as numeric values in two variables.
  - Display the result of adding, subtracting, dividing, and multiplying these numbers.

```
?> ./calculator.py
Give me the first number: 10
Give me the second number: 2
Thank you!
10 + 2 = 12
10 - 2 = 8
10 / 2 = 5
10 * 2 = 20
?>
```

Chapter VII

Exercise 03: Decimal

	Exercise 03
Working with Floating-Point Numbers	
Turn-in directory: $ex03/$	
Files to turn in: `float.py`	
Allowed functions: `All`	

- Create a program called `float.py`.
- Ensure this program is executable.
- This program should:
  - Prompt the user to enter a number.
  - Determine if the entered number is a decimal or not and display the result.

```
?> ./float.py
Give me a number: 42
This number is an integer.
?>

?> ./float.py
Give me a number: 42.00
This number is an integer.
?>

?> ./float.py
Give me a number: 42.42
This number is a decimal.
?>
```

`Google string to float.`

Chapter VIII

Exercise 04: I Don't Like Commas

	Exercise 04
I Don't Like Commas	
Turn-in directory: $ex04/$	
Files to turn in: `round_up.py`	
Allowed functions: `All`	

- Create a program called `round_up.py`.
- Ensure this program is executable.
- This program should:
  - Prompt the user to enter a number.
  - Display the number rounded up.

```
?> ./round_up.py
Give me a number: 41.42
42
?>

?> ./round_up.py
Give me a number: 42
42
?>

?> ./round_up.py
Give me a number: 0.001
1
?>
```

Chapter IX

Exercise 05: Uppercase and Lowercase

	Exercise 05
The Merge of Uppercase and Lowercase	
Turn-in directory: $ex05/$	
Files to turn in: `up_low.py`	
Allowed functions: `All`	

- Create a program called `up_low.py`.
- Ensure this program is executable.
- This program should:
  - Prompt the user to enter a string.
  - Display the string with uppercase letters changed to lowercase and vice versa.

```
?> ./up_low.py
Hello World
hELLO wORLD
?>
?> ./up_low.py
aaaaAAAA
AAAAaaaa
?>
?> ./up_low.py
hello 42
HELLO 42
?>
```

Chapter X

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 `module4`.
- Inside the `module4` 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.`