Piscine 101

Python 04

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

Contents

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.
- 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 - python3.9.0`.
- Catch every exceptions.
- Each exercises must be done using virtualenv.

```
?>python3 -m venv myenv
?>source myenv/bin/activate
```

- You should use guacamole.42tokyo.jp to validate exercises.

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 : *ex*00/	
Files to turn in : `requirements.txt, pokewiki.py, main.py`	
Allowed functions : `json, requests`	

You want to get Pokemon's information somehow.

Create the `PokeWiki` class containing the following functionalities. Use variables and name methods as you like:

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

Example of main.py:

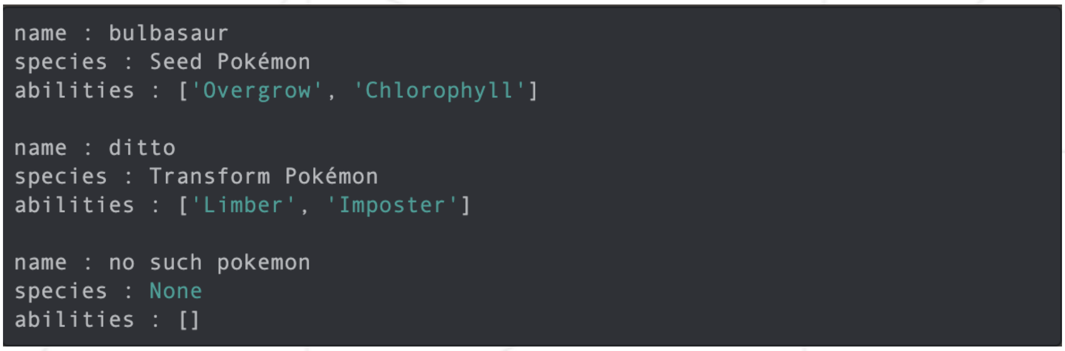
```
from pokewiki import PokeWiki
from requests.exceptions import RequestException

def main():
    test_pokemon("bulbasaur")
    test_pokemon("ditto")
    test_pokemon("no such pokemon")

def test_pokemon(name: str):
    p = PokeWiki()
    try:
        print(f"name : {name}")
        print(f"species : {p.get_species(name)}")
        print(f"abilities : {p.get_abilities(name)}")
        print()
    except RequestException as e:
        print(e)

if __name__ == "__main__":
    main()
```

Example of output:



```
name : bulbasaur
species : Seed Pokémon
abilities : ['Overgrow', 'Chlorophyll']

name : ditto
species : Transform Pokémon
abilities : ['Limber', 'Imposter']

name : no such pokemon
species : None
abilities : []
```

Chapter IV

Exercise 01 : easier!

	Exercise 01
it's more beautiful.	
Turn-in directory : *ex*01/	
Files to turn in : `requirements.txt, pokewiki.py, main.py`	
Allowed functions : `json, requests, beautifulsoup4(BeautifulSoup), random`	

It is the same thing, but use BeautifulSoup.

Re-create the `PokeWiki` class containing the following functionalities.Use variables and name methods as you like:

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

Exercise 02 : even easier!

	Exercise 02
it’s api.	
Turn-in directory : *ex*02/	
Files to turn in : `requirements.txt, pokewiki.py or pokeapi.py, main.py`	
Allowed functions : `json, requests, random`	

It is the same thing, but use pokeapi endpoints(https://pokeapi.co/api/v2/...).

Re-create the `PokeWiki` class containing the following functionalities.Use variables and name methods as you like:

- `A method` that take an entire information of a specified pokemon by name.
- `A method` that will return its species of a spacified pokemon by name.
- `A method` that will return the names of its abilities of a spacified 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

