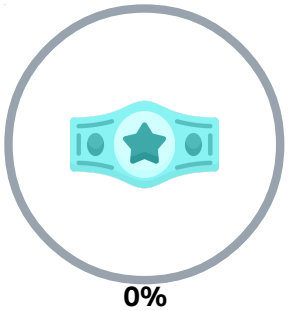
(/)



0%

Python - More Data Structures: Set, Dictionary

Python

Amateur

By: Guillaume, CTO at Holberton School

Weight: 1

Your score will be updated once you launch the project review.

Resources

Read or watch:

- Data structures (/rltoken/gMupLEVx--wpeBGaXolQzA)
- Lambda, filter, reduce and map (/rltoken/Gu5vy0GcihvtPt3lg0K8Jg)
- Learn to Program 12 Lambda Map Filter Reduce (/rltoken/-Gve48yvKfgK0SOKtSG6KQ)

man or help:

- `python3`

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/fLUVHvuQptOcve-mPlQZcQ), **without the help of Google**:

General

- Why Python programming is awesome
- What are set and how to use them
- What are the most common methods of set and how to use them
- When to use sets versus lists
- How to iterate into a set
- What are dictionary and how to use them

- When to use dictionaries versus lists or sets
- What is a key in a dictionary
- How to iterate into a dictionary
- What is a lambda function
- What is map, reduce and filter functions

Requirements

General

- Recommended editor: `Visual studio code`
- All your files will be interpreted/compiled on Ubuntu 20.04 LTS using `python3` (version 3.4.3)
- All your files should end with a new line
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should use the `PEP 8` style (version 1.7.*)
- The length of your files will be tested using `wc`

Quiz questions

Great! You've completed the quiz successfully! Keep going! (Show quiz)

Tasks

0. Squared simple

mandatory

Write a function that computes the square value of all integers of a matrix.

- Prototype: `def square_matrix_simple(matrix=[]):`
- `matrix` is a 2 dimensional array
- Returns a new matrix:
  - Same size as `matrix`
  - Each value should be the square of the value of the input
- Initial matrix should not be modified
- You are not allowed to import any module
- You are allowed to use regular loops, `map`, etc.

0)

```
guillaume@ubuntu:~/$ cat 0-main.py
#!/usr/bin/python3
square_matrix_simple = __import__('0-square_matrix_simple').square_matrix_simple

matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]

new_matrix = square_matrix_simple(matrix)
print(new_matrix)
print(matrix)

guillaume@ubuntu:~/$ ./0-main.py
[[1, 4, 9], [16, 25, 36], [49, 64, 81]]
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx_python`
- Directory: `python-more_data_structures`
- File: `0-square_matrix_simple.py`

Help | Check your code | >_ Get a sandbox **0/10** pts

1. Present in both

mandatory

Write a function that returns a set of common elements in two sets.

- Prototype: `def common_elements(set_1, set_2):`
- You are not allowed to import any module

```
guillaume@ubuntu:~/$ cat 1-main.py
#!/usr/bin/python3
common_elements = __import__('1-common_elements').common_elements

set_1 = { "Python", "C", "Javascript" }
set_2 = { "Bash", "C", "Ruby", "Perl" }
c_set = common_elements(set_1, set_2)
print(sorted(list(c_set)))

guillaume@ubuntu:~/$ ./1-main.py
['C']
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx_python`
- Directory: `python-more_data_structures`
- File: `1-common_elements.py`

Help | Check your code | >_ Get a sandbox

0/12 pts

2. Update dictionary

mandatory

Write a function that replaces or adds key/value in a dictionary.

- Prototype: `def update_dictionary(a_dictionary, key, value):`
- `key` argument will be always a string
- `value` argument will be any type
- If a key exists in the dictionary, the value will be replaced
- If a key doesn't exist in the dictionary, it will be created
- You are not allowed to import any module

7)

```
guillaume@ubuntu:~/$ cat 2-main.py
#!/usr/bin/python3
update_dictionary = __import__('2-update_dictionary').update_dictionary

def print_sorted_dictionary(my_dict):
    """ Print sorted dictionary """
    keys = sorted(my_dict.keys())
    for k in keys:
        print("{}: {}".format(k, my_dict[k]))

a_dictionary = { 'language': "C", 'number': 89, 'track': "Low level" }
new_dict = update_dictionary(a_dictionary, 'language', "Python")
print_sorted_dictionary(new_dict)
print("--")
print_sorted_dictionary(a_dictionary)

print("--")
print("--")

new_dict = update_dictionary(a_dictionary, 'city', "San Francisco")
print_sorted_dictionary(new_dict)
print("--")
print_sorted_dictionary(a_dictionary)

guillaume@ubuntu:~/$ ./2-main.py
language: Python
number: 89
track: Low level
--
language: Python
number: 89
track: Low level
--
--
city: San Francisco
language: Python
number: 89
track: Low level
--
city: San Francisco
language: Python
number: 89
track: Low level
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx_python`
- Directory: `python-more_data_structures`

- File: `2-update_dictionary.py`

(/)

Help | Check your code | >_ Get a sandbox **0/10** pts

3. Best score

mandatory

Write a function that returns a key with the biggest integer value.

- Prototype: `def best_score(a_dictionary):`
- You can assume that all values are only integers
- If no score found, return `None`
- You can assume all students have a different score
- You are not allowed to import any module

```
guillaume@ubuntu:~/$ cat 3-main.py
#!/usr/bin/python3
best_score = __import__('3-best_score').best_score

a_dictionary = {'John': 12, 'Bob': 14, 'Mike': 14, 'Molly': 16, 'Adam': 10}
best_key = best_score(a_dictionary)
print("Best score: {}".format(best_key))

best_key = best_score(None)
print("Best score: {}".format(best_key))

guillaume@ubuntu:~/$ ./3-main.py
Best score: Molly
Best score: None
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx_python`
- Directory: `python-more_data_structures`
- File: `3-best_score.py`

Help | Check your code | >_ Get a sandbox **0/9** pts

4. Multiply by using map

mandatory

Write a function that returns a list with all values multiplied by a number without using any loops.

- Prototype: `def multiply_list_map(my_list=[], number=0):`
- Returns a new list:
  - Same length as `my_list`

    - Each value should be multiplied by `number`
- Initial list should not be modified
- ~~You are not allowed to import any module~~
- You have to use `map`
- Your file should be max 3 lines

```
guillaume@ubuntu:~/$ cat 4-main.py
#!/usr/bin/python3
multiply_list_map = __import__('4-multiply_list_map').multiply_list_map

my_list = [1, 2, 3, 4, 6]
new_list = multiply_list_map(my_list, 4)
print(new_list)
print(my_list)

guillaume@ubuntu:~/$ ./4-main.py
[4, 8, 12, 16, 24]
[1, 2, 3, 4, 6]
guillaume@ubuntu:~/$
```

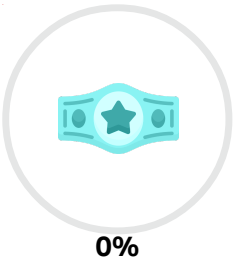
Repo:

- GitHub repository: `alx_python`
- Directory: `python-more_data_structures`
- File: `4-multiply_list_map.py`

Help | Check your code | >_ Get a sandbox

0/10 pts

Score



0%

Your score will be updated once you launch the project review.

Please review **all the tasks** before you start the peer review.

Review all the tasks | Skip this project

Previous project (/projects/2057)

Copyright © 2023 ALX, All rights reserved.

(/)