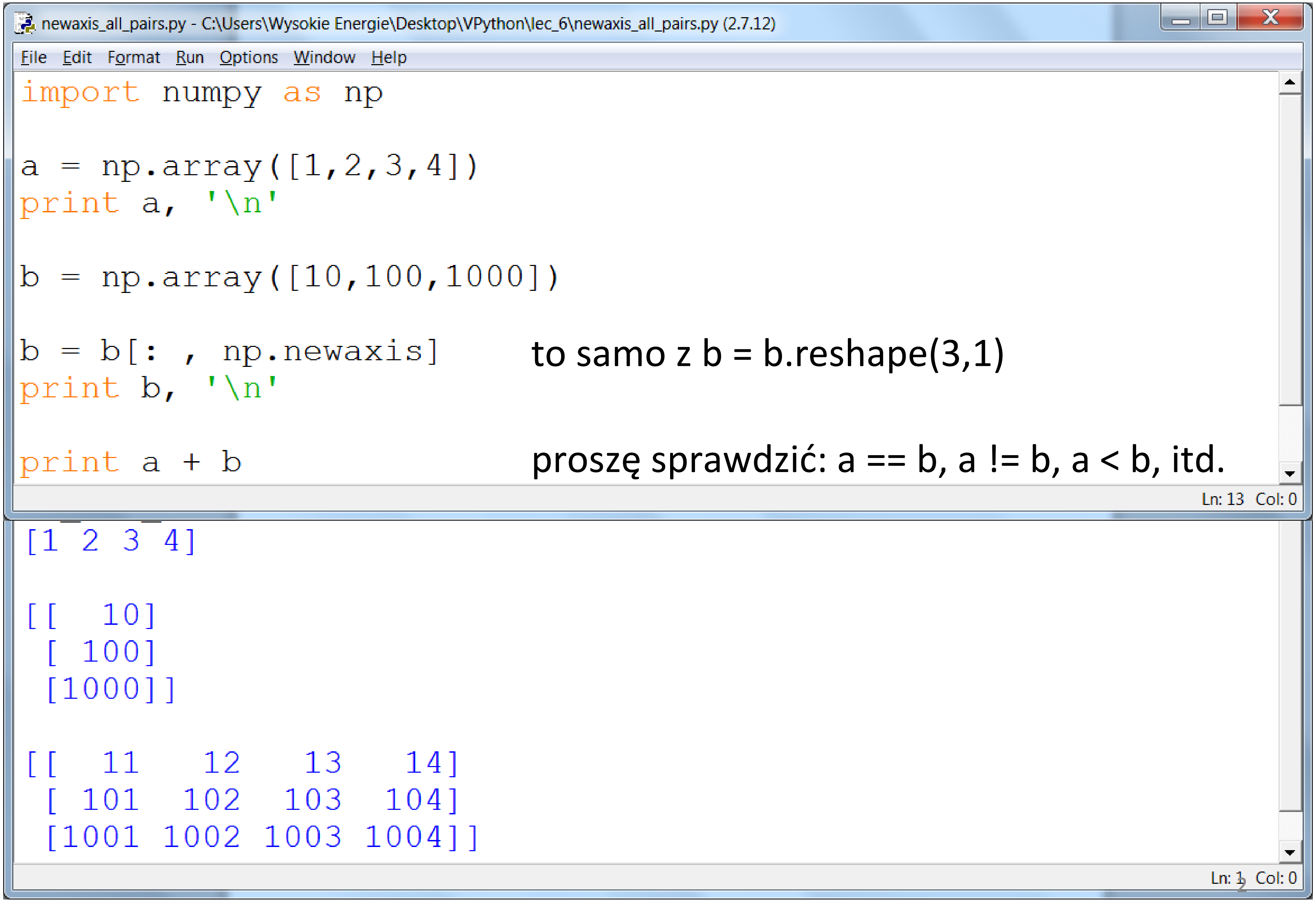
VPython - symulacje fizyczne z grafiką 3D dla każdego

wykład 6

Dr hab. Adam Bzdak

newaxis, wszystkie pary


newaxis_all_pairs.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\newaxis_all_pairs.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 13 Col: 0
Ln: 1 Col: 0

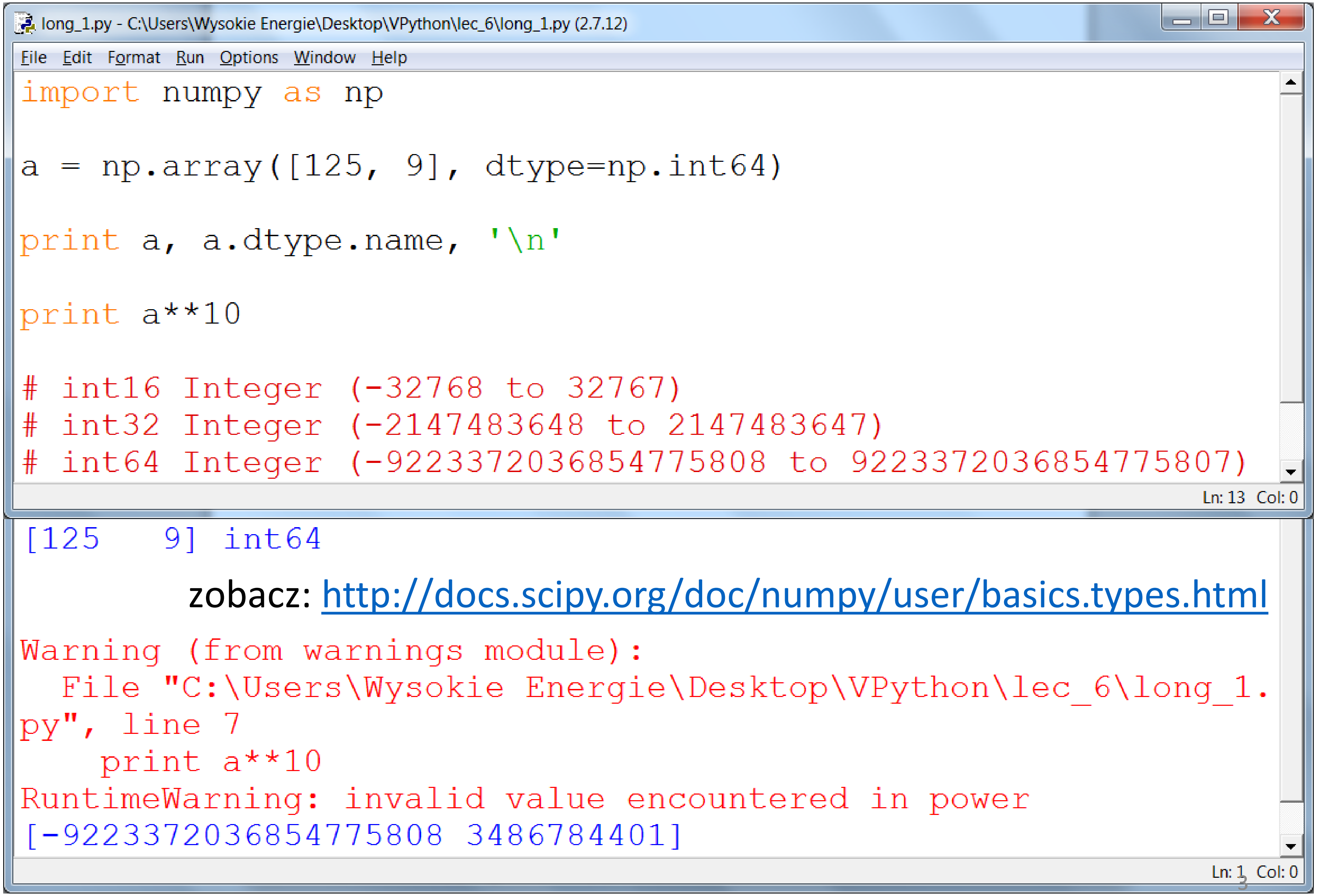
```
import numpy as np

a = np.array([1,2,3,4])
print a, '\n'

b = np.array([10,100,1000])

b = b[: , np.newaxis]
print b, '\n'

print a + b
```

to samo z b = b.reshape(3,1)

proszę sprawdzić: a == b, a != b, a < b, itd.

```
[1 2 3 4]

[[  10]
 [ 100]
 [1000]]

[[  11   12   13   14]
 [ 101  102  103  104]
 [1001 1002 1003 1004]]
```

