KURS PROGRAMOWANIA W JĘZYKU PYTHON

TYDZIEŃ 4 – MODUŁY I BIBLIOTEKI

1. Moduły i importy

Moduły

Rzeczywiste programy są złożone. Nawet małe oprogramowanie zawiera tysiące linii kodu, dlatego też pisanie kodu w ciągłym przepływie utrudnia programistom i deweloperom jego zrozumienie. Aby uprościć zrozumienie i logicznie je podzielić, programiści używają programowania modułowego. Jest to technika rozdzielania dużych zadań kodowania na krótsze, logiczne i bardziej elastyczne podzadania.

Moduły są kolekcją powiązanych kodów, które są spakowane w ramach programu Python. Są to pliki Pythona zawierające prawidłowe definicje i instrukcje Pythona. Są to normalne nazwy plików, które podczas tworzenia używają przyrostka .py.

Dzielą się na:

- Wbudowane np. math, datetime, statistics, random, os, sys
- Zdefiniowane – czyli te stworzone przez użytkownika

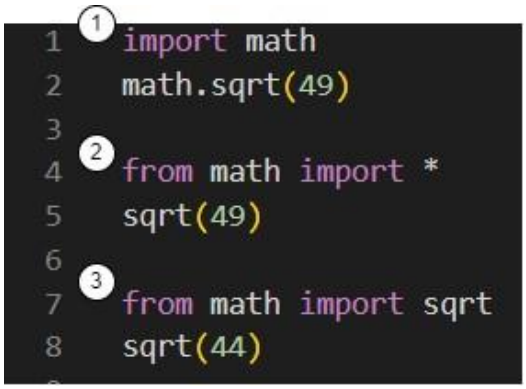
Programiści mogą używać tych modułów w programach Pythona poprzez bezpośrednie importowanie ich poprzez wywołanie ich nazwy wraz ze słowem kluczowym "import". Np. import math. Przed importem może być konieczne zainstalowanie modułu za pomocą **pip** – instalatora bibliotek, pakietów, modułów w pythonie. Instalator pip jest zaprojektowany do użycia z linii poleceń (command line):

Poniższe polecenie zainstaluje najnowszą wersję wybranego modułu.

```
PS C:\Users\MJ8079> python -m pip install SomePackage
```

Importowanie modułów

W zależności od sposobu importu, późniejsze użycie funkcji zawartych w module nieco się różni.



1. Importowanie całego modułu z całą jego zawartością za pomocą **import math** – w takim wypadku, aby użyć funkcji zawartej w module przed jej wywołaniem musimy dodać prefiks **math.**
2. Importowanie całego modułu z całą jego zawartością za pomocą **from math import *** - w takim wypadku nie musimy używać prefiksu przed wywołaniem funkcji.
3. Importowanie konkretnej funkcji z modułu – w takim wypadku nie musimy używać prefiksu math przed wywołaniem wybranej przez nas funkcji. Należy jednak pamiętać, że zaimportowaliśmy tylko tą jedną funkcję **sqrt()** i nie mamy dostępu do innych funkcji tego modułu.

Sqrt() – funkcja służąca do obliczenia pierwiastka kwadratowego z wybranej liczby.

Dodatkowo przy importowaniu modułów możemy używać **aliasów**. Są to niejako skróty nazw importowanego modułu ustalone przez nas do późniejszego łatwiejszego użycia. Alias definiujemy przy imporcie i użuwamy później w następujący sposób:

```
import math as m
m.sqrt(44)
```

2. Korzystanie z gotowych bibliotek

Biblioteka to ogólny termin, który obejmuje zestaw kodu/instrukcji Pythona wielokrotnego użytku. Zazwyczaj biblioteka Pythona jest zbiorem powiązanych modułów połączonych pod jedną nazwą. Jest ona powszechnie używana przez programistów do udostępniania społeczności kodu wielokrotnego użytku. Eliminuje to konieczność pisania kodu Pythona od zera.

Standardowe biblioteki są dołączane do interpretera Pythona, gdy programiści i deweloperzy instalują go w swoim systemie.

