



Piscine 101

Python 04

Summary: This document is the subject for the PYTHON module 04 of the Piscine 101 @ 42Tokyo.

Contents

I	Instructions	2
II	Foreword	4
III	Exercise 00 : where is abilities?	5
IV	Exercise 01 : easier!	6
V	Exercise 02 : even easier!	7
VI	Exercise 03 : pokequiz	8
VII	Exercise 04 : pokedex	9
VIII	Exercise 05 : bonus	10

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.
- On top of that, your exercises will be checked and graded by a program called Moulinette.
- 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`.
- No code in the global scope(only import). We want functions!
- you're free to define as many function as you like and name them as you like also.

- Each turned-in file must end with a function call in a condition identical to:

```
if __name__ == '__main__':  
    your_function( whatever, parameter, is, required )
```

- You can set an error management in this condition.
- No import will be authorized, except the ones explicitly mentioned in the 'Authorized functions' in each exercise's description.
- You won't have to manage the exceptions raised by the `open` function.


Chapter II

Foreword

Them - You are the semicolon to my statement;
You - But I code in python.

Chapter III

Exercise 00 : where is abilities?

	Exercise 00
Get all of it, must be there.	
Turn-in directory : <i>ex00/</i>	
Files to turn in : <code>pokewiki.py</code> , <code>main.py</code>	
Allowed functions : <code>requests</code>	

You want to get Pokemon's information somehow.

Create the `PokeWiki` class containing the following functionalities:

- A method that take an entire page of a specified pokemon by name from `https://pokemon.fandom`
- A method that will return its species of a specified pokemon by name.
- A method that will return the names of its abilities of a specified pokemon by name.


For example,

```
name : bulbasaur
species : Seed Pokemon
abilities : ["Overgrow", "Chlorophyll"]
```

Turn in your `main.py` file to test this exercise. Your `main.py` will be reviewed at the time of your defense.

Chapter IV

Exercise 01 : easier!

	Exercise 01
it's more beautiful.	
Turn-in directory : <i>ex01/</i>	
Files to turn in : <code>pokewiki.py</code> , <code>main.py</code>	
Allowed functions : <code>requests</code> , <code>BeautifulSoup</code>	

It is the same thing, but use BeautifulSoup.


Re-create the `PokeWiki` class containing the following functionalities:

- A method that take an entire page of a specified pokemon by name from `https://pokemon.fandom`
- A method that will return its species of a specified pokemon by name.
- A method that will return the names of its abilities of a specified pokemon by name.

Turn in your `main.py` file to test this exercise. Your `main.py` will be reviewed at the time of your defense.

Chapter V

Exercise 02 : even easier!

	Exercise 02
it's api.	
Turn-in directory : <i>ex02/</i>	
Files to turn in : <code>pokewiki.py</code> , <code>main.py</code>	
Allowed functions : <code>requests</code> , <code>pokeapi</code>	

It is the same thing, but use `pokeapi`.


Re-create the `PokeWiki` class containing the following functionalities:

- A method that take an entire page of a specified pokemon by name from `https://pokemon.fandom.com`
- A method that will return its species of a specified pokemon by name.
- A method that will return the names of its abilities of a specified pokemon by name.
- A method that will return a random name pokemon.

Turn in your `main.py` file to test this exercise. Your `main.py` will be reviewed at the time of your defense.

Chapter VI

Exercise 03 : pokequiz

	Exercise 03
it's a bot.	
Turn-in directory : <i>ex03/</i>	
Files to turn in : <code>pokeapi.py</code> , <code>pokequiz.py</code> , <code>main.py</code>	
Allowed functions : <code>pokeapi</code> , <code>discord</code>	


Create a discord server and a bot that tests people's pokemon knowledge. Your bot must:

- give a player a quiz when member send a message `!pokemontest`, showing the image of a pokemon.
- give some hints every time user fail to answer the correct pokemon's name. The hints can be its abilities, initials, or species.
- give the correct pokemon's name after the member failed to answer its name.

Be creative.

Chapter VII

Exercise 04 : pokedex


	Exercise 04
!pokedex	
Turn-in directory : <i>ex04/</i>	
Files to turn in : <code>pokeapi.py</code> , <code>pokequiz.py</code> , <code>main.py</code>	
Allowed functions : whatever you need	

Re-create a discord bot with following functionalities.

- all the functionalities from the previous exercise.
- save at least 100 different player's progress using spreadsheet and its api.
- when a member send a message **!pokedex**, show the list of pokemons of the member answered correctly for the quiz.

Chapter VIII

Exercise 05 : bonus

	Exercise 05
!pokegame	
Turn-in directory : <i>ex05/</i>	
Files to turn in : <code>pokeapi.py</code> , <code>pokequiz.py</code> , <code>main.py</code>	
Allowed functions : <code>whatever you need</code>	

Add following functionalities or your original idea to the discord bot from the last exercise in order to make it into a little pokemon game.

- add some images and messages before a quiz to make a little bit of story.
- add a concept of items like pokemon ball.
- re create a quiz so that it resembles to a battle.

Be creative.