

# Laboratorium programistyczne — Python: obsługa plików i wykresy

Projekt „Matematyka dla ciekawych świata”  
Robert Paciorek, Łukasz Mazurek

2018-03-18

## 1 Zapisywanie i wczytywanie danych z pliku

Do tej pory wszystkie dane, z których korzystały nasze programy, wprowadzaliśmy bezpośrednio do kodu programu. Tym razem zaczniemy wykorzystywać dane zapisane w pliku.

### 1.1 Pierwszy program — obliczanie wartości funkcji i zapis do pliku

Napišemy program, który obliczy wartości zadanej funkcji (np.  $f(x) = x^2$ ) w podanym przedziale zadaną gęstością, a otrzymane wyniki zapisze do pliku tekstowego. Każdy wiersz w pliku odpowiada jednemu punktowi funkcji i ma postać:

```
x f(x)
```

Jako porządni programiści zadanie to podzielimy na kilka funkcji realizujących jego wybrane fragmenty, aby zapewnić czytelność i łatwość modyfikacji kodu.

**Zadanie 1** *Napisz funkcję która oblicza wartości  $f(x)$  dla  $n$  równomiernie rozłożonych  $x$  z zakresu od  $x_{poczetek}$  do  $x_{koniec}$ . Funkcja powinna przyjmować jako swoje argumenty: funkcję której wartości oblicza, początek i koniec przedziału ( $x_{poczetek}$ ,  $x_{koniec}$ ) oraz wartość związaną z gęstością punktów w przedziale ( $n$ ). Funkcja powinna zwrócić słownik w którym kluczami są wartości  $x$  a wartościami obliczone wartości  $f(x)$ . Wskazówka: **range** nie obsługuje wartości zmiennoprzecinkowych, a w zależności od wartości  $x_{poczetek}$ ,  $x_{koniec}$  i  $n$  możemy mieć takie wartości  $x$ , więc może warto użyć pętli **while***

#### 1.1.1 Zapisywanie tekstu do pliku

Zapis do pliku tekstowego możemy zrealizować w sposób następujący:

```
plik = open('dane.txt', 'wt', encoding='utf8')
plik.write("teskt1\n")
plik.write("teskt2\nteskt3")
plik.close()
```

Jak to działa?

- Polecenie z pierwszej linijki otwiera plik `dane.txt` i zapewnia dostęp do niego poprzez zmienną `plik`. Opcja `'w'` oznacza, że plik jest otwarty „do zapisu” (od angielskiego *write*). Opcja `'t'` oznacza, że plik traktowany jako plik tekstowy<sup>1</sup>. Argument `encoding` pozwala na określenie kodowania użytego do zapisu pliku tekstowego, jest on opcjonalny i gdy nie zostanie podany kodowanie pliku zależne jest od ustawień systemowych.
- Druga i trzecia komenda zapisuje podany jako argument tekst do pliku `dane.txt` (zwróć uwagę na wstawianie nowej linii przy pomocy `'\n'`)

<sup>1</sup> Tekst możemy zapisywać także do plików otwieranych jako binarne, w takim wypadku argument funkcji `write` musi mieć typ *bytes* a nie *str*, czyli być jawnie zakodowany w jakimś standardzie.

- Ostatnie polecenie zamyka dostęp do pliku `dane.txt`.

Po uruchomieniu powyższego kodu powinien zostać utworzony plik „dane.txt”<sup>2</sup>, zawierający 3 linie tekstu. Jeżeli plik taki wcześniej istniał zostanie on nadpisany.

**Zadanie 2** *Napisz funkcję która zapisze przekazany do niej słownik do pliku tekstowego w taki sposób, że w pierwszej kolumnie będzie znajdował się klucz, a w drugiej (rozdzielonej spacją) będzie związana z nim wartość. Zakładamy że dane w słowniku nie zawierają spacji, ani znaków nowej linii.*

## 1.2 Drugi program — wczytywanie danych z pliku i generowanie wykresów

Spróbujmy teraz zrobić operację odwrotną czyli wczytać dane z pliku do słownika.

