

# Python - Przeszukiwanie tekstu

© Copyright by 3bird Projects 2022, <http://edukacja.3bird.pl>

## Uwagi ogólne

Tekst (jako łańcuch *string*) czyszczony jest ze wszystkich znaków interpunkcyjnych, a następnie dzielony jest na poszczególne wyrażenia, które tworzą listę elementów. W liście tej wyszukujemy dane wyrażenie.

Uwaga! Wcięcia w kodzie, mają dla Pythona znaczenie (są konieczne w odpowiednich miejscach)! Python do pobrania (dla Windows): <https://www.python.org/downloads/windows/>  
Uruchamianie skryptu:

```
C:\> python naszSkrypt.py
```

**Uwaga:** Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstępy. Kod należy przepisać ze zrozumieniem.

## Kod skryptu - wersja podstawowa

```
zdanie = """
```

```
Sing, O goddess, the anger of Achilles son of Peleus, that brought countless ills upon the  
Achaean. Many a brave soul did it send hurrying down to Hades, and many a hero did it yield  
a prey to dogs and vultures, for so were the counsels of Jove fulfilled from the day on which the  
son of Atreus, king of men, and great Achilles, first fell out with one another.
```

```
And which of the gods was it that set them on to quarrel? It was the son of Jove and Leto; for  
he was angry with the king and sent a pestilence upon the host to plague the people, because  
the son of Atreus had dishonoured Chryses his priest. Now Chryses had come to the ships of  
the Achaean to free his daughter, and had brought with him a great ransom: moreover he  
bore in his hand the sceptre of Apollo wreathed with a suppliant's wreath and he besought the  
Achaean, but most of all the two sons of Atreus, who were their chiefs.
```

```
"Sons of Atreus," he cried, "and all other Achaean, may the gods who dwell in Olympus grant  
you to sack the city of Priam, and to reach your homes in safety; but free my daughter, and  
accept a ransom for her, in reverence to Apollo, son of Jove."
```

```
On this the rest of the Achaean with one voice were for respecting the priest and taking the  
ransom that he offered; but not so Agamemnon, who spoke fiercely to him and sent him  
roughly away. "Old man," said he, "let me not find you tarrying about our ships, nor yet coming  
hereafter. Your sceptre of the god and your wreath shall profit you nothing. I will not free her.  
She shall grow old in my house at Argos far from her own home, busying herself with her loom  
and visiting my couch; so go, and do not provoke me or it shall be the worse for you."
```

```
"""
```

```
# CZYSZCZENIE TEKSTU (z niepożądanych znaków typu: przecinek, apostrof, nowa linia,  
kropka, itp.; każdy z nich zamieniamy na pusty element [dwa apostrofy]):
```

```
zdanie = zdanie.replace(';', '')
```

```
zdanie = zdanie.replace("'", '')
```

```
zdanie = zdanie.replace('\n', ' ')
```

```
zdanie = zdanie.replace('.', '')
```

```
zdanie = zdanie.replace('?', '')
```

```
zdanie = zdanie.replace(':', '')
```

```
print('Przeszukiwane zdanie po wyczyszczeniu:\n\n===== \n', zdanie, '\n\n===== \n\n')
```

# Nie da się przeszukiwać w *stringu*, więc tworzymy z tekstu listę wyrazów. *String* będzie dzielony na elementy (wyrazy) w oparciu o wykrytą spację:

```
wyrazy = zdanie.split(' ')
```

# Z ciekawości możemy włączyć sobie widok tej listy:

```
print("\n\nZdanie po transformacji na listę:\n\n=====\n", wyrazy, '\n\n=====\n\n')
```

# Teraz przystępujemy do szukania wyrazu w liście i wyliczenia, ile razy występuje.

```
# poszukiwanyWyraz = 'Apollo'
```

```
poszukiwanyWyraz = input('Wpisz wyraz, który chcesz wyszukać: ')
```

# Zakładamy na początku, że będzie 0 wystąpień szukanego wyrażenia.

```
liczbaWystapien = 0
```

```
for kazdyWyraz in wyrazy:
```

```
    if kazdyWyraz == poszukiwanyWyraz:
```

```
        liczbaWystapien += 1
```

```
print(f'\n\nWyraz \'{poszukiwanyWyraz}\'' wystąpił w tekście {liczbaWystapien} raz(y).\n')
```

```
input('\n\nNaciśnij ENTER, aby zakończyć\n')
```