Liczby całkowite w NumPy

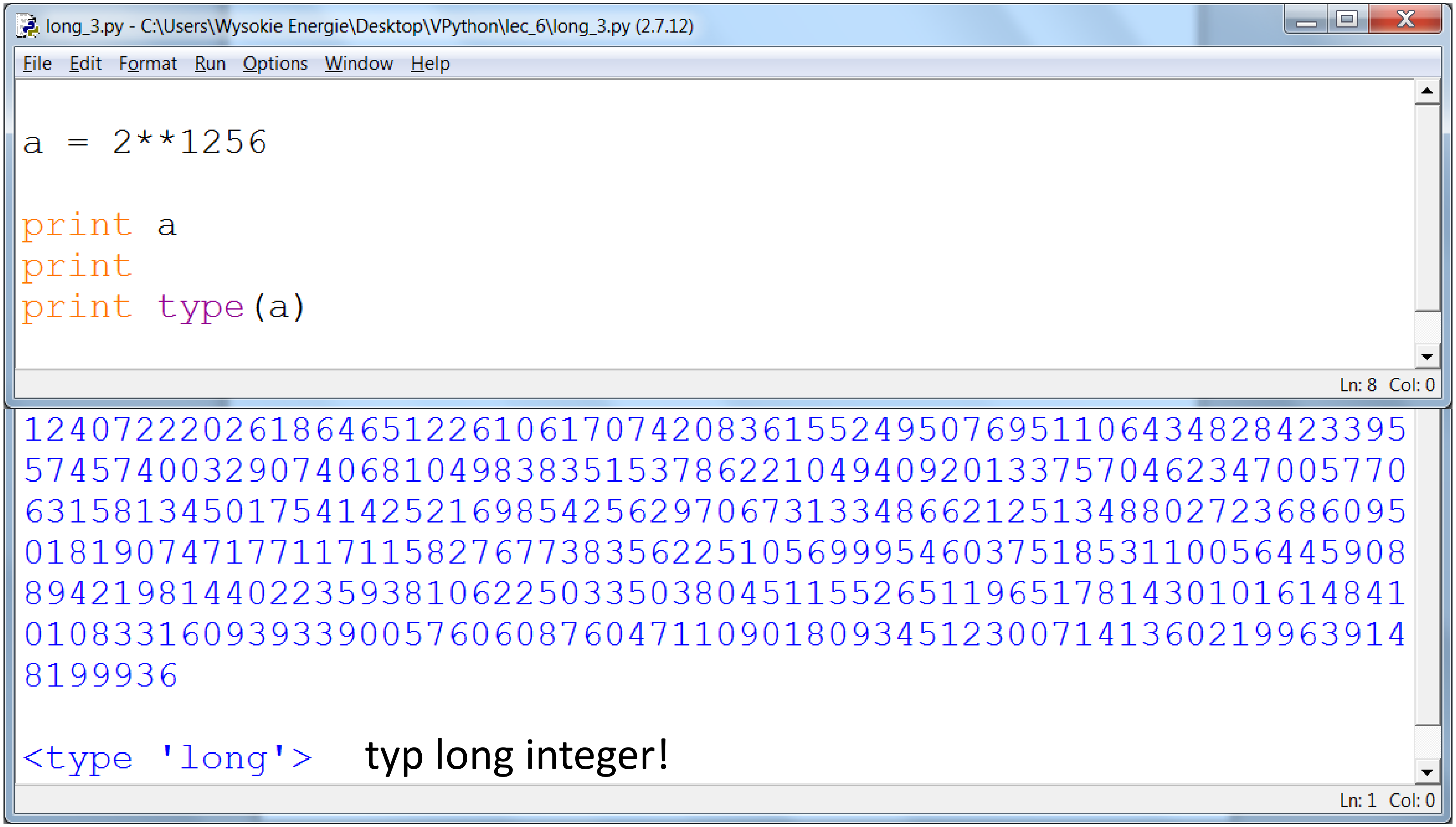

long_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\long_1.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 13 Col: 0
Ln: 1 Col: 0

```
import numpy as np

a = np.array([125, 9], dtype=np.int64)

print a, a.dtype.name, '\n'

print a**10

# int16 Integer (-32768 to 32767)
# int32 Integer (-2147483648 to 2147483647)
# int64 Integer (-9223372036854775808 to 9223372036854775807)
```

```
[125    9] int64

Warning (from warnings module):
  File "C:\Users\Wysokie Energie\Desktop\VPython\lec_6\long_1.py", line 7
    print a**10
RuntimeWarning: invalid value encountered in power
[-9223372036854775808 3486784401]
```

zobacz: http://docs.scipy.org/doc/numpy/user/basics.types.html

Python, long


long_3.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\long_3.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 8 Col: 0
Ln: 1 Col: 0

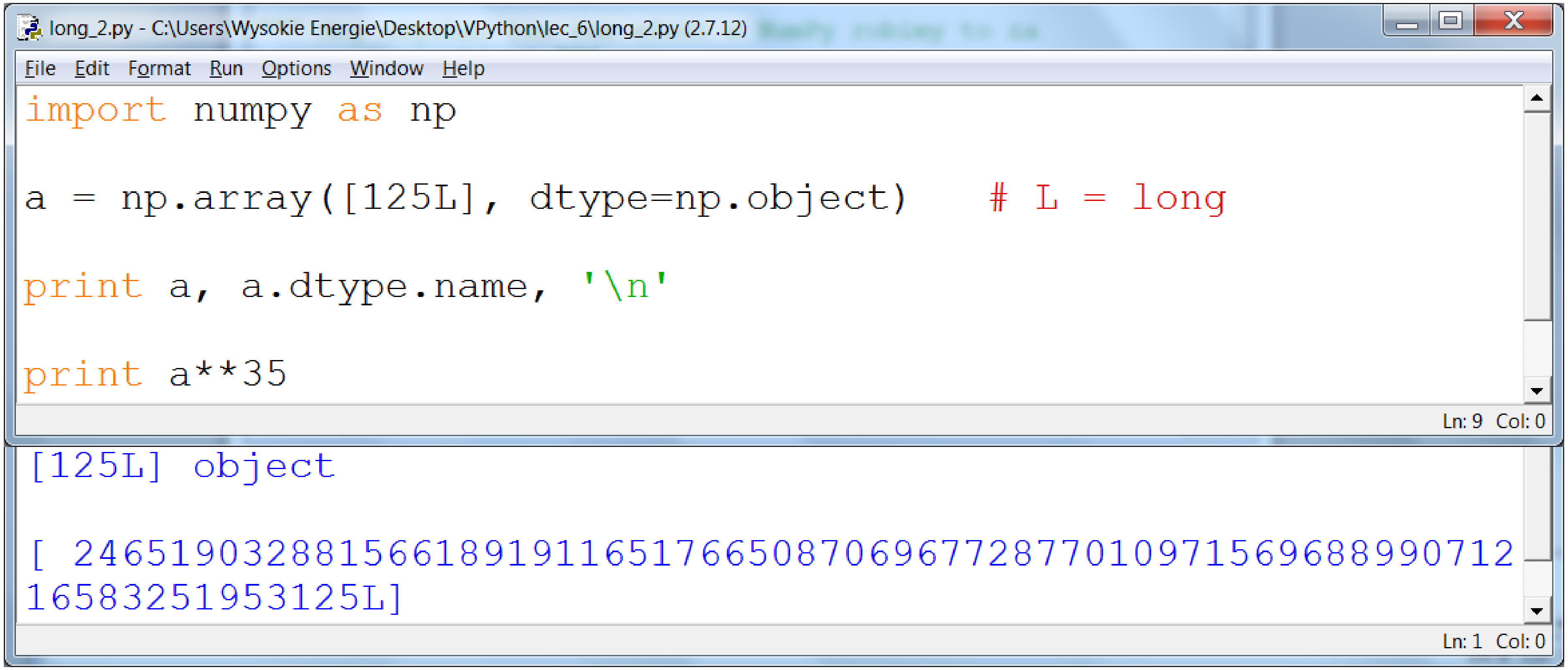
```python
a = 2**1256

print a
print
print type(a)
```

```
12407222026186465122610617074208361552495076951106434828423395
```

```
12407222026186465122610617074208361552495076951106434828423395
57457400329074068104983835153786221049409201337570462347005770
63158134501754142521698542562970673133486621251348802723686095
018190747177111711582767738356225105699954603751853110056445908
89421981440223593810622503350380451155265119651781430101614841
01083316093933900576060876047110901809345123007141360219963914
8199936

<type 'long'>
```

