

# Wizualizacja danych

Dr Anna Muranova  
Semestr letni 2024, UWM w Olsztynie

## 3. Zajęcie (08.03.2024)

---

- **List (Lista)** jest kolekcją uporządkowaną i zmienną. Zezwala na duplikowanych członków.
- **Tuple (Krotka)** to kolekcja uporządkowana i niezmienna. Zezwala na duplikowanych członków.
- **Set (Zbiór)** to kolekcja, która jest nieuporządkowana, niezmienna (ale można dodawać i usuwać elementy) i nieindeksowana. Brak duplikatów członków.
- **Dictionary (Słownik)** jest kolekcją zmienną. Brak duplikatów członków. Uporządkowany po Python 3.7 i wyżej, nie uporządkowany w Python 3.6 i niżej.

<https://docs.python.org/3/library/stdtypes.html#typeseq>

<https://docs.python.org/3/library/stdtypes.html#mapping-types-dict>

<https://docs.python.org/3/library/stdtypes.html#set-types-set-frozenset>

<https://docs.python.org/3/tutorial/datastructures>.

---

**Ćwiczenie 1** (Listy). (a) • Przećwicz podstawowe funkcje dla **List** w *Python Console* (**append**, **extend**, **insert**, **remove**, **pop**, **clear**, **index**, **count**, **sort**, **reverse**, **copy**).

- Wypisz listę liczby naturalnych od 0 do 14.

(b) Zrób następujące zadania w *Python Console* przy pomocy **List comprehensions** a potem w pliku **program31.py**

- Wypisz piątą potęgę liczb od 0 do 14 naturalnych jako listę.
- Wypisz silnię pierwszych 20 liczb naturalnych jako listę.
- $1, e^1, e^2, e^3, \dots, e^{18}, e^{19}$  jako listę.
- Dla listy nazwisk wypisz listę ich długości.

<https://docs.python.org/3/tutorial/datastructures>

[https://www.w3schools.com/python/python\\_lists\\_comprehension.asp](https://www.w3schools.com/python/python_lists_comprehension.asp)

(c) Utwórz program **program32.py**, w którym:

- Utwórz dwa listy:

```
list1=[1,2,3,4,5,6,7,8,9,10]
list2=[10,20,30,40,50,60,70,80,90,100]
```

Co będzie wynikiem `list1+list2`?

Jak zapisać sumą listów jako sumą wektorów?

- Posortuj listę zawierająca nazwy miesięcy kalendarzowych za pomocą funkcje `sort()`. [https://www.w3schools.com/python/ref\\_list\\_sort.asp](https://www.w3schools.com/python/ref_list_sort.asp)  
Wskazówka Napisz funkcję, która zwraca numer miesiąca po nazwie.
- Napisz funkcję, której parametrami są lista nazwisk i litera. Funkcja musi zwracać listę tych nazwisk, który zaczynają się od liter późniejszych w alfabecie niż podana litera.
- Wybierz z listy nazwisk nazwiska, których długość jest większa niż 6 symbolów. Nie korzystaj z pętli, użyj tzw. **List Comprehensions** we wnętrzu funkcji.
- Napisz funkcję, która sprawdza czy na liście elementy są posortowane od największych do najmniejszych (wtedy zwraca **True**, w przeciwnym wypadku – **False**). Stwórz dwa przypadki testowe.
- Napisz funkcję, która zwraca ciąg wejściowy podniesiony do trzeciej potęgi. Nie korzystaj z pętli, użyj tzw. **List Comprehensions** we wnętrzu funkcji. Stwórz dwa przypadki testowe.
- Napisz funkcję `func(list, n1, n2)` która pobiera listę liczb typu **float** i zamienia w całej liście liczbę `n1` na liczbę `n2`.
- Zmodyfikuj funkcję tak żeby zmieniała na `n2` wszystkie liczby bliskie do `n1` (`isclose`). Parametry `isclose()` muszą być parametrami funkcje.

**Ćwiczenie 2** (Zbiory (set)). <https://docs.python.org/3/tutorial/datastructures>.  
Przećwicz funkcji z sekcji 5.4. na konsoli (Zbiory).

- stwórz zbiór (set) z 5 nazwami Państw
- spróbuj dodać do zbioru istniejące element w zbiorze
- sprawdź, czy *Polska* jest w zbiorze
- usuń ze zbioru istniejący element
- stwórz jeszcze dwa zbiory z nazwami Państw. Wyświetl:
  - miasta, które są w przynajmniej jednym ze zbiorów (odpowiednik sumy zbiorów),
  - miasta, które są jednocześnie we wszystkich zbiorach (odpowiednik części wspólnej zbiorów),
  - miasta, które są w pierwszym zbiorze, ale nie ma ich w drugim zbiorze (odpowiednik różnicy zbiorów),

- wyszukaj sposoby sprawdzenia czy zbiór  $x$  jest podzbiorem zbioru  $y$ . Sprawdź na przykładzie zbiorów Państw.

**Ćwiczenie 3** (Krotki (tuples)). <https://docs.python.org/3/tutorial/datastructures.html#tuples-and-sequences>

(a) Przećwicz funkcji z sekcji 5.3. na konsoli (Krotki).

- stwórz krotkę o 7 elementach różnych typów
- spróbuj zmienić element o indeksie 2
- wyświetl na konsoli element krotki o indeksie 3
- wyświetl na konsoli elementy krotki o indeksach 3,4,5 w jednym poleceniu.
- wyświetl na konsoli element trzeci od końca

(b) Utwórz program **program33.py**

- Napisz funkcję, której parametrem jest krotka i element, a wynikiem – krotką z dodanym elementem (2 możliwości). Zrób to samo z usunięciem elementu.  
[https://www.w3schools.com/python/python\\_tuples\\_update.asp](https://www.w3schools.com/python/python_tuples_update.asp)
- Napisz funkcję, która wyświetla na konsoli elementy listy bez powtórzeń (2 możliwości).

**Ćwiczenie 4** (Słownik (dictionary)). <https://docs.python.org/3/tutorial/datastructures.html> [https://www.w3schools.com/python/python\\_dictionaries\\_loop.asp](https://www.w3schools.com/python/python_dictionaries_loop.asp)

(a) • Utwórz plik **phonenumbers.py** Stwórz słownik , który będzie przechowywał jako klucz imiona, a jako wartość jakieś liczby (odpowiadające numerom telefonu). Następnie wykonaj kilka operacji na słowniku.

- Napisz funkcją `print_phone_numbers`, która wypisuje na konsoli dla podanego słownika wszystkie numery w postaci  
`...(Imie Nazwisko)... ma numer ...()....`
- Zaimportuj plik `phonenumbers.py` do konsoli i wykonaj kilka operacje na stworzonym słowniku (dodaj osobą, usuń osobą, zmień numer)

(b) • Napisz funkcję, która zamienia podaną listą dni tygodnia angielskimi odpowiednikami i funkcję odwrotną.

- Napisz funkcję, która posortuje listę zawierająca nazwy miesięcy kalendarzowych przy pomocy słownika.
- Napisz funkcję, która jako parametr pobiera liczbę w systemie rzymskim (typ **string**), a zwraca wartość liczby w systemie arabskim.

[https://pl.wikipedia.org/wiki/Rzymski\\_system\\_zapisywania\\_liczb](https://pl.wikipedia.org/wiki/Rzymski_system_zapisywania_liczb)

- Napisz odwrotną funkcję, której parametrem jest liczba arabska, a wynikiem – liczba rzymska (nie obowiązkowo używać słownik)