## Kod skryptu - wersja rozbudowana

```
#!/usr/bin/env python
```

```
# Powyższa linia tylko dla systemu Linux.
```

```
from os import system # Wymagane do kolorowania składni w systemie Windows 10/11:
```

```
system("")
```

```
# Uwaga: Składnia nie będzie kolorowana, gdy uruchomimy kod w IDLE Shell (to nie jest
```

```
# prawdziwy terminal) oraz w Windows 7/8.
```

```
from os.path import exists # Wymagane do sprawdzenia, czy plik istnieje
```

```
import mimetypes # Do sprawdzania, czy otwierany plik jest typu "plain-text"
```

```
# Uwaga: Moduł "magic" jest modułem zewnętrznym, którego import
```

```
# wymaga wiedzy informatycznej (a więc nie dla zwykłych użytkowników).
```

```
# Z kolei moduł "bs4 / BeautifulSoup" - i tak najpierw otwiera plik,
```

```
# aby go zbadać (więc jest to bez sensu). Wszystko wskazuje na to,
```

```
# że nie ma pewnego sposobu.
```

```
czyWszystkoOK = False
```

```
licznikPetliWhile = 0
```

```
while not czyWszystkoOK:
```

```
    nazwaPlikuDoOtwarcia = input('\n\nPodaj nazwę pliku tekstowego razem z rozszerzeniem\n\n(*.txt).\n\nPlik powinien znajdować się w tym samym folderze, co skrypt Pythona: ')
```

```
    czyPlikIstnieje = exists(nazwaPlikuDoOtwarcia)
```

```
    # Poniżej, indeks [0] oznacza pierwszy zwracany wyraz, czyli 'text/plain':
```

```
    typPliku = mimetypes.guess_type(nazwaPlikuDoOtwarcia)[0]
```

```
    # Poniżej, indeks [0:4] oznacza zwracane pierwsze 4 litery, czyli 'text'
```

```
    # z pierwotnego 'text/plain'. Stosowane, gdyż inne pliki tekstowe mają postać
```

```

# typu 'text/x-python', itp.
if ((czyPlikIstnieje) and (typPliku[0:4] == 'text')):
    try:
        otwartyPlik = open(nazwaPlikuDoOtwarcia, "r", encoding="UTF-8")
        zdanie = otwartyPlik.read()
        czyWszystkoOK = True # Przerwywamy wykonywanie kolejnej pętli WHILE
    except UnicodeDecodeError:
        print('\n\n\033[1;37;41m BŁĄD: \033[0m')
        print('\033[1;31;40mRozszerzenie pliku wskazuje, że jest to plik tekstowy...\nale jego
zawartość już nie! Wybierz inny plik.\033[0m')
    else:
        print('\n\n\033[1;37;41m BŁĄD: \033[0m')
        print('\033[1;31;40mNazwa pliku lub ścieżka do niego są nieprawidłowe.\nSpróbuj
jeszcze raz.\n\033[0m')

licznikPetliWhile += 1
if licznikPetliWhile >= 3:
    input('\033[1;33;40mCoś chyba nie za bardzo Ci to wychodzi.\nPrzemyśl na spokojnie i
spróbuj później.\n\nNaciśnik ENTER...\033[0m')
    raise SystemExit # Nie używać exit() ani quit() - one zamykają interpreter, os.exit() nie
czyści buffora, zaś sys.exit() wymaga importu biblioteki.

# CZYSZCZENIE TEKSTU (z niepożądanych znaków).
# Można tak, jak poniżej (ale to brzydkie rozwiązanie).

# zdanie = zdanie.replace(',','')
# zdanie = zdanie.replace(' ','')
# zdanie = zdanie.replace('\n',' ')
# zdanie = zdanie.replace('.', '')
# zdanie = zdanie.replace('?', '')
# zdanie = zdanie.replace(':', '')

# Lepiej w tym przypadku zrobić listę elementów do usunięcia:

elementyDoUsuniecia = [',', ' ', '.', '?', ':', '\n', '(', ')']
for kazdyElement in elementyDoUsuniecia:
    zdanie = zdanie.replace(kazdyElement, '') # Każdy z tych elementów zamieniamy na
pusty znak.

print('\n\n\033[1;37;40mPrzeszukiwane zdanie po wyczyszczeniu: \033[0m\n\n\
033[1;30;40m===== \n', zdanie, '\n===== \n\n\033[0m')

# Nie da się przeszukiwać w stringu, więc tworzymy z tekstu listę wyrazów.
# String będzie dzielony na elementy (wyrazy) w oparciu o wykrytą spację:
wyrazy = zdanie.split(' ')

# Z ciekawości możemy włączyć sobie widok listy:
print("\n\033[1;37;40mZdanie po transformacji na listę:\033[0m\n\n\
033[1;30;40m===== \n", wyrazy, '\n===== \n\n\033[0m')

```

```
# Teraz przystępujemy do szukania wyrazu w liście i wyliczenia, ile razy występuje.
# poszukiwanyWyraz = 'Apollo'
poszukiwanyWyraz = input('Wpisz wyraz, który chcesz wyszukać: ')

# Zakładamy na początku, że będzie 0 wystąpień szukanego wyrażenia.
liczbaWystapien = 0

for kazdyWyraz in wyrazy:
    if kazdyWyraz == poszukiwanyWyraz:
        liczbaWystapien += 1

print(f'\n\n\033[1;32;40mWyraz \'{poszukiwanyWyraz}\'' wystąpił w tekście {liczbaWystapien}
raz(y).\n\n\033[0m')

otwartyPlik.close()
input('\n\nNaciśnij ENTER, aby zakończyć...\n')
```

Ostatnia aktualizacja: 6 grudnia 2022.