typ long integer!

Long integer w NumPy


long_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\long_2.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 9 Col: 0
Ln: 1 Col: 0

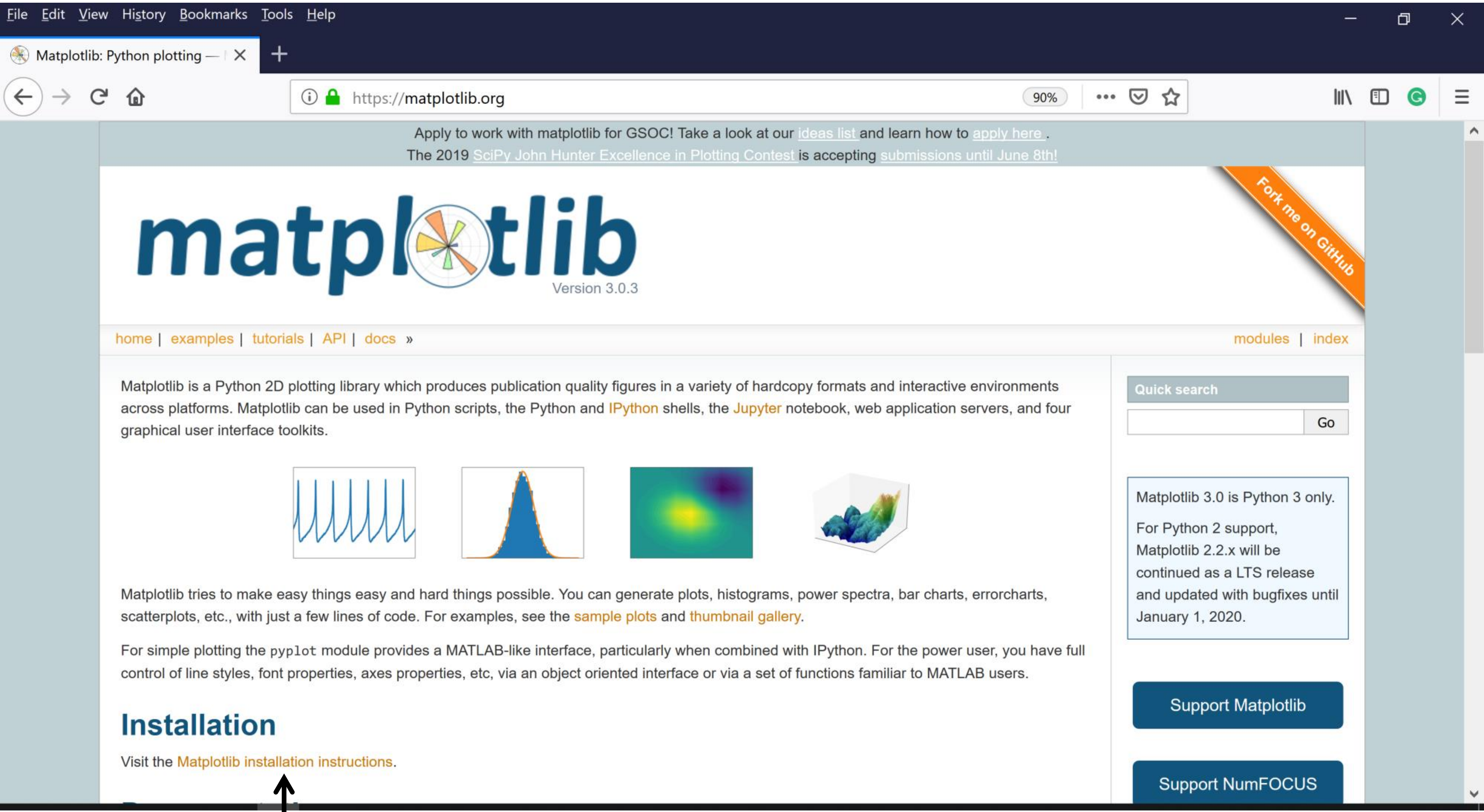
```
import numpy as np

a = np.array([125L], dtype=np.object)    # L = long

print a, a.dtype.name, '\n'

print a**35
```

```
[125L] object

[ 246519032881566189191165176650870696772877010971569688990712
1658325195312 5L]
```

Matplotlib

Matplotlib
http://matplotlib.org/

Matplotlib 3.0 jest dla Python 3


File Edit View History Bookmarks Tools Help
Matplotlib: Python plotting —
https://matplotlib.org
90%
Fork me on GitHub
home | examples | tutorials | API | docs »
modules | index
Quick search
Go
Support Matplotlib
Support NumFOCUS

Apply to work with matplotlib for GSOC! Take a look at our ideas list and learn how to apply here .
The 2019 SciPy John Hunter Excellence in Plotting Contest is accepting submissions until June 8th!

matplotlib
Version 3.0.3

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the `pyplot` module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Matplotlib 3.0 is Python 3 only.

For Python 2 support, Matplotlib 2.2.x will be continued as a LTS release and updated with bugfixes until January 1, 2020.

