



# Pre-open Piscine

## Pre open 02

*Summary: This document is the subject for the Pre open 02 module of the Pre-open Piscine @ 42Tokyo.*

# Contents

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>Foreword</b>	<b>3</b>
<b>III</b>	<b>Exercise 00 : 42 API</b>	<b>4</b>
<b>IV</b>	<b>Exercise 01 : Discord Bot</b>	<b>5</b>
<b>V</b>	<b>Bonus for Discord Bot</b>	<b>7</b>

# Chapter I

## Instructions

- Only this page will serve as reference; do not trust rumors.
- Watch out! This document could potentially change up to an hour before submission.
- These exercises are carefully laid out by order of difficulty - from easiest to hardest. We **will not** take into account a successfully completed harder exercise if an easier one is not perfectly functional.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for every exercise.
- Your exercises will be checked and graded by your fellow classmates.
- Moulinette is very meticulous and strict in its evaluation of your work. It is entirely automated and there is no way to negotiate with it. So if you want to avoid bad surprises, be as thorough as possible.
- Exercises in Shell must be executable with `/bin/sh`.
- You cannot leave any additional file in your directory than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called `Google / man / the Internet / ....`
- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...
- If no other explicit information is displayed, you must assume the following versions of languages : `Python >= python3.8.2`, `Ruby >= 2.3.7`


# Chapter II

## Foreword

"Yesterday It worked"

# Chapter III

## Exercise 00 : 42 API

	Exercise 00
42 API	
Turn-in directory : <i>ex00/</i>	
Files to turn in : <i>ft_api.py</i> or <i>ft_api.rb</i> or both	
Allowed functions : *	

Create a script which will print out the user information on terminal.


- Your script will accept one argument which will be the login.
- If the user exist, print out `login`, `first_name`, `last_name`, `id`, `email` on the terminal.
- If the user doesn't exist, then print out 'No such user!' on the terminal.
- The format for the output is shown below. please follow this format.

```
$>python3 ft_api.py nop
login: nop
first_name: Ryota
last_name: Yaoi
id: 59728
email: nop@42tokyo.jp
$>python3 api.py this_user_doesnt_exist
No such user!
$>python3 ft_api.py
$>
```

```
$>ruby ft_api.rb nop
login: nop
first_name: Ryota
last_name: Yaoi
id: 59728
email: nop@42tokyo.jp
$>ruby ft_api.rb this_user_doesnt_exist
No such user!
$>ruby ft_api.rb
$>
```

# Chapter IV

## Exercise 01 : Discord Bot

	Exercise 01
Discord Bot	
Turn-in directory : <i>ex01/</i>	
Files to turn in : <code>discord_bot.py</code> or <code>discord_bot.rb</code> or both	
Allowed functions : *	

I guess you know how to use discord. If you don't know how to use it then now it's the right time to get used to it.

- Create your own discord server.
- Create a script which will act as a bot on your own discord server.
- when a user type `/covid JPN` it will print out yesterdays data retrieved from the api <https://covidapi.info> with japan as the country.
- Your bot must show the number of **confirmed**, **deaths** and **recovered**.
- You are free to add more detail or changed the output format but don't forget to show the 3 values above.
- Example of the output of discord bot





<https://documenter.getpostman.com/view/2568274/SzS8rjbe>

# Chapter V

## Bonus for Discord Bot

- User can precise the country.
- User can precise the day, week, month etc...
- A better visual.