Wczytywanie danych z pliku możemy zrealizować w sposób następujący:

```
plik = open('dane.txt', 'rt', encoding='utf8')
for linia in plik:
    print(linia, end=" ")
plik.close()
```

**Zadanie 3** *Napisz funkcję która wczyta dane z pliku do słownika, w taki sposób że każda linia pliku odpowiada jednemu wpisowi w słowniku, pierwsza kolumna odpowiada kluczowi, a druga (rozdzielonej spacją) wartości.*

*Wskazówka: do podziału linii możesz użyć metody `split()`, pamiętaj aby przekonwertować napisy na wartości liczbowe.*

### 1.2.1 Wykresy

Do rysowania wykresów w Pythonie posłużyć może np. pakiet `matplotlib`. Aby go używać, na początku programu musimy go zaimportować i wybrać odpowiedni tryb poleceniami:

```
import matplotlib
matplotlib.use('Agg')
import matplotlib.pyplot as plt
```

Użyta w drugiej linii funkcja `use` ustawia bibliotekę<sup>3</sup> używaną do rysowania wykresów przez `matplotlib`. Instrukcja `import` ze słowem kluczowym `as` pozwala posługiwać się podanym aliasem zamiast pełną nazwą zaimportowanego modułu. Dzięki temu w dalszej części możemy pisać `plt`, zamiast `matplotlib.pyplot`.

Do tworzenia wykresów będzie nam służyła funkcja `plot()` z biblioteki zaimportowanej jako `plt`. Będziemy ją wywoływać w sposób następujący: `plt.plot(x, y, format)`, gdzie `x` jest listą kolejnych wartości osi X, `y` jest listą odpowiadających im wartości osi Y, a `format` określa typ rysowanego wykresu. Na przykład:

```
x = [1, 3, 13]
y = [1, 2, 0]
plt.plot(x, y, ".r")
```

Po wygenerowaniu wykresu musimy zapisać wykres w pliku graficznym<sup>4</sup>:

```
plt.savefig('wykres.png')
```

<sup>2</sup> Jeżeli w `repl.it` nie widać tego pliku należy kliknąć przycisk *Add new file* w lewym górnym rogu strony i utworzyć nowy plik o dowolnej nazwie. Od tej pory pliki tworzone z poziomu uruchamianego kodu będą widoczne w pasku z plikami należącymi do projektu, zatem uruchom kod jeszcze raz.

<sup>3</sup> `'Agg'` (*Anti-Grain Geometry*) obsługuje grafikę rastrową i wspiera format plików png. Pełna lista obsługiwanych bibliotek dostępna w dokumentacji: [https://matplotlib.org/faq/usage\\_faq.html#what-is-a-backend](https://matplotlib.org/faq/usage_faq.html#what-is-a-backend) Jeżeli używasz Pythona zainstalowanego na lokalnym komputerze możesz spróbować np. `'Qt4Agg'` lub innych bibliotek (w zależności od zainstalowanych modułów) celem rysowania wykresu w okienku GUI, zamiast zapisywania go do pliku.

<sup>4</sup> Jeżeli używasz biblioteki rysującej w okienku GUI, zamiast `plt.savefig('wykres.png')` powinieneś użyć `plt.show()`

**Zadanie 4** *Napisz funkcję która narysuje wykres w oparciu o przekazany do niej słownik, w którym klucze traktowane są jako wartości osi X, a wartości jako wartości osi Y*

*Wskazówka: musisz dokonać konwersji słownika na dwie listy, dane w listach powinny być posortowane wg. wartości x'ów.*

## 1.2.2 Podsumowanie

Spróbujmy teraz złożyć wszystkie elementy w całość. Zauważ że uzyskaliśmy zestaw funkcji które pozwalają na:

- obliczanie wartości zadanej funkcji
- zapisywanie wyniku tych obliczeń do pliku
- wczytywanie danych potrzebnych do narysowania wykresu z pliku
- rysowanie wykresów

a dzięki podziałowi zadania na osobne funkcję także na bezpośrednie rysowanie wykresu w oparciu o obliczone wartości funkcji (bez zapisywania danych do pliku).

