

# Laboratorium 3 – Podstawowe struktury danych Pythona.

## Cele dydaktyczne

Zapoznanie z podstawowymi strukturami danych Pythona: listami, krotkami, słownikami.

**Program można zgłosić jako zrobiony, jeśli spełnione są poniższe warunki:**

1. Program jest zgodny z podaną specyfikacją.
2. Został przetestowany.
3. Student go rozumie i potrafi wyjaśnić.

### UWAGA:

- 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.
- **Z wykorzystaniem *mechanizmu obsługi wyjątków* zadbaj o poprawność działania funkcji**
- **Do zadań, gdzie jest to możliwe, przygotować kilka testów sprawdzających poprawność działania**

## Zadania

### Przygotowanie i czytanie danych

Wykorzystaj plik z logami serwera webowego z poprzedniego laboratorium i przygotuj skrypt pozwalający odczytywać je z wejścia standardowego.

#### 1. Listy i krotki

- a. Utwórz funkcję **read\_log**, która czyta wszystkie linie z wejścia standardowego i zwraca listę zawierającą poszczególne wpisy.

Funkcja powinna:

- ✓ Podzielić każdą linię (wpis) na niezależne elementy.
- ✓ Dokonać konwersji ciągów znaków na odpowiednie typy danych (np. kod statusu – int, data i

- czas żądania – [datetime.datetime](#).
- ✓ Zapisać każdy wpis (niepustą linię) jako krotkę na liście.
  - ✓ Zwrócić listę krotek.
- b. Napisz funkcję **sort\_log**, która przyjmuje dwa parametry:
- listę krotek reprezentującą log
  - liczbę określającą element krotki, według którego zostanie wykonane sortowanie.
- ✓ Zaimplementuj mechanizm sortowania listy korzystając z funkcji `sorted()` lub `sort()`.
- c. Napisz funkcję **get\_entries\_by\_addr**, która:
- ✓ przyjmuje jako parametr listę krotek reprezentującą log,
  - ✓ przyjmuje jako parametr ciąg znaków reprezentujący adres IP lub nazwę domenową hosta wykonującego żądanie,
  - ✓ waliduje podany IP lub nazwę domenową hosta
  - ✓ zwraca listę wpisów z danym ip lub nazwą domenową hosta .
- d. Napisz funkcję **get\_failed\_reads**, która:
- ✓ przyjmuje jako parametr listę krotek reprezentującą log
  - ✓ tworzy dwie oddzielne listy zawierające wpisy z kodami statusu HTTP 4xx oraz 5xx.
  - ✓ przyjmuje opcjonalny parametr logiczny, który określa, czy zwrócić jedną, połączoną listę czy dwie osobne,
  - ✓ opcjonalnie, łączy dwie listy w jedną
  - ✓ zwraca odpowiedni wynik
- e. Napisz funkcję **get\_entries\_by\_extension**, która:
- ✓ przyjmuje jako parametr listę krotek reprezentującą log,
  - ✓ przyjmuje jako parametr ciąg znaków reprezentujący rozszerzenie pliku (np. "jpg")
  - ✓ zwraca wszystkie wpisy zawierające zapytania o zasoby z danym rozszerzeniem.
- f. Napisz funkcję **print\_entries**, która wypisuje kolejne krotki z listy przekazanej do niej jako parametr do N krotki, której numer zostanie podany jako parametr.

**Punkty: 6**

## 2. Słowniki

- a. Utwórz funkcję **entry\_to\_dict**, która:
- ✓ przyjmuje na wejściu krotkę reprezentującą pojedynczy wpis
  - ✓ tworzy słownik, w którym kluczem jest ciąg znaków reprezentujący znaczenie kolejnych pól wpisu, np. "ip", "status code", "datetime", etc. Wartością w słowniku jest wartość danego pola.
  - ✓ zwraca słownikową reprezentację wpisu
- b. Utwórz funkcję **log\_to\_dict**, która:
- ✓ przyjmuje na wejściu listę krotek reprezentującą log,
  - ✓ tworzy słownik, w którym kluczem jest ciąg znaków reprezentujący adres IP lub nazwę domenową hosta wykonującego żądanie. Wartością w tym słowniku powinna być lista wpisów w postaci zwracanej przez funkcję `entry_to_dict`.

- ✓ zwraca słownik zawierający wszystkie wpisy z listy
- c. Utwórz funkcję **get\_addrs**, która:
  - ✓ przyjmuje na wejściu słownik zawierający wszystkie wpisy z listy
  - ✓ zwraca na wyjściu listę adresów ip lub nazw domenowych, które stanowią klucze słownika
- d. Utwórz funkcję **print\_dict\_entry\_dates**, która:
  - ✓ przyjmuje na wejściu słownik zawierający wszystkie wpisy z listy
  - ✓ iteruje po elementach słownika
  - ✓ wyświetla na wyjściu standardowym w czytelny sposób:
    1. adres ip/nazwę domenową hosta,
    2. liczbę żądań wykonanych przez danego hosta,
    3. datę pierwszego i ostatniego żądania wykonanego przez hosta,
    4. stosunek liczby żądań z podanym kodem do liczby pozostałych żądań.

**Punkty: 4**

## **Materiały dodatkowe**

1. 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/>]
2. <https://docs.python.org/3/tutorial/datastructures.html>