

# MATPLOTLIB.

K. S. M. G. T. P.

Wydział Fizyki, Matematyki i Informatyki  
Politechnika Krakowska im. Tadeusza Kościuszki

17.05.2019

# Plan.

- 1 Wstęp.
- 2 Architektura - omówienie.
- 3 Warstwa - Backend.
- 4 Warstwa - Artist.
- 5 Warstwa - Scripting.
- 6 Przykłady.
- 7 Źródła.

# Wstęp.

Matplotlib to biblioteka do tworzenia wykresów dla języka programowania Python i jego rozszerzenia numerycznego NumPy. Zawiera ona API Pylab zaprojektowane tak aby było jak najbardziej podobne do MATLABa, przez co jest łatwy do nauczenia przez jego użytkowników. Matplotlib został napisany i jest utrzymywany głównie przez Johna Huntera, i jest dostępny na licencji przypominającej licencję BSD (licencji zgodnych z zasadami wolnego oprogramowania).



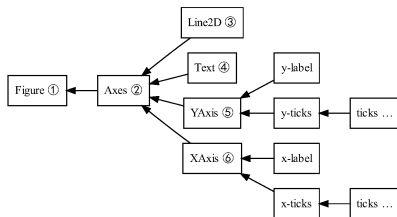
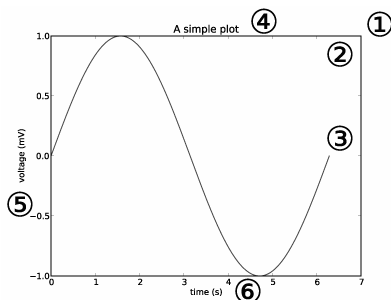
**Figure** jest to obiekt najwyższego poziomu w bibliotece Matplotlib. Zawiera on i zarządza wszystkimi elementami danej grafiki (np. wykresu). Jednym z podstawowych zadań architektonicznych, które Matplotlib musi rozwiązać, jest implementacja struktury do reprezentowania i manipulowania obiektem **Figure**, niezależnie od czynności renderowania tego obiektu, dzięki czemu można budować coraz bardziej wyrafinowane konstrukcje w obrębie obiektu **Figure**. Matplotlib enkapsuluje nie tylko interfejsy rysowania, ale także podstawową obsługę zdarzeń i okienkowanie najpopularniejszych zestawów narzędzi interfejsu użytkownika. W celu uzyskania powyższego efektu architektura została podzielona na trzy warstwy, które można traktować jako stos. Każda warstwa, która znajduje się nad inną warstwą, wie, jak rozmawiać z warstwą pod nią, ale dolna warstwa nie jest świadoma warstw nad nią.

Na dole stosu znajduje się warstwa Backend, która zapewnia konkretne implementacje abstrakcyjnych klas interfejsów:

- **FigureCanvas** - powierzchnia do rysowania,
- **Renderer** - wykonuje rysunek,
- **Event** - obsługuje zdarzenia wejściowe (np. mysz).

# Warstwa - Artist.

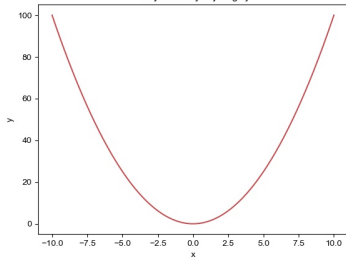
Warstwa Artist jest środkową warstwą stosu. Wszystkie elementy widoczne w obiekcie **Figure** takie jak: tytuł, linie, etykiety, obrazy itp. odpowiadają poszczególnym instancjom warstwy Artist. Klasą bazową jest klasa `matplotlib.artist.Artist`. Sprzężenie pomiędzy warstwą Artist a warstwą Backend jest uzyskiwane za pomocą metody **draw**.



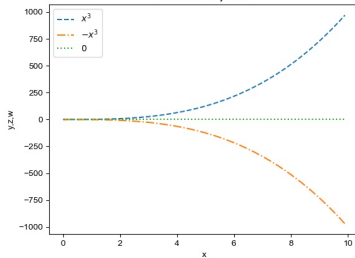
Większość języków specjalnego przeznaczenia, do analizy danych i wizualizacji zapewnia prosty interfejs skryptowy, który ma na celu uproszczenie typowych zadań. Matplotlib robi to również w interfejsie `matplotlib.pyplot`. Interfejs ten obsługuje wiele szablonów służących do tworzenia wykresów i osi i operacji na nich. Dzięki `matplotlib.pyplot`, w relatywnie prosty sposób można dokonywać odpowiedniej dla użytkownika wizualizacji danych.

