

# Laboratorium 3 – Podstawowe struktury danych Pythona.

## Cele dydaktyczne

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

## Zadania

### 1. Przygotowanie i czytanie danych

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

### 2. Listy i krotki

- a. Utwórz funkcję **read\_log**, która czyta wszystkie linie z wejścia standardowego i zwraca listę zawierającą poszczególne wpisy. Funkcja powinna:
  - i. Podzielić każdą linię (wpis) na niezależne elementy.
  - ii. Dokonać konwersji ciągów znaków na odpowiednie typy danych (np. kod statusu – int, data i czas żądania – [datetime.datetime](#)).
  - iii. Zapisać każdy wpis (niepustą linię) jako krotkę na liście.
  - iv. Zwrócić listę krotek.
- b. Napisz funkcję **sort\_log**, która:
  - i. przyjmuje dwa parametry:
    1. listę krotek reprezentującą log
    2. liczbę określającą element krotki, według którego zostanie wykonane sortowanie.
  - ii. Zaimplementuj mechanizm sortowania listy korzystając z funkcji `sorted()` lub `sort()`.
  - iii. Z wykorzystaniem *mechanizmu obsługi wyjątków* zadбай o poprawność działania funkcji (np. gdy zostanie podana liczba przekraczająca rozmiar krotki).
- c. Napisz funkcję **get\_entries\_by\_addr**, która:
  - i. przyjmuje jako parametr listę krotek reprezentującą log,
  - ii. przyjmuje jako parametr ciąg znaków reprezentujący adres IP lub nazwę domenową hosta wykonującego żądanie,

- iii. waliduje podany kod statusu,
  - iv. zwraca listę wpisów z danym ip lub nazwą domenową hosta .
- d. Napisz funkcję **get\_entries\_by\_code**, która:
  - i. przyjmuje jako parametr listę krotek reprezentującą log,
  - ii. przyjmuje jako parametr kod statusu HTTP (np. 200, 302, 404),
  - iii. waliduje podany kod statusu,
  - iv. zwraca listę wpisów z danym kodem statusu.
- e. Napisz funkcję **get\_failed\_reads**, która:
  - i. przyjmuje jako parametr listę krotek reprezentującą log
  - ii. tworzy dwie oddzielne listy zawierające wpisy z kodami statusu HTTP 4xx oraz 5xx.
  - iii. przyjmuje opcjonalny parametr logiczny, który określa, czy zwrócić jedną, połączoną listę czy dwie osobne,
  - iv. opcjonalnie, łączy dwie listy w jedną
  - v. zwraca odpowiedni wynik
- f. Napisz funkcję **get\_entries\_by\_extension**, która:
  - i. przyjmuje jako parametr listę krotek reprezentującą log,
  - ii. przyjmuje jako parametr ciąg znaków reprezentujący rozszerzenie pliku (np. "jpg")
  - iii. zwraca wszystkie wpisy zawierające zapytania o zasoby z danym rozszerzeniem.
- g. Napisz funkcję **print\_entries**, która wypisuje kolejne krotki z listy przekazanej do niej jako parametr.

**Punkty: 7**

### 3. Słowniki

- a. Utwórz funkcję **entry\_to\_dict**, która:
  - i. przyjmuje na wejściu krotkę reprezentującą pojedynczy wpis
  - ii. 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.
  - iii. zwraca słownikową reprezentację wpisu
- b. Utwórz funkcję **log\_to\_dict**, która:
  - i. przyjmuje na wejściu listę krotek reprezentującą log,
  - ii. 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**.
  - iii. zwraca słownik zawierający wszystkie wpisy z listy

- c. Utwórz funkcję **get\_addrs**, która:
  - i. przyjmuje na wejściu słownik zawierający wszystkie wpisy z listy
  - ii. zwraca na wyjściu listę adresów ip lub nazw domenowych, które stanowią klucze słownika
- d. Utwórz funkcję **print\_dict\_entry\_dates**, która:
  - i. przyjmuje na wejściu słownik zawierający wszystkie wpisy z listy
  - ii. iteruje po elementach słownika
  - iii. 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 kodem 200 do liczby wszystkich żądań.

**Punkty: 3**

## **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>