

Przetwarzanie tekstów

1. Podstawowe operacje na tekście

1 pkt za wykonanie wszystkich zadań z tej sekcji

1. Przeczytaj artykuł na temat [operacji na tekście](#)
2. Zdefiniuj dwie zmienne napisowe (*lancuch1*, *lancuch2*), a każda z nich ma zawierać wielolinijkowy tekst w języku polskim, kodowanie UTF-8
3. Spowoduj aby połączona zawartość zmiennych *lancuch1* oraz *lancuch2* została trzykrotnie wypisana (powielona) na ekran — proszę nie używać pętli. Ponadto operacja łączenia i powielania ma być zapisana w jednej linii
4. Zdefiniuj zmienną *lancuch* zawierającą dowolny tekst
5. Za pomocą funkcji 'print', bez użycia pętli, wypisz:
 1. pierwszy znak łańcucha
 2. dwa pierwsze znaki
 3. wszystkie znaki, za wyjątkiem dwóch pierwszych znaków
 4. przedostatni znak łańcucha
 5. trzy ostatnie znaki
 6. wszystkie znaki na pozycjach parzystych
6. Proszę sprawdzić czy łańcuchy znaków mogą być modyfikowane tzn. czy można przypisać nową wartość do zaindeksowanej pozycji łańcucha

2. Wyrażenia regularne

1. (**1 pkt**) Stwórz **skrypt** oraz **test** weryfikujący poprawność działania tego skryptu. Skrypt, w oparciu o moduł [re](#), rozpoznaje ciąg napisów (zawierających polskie znaki Unicode) wczytany z klawiatury (jeden napis w pojedynczej linii) i wypisuje rozpoznany napis wraz informacją o jego rodzaju (liczba lub wyraz), a w przypadku tekstu mieszanego, wypisuje tekst tworzący liczbę oraz tekst tworzący wyraz. Przykład działania:

```
$ python3 skrypt.py
Ala
  Wyraz: Ala
ma
  Wyraz: ma
1kota
  Liczba: 1
  Wyraz: kota
oraz
  Wyraz: oraz
psów20
```

```
Wyraz: psów
Liczba: 20
ponadto
Wyraz: ponadto
50
Liczba: 50
chomików
Wyraz: chomików
^D
```

Wskazówka: pomocne może być użycie konstrukcji opisanej w [Python Switch Statement – Switch Case Example](#), a w przypadku Pythona 3.10+, [dopasowywanie wzorców](#)

3. Obsługa linii komend

1 pkt za wykonanie wszystkich zadań z tej sekcji

1. Proszę utworzyć dwa pliki: "lista.py" oraz "słownik.py"
2. Proszę umieścić, w każdym z nich, następujący kod:

```
print('Ładowanie modułu "{0}"'.format(__name__))
#####
def wypisz():
    print('Wywołano funkcję "wypisz()" modułu "{0}"'.format(__name__))
#####
print('Załadowano moduł "{0}"'.format(__name__))
```

3. Uruchom konsolę Python — `python3`
4. Sprawdź (w konsoli) działanie następujących instrukcji:

```
import lista
import lista
import słownik
import słownik
lista.wypisz()
słownik.wypisz()
```

5. Zmodyfikuj zawartość ww. plików:

- Plik "lista.py" ma zawierać definicję listy o nazwie *lista*, a plik "słownik.py", definicję słownika o nazwie *słownik* — zarówno lista jak i słownik, początkowo, nie zawierają żadnych elementów.
- Każdy z plików powinien zawierać dwie funkcje: `zapisz()` oraz `wypisz()`
- Pierwsza z tych funkcji aktualizuje listę / słownik informacjami o ilości wystąpień poszczególnych liczb zawartych w linii komend, zaś druga zwraca zawartość tej listy / słownika w postaci łańcucha znaków:
"liczba₁:liczbawystąpień, liczba₂:liczbawystąpień, ...".
UWAGA: Lista powinna być możliwie najkrótsza tzn. nie

powinna przechowywać informacji o krotności jeżeli dana liczba nie występuje w ciągu wejściowym

6. Proszę utworzyć skrypt i użyć w nim, odpowiednio, funkcji z pliku "lista.py" / "słownik.py", w zależności od opcji z którą skrypt został wywołany ('--lista' / '--słownik')
7. Utwórz drugi skrypt — w oparciu o moduł [getopt](#) spowoduj, aby drugi skrypt, zamiast dwóch opcji ('--lista' lub '--słownik'), obsługiwał jedną: '--moduł=nazwa', gdzie *nazwa* to: lista lub słownik

4. Zadanie

(2 pkt.) Napisz skrypt, który wczytuje z klawiatury wielolinijkowy tekst (kilka napisów w pojedynczej linii) i przetwarza go w sposób przedstawiony na początku ćwiczeń