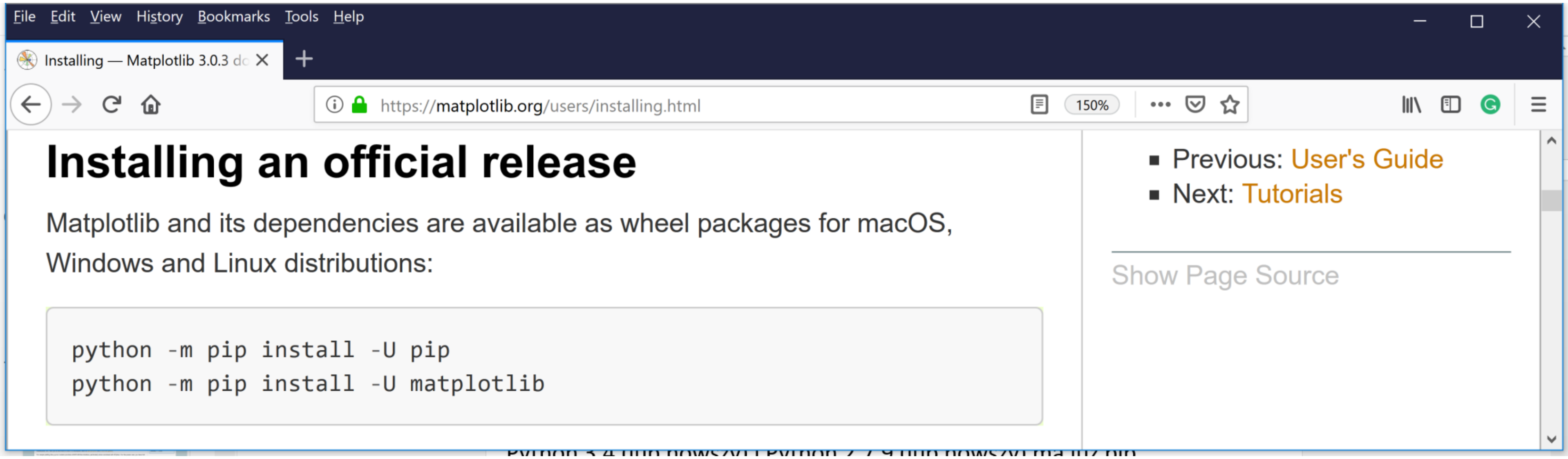
Installation

Visit the Matplotlib installation instructions.

↑

i klikamy

Instalacja


File Edit View History Bookmarks Tools Help
Installing — Matplotlib 3.0.3 dc
https://matplotlib.org/users/installing.html
150%

Installing an official release

Matplotlib and its dependencies are available as wheel packages for macOS, Windows and Linux distributions:

```
python -m pip install -U pip
python -m pip install -U matplotlib
```

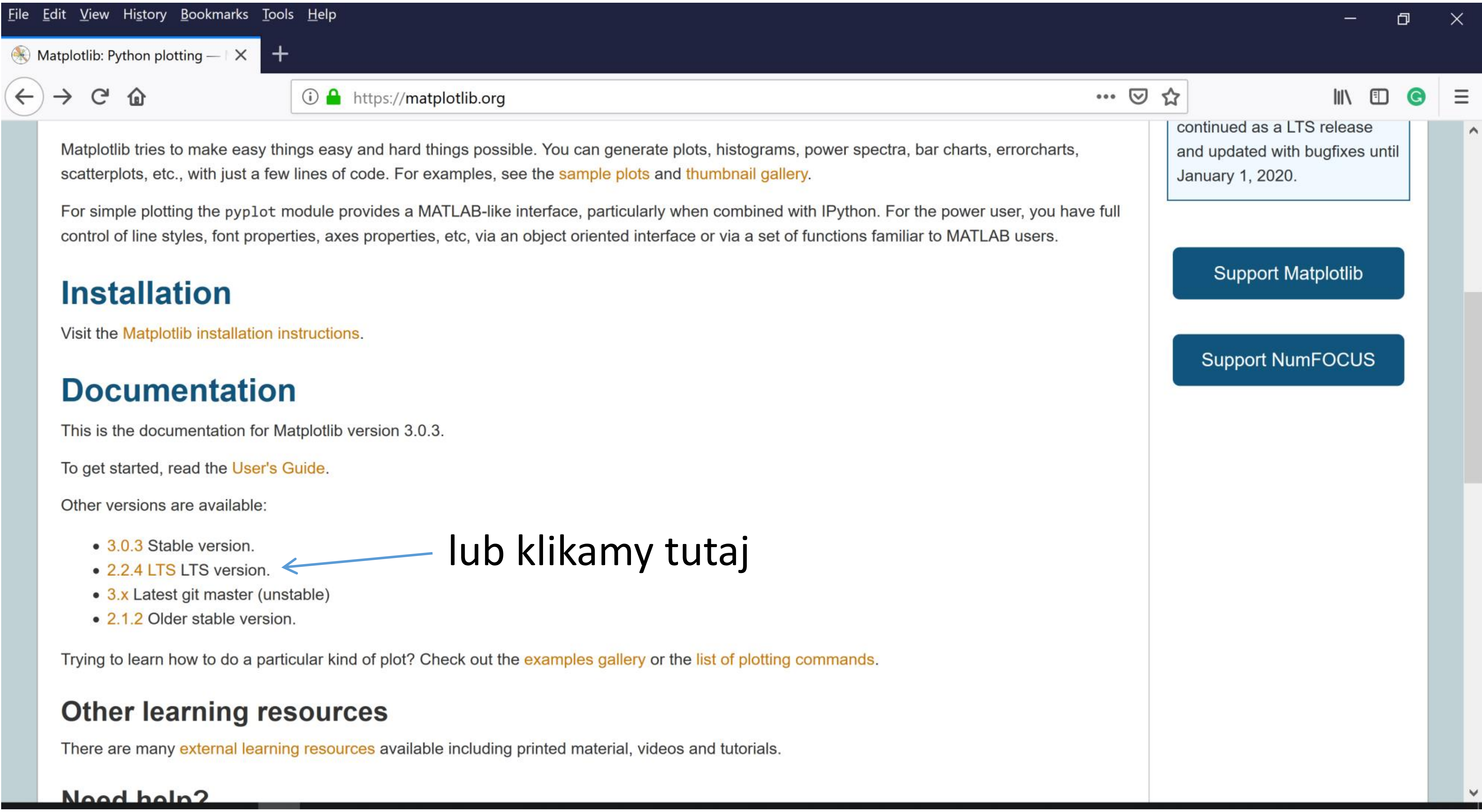
- Previous: User's Guide
- Next: Tutorials

Show Page Source

Python 3.4 (lub nowszy) i Python 2.7.9 (lub nowszy) ma już pip.

Dla Python 2 ≤ 2.7.8 i Python 3 ≤ 3.3 należy zainstalować pip
http://stackoverflow.com/questions/4750806/how-do-i-install-pip-on-windows

Dla Python 2
https://matplotlib.org/2.2.4/index.html



continued as a LTS release and updated with bugfixes until January 1, 2020.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the `pyplot` module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Installation

Visit the Matplotlib installation instructions.

Documentation

This is the documentation for Matplotlib version 3.0.3.

To get started, read the User's Guide.

Other versions are available:

- 3.0.3 Stable version.
- 2.2.4 LTS LTS version.
- 3.x Latest git master (unstable)
- 2.1.2 Older stable version.

Trying to learn how to do a particular kind of plot? Check out the examples gallery or the list of plotting commands.