# Przykłady.

Przykład krzywej ciągłej

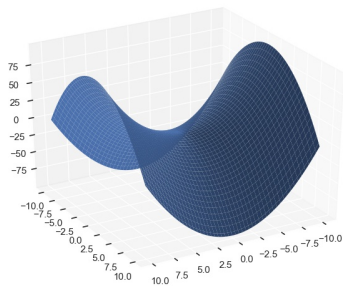
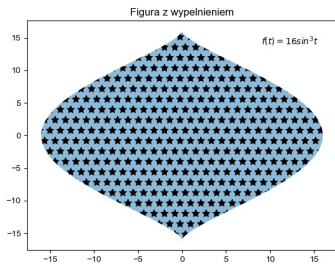


Różne rodzaje linii

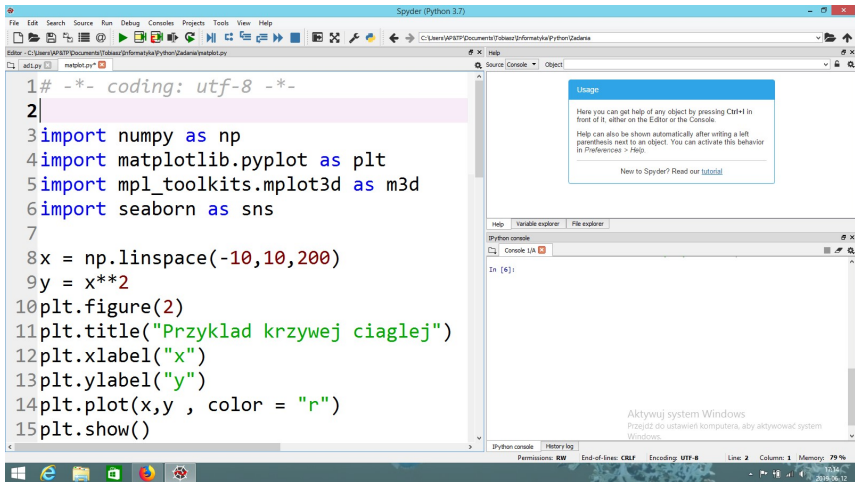




# Przykłady.



# Przykłady.



The image shows a screenshot of the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.7)". The code editor contains the following Python code:

```
1# -*- coding: utf-8 -*-
2|
3import numpy as np
4import matplotlib.pyplot as plt
5import mpl_toolkits.mplot3d as m3d
6import seaborn as sns
7
8x = np.linspace(-10,10,200)
9y = x**2
10plt.figure(2)
11plt.title("Przykład krzywej ciaglej")
12plt.xlabel("x")
13plt.ylabel("y")
14plt.plot(x,y , color = "r")
15plt.show()
```

The code is written in a light blue font on a white background. The first line is a comment indicating the encoding. The code imports several libraries: numpy, matplotlib.pyplot, mpl\_toolkits.mplot3d, and seaborn. It then defines two variables, x and y, and uses matplotlib to create a 2D plot of a parabola. The plot title is "Przykład krzywej ciaglej" (Example of a continuous curve). The x-axis is labeled "x" and the y-axis is labeled "y". The plot is drawn in red. Finally, plt.show() is called to display the plot.

On the right side of the IDE, there is a "Help" window titled "Usage" with the following text:

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our [tutorial](#)

Below the help window, there is a "Python console" window with the prompt "In [6]:". At the bottom of the IDE, there is a status bar showing "Permissions: RW", "End-of-lines: CRLF", "Encoding: UTF-8", "Line: 2", "Column: 1", and "Memory: 79 %".

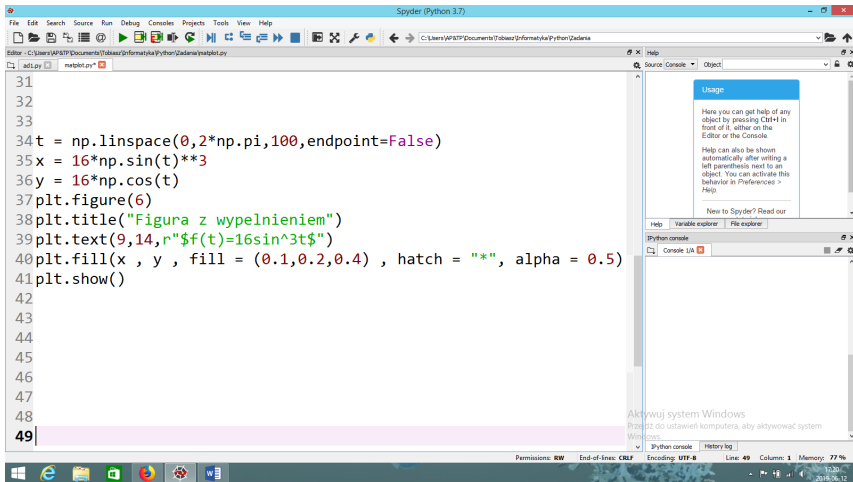
# Przykłady.

