Python Bootcamp {2019}

Kirirom Institute of Technology

EXERCISES WEEK 3

EXERCISE 51

Description :

You will create a function that take one string in argument that's represent a valid a url and then return the HTML code of the page as a string. (No need output for this function but you might need to print your results before submitting your work).

You have to make a GET Request.

Requirements :

- Program name : **51_get_html**
- Function name : **get_html**
- Directory : **week03/ex** folder

Hint :

- requests
- get

Python Bootcamp {2019}

EXERCISE 52

Description :

You will create a function that take one string in argument (that's represent a valid a facebook url or fb username and then a tuple with the request status_code (usually it will be 200) in the first element the associated ID in the second element) If no ID is found, you will return a tuple with the status_code as first element and 0 in the second element.

The url endpoint is : https://findmyfbid.com/

The params you have to provide is : {'url' : 'your_fb_url_here'}

Your have to make a POST request.

EXAMPLE:

get_fbid("https://facebook.com/JohnDoe") ⇒ (200, 1213161789)

get_fbid("zuck") ⇒ (200, 4)

get_fbid("adwopawpodawodpwdpwoawopdwopawopdw") ⇒ (200, 0)

Requirements :

- Program name : **52_get_fbid**
- Function name : **get_fbid**
- Directory : **week03/ex** folder

Hint :

- ❖ requests
- ❖ post

Python Bootcamp {2019}

EXERCISE 53

Description :

You will create a function, that will take one string list as parameters and return a list of tuple. Each string element of the list represent a currency pair: "EURUSD", "EURGBP", "GBPUSD", "USDJPY"...

For each pair you have inside your array, you will add one element in your tuple list :

1. Pair name (example EURUSD)
2. Rate (example 0.549013)
3. Timestamp (example 1558650441)
4. Readable datetime string (example: 14/06/2019 14:33:25)

The first three element are found inside the JSON that is return from the API, the last element (datetime string) must be generate by your program using the given timestamp)

The url endpoint is : https://www.freeforexapi.com/api/live
The params you have to provide is : {'pairs' : 'PAIR,PAIR2,PAIR3'}
Your have to make a POST request.

EXAMPLE :

get_currency('EURUSD','EURGBP', 'USDGBP')
⇒ [('EURUSD', 1.123501, 1559650745, '06/04/2019 00:19'), ('EURGBP', 0.886762, 1559650745, '06/04/2019 00:19'), ('USDGBP', 0.789285, 1559650745, '06/04/2019 00:19')]

get_currency('ABCDEF'):
⇒ 'The currency pair 'ABCDEF' was not recognised or supported'

Documentation : https://www.freeforexapi.com/Home/Api

Python Bootcamp {2019}

Kirirom Institute of Technology

Requirements :

- Program name : **53_get_currency**
- Function name : **get_currency**
- Directory : **week03/ex** folder

Hint :

- ❖ requests
- ❖ post

EXERCISE 54

Description :

You will create a function that take a string list in parameter and return a list of tuple with two elements: (1. ID, 2. NAME).

The string list parameter will represent a Pokemon Type. (for example ‘Fire’, ‘Water’, ‘Psychic’...)

To do so, you will make a get request to the endpoint, and find every Pokemon that got the desired type inside the ‘type’ Key (it’s represent a string array).

ALL TYPES MUST BE FOUND IN TYPE TO BE ACCEPTED (watch the output)
If no Pokemons are found you will return an empty list: []

Python Bootcamp {2019}

Endpoint:
https://raw.githubusercontent.com/Biuni/PokemonGO-Pokedex/master/pokedex.json

Your have to make a GET request.

EXAMPLE:

poke_type(['Psychic'])
⇒ [(63, 'Abra'), (64, 'Kadabra'), (65, 'Alakazam'), (79, 'Slowpoke'), (80, 'Slowbro'), (96, 'Drowzee'), (97, 'Hypno'), (102, 'Exeggcute'), (103, 'Exeggutor'), (121, 'Starmie'), (122, 'Mr. Mime'), (124, 'Jynx'), (150, 'Mewtwo'), (151, 'Mew')])

poke_type(['Water', 'Psychic'])
⇒ [(79, 'Slowpoke'), (80, 'Slowbro'), (121, 'Starmie')]

poke_type(['Water', 'Fire'])
⇒ []

Requirements :

- Program name : **54_poke_type**
- Function name : **poke_type**
- Directory : **week03/ex** folder

EXERCISE 55

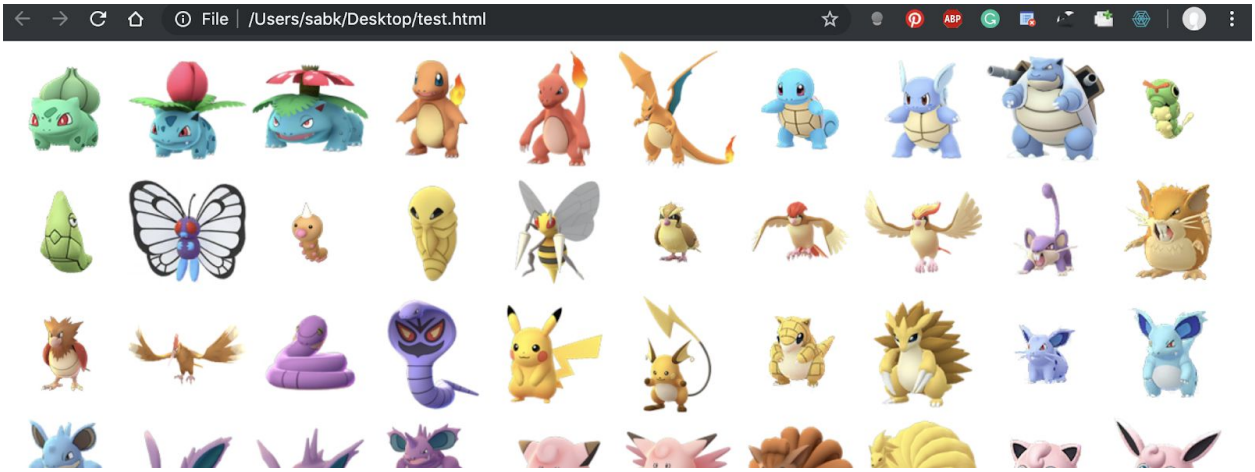
Description :