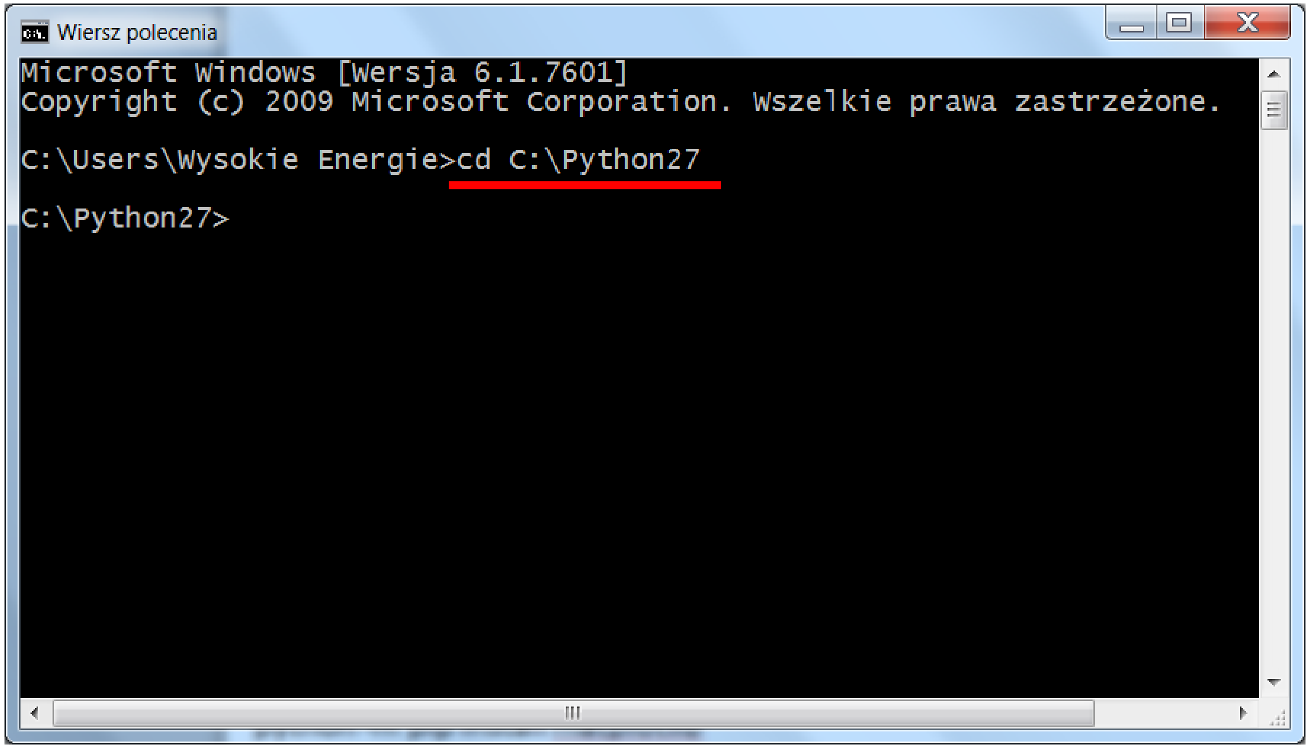
Other learning resources

There are many external learning resources available including printed material, videos and tutorials.

piszemy

cd C:\Python27

jeśli tam jest zainstalowany Python. Dla innej wersji Pythona piszemy np. Python34 lub coś innego (w zależności od wersji)



```
Microsoft Windows [Wersja 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Wysokie Energie>cd C:\Python27

C:\Python27>
```

następnie piszemy

python –m pip install -U pip

i następnie

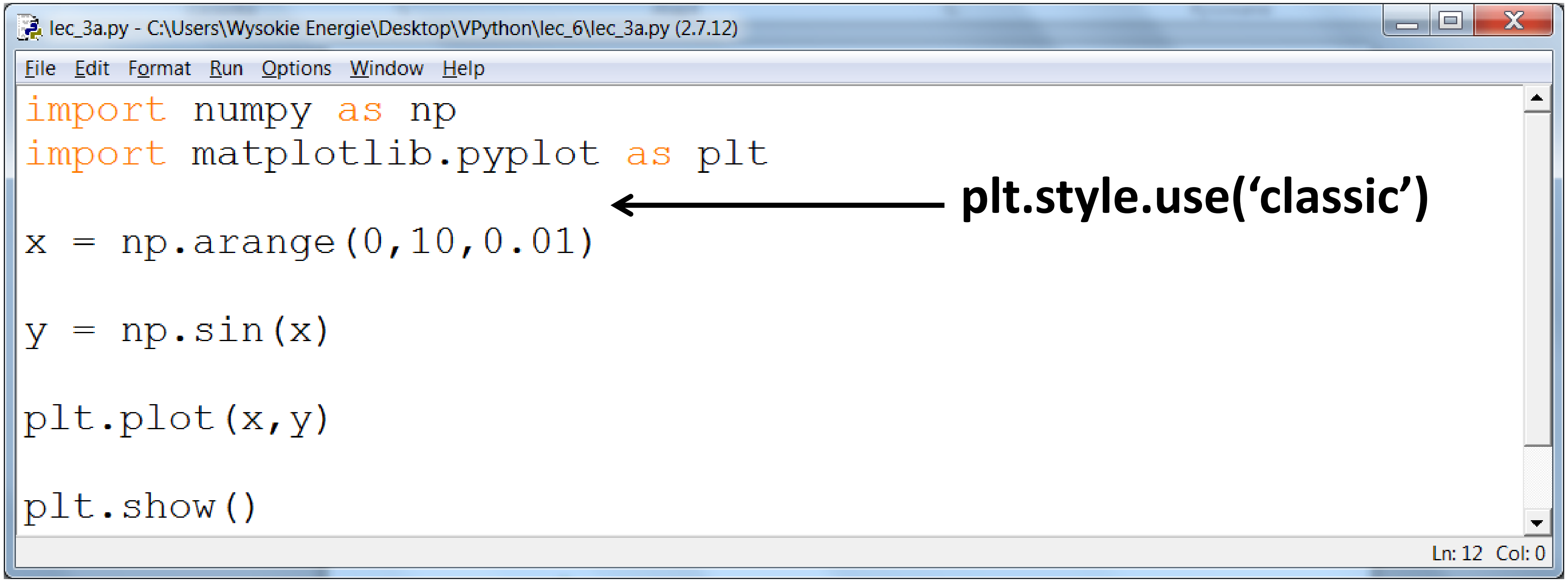
python –m pip install -U matplotlib

Matplotlib ma teraz nowy styl

Moje przykłady są w *classic view*

plt.style.use(‘classic’)