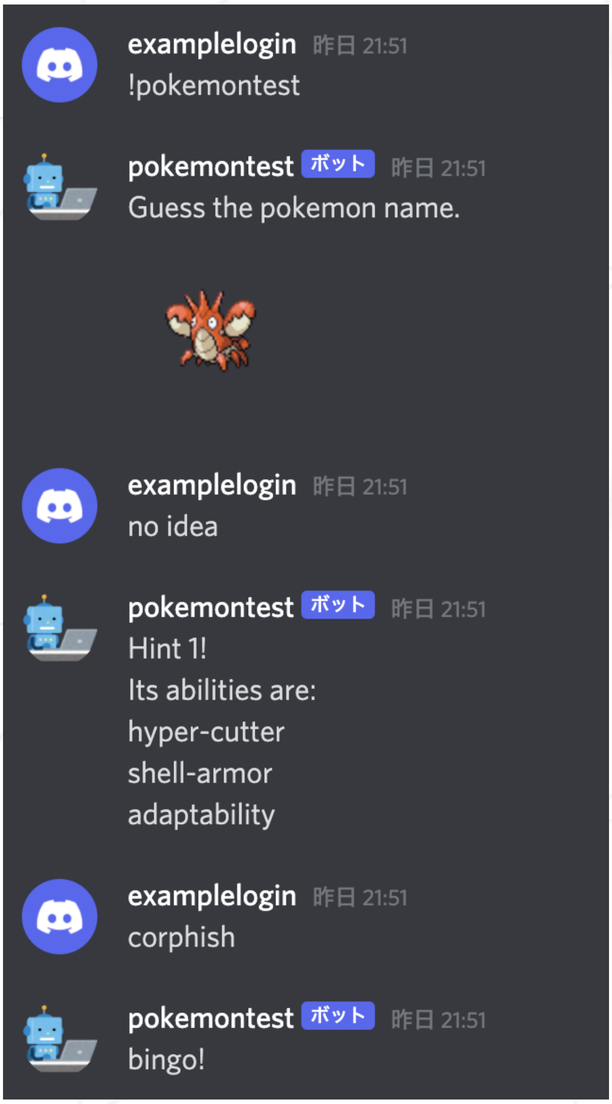
Exercise 03 : pokequiz

	Exercise 03
it's a bot.	
Turn-in directory : *ex*03/	
Files to turn in : `requirements.txt, pokeapi.py, pokequiz.py, main.py`	
Allowed functions : `json, requests, discord.py, random`	

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.

for example,



examplelogin 昨日 21:51
!pokemontest

pokemontest ボット 昨日 21:51
Guess the pokemon name.

examplelogin 昨日 21:51
no idea

pokemontest ボット 昨日 21:51
Hint 1!
Its abilities are:
hyper-cutter
shell-armor
adaptability

examplelogin 昨日 21:51
corphish

pokemontest ボット 昨日 21:51
bingo!

Chapter VII

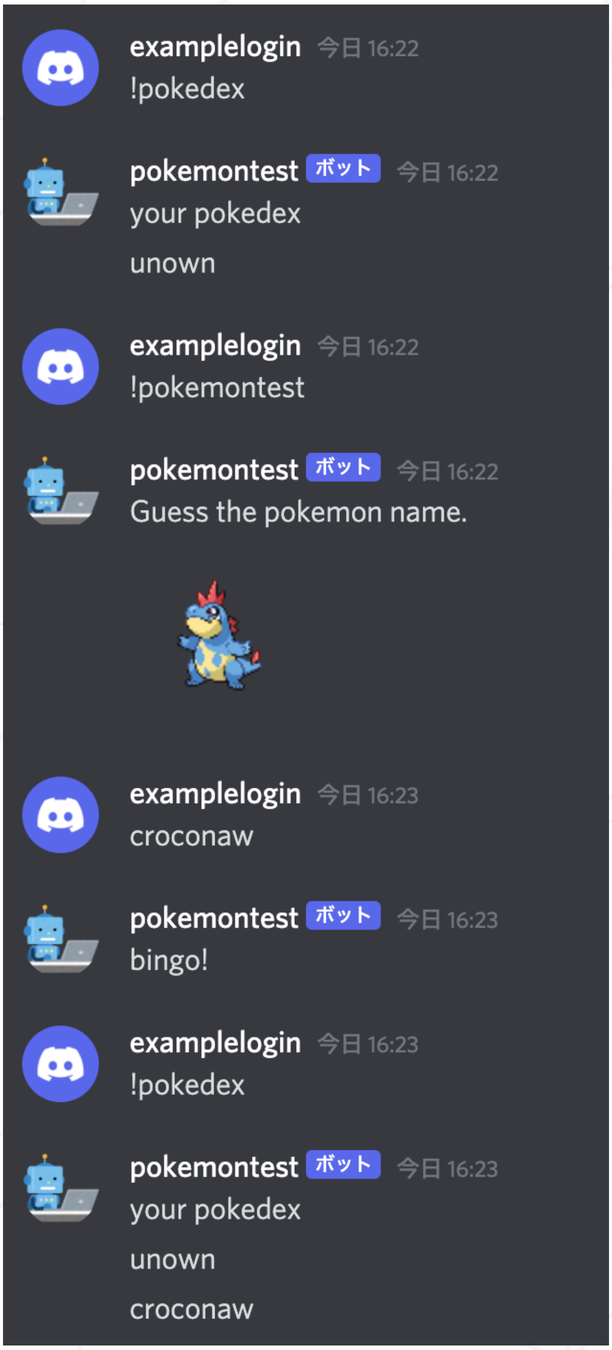
Exercise 04 : pokedex

	Exercise 04
!pokedex	
Turn-in directory : *ex*04/	
Files to turn in : `requirements.txt, spreadsheet.py pokeapi.py, pokequiz.py, main.py, *.json`	
Allowed functions : `*`	

Re-create a discord bot with following functionalities.

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

for example,



examplelogin 今日 16:22
!pokedex

pokemontest ボット 今日 16:22
your pokedex
unown

examplelogin 今日 16:22
!pokemontest

pokemontest ボット 今日 16:22
Guess the pokemon name.

examplelogin 今日 16:23
croconaw

pokemontest ボット 今日 16:23
bingo!

examplelogin 今日 16:23
!pokedex

pokemontest ボット 今日 16:23
your pokedex
unown
croconaw

⚠ `You should remove or delete the credentials which are used for this subject when you are done with this subject.`

It is not recommended to push credentials to git repository.

Chapter VIII

Exercise 05 : bonus

	Exercise 05
!pokegame	
Turn-in directory : *ex*05/	
Files to turn in : *	
Allowed functions : *	

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.