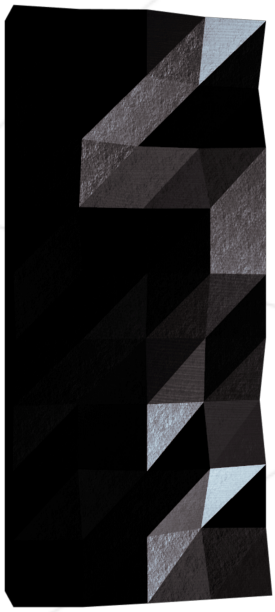

Discovery Piscine

Rush - Python

Summary: *How far we've come.*

Version:

Contents

Chapter I

A word about this Discovery Pool

Welcome !

You will begin the first cell of this discovery pool of computer programming. We want to both show you what the code is that makes up the software you use every day, and at the same time experience peer-learning, an educational model of 42.

Programming involves logic (not math). It provides you with elementary bricks, which you assemble as you wish. There is never THE solution to a problem. There will be your solution, there will be those of each of your neighbors. Slow or fast, ugly or beautiful, if that gets the job done, that's all it takes! This assembly of bricks will constitute a series of orders (calculation, display, ...) that the computer will perform, in the order you have chosen.

Rather than giving you a course with only one solution for each problem, and which will probably be outdated in a few years, we have chosen to put you in a peer-learning situation. You are going to look for the elements that could serve you for your challenge, sort out those that are actually interesting by testing and manipulating them, and create your own program. To do this, discuss with others, exchange your points of view, find new ideas together, and finally test for yourself even to convince you that it works.

Peer-evaluation is a key moment to discover other ways of doing things, as well as special cases that you have not thought of and that could undermine your program (think about your degree of nervousness with software which crashes). Like different clients who don't pay attention to the same things, each reviewer will be different from the last. And who knows, you might have made new acquaintances for later collaborations.

At the end of this pool, you will not have done the same things as the other participants, you will not have validated the same projects, you will have chosen to do one challenge rather than another ... .and that's normal! It's both a collective and a personal experience. Everyone will benefit from what he or she experiences during this time.

Good luck to all, we hope you will like this discovery.

Chapter II

Introduction

What this cell will show you:

- Creating a complex class with methods and variables.

Chapter III

General instructions

Except for specific exceptions, the following rules will be in effect for the entire duration of the piscine.

- This subject is the only reliable source. Do not believe rumors.
- Pay attention to the access permissions of your files and folders.
- Your projects will be evaluated by your fellow Piscine participants.
- You must not include any files or folders in the submission that are not explicitly requested by the subject.
- Have questions? Ask the person on your left. If not, ask the person on your right.
- Any technical question can be resolved using the `man` or through the Internet.
- Read the examples carefully. They may reveal details about the exercise that are not explicitly mentioned in the subject.
- In the examples, you will often find `"?>"`, which represents the terminal prompt. In your case, the prompt will appear with your workstation number `"c1r1p1"`.
- By Thor, by Odin! Use your brain!!!

Chapter IV

Exercise 00: Phonebook

- Create the script `Phonebook.py`.
- This script will contain the declaration for the Phonebook class.
- This class will contain a `dict` type attribute that will hold the phone numbers of the contacts as keys and the names as values
- You will also need to implement some methods to allow you to interact with the contacts:
- The methods will be called: `add_contact`, `remove_contact` and `contact_list`
- When executed the script will prompt the user for the next thing to do: ADD, REMOVE, LIST or CLOSE. With each calling their respective method or closing the program
- The add_contact method will ask the user to insert a new contact by first asking the phone number, then the name and adding them to the phonebook dict
- The remove_contact method will ask for a phone number and if it exists remove it from the dict
- The contact_list method will print out the contacts stored in the dict

```
?> python3 whatsyourname.py
what do you want to do next (ADD, REMOVE, LIST, CLOSE): ADD
phone number: 4242424242
name: Pasquale
what do you want to do next (ADD, REMOVE, LIST, CLOSE): LIST
{'4242424242': 'Pasquale'}
what do you want to do next (ADD, REMOVE, LIST, CLOSE): REMOVE
phone number: 4242424242
what do you want to do next (ADD, REMOVE, LIST, CLOSE): LIST

what do you want to do next (ADD, REMOVE, LIST, CLOSE): CLOSE
?>
```

Look up classes and their implementation in python.

Note how the script never closes by itself, the only way to close it will be the CLOSE command. search for "while true"

Chapter V

Bonus

If you are up to the task we ask you to implement some extra functionalities to your class

- A new method that allows you to interact with the contacts in a different way, be it edit them or something else
- Parsing for the methods that exist. checking if the input given is actually a number, names can't have numbers in them, etc...
- Improving the data storage for the class to make it so it also accepts email and surname of the person, how you manage it is up to you

Chapter VI

Submission and peer-evaluation

- In the `discovery_piscine` folder at the root of your home, create a new `Rush` folder and navigate to it.
- Utilize this folder for the exercise

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