Pierwszy wykres

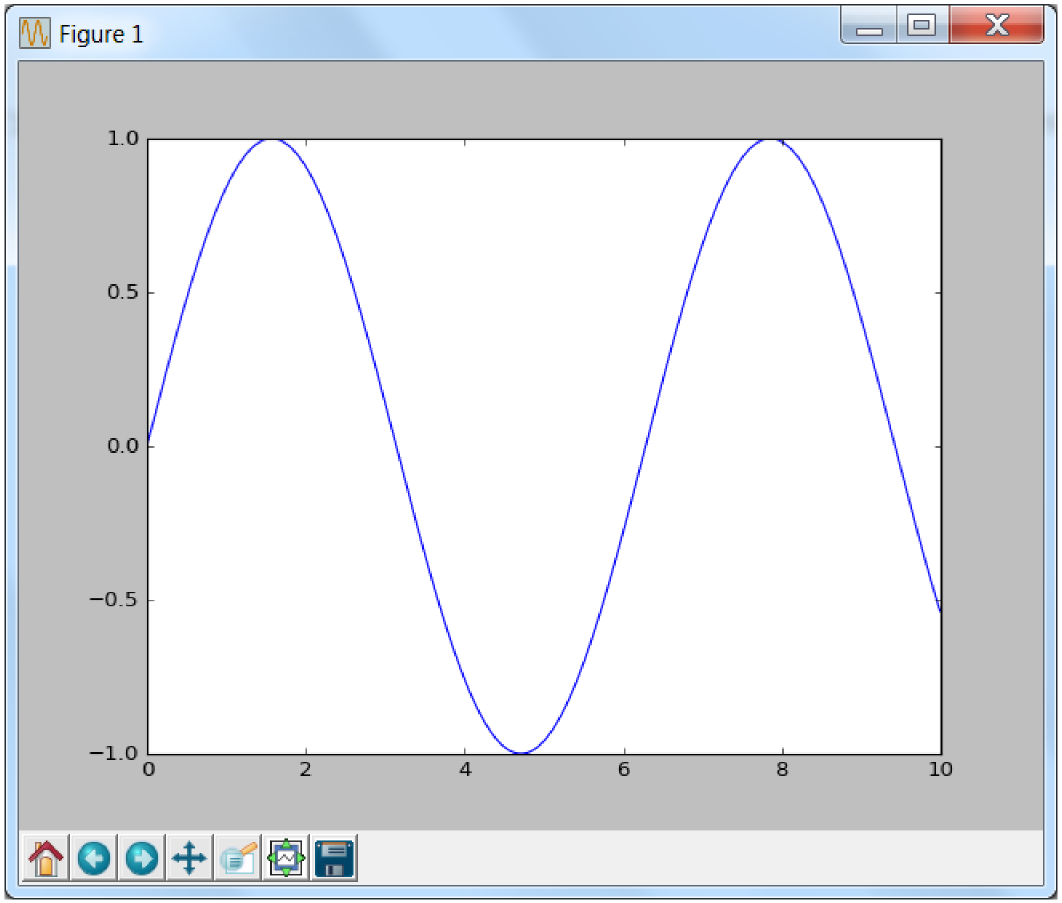

lec_3a.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\lec_3a.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 12 Col: 0

```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0,10,0.01)

y = np.sin(x)

plt.plot(x,y)

plt.show()
```

plt.style.use('classic')

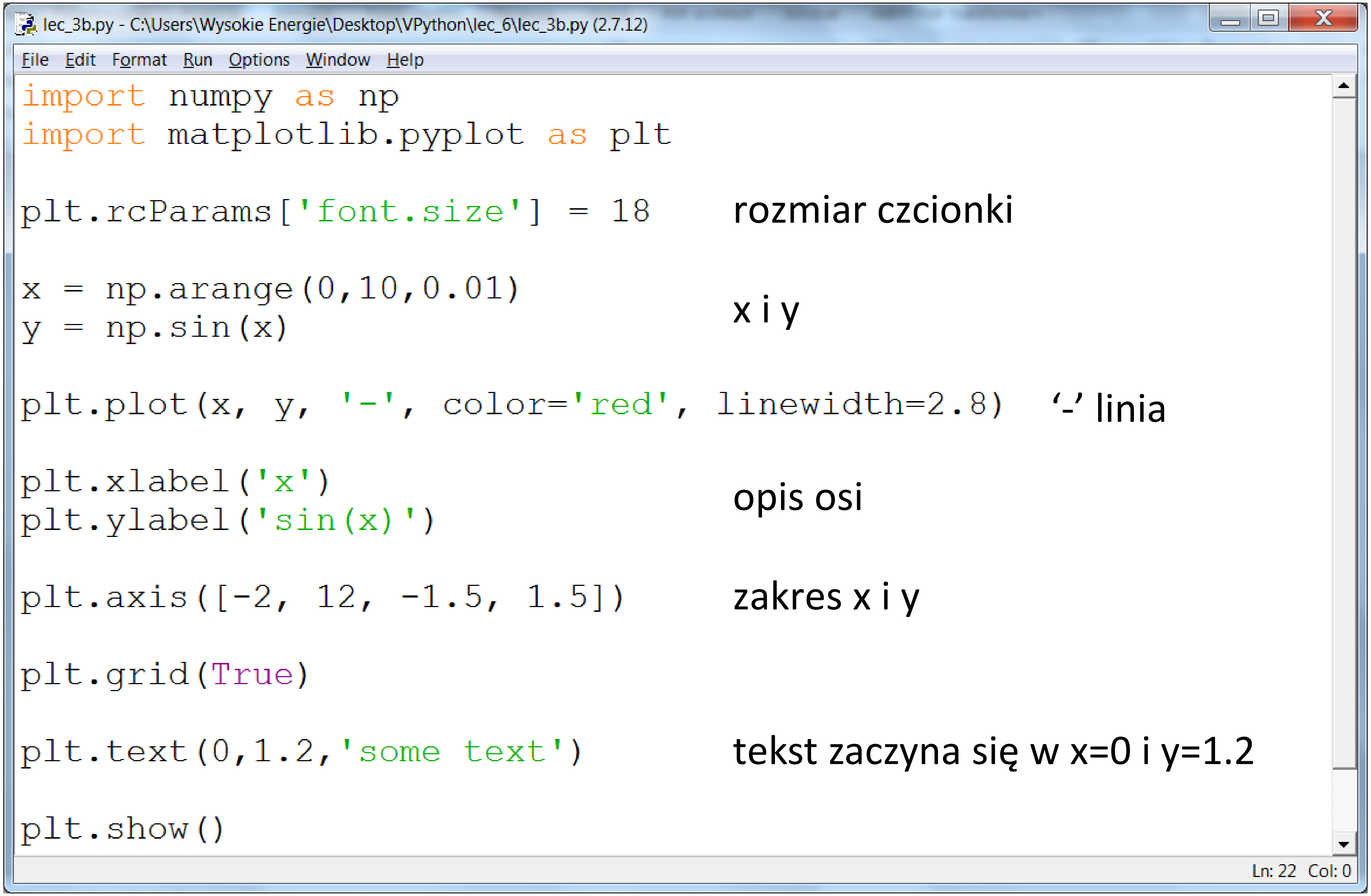
x = [0, 0.01, 0.02, ..., 9.99], to jest array

y = np.sin(x) to też jest array



można zapisać jako .pdf, .ps, .eps, etc.
można powiększyć, przesunąć itp.

Lepszy wykres


lec_3b.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\lec_3b.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 22 Col: 0

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 18

x = np.arange(0,10,0.01)
y = np.sin(x)

plt.plot(x, y, '-', color='red', linewidth=2.8)

plt.xlabel('x')
plt.ylabel('sin(x)')

plt.axis([-2, 12, -1.5, 1.5])

plt.grid(True)

plt.text(0,1.2,'some text')