Niektóre popularne przykłady bibliotek Pythona to: matplotlib, Pygame, Pytorch, Requests, Beautifulsoup itp.

Biblioteki importujemy w taki sam sposób jak moduły. Przed importem może być konieczne zainstalowanie biblioteki za pomocą **pip** – instalatora bibliotek, pakietów, modułów w pythonie – podobnie jak to miało miejsce przy instalowaniu modułu.

Przykładem biblioteki może być matplotlib. Matplotlib to wszechstronna biblioteka do tworzenia statycznych, animowanych i interaktywnych wizualizacji w Pythonie.

Przykład:

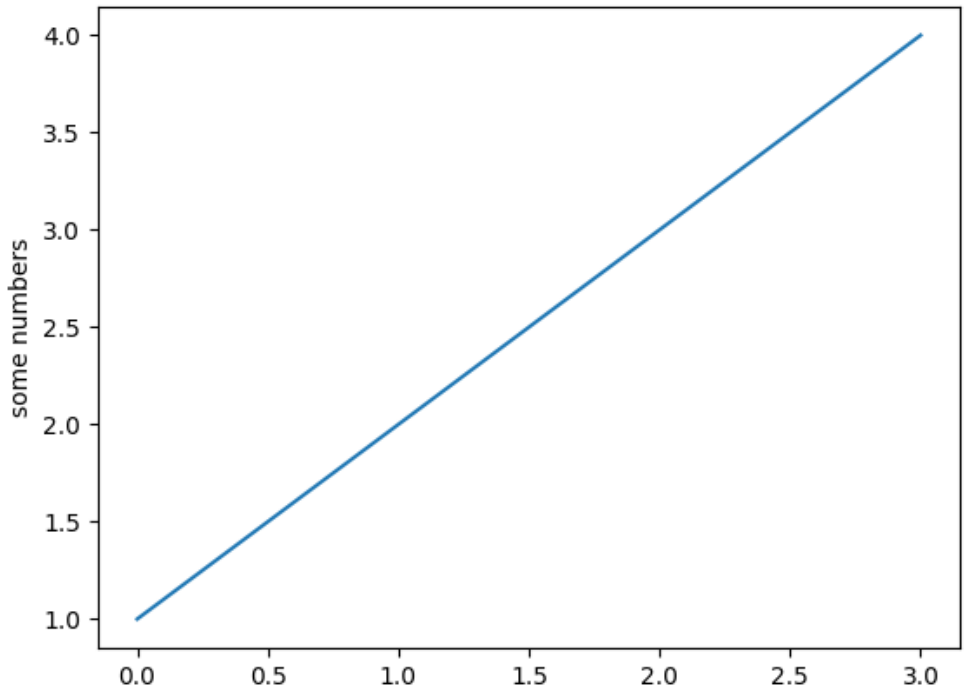
W tym wypadku wraz z biblioteką matplotlib zaimportowaliśmy pyplot.

Pyplot to moduł Matplotlib, który zapewnia interfejs podobny do MATLABa (MATLAB - program komputerowy będący interaktywnym środowiskiem do wykonywania obliczeń naukowych i inżynierskich, oraz do tworzenia symulacji komputerowych.

W skrócie: Pyplot to moduł biblioteki matplotlib, który pozwala na tworzenie różnorakich wykresów i wprowadzanie do nich zmian: dodanie etykiet bądź opisów osi x i y.

```
14  import matplotlib.pyplot as plt
15
16  plt.plot([1, 2, 3, 4])
17  plt.ylabel('some numbers')
18  plt.show()
19
```

Output:



3. Generatory liczb pseudolosowych

Z racji, że generatory liczb pseudolosowych to temat, który może być łatwiejszy do pokazania aniżeli opisania słowami zachęcamy do zapoznania się z krótkim filmikiem o generatorach liczb pseudolosowych 😊.

https://pl.khanacademy.org/computing/computer-science/cryptography/crypt/v/random-vs-pseudorandom-number-generators