

Laboratorium 2 – Podstawy Pythona. Praca z funkcjami. Wejście/wyjście standardowe

Cele dydaktyczne

1. Zapoznanie z podstawowymi metodami budowania abstrakcji w Python
2. Zapoznanie z przetwarzaniem danych z wejścia standardowego

Zadania

1. Przygotowanie danych

Pobierz plik z [logami serwera *www.Apache.NASA.Kennedy.Space.Center.WWW* we Florydzie](http://logami.serwera.www.Apache.NASA.Kennedy.Space.Center.WWW.we.Florydzie). Rozpakuj plik. Każda linia logu zawiera:

- adres/nazwę hosta, który wykonywał żądanie,
- znacznik czasu w formacie "DAY/MON/YYY DD HH:MM:SS [TZ]",
- metodę protokołu HTTP wraz z ścieżką do żadanego zasobu,
- kod odpowiedzi HTTP (200 – w przypadku dostępności zasobu, 302 – zasób przeniesiony tymczasowo, 404 – gdy nie znaleziono),
- liczbę bajtów w odpowiedzi.

2. Czytanie danych z wejścia standardowego

Napisz program, który pozwoli na przetwarzanie kolejnych linii z danych odczytanych z wejścia standardowego:

python

lab_2.py

<

NASA

Niech program wypisuje na wyjście standardowe dane, które zostały przesłane na wejście, a po odczytaniu EOF, niech zakończy działanie.

3. Funkcje

Rozszerz program z zadania 2 o opisane poniżej funkcjonalności w języku Python z uwzględnieniem następujących uwag:

- Niech każda z funkcjonalności będzie zaimplementowana jako funkcja w osobnym pliku.
- Zadbaj o to, aby każda funkcja w programie miała tylko jedną odpowiedzialność.
- Zadbaj o rozdzielenie funkcji przetwarzających dane od funkcji najwyższego poziomu wypisujących tekst na wyjście standardowe.
- W przypadku, gdy kilka funkcjonalności wymaga skorzystania z samych funkcji, umieść je w osobnym module, który będzie ponownie użyty.
- Przed potencjalnie źle sformatowanymi wierszami zabezpiecz się wykorzystując mechanizm wyjątków.

Funkcje redukujące:

- a. Funkcja, która wypisuje na wyjście standardowe liczbę żądań z kodem:
 - a. 200,
 - b. 302
 - c. 404.
- b. Funkcja, która wypisuje sumaryczną liczbę danych wysłanych do hostów podaną w gigabajtach.
- c. Funkcja, która wypisuje na wyjście standardowe ścieżkę i rozmiar największego zasobu.
- d. Funkcja, wypisująca stosunek pobrań grafiki (zasoby kończące się na *.gif, *.jpg, *.jpeg, *.xbm) do pozostałych zasobów.

Funkcje filtrujące:

- e. Funkcja, która wypisuje na wyjście standardowe tylko wiersze z kodem odpowiedzi 200.
- f. Funkcja, która wypisuje na wyjście standardowe zasoby pobierane pomiędzy 22 a 6 rano.
- g. Funkcja, które wypisuje na wyjście standardowe tylko zasoby pobierane w piątek.
- h. Funkcja, która wypisuje na wyjście standardowe tylko żądania z Polski (tzn. hostów z nazwą domenową kończącą się .pl).

4. `__main__`

Zapoznaj się z rozdziałem dokumentacji https://docs.python.org/3/library/__main__.html

Zmodyfikuj programy w taki sposób, aby każde wywołanie skryptu z terminalu odbywało się przy użyciu konstrukcji `if __name__ == '__main__':`.

5. Przesyłanie potokowe

Pokaż, że opracowane przez Ciebie funkcje umożliwiają na potokowe przesyłanie danych pomiędzy sobą. Na przykład, w celu wypisania liczby żądań z kodem 404 z Polski.

```
cat NASA | python lab_3_h.py | python lab_3_a_c.py
```

(Uwaga, polecenie powyżej nie będzie działać w CMD na windows ze względu na brak polecenia `cat` w tej powłoce. W przypadku korzystania z CMD należy użyć np. `type`).

Materiały dodatkowe

1. <https://www.digitalocean.com/community/tutorials/read-stdin-python>
2. Alex Martelli, Anna Martelli Ravenscroft, Steve Holden, Paul McGuire, Python in a Nutshell, 4th Edition, Published by O'Reilly Media, Inc., *Rozdział 3*
[URL: <https://learning.oreilly.com/library/view/python-in-a/9781098113544/>]
3. Ryan's tutorial, Piping & Redirection
[URL: <https://ryantutorials.net/linuxtutorial/piping.php>]