plt.show()
```

rozmiar czcionki

x i y

'-' linia

opis osi

zakres x i y

tekst zaczyna się w x=0 i y=1.2

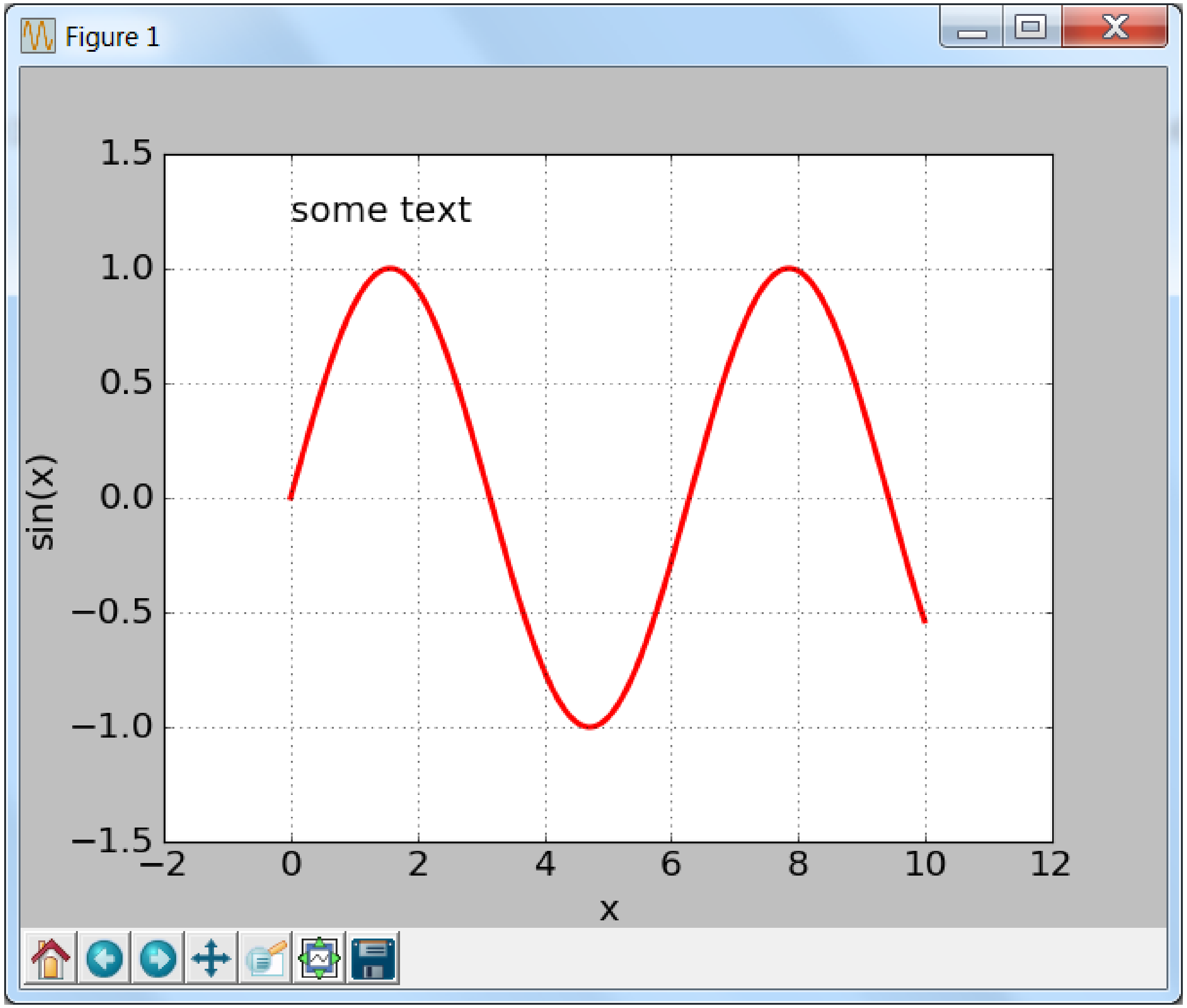
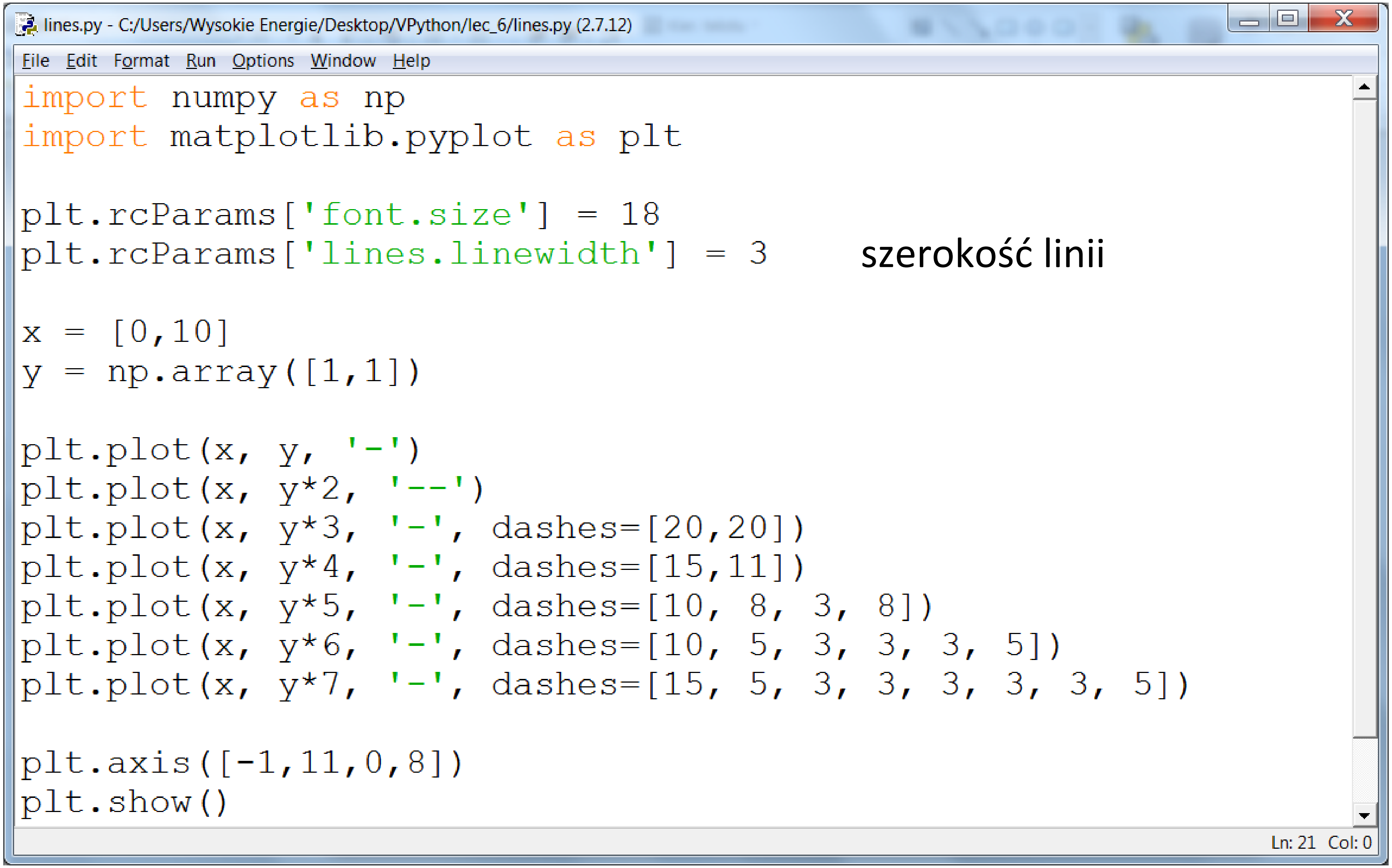


Figure 1
some text
sin(x)
x
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2
0
2
4
6
8
10
12

linie


lines.py - C:/Users/Wysokie Energie/Desktop/VPython/lec_6/lines.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 21 Col: 0

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 18
plt.rcParams['lines.linewidth'] = 3

x = [0,10]
y = np.array([1,1])

plt.plot(x, y, '-')
plt.plot(x, y*2, '--')
plt.plot(x, y*3, '-', dashes=[20,20])
plt.plot(x, y*4, '-', dashes=[15,11])
plt.plot(x, y*5, '-', dashes=[10, 8, 3, 8])
plt.plot(x, y*6, '-', dashes=[10, 5, 3, 3, 3, 5])
plt.plot(x, y*7, '-', dashes=[15, 5, 3, 3, 3, 3, 3, 5])

plt.axis([-1,11,0,8])
plt.show()
```

szerokość linii

dashes = [linia, przerwa, linia, przerwa, ...]