The screenshot shows the Spyder Python IDE interface. The main editor window contains the following Python code:

```
16
17 x = np.arange(0,10,0.1)
18 y = x**3
19 z = -(x**3)
20 w = 0*x
21 plt.figure(7)
22 plt.title("Rozne rodzaje linii")
23 plt.xlabel("x")
24 plt.ylabel("y,z,w")
25 plt.plot(x,y, "--", label = "$x^3$")
26 plt.plot(x,z, "-.", label = "$-x^3$")
27 plt.plot(x,w, ":", label = "$0$")
28 plt.legend()
29 plt.show()
30
```

The right-hand side of the IDE features a Help panel with a 'Usage' section, a Variable explorer, a File explorer, and a Python console. The Python console shows the prompt 'In [6]:' and is currently empty. At the bottom of the IDE, the status bar displays 'Permissions: RW', 'End-of-lines: CRLF', 'Encoding: UTF-8', 'Line: 2', 'Column: 1', and 'Memory: 78 %'. The Windows taskbar is visible at the bottom of the screen.

# Przykłady.



The image shows a screenshot of the Spyder Python IDE interface. The main editor window displays a Python script with the following code:

```
31
32
33
34 t = np.linspace(0, 2*np.pi, 100, endpoint=False)
35 x = 16*np.sin(t)**3
36 y = 16*np.cos(t)
37 plt.figure(6)
38 plt.title("Figura z wypelnieniem")
39 plt.text(9, 14, r"$f(t)=16\sin^3t$")
40 plt.fill(x, y, fill = (0.1, 0.2, 0.4), hatch = "*", alpha = 0.5)
41 plt.show()
42
43
44
45
46
47
48
49
```

The right-hand side of the interface features a help window titled "Usage" with the following text:

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.

New to Spyder? Read our

Help Variable explorer File explorer

Python console

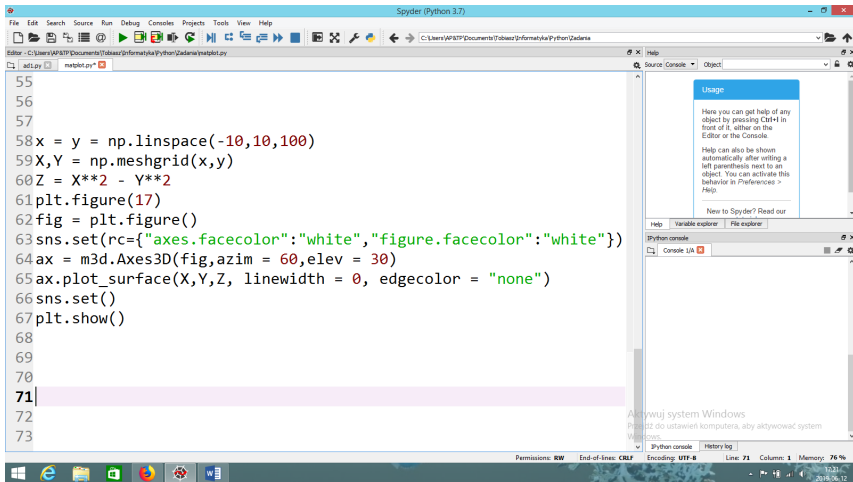
Console 1/A

Python console History log

At the bottom of the IDE, a Windows watermark is visible: "Aktywuj system Windows. Przejdź do ustawień komputera, aby aktywować system Windows."

The status bar at the bottom indicates: Permissions: RW End-of-line: CRLF Encoding: UTF-8 Line: 49 Column: 1 Memory: 77%

# Przykłady.



The image shows a screenshot of the Spyder Python IDE interface. The main editor window displays a Python script with the following code:

```
55
56
57
58 x = y = np.linspace(-10,10,100)
59 X,Y = np.meshgrid(x,y)
60 Z = X**2 - Y**2
61 plt.figure(17)
62 fig = plt.figure()
63 sns.set(rc={"axes.facecolor":"white", "figure.facecolor":"white"})
64 ax = m3d.Axes3D(fig,azim = 60,elev = 30)
65 ax.plot_surface(X,Y,Z, linewidth = 0, edgecolor = "none")
66 sns.set()
67 plt.show()
68
69
70
71 |
72
73
```

The interface includes a menu bar (File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help), a toolbar with various icons, and a status bar at the bottom showing "Permissions: RW", "End-of-line: CRLF", "Encoding: UTF-8", "Line: 71", "Column: 1", and "Memory: 76 %". The system tray at the bottom right shows the date and time as "17:21 2018.06.12".

# Źródła.



<http://www.aosabook.org/en/matplotlib.html>



<https://matplotlib.org/>