You will create a function, that take no argument and generate a HTML file. Your html will display the 151 pokemons. To be able to do so, for each pokemon you will get the value:
{"img": "http://www.serebii.net/pokemongo/pokemon/002.png"}

And generate a HTML Images:
<img src="your_url.png" height="100px" width="100px">

You just have to replace “your_url.png” with the real Pokemon image URL. Finally, use append or write function to save it into **pokemon.html**

If you did this exercises successfully, you can now open **pokemon.html** inside a web browser and you will see something like that :



Endpoint:

https://raw.githubusercontent.com/Biuni/PokemonGO-Pokedex/master/pokedex.json

Your have to make a GET request.

Requirements :

- Program name : **55_poke_gallery**
- Function name : **poke_gallery**
- Directory : **week03/ex** folder

Python Bootcamp {2019}

EXERCISE 56

Description :

You will create a function that detect if string contains only value from a-z A-Z and 0-9.

You function will take one argument in parameter (a string) and will return True or False (boolean)

If the string in empty, you will return False.

You need to do it using REGEX only.

Example :

regex_alpha("abc123")
⇒ True

regex_alpha("AAA123")
⇒ True

regex_alpha("abc123!")
⇒ False

regex_alpha("")
=> False

Requirements :

- Program name : **56_regex_alpha**
- Function name : **regex_alpha**
- Directory : **week03/ex** folder

Hint :

- ❖ re

Python Bootcamp {2019}

Kirirom Institute of Technology

EXERCISE 57

Description :

You will create a function that remove all words from a string length of between 1 and give number. This function will take one integer in parameter that represent the length and one string that represent the text to be formatted. You need to return the new string. If the number is negative or 0 you will return an empty string. The special characters will not be removed.

You need to do it using REGEX only.

EXAMPLE :

regex_word(3, “My name is Kevin”)
⇒ “name Kevin”

regex_word(5, “How are you today?”)
⇒ “?”

regex_word(2, “Is it a cat or is it a dog?”)
⇒ “cat dog?”

Requirements :

- Program name : **57_regex_word**
- Function name : **regex_word**
- Directory : **week03/ex** folder

Hint :

- ❖ re
- ❖ sub

Python Bootcamp {2019}

EXERCISE 58

Description :

You need to create a function that take a string in parameters and return a string without any parenthesis content.

You need to do it using REGEX only.

EXAMPLE:

regex_par("(hello) Welcome to KIT (Kirirom Institute of Technology)!"
⇒ " Welcome to KIT!"

regex_par("((not ok)))ok( )))(not_ok)(((ok")
⇒ "))ok))(((ok"

Requirements :

- Program name : **58_regex_par**
- Function name : **regex_par**
- Directory : **week03/ex** folder

Hint :

- ❖ re
- ❖ sub

Python Bootcamp {2019}

EXERCISE 59

Description :

You will create a function that take a string in parameter and remove every HTML content (everything between ‘<’ and ‘>’) You will return the new formatted string.

You need to do it using REGEX only.

EXAMPLE :

regex_html("<html lang = 'pl' ><body> content of body </body> ... </html>")
⇒ " content of body ..."

regex_html("<h1>hello</h1> hello ")
⇒ "hello hello"

regex_html("")
⇒ ""

regex_html("hello")
⇒ "hello"

regex_html("<<><>>><>")
⇒ ">>"

Requirements :

- Program name : **59_regex_html**
- Function name : **regex_html**
- Directory : **week03/ex** folder

Hint :

- ❖ re
- ❖ sub

Python Bootcamp {2019}

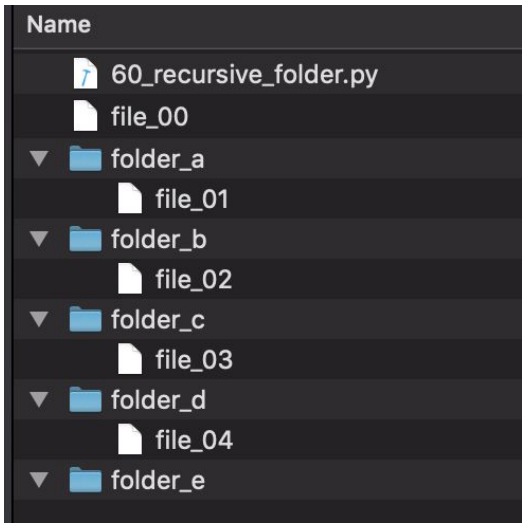
Kirirom Institute of Technology

EXERCISE 60

Description :

You will write a function that take a string in parameter that represent a given path and return files path recursively. The paths must be sorted. If the folder is empty, you must not display it.
Also, don't display your program name.

EXAMPLE:



60_recursive_folder('.')
⇒ ['./file_00', './folder_a/file_01', './folder_b/file_02', './folder_c/file_03', './folder_d/file_04']

Requirements :

- Program name : **60_recursive_folder.py**
- Function name : **recursive_folder**
- Directory : **week03/ex** folder

Hint :

- os.walk