## 1.2.3 Jeszcze kilka słów o wykresach

Możliwe jest rysowanie kilku wykresów na tym samym obrazku, w tym celu należy kilkakrotnie wywołać funkcję `plot()` z różnymi zestawami danych lub opcjami formatowania. Na przykład:

```
x = [1, 3, 13]
y = [1, 2, 0]
plt.plot(x, y, ".r")
plt.gcf().clear()
plt.plot(x, y, "-b")
plt.savefig(nazwaPliku)
```

Opcje formatowania opisane są w dokumentacji funkcji `plot()`, aby się z nią zapoznać użyj `help(plt.plot)`. Jeżeli chcemy wyczyścić wykres i zacząć rysować go od nowa, możemy skorzystać z `plt.gcf().clear()`.

## 2 Zadania dodatkowe

**Zadanie 5** *Zastanów się czy (biorąc pod uwagę format argumentów funkcji `plot()` oraz rozwiązania poprzednich zadań) wybór słownika jako formatu danych używanych do rysowania wykresów był słuszny?*

## 3 Praca domowa

Rozwiązania zadań domowych należy przesłać na adres [licealisci.pracownia@icm.edu.pl](mailto:licealisci.pracownia@icm.edu.pl) wpisując jako temat wiadomości `gx PD3`, gdzie `x` to numer grupy, np. `g3 PD3` dla grupy nr. 3, itd. Zadania domowe są nie obowiązkowe, jednak zachęcamy do ich robienia i wysyłania rozwiązań (nawet niekompletnych). Na ten adres można także nadsyłać ewentualne pytania do zadań (zarówno domowych jak i innych zamieszczonych w skrypcie), w tym wypadku także prosimy o umieszczenie w temacie wiadomości `gx`, gdzie `x` to numer grupy.

Prosimy o załączanie rozwiązań (kodu programów) w postaci załączników lub odnośnika do rozwiązań na [repl.it](http://repl.it), w tym drugim przypadku prosimy dodatkowo o wklejenie kodu w treść wiadomości.

Termin nadsyłania zdań domowych to ..... godz. 16<sup>59</sup>.

**Zadanie domowe 1 (1 pkt)** Napisz program, który wczyta tekst z pliku `tekst.txt`, i wypisze na ekran co piąty znak tego tekstu, począwszy od pierwszego (liczymy wszystkie znaki, również spacje i znaki interpunkcyjne).

**Zadanie domowe 2 (2 pkt)** Napisz program rysujący wykres funkcji na podstawie danych wczytanych z pliku `"dane.txt"`. Pierwszy wiersz pliku zawiera kolejne wartości  $x$  rozdzielane spacjami, a drugi wartości  $f(x)$  również rozdzielane spacjami. Przykład pliku (zauważ że może być więcej niż jedna spacja do rozdzielania pól):

```
1 2      3      7 9      11 13
3 2.3 2.1 2 2.9 2.9 3
```

Wskazówka: do wczytania pojedynczej linii pliku można użyć metody `readline()` obiektu zwróconego przez funkcję `open()`

**Zadanie domowe 3 (3 pkt)** Napisz funkcję otrzymującą dwie listy  $x$  i  $y$  która posortuje ich elementy wg porządku elementów w liście  $x$ . Dla list postaci:

```
x = [1, 9, 5, 3, 7]
y = ['a', 'b', 'c', 'd', 'e']
```

Funkcja powinna zwrócić (np. w formie słownika lub listy list) listy w postaci:

```
x = [1, 3, 5, 7, 9]
y = ['a', 'd', 'c', 'e', 'b']
```

Wskazówka 1: takie dwie listy można zamienić np. na słownik.

Wskazówka 2: przypomnij sobie iterowanie po indeksach listy