

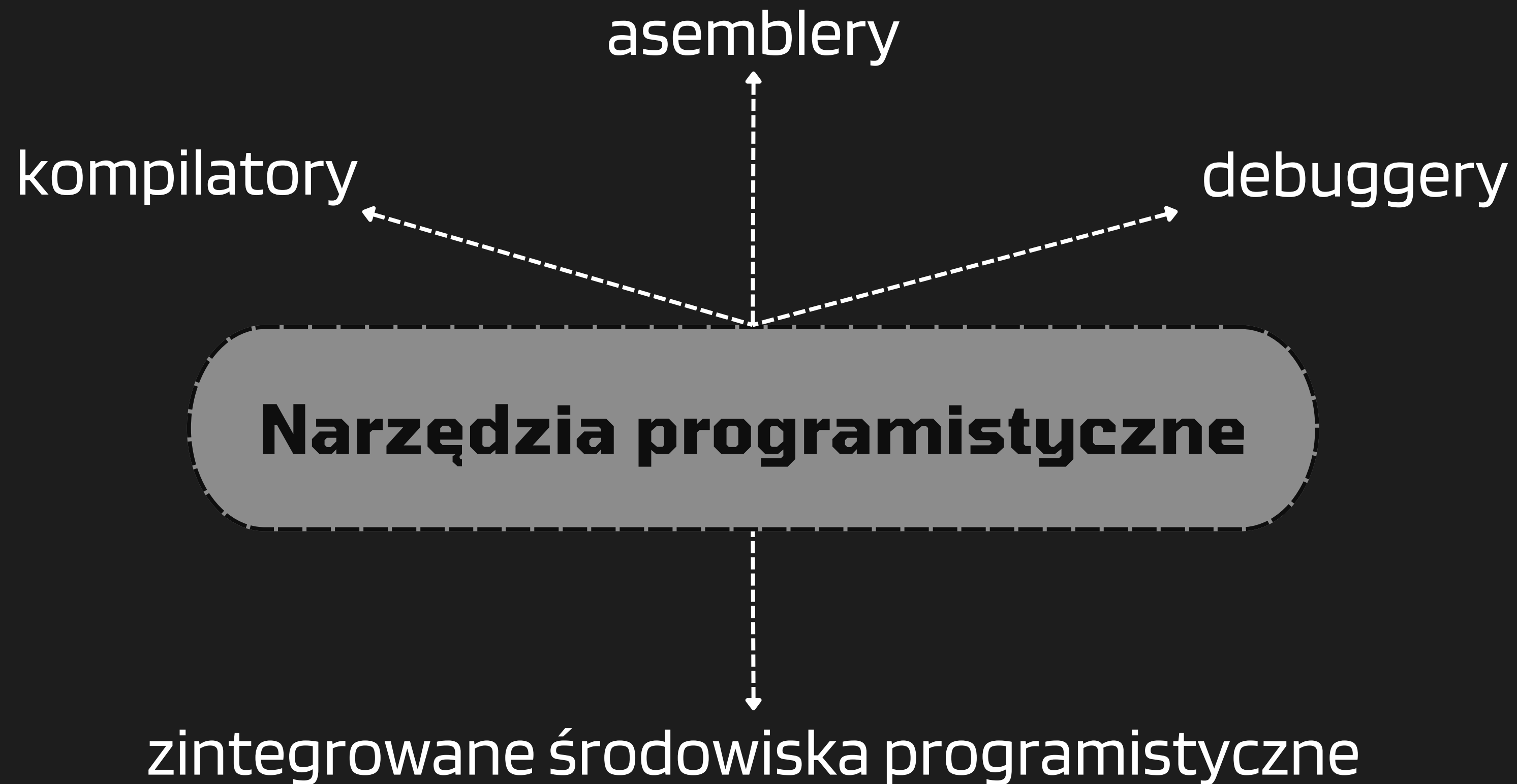


ZAAWANSOWANE NARZĘDZIA I BIBLIOTEKI PROGRAMISTYCZNE



Czym jest narzędzie programistyczne?

Jest to program komputerowy służący do tworzenia, modyfikowania, testowania i konserwacji oprogramowania.



Kompilator

kod źródłowy

tłumacz

kod maszynowy
/kod assemblera

Asembler

kod assemblera

Asembler

kod maszynowy

Debugger

Program komputerowy służący do dynamicznej analizy innych programów, w celu odnalezienia i identyfikacji zawartych w nich błędów, zwanych z angielskiego bugami.