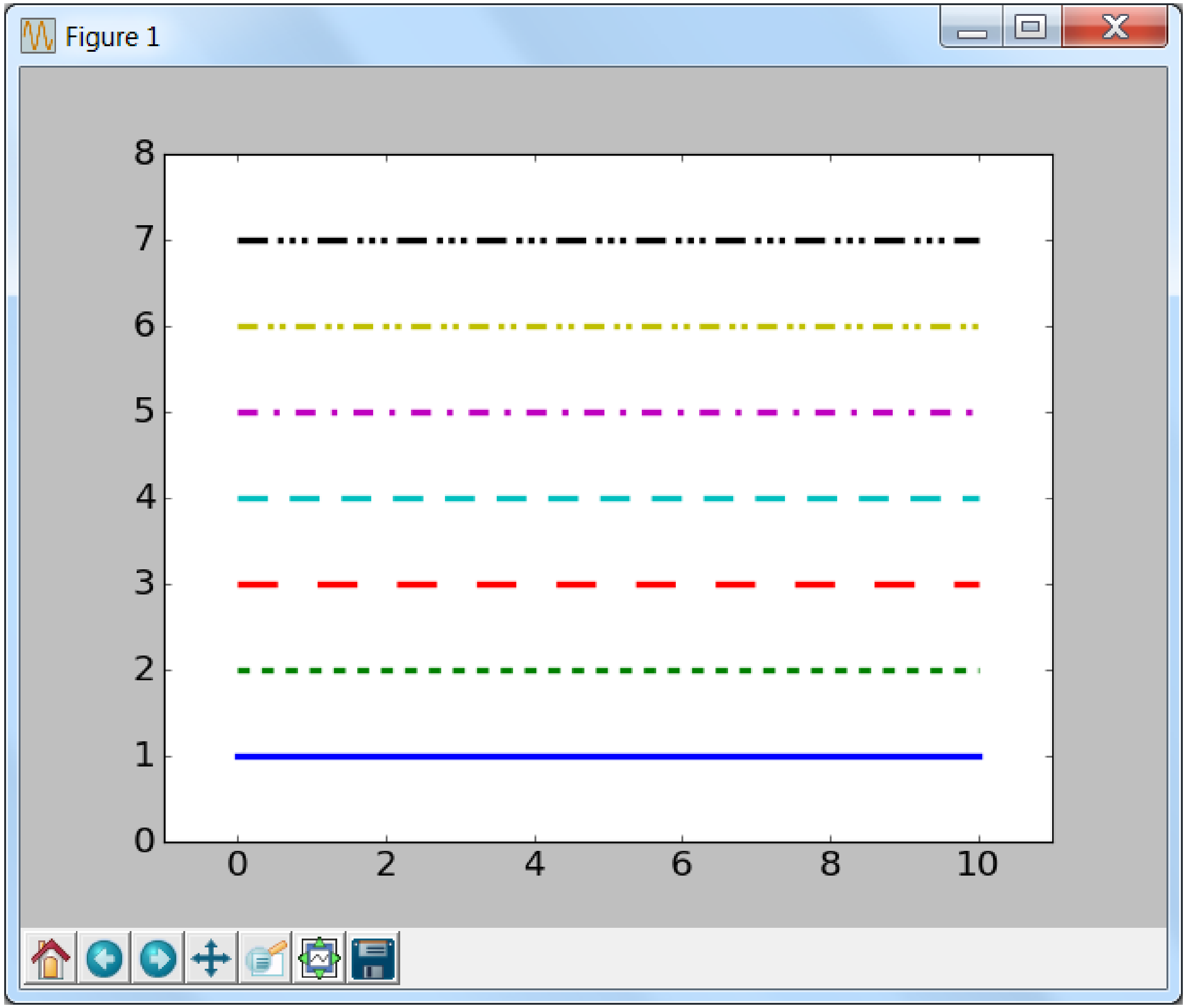
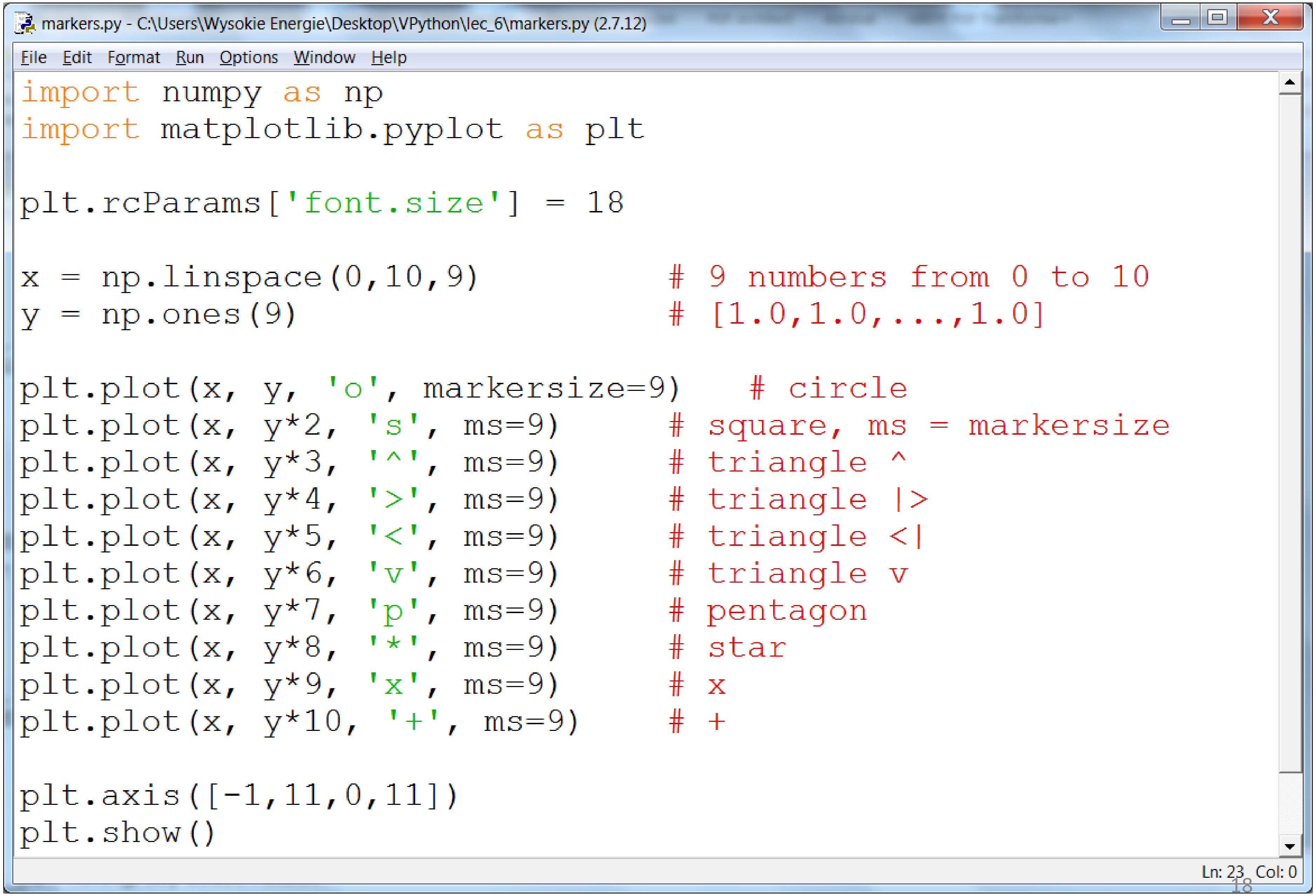


Figure 1
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10

Symbole (markers)


markers.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\markers.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 23 Col: 0

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 18

x = np.linspace(0,10,9)               # 9 numbers from 0 to 10
y = np.ones(9)                        # [1.0,1.0,...,1.0]

plt.plot(x, y, 'o', markersize=9)       # circle
plt.plot(x, y*2, 's', ms=9)           # square, ms = markersize
plt.plot(x, y*3, '^', ms=9)           # triangle ^
plt.plot(x, y*4, '>', ms=9)           # triangle |>
plt.plot(x, y*5, '<', ms=9)           # triangle <|
plt.plot(x, y*6, 'v', ms=9)           # triangle v
plt.plot(x, y*7, 'p', ms=9)           # pentagon
plt.plot(x, y*8, '*', ms=9)           # star
plt.plot(x, y*9