



CHALLENGES

MEDIA

SQUADS

INACTIVITY

CLUSTER

STATISTICS

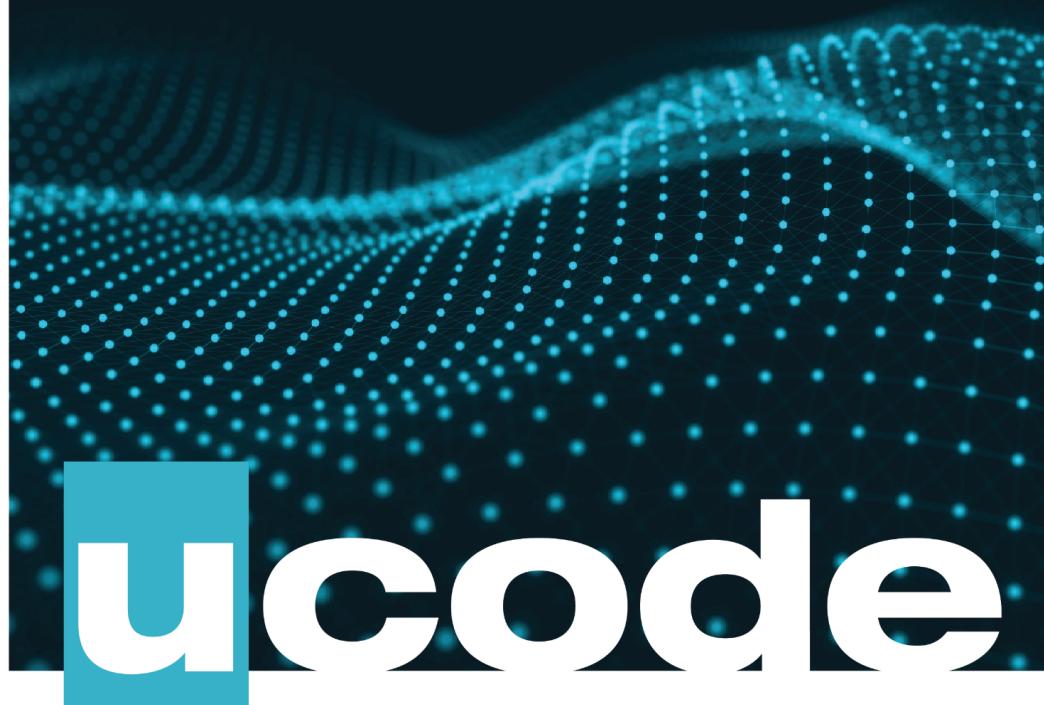


Sprint 03

Marathon Python



May 5, 2021



Contents

Engage	2
Investigate	3
Act Basic: Task 00 > Lambda	5
Act Basic: Task 01 > Clear words	7
Act Basic: Task 02 > The extractor	10
Act Basic: Task 03 > Read file	14
Act Basic: Task 04 > Write	17
Act Basic: Task 05 > Load JSON	20
Act Basic: Task 06 > Dump JSON	23
Act Advanced: Task 07 > Calculambdor	27
Act Advanced: Task 08 > Multiplier	29
Act Advanced: Task 09 > Raise	31
Act Advanced: Task 10 > Group	34
Share	38

ucode

Engage

DESCRIPTION

Hello!

Hello..

There are many different programming paradigms. For example, imperative, object-oriented, procedural, etc. One such paradigm is functional programming, inspired by Lambda Calculus.

Functional programming embraces mathematical-style computation using functions and avoids changing state and data mutability. Functional paradigm focuses on *what the code does*, compared to *how the code does it* (imperative programming).

Python is not a purely functional language, but it does contain certain features that allow programming in a functional style. One such feature that you will be learning in this challenge is lambda functions.

Also in this **Sprint**, you will learn how to work with files in Python. How to open, close, write information in them, and receive it. You will also learn how to work with the JSON format.

So... Let's go?!

BIG IDEA

Functional programming and work with files.

ESSENTIAL QUESTION

How to write functional-style programs in Python?
How to interact with files?

CHALLENGE

Learn functional-style tools in Python.
Learn the basics of working with files in Python

Sprint 03 | Marathon Python > 2



Investigate



GUIDING QUESTIONS

We invite you to find answers to the following questions. By researching and answering them, you will gain the knowledge necessary to complete the challenge. To find answers, ask the students around you and search the internet. We encourage you to ask as many questions as possible. Note down your findings and discuss them with your peers.

- What are the main concepts of functional programming?
- What are the benefits of functional programming?
- What are the features of functional programming in Python?
- What is a file in programming?

- What file formats are there?

GUIDING ACTIVITIES

Complete the following activities. Don't forget that you have a limited time to overcome the challenge. Use it wisely. Distribute tasks correctly.

- Read about [Python lambdas](#).
- [Here](#) you can find more info around lambdas in general.
- Learn how to read and write to files. For example, read this [article](#).
- Familiarize yourself with [JSON](#).
- Attentively watch and investigate learning videos available on the challenge page. Try to repeat all actions.
- Clone your git repository issued on the challenge page in the LMS.
- Proceed with tasks.

ANALYSIS

Analyze your findings. What conclusions have you made after completing guiding questions and activities? In addition to your thoughts and conclusions, here are some more analysis results.

- Be attentive to all statements of the story.
- All tasks are divided into [Act Basic](#) and [Act Advanced](#). You need to complete all basic tasks to validate the [Sprint](#). But to achieve maximum points, consider accomplishing advanced tasks also.
- Analyze all information you have collected during the preparation stages. Try to define the order of your actions.
- Perform only those tasks that are given in this document.
- Submit only those files that are described in the story. Only useful files allowed, garbage shall not pass!
- Run the scripts using [python3](#).
- Make sure that you have [Python](#) with a [3.8](#) version, or higher.

Sprint 03 | Marathon Python > 3



- Use the standard library available after installing [Python](#). You may use additional packages/libraries that were not previously installed only if they are specified in the task.
- To figure out what went wrong in your code, use [PEP 553 -- Built-in breakpoint\(\)](#).
- Complete tasks according to the rules specified in the [PEP8 conventions](#).
- The solution will be checked and graded by students like you. [Peer-to-Peer learning](#).
- Also, the challenge will pass automatic evaluation which is called [Oracle](#).
- If you have any questions or don't understand something, ask other students or just Google it.



Sprint 03 | Marathon Python > 4

Act Basic: Task 00

>

NAME
Lambda

DIRECTORY
`t00_lambda/`

SUBMIT
`expression.py`

BEFORE YOU BEGIN

In order to complete this task, you must be able to answer the following questions:

- What is a lambda function in Python?
- What is another name for lambda functions and why?
- What are the main features and limitations of lambda functions?
- What is the keyword used to create a lambda function?
- What does the following expression do `lambda a, b: a + b`? What will be the result if you pass in the values `(2, 2)` ?

LEGEND

ENCYCLOPEDIA SALESMAN: Burglar! *[rings again]* Burglar!
[woman appears at other side of door]
WOMAN: Yes?
ENCYCLOPEDIA SALESMAN: Burglar, madam.
WOMAN: What do you want?
ENCYCLOPEDIA SALESMAN: I want to come in and steal a few things, madam.

WOMAN: Are you an encyclopaedia salesman?
ENCYCLOPEDIA SALESMAN: No madam, I'm a burglar, I burgle people.
WOMAN: I think you're an encyclopaedia salesman.
ENCYCLOPEDIA SALESMAN: Oh I'm not, open the door, let me in please.
WOMAN: If I let you in, you'll sell me encyclopedias.
ENCYCLOPEDIA SALESMAN: I won't, madam. I just want to come in and ransack the flat.
Honestly.
WOMAN: Promise? No encyclopedias?
ENCYCLOPEDIA SALESMAN: None at all.
WOMAN: All right. [she opens door] You'd better come in then.
ENCYCLOPEDIA SALESMAN: Mind you, I don't know whether you've really considered the advantages
of owning a really fine set of modern encyclopedias... You know, they can really do you
wonders.
-- Monty Python's Flying Circus

DESCRIPTION

Create a script that contains a lambda function.

The lambda function must return `True` or `False` based on whether `n` is divisible by both `a`



Sprint 03 | Marathon Python > 5

and `b` with a remainder of 0.

Use the code snippet in the **SYNOPSIS**, but instead of `...` put the needed code to get the correct results.

See examples of the script working in the **CONSOLE VIEW**.

SYNOPSIS

```
n = int(input('n: '))
a = int(input('a: '))
b = int(input('b: '))

# Only edit the following line
result = (lambda ...)(...)

print(result)
```

CONSOLE VIEW

```
>python3 expression.py
n: 15
a: 3
b: 5
True
>python3 expression.py
n: 64
a: 2
b: 14
False
>python3 expression.py
n: 10
a: 4
b: 5
False
>
```

SEE ALSO

Python lambda

ucode

Sprint 03 | Marathon Python > 6

Act Basic: Task 01

NAME
Clear words

DIRECTORY
t01_clear_words/

SUBMIT
clear_words.py

BEFORE YOU BEGIN

In order to complete this task, you must be able to answer the following questions:

- What does the Python `map()` function do?
- What parameters does the function `map()` take?
- How to use lambda expressions with `map()` ?

DESCRIPTION

Create a function `clear_words()` that:

- takes some text as a string with punctuation marks. Punctuation marks must be `[?!.:;,-]` (use regex)
- splits the string by spaces
- uses the function `map()` with a lambda expression to remove all punctuation marks in the split words
- returns the result as a list

The script in the EXAMPLE tests your function. If everything is correct, it should generate output as seen in the CONSOLE VIEW. Also, you can see examples in the PYTHON INTERPRETER. Pay attention that you must only submit the file `clear_words.py`, not the test script.

EXAMPLE

```
from clear_words import clear_words

text_example_1 = 'WOMAN: Yes?, ENCYCLOPEDIA SALESMAN: Burglar, madam. WOMAN: \'\
    'Are you an encyclopaedia salesman?'

text_example_2 = 'ENCYCLOPEDIA SALESMAN: No madam, I\'m a burglar, \'\
    'I burgle people. WOMAN: I think you\'re an encyclopaedia \'\
    'salesman.'

text_example_3 = 'ENCYCLOPEDIA SALESMAN: Oh I\'m not, open the door, \'\
    'let me in please. WOMAN: If I let you in, you\'ll sell \'\
    'me encyclopedias.'
```



Sprint 03 | Marathon Python > 7



```

if __name__ == '__main__':
    result = clear_words(text_example_1)
    print(result)

    result = clear_words(text_example_2)
    print(result)

    result = clear_words(text_example_3)
    print(result)

```

CONSOLE VIEW

```
>python3 s03t01_clear_words_main.py
['WOMAN', 'Yes', 'ENCYCLOPEDIA', 'SALESMAN', 'Burglar', 'madam', 'WOMAN', 'Are', 'you',
 'an', 'encyclopaedia', 'salesman']
['ENCYCLOPEDIA', 'SALESMAN', 'No', 'madam', "I'm", 'a', 'burglar', 'I', 'burgle',
 'people', 'WOMAN', 'I', 'think', "you're", 'an', 'encyclopaedia', 'salesman']
['ENCYCLOPEDIA', 'SALESMAN', 'Oh', "I'm", 'not', 'open', 'the', 'door', 'let', 'me',
 'in', 'please', 'WOMAN', 'If', 'I', 'let', 'you', 'in', "you'll", 'sell', 'me',
 'encycopedias']
>
```

PYTHON INTERPRETER

```
>python3
>>> from clear_words import clear_words
>>> text_example_1 = 'Now, you listen here: \'e\'s not the Messiah, \'e\'s a very naughty
   boy! Now piss off!'
>>> result = clear_words(text_example_1)
>>> result
['Now', 'you', 'listen', 'here', "'e's", 'not', 'the', 'Messiah', "'e's", 'a', 'very',
 'naughty', 'boy', 'Now', 'piss', 'off']
>>> text_example_2 = 'There\'s a lot of punctuation h-e-r-e! And ,,,, not @ll of
   --- it#is corr>ct.'
>>> result = clear_words(text_example_2)
>>> result
['There', 's', 'a', 'lot', 'of', 'punctuation', 'here', 'And', ',', 'not', '@ll', 'of',
 '---', 'it#is', 'corr>ct']
>>> text_example_3 = ''
>>> result = clear_words(text_example_3)
>>> result
[]
>>> text_example_4 = ''
>>> result = clear_words(text_example_4)
```

Sprint 03 | Marathon Python > 8



```
>>> result
[]
>>> text_example_5 = ',,,,'
>>> result = clear_words(text_example_5)
>>> result
['']
>>>
```

SEE ALSO

[Python - map\(\) Function](#)
[Python Lambda Functions with Examples](#)

Sprint 03 | Marathon Python > 9

ucode

Act Basic: Task 02

NAME
The extractor

DIRECTORY
`t02_the_extractor/`

SUBMIT
`extractor.py`

BEFORE YOU BEGIN

In order to complete this task, you must be able to answer the following questions:

- How does the method `filter()` work?
- What are the requirements for the function that is passed to `filter()`?
- What happens if you pass `None` instead of a function to `filter()`?
- Does `filter()` modify an iterable in-place, or return the result?
- How to check if a value is of a certain data type?
- Why does this expression `isinstance(False, int)` return `True`?

LEGEND

Now, you listen here: 'e's not the Messiah, 'e's a very naughty boy! Now piss off!
-- Monty Python's Life of Brian

DESCRIPTION

Create a script with a function `extractor()` that filters a dictionary according to a certain data type of the values. It takes two arguments: a dictionary `extractable` and a data type `value_type` (an instance of class `type`, e.g., `int`, `bool`, etc). `value_type` has the default value of `str`.

The function uses `filter()` to create a new dictionary that only contains the items from `extractable` that have a value of type `value_type`.

The script in the EXAMPLE tests your function. Use this as an example, add your own values, and also test your function as shown in the PYTHON INTERPRETER. If everything is correct, it should generate output as shown below. Pay attention that you must only submit the file `extractor.py`, not the test script, and Oracle will check your function with random values.



Sprint 03 | Marathon Python > 10