Proces nadzorowania wykonania programu za pomocą debuggera określa się mianem debugowania.

Zintegrowane środowisko programistyczne IDE

Program lub zespół programów (środowisko) służących do tworzenia, modyfikowania, testowania i konserwacji oprogramowania.

IDE udostępniają funkcjonalności takie jak:

-> edycję kodu źródłowego, kompilowanie kodu źródłowego,
-> tworzenie zasobów programu (tzn. formatek/ekranów/okien dialogowych,
-> menu, raportów, elementów graficznych jak ikony, obrazy),
-> tworzenie baz danych, komponentów i innych.

Listopad 2022
porównywany
do zeszłego
roku

TOP IDE Top Integrated Development
Environment index
TOP IDE Top Integrated Development Environment

Rank	Change	IDE	Best for	Share	Trend
1		Visual Studio	C/C++/C#	28.01 %	-0.6 %
2	↑	Visual Studio Code	C/C++/C#	13.46 %	+1.7 %
3	↓	Eclipse	C/C++	12.49 %	-1.2 %
4		Android Studio	Kotlin/Java/C/C++	8.69 %	-0.4 %
5		pyCharm	Python	8.51 %	+0.6 %
6		IntelliJ	Java	7.31 %	+1.0 %
7		NetBeans	Java/JS/C/C++/HTML/CSS	4.69 %	-0.2 %
8	↑	Sublime Text	JS/HTML/CSS/PHP/Python	3.72 %	+0.4 %
9	↓	Xcode	iPhone apps development	3.0 %	-0.5 %
10		Atom	duża zgodność ze wszystkimi systemami operacyjnymi	2.74 %	-0.2 %

Czym jest biblioteka programistyczna?

Jest to plik dostarczający **podprogramy**, **dane** oraz **typy danych**, które mogą zostać wykorzystane z poziomu kodu źródłowego programu.

Użycie bibliotek to sposób na ponowne wykorzystanie tego samego kodu.

Python – biblioteki programistyczne ułatwiające pracę z danymi

Matplotlib

Służy do tworzenia
różnego rodzaju
wykresów.

Działa na podobieństwo
środowisko Matlab.

Pandas/Numpy

Biblioteka
przeznaczona do
ładowania, czyszczenia,
przygotowywania i
analizowania danych

Scikit-learn

Umożliwia
przeprowadzanie
algorytmów
klasyfikacji, regresji i
klastrowania.

Poniżej przykład użycia bibliotek:

- Matplotlib
- Numpy

do wygenerowania wykresu w pythonie

```
# WYKRES 2D
import matplotlib.pyplot as plt
import numpy as np

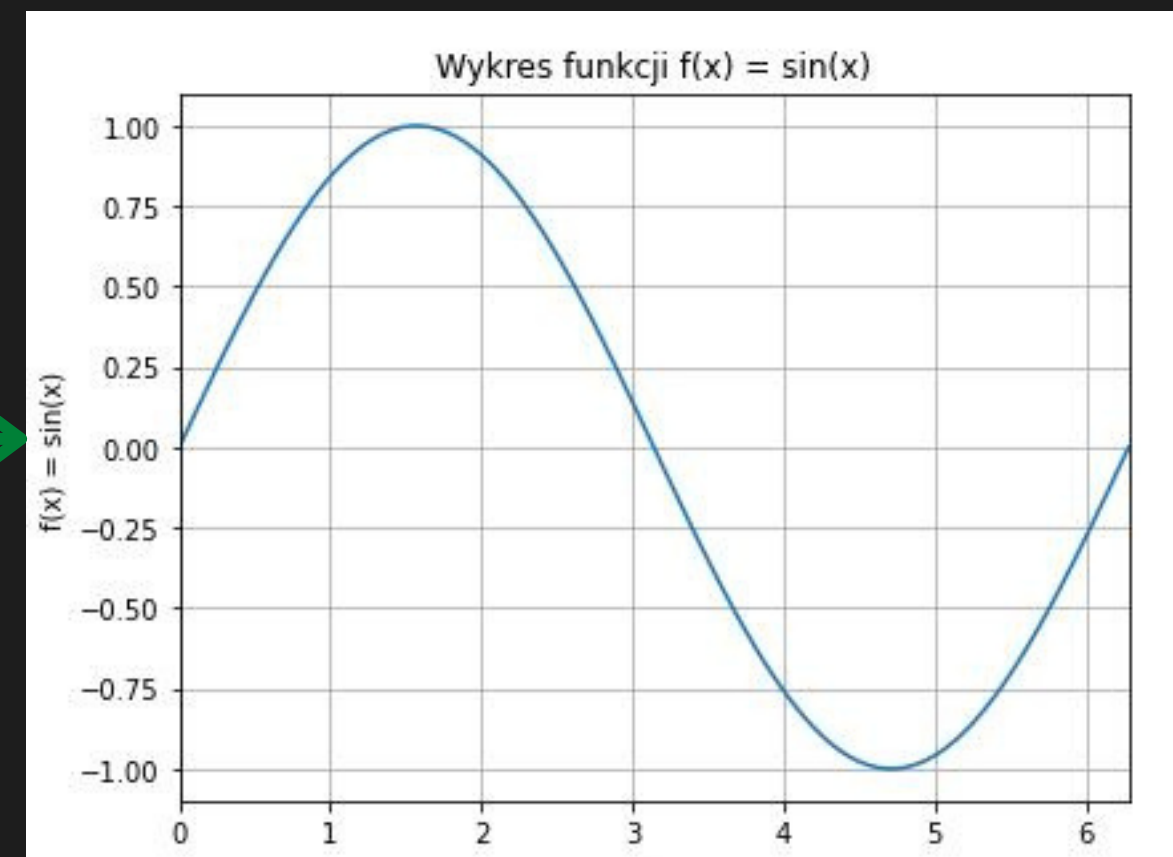
# tworzę listę punktów x oraz wyliczam y=f(x):
x = np.linspace(0, np.pi * 2, 100)
print(x)
y = np.sin(x)

# tworzę wykres:
plt.plot(x, y)
plt.grid(True)
plt.xlim(0, np.pi * 2)
plt.xlabel("x")
plt.ylabel("f(x) = sin(x)")
plt.title("Wykres funkcji f(x) = sin(x)")
plt.savefig("fig1.jpg", dpi = 72)
plt.show()
```

```
# tworzenie wykresow
# generowanie danych do wykresu

# lista 100 elementow od 0 do 3,14*2
# funkcją f(x) jest cos(x)

# wykres z osiami x i y
# włączam wyświetlanie na wykresie siatki
# przedział danych dla x
# tytuł osi x
# tytuł osi y
# tytuł wykresu
# zapisuję wykres do pliku fig1.jpg
# wyświetlam wykres
```



Przykłady użycia biblioteki pandas:

```
import pandas as pd
dictionary = {                                     # słownik z danymi
    'Country': ['Afganistan', 'Albania', 'Algeria', 'Andorra'],
    'Region': ['ASIA', 'EASTERN EUROPE', 'NORTHERN AFRIKA', 'WESTERN EUROPE'],
    'Population_[mln]': [31.056997, 3.581655, 32.930091, 0.071201]
}
df = pd.DataFrame(dictionary)                     # ramka danych ze słownika
```

	Country	Region	Population_[mln]
0	Afganistan	ASIA	31.056997
1	Albania	EASTERN EUROPE	3.581655
2	Algeria	NORTHERN AFRIKA	32.930091
3	Andorra	WESTERN EUROPE	0.071201

```
# wiersz, ktorego wartosc dla kolumny Region wynosi 'ASIA'
print(df[df['Region'] == 'ASIA'])
# wiersze, dla ktorych wartosc populacji jest miedzy 3 a 32mln
print(df[(df['Population_[mln]'] > 3) & (df['Population_[mln]'] < 32)])
```

	Country	Region	Population_[mln]
0	Afganistan	ASIA	31.056997

	Country	Region	Population_[mln]
0	Afganistan	ASIA	31.056997
1	Albania	EASTERN EUROPE	3.581655

```
print(df['Population_[mln]'].sum()) # zsumuj wszystkie wartosci dla kolumny Population
print(df['Population_[mln]'].max()) # największa wartosc z kolumny Population
```

67.639944
32.930091

```
print(df.groupby('Population_[mln]').min()) # grupuj według Populacji zaczynając od najmniejszej
print(df.groupby('Region')['Population_[mln]'].min()) # grupuj po według Regionu i populacji alfabetycznie
```

```
df['Nowa_kolumna'] = 1 # dodaj kolumnę z wierszami o wartościach 1
```

Przykład użycia biblioteki scikit-learn:

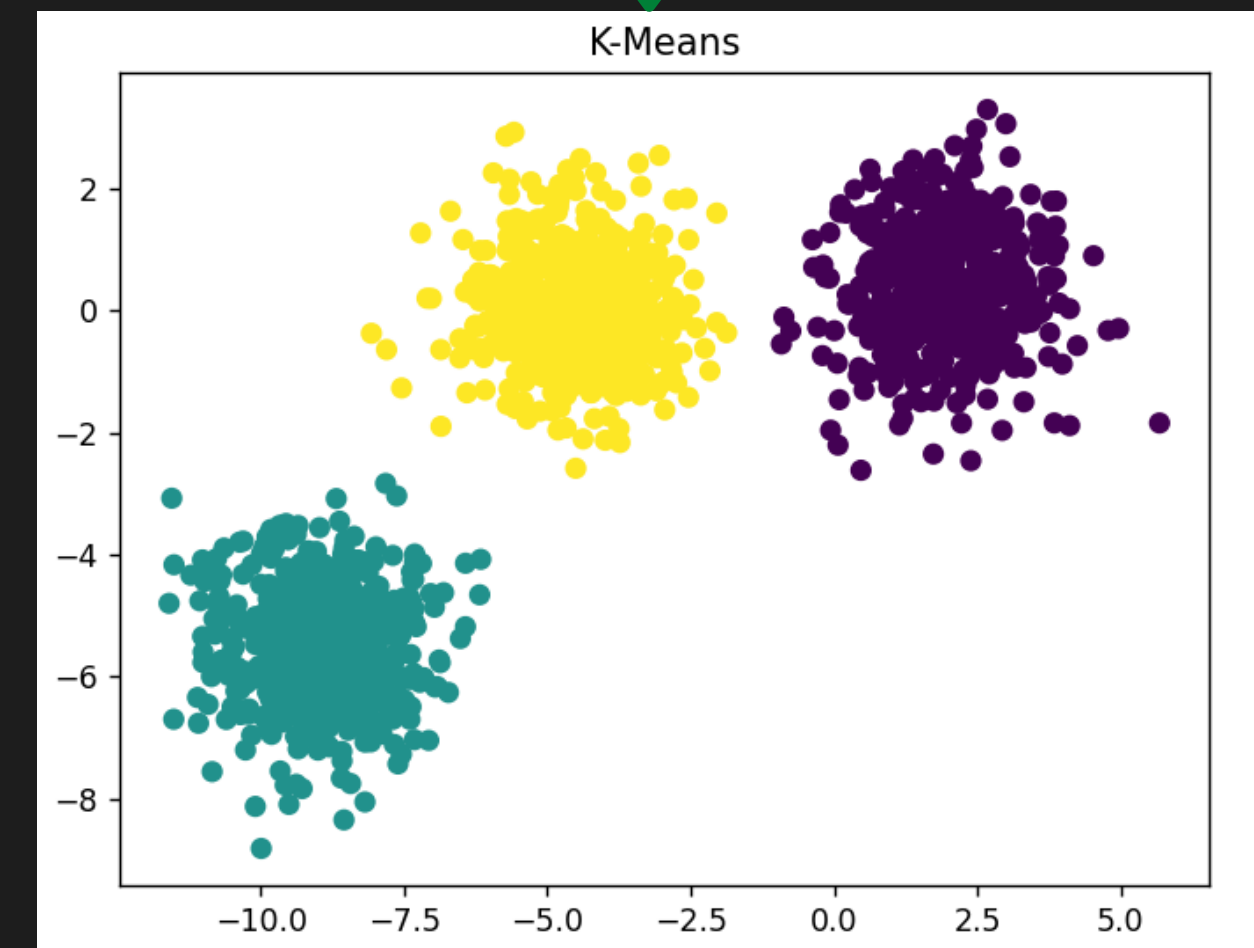
```
# SCIKIT LEARN
import numpy as np
import matplotlib.pyplot as plt

from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

n_samples = 1500
random_state = 170
X, y = make_blobs(n_samples=n_samples, random_state=random_state) # Generowanie plam gaussowskich do grupowania

# KMeans:
# liczba klastrow = 3,
# uruchamia się 10 razy z innymi centroidami,
# losowość deterministyczna dla centroidow (pseudolosowość)
y_pred = KMeans(n_clusters=3, n_init="auto", random_state=random_state).fit_predict(X)

# Tworzenie wykresu
plt.plot(X.min(), y.max())          # wykres
plt.scatter(X[:, 0], X[:, 1], c=y_pred) # rozrzut zaleznosci y od x
plt.title("K-Means")                # tytuł
plt.show()
```





DZIĘKUJĘ :)