



# Discovery Piscine

## Module2 - Python

*Summary: In this Module2 we will see how to use conditions.*

*Version: 1.00*

# Contents

<b>I</b>	<b>A word about this Discovery Pool</b>	<b>2</b>
<b>II</b>	<b>Introduction</b>	<b>3</b>
<b>III</b>	<b>General instructions</b>	<b>4</b>
<b>IV</b>	<b>Exercise 00: Am I Null?</b>	<b>5</b>
<b>V</b>	<b>Exercise 01: Am I Negative?</b>	<b>6</b>
<b>VI</b>	<b>Exercise 02: Passwords</b>	<b>7</b>
<b>VII</b>	<b>Exercise 03: Multiplication</b>	<b>8</b>
<b>VIII</b>	<b>Submission and peer-evaluation</b>	<b>10</b>

# 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 conditions.

# 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: Am I Null?

	Exercise 00
Am I Null?	
Turn-in directory : <i>ex00/</i>	
Files to turn in : <b>iszero.py</b>	
Allowed functions : All	

- Create a program called `iszero.py`.
- Ensure this program is executable (pay attention to the file permissions).
- When executed, the program should prompt the user to enter a number
- If the number is equal to zero, the program should display: "This number is equal to zero."
- If the number is not equal to zero, the program should display: "This number is different from zero."


```
?> ./iszero.py
42
This number is different from zero.
?> ./iszero.py
0
This number is equal to zero.
?>
```



There's a small line to add at the beginning of your file to let the shell know how to interpret the the script. Search for "Making a Python Script Executable".

# Chapter V

## Exercise 01: Am I Negative?


	Exercise 01
Am I Negative?	
Turn-in directory : <i>ex01/</i>	
Files to turn in : <b>isneg.py</b>	
Allowed functions : All	

- Create a program called **isneg.py**.
- Ensure this program is executable.
- When executed, the program should prompt the user to enter a number.
- If the number is negative, the program should display "This number is negative."
- If the number is positive, the program should display "This number is positive."
- If the number is equal to zero, the program should display "This number is both positive and negative."

```
?> ./isneg.py
42
This number is positive.
?> ./isneg.py
-42
This number is negative.
?> ./isneg.py
0
This number is both positive and negative.
?>
```

# Chapter VI

## Exercise 02: Passwords

	Exercise 02
Access denied	
Turn-in directory : <i>ex02/</i>	
Files to turn in : <b>password.py</b>	
Allowed functions : A11	

- Create a program called `password.py`.
- Ensure this program is executable.
- Define a variable containing a password.

```
password = "Python is awesome"
```


- When executed, the program should prompt the user to enter a password.
- If the password is correct, the program should display: "ACCESS GRANTED".
- If the password is incorrect, the program should display: "ACCESS DENIED".

```
?> ./password.py
1234
ACCESS DENIED
?> ./password.py
Python is awesome
ACCESS GRANTED
?>
```



# Chapter VII

## Exercise 03: Multiplication

	Exercise 03
Multiplication	
Turn-in directory : <i>ex03/</i>	
Files to turn in : <code>mult.py</code>	
Allowed functions : All	

- Create a program called `mult.py`.
- Ensure this program is executable.
- When executed, the program should prompt the user to enter 2 numbers.
- The program should:
  - Display whether the result of multiplying the two numbers is positive, negative, or zero.
  - Display the result of the multiplication.

```
$ ./mult.py
Enter the first number:
42
Enter the second number:
42
42 x 42 = 1764
The result is positive.
```

```
$ ./mult.py
Enter the first number:
78
Enter the second number:
-1
78 x -1 = -78
The result is negative.
```

```
$ ./mult.py
Enter the first number:
72
Enter the second number:
0
```

```
72 x 0 = 0  
The result is positive and negative.
```

# 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 `module2`.
- Inside the `module2` 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 day will not be checked.