(/)

Curriculum
Short Specializations
Average: **97.3%**

0x03. Log Parsing

Algorithm **Python**

By: Alexa Orrico, Software Engineer at Holberton School

Weight: 1

Project over - took place from Dec 18, 2023 6:00 AM to Dec 22, 2023 6:00 AM

An auto review will be launched at the deadline

In a nutshell...

- **Auto QA review:** 1.0/11 mandatory
- **Altogether: 9.09%**
  - Mandatory: 9.09%
  - Optional: no optional tasks

For the "0x03. Log Parsing" project, you will need to apply your knowledge of Python programming, focusing on parsing and processing data streams in real-time. This project involves reading from standard input (stdin), handling data in a specific format, and performing calculations based on the input data. Here's a list of concepts and resources that you might find useful:

Concepts Needed:

1. **File I/O in Python**:
   - Understand how to read from `sys.stdin` line by line.
   - Python Input and Output (/rltoken/f7U2MDsBT_rd9AfUUaqVnQ)
2. **Signal Handling in Python**:
   - Handling keyboard interruption (CTRL + C) using signal handling in Python.
   - Python Signal Handling (/rltoken/1nDqPJe80rSD-NMulzjJBw)
3. **Data Processing**:
   - Parsing strings to extract specific data points.
   - Aggregating data to compute summaries.

4. **Regular Expressions**:
(/)
   - Using regular expressions to validate the format of each line.
   - Python Regular Expressions (/rltoken/ZsD-YLisfaHFeMT_sZxX1Q)
5. **Dictionaries in Python**:
   - Using dictionaries to count occurrences of status codes and accumulate file sizes.
   - Python Dictionaries (/rltoken/JM-RpavKkb8yanxWEnNYJw)
6. **Exception Handling**:
   - Handling possible exceptions that may arise during file reading and data processing.
   - Python Exceptions (/rltoken/OA2PlryrYA2gyCCKlsdgUw)

By studying these concepts and utilizing the resources provided, you will be well-prepared to tackle the log parsing project, effectively handling data streams, parsing log entries, and computing metrics based on the processed data.

Additional Resources

- Mock Technical Interview (/rltoken/VlOaXKkbecRYdnTLaLU1lg)

Requirements

General

- Allowed editors: `vi, vim, emacs`
- 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
- The first line of all your files should be exactly `#!/usr/bin/python3`
- 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.x)
- All your files must be executable
- The length of your files will be tested using `wc`

Tasks

0. Log parsing

mandatory

Score: 9.09% (*Checks completed: 9.09%*)

Write a script that reads `stdin` line by line and computes metrics:

- Input format: `<IP Address> - [<date>] "GET /projects/260 HTTP/1.1" <status code> <file size>` (if the format is not this one, the line must be skipped)

- After every 10 lines and/or a keyboard interruption (`CTRL + C`), print these statistics from the beginning:
  - Total file size: `File size: <total size>`
  - where `<total size>` is the sum of all previous `<file size>` (see input format above)
  - Number of lines by status code:
    - possible status code: `200` , `301` , `400` , `401` , `403` , `404` , `405` and `500`
    - if a status code doesn't appear or is not an integer, don't print anything for this status code
    - format: `<status code>: <number>`
    - status codes should be printed in ascending order

Warning: In this sample, you will have random value - it's normal to not have the same output as this one.

```
alexa@ubuntu:~/0x03-log_parsing$ cat 0-generator.py
#!/usr/bin/python3
import random
import sys
from time import sleep
import datetime

for i in range(10000):
    sleep(random.random())
    sys.stdout.write("{:d}.{:d}.{:d}.{:d} - [{}] \"GET /projects/260 HTTP/1.1\" {}
{}\n".format(
        random.randint(1, 255), random.randint(1, 255), random.randint(1, 255), rand
om.randint(1, 255),
        datetime.datetime.now(),
        random.choice([200, 301, 400, 401, 403, 404, 405, 500]),
        random.randint(1, 1024)
    ))
    sys.stdout.flush()

alexa@ubuntu:~/0x03-log_parsing$ ./0-generator.py | ./0-stats.py
File size: 5213
200: 2
401: 1
403: 2
404: 1
405: 1
500: 3
File size: 11320
200: 3
301: 2
400: 1
401: 2
403: 3
404: 4
405: 2
500: 3
File size: 16305
200: 3
301: 3
400: 4
401: 2
403: 5
404: 5
405: 4
500: 4
^CFile size: 17146
200: 4
301: 3
400: 4
401: 2
403: 6
404: 6
```

```
405: 4
500: 4
(0)
Traceback (most recent call last):
  File "./0-stats.py", line 15, in <module>
Traceback (most recent call last):
  File "./0-generator.py", line 8, in <module>
    for line in sys.stdin:
KeyboardInterrupt
    sleep(random.random())
KeyboardInterrupt
alexa@ubuntu:~/0x03-log_parsing$
```

Repo:

- GitHub repository: `alx-interview`
- Directory: `0x03-log_parsing`
- File: `0-stats.py`

☐ Done? | Help | Check your code | Ask for a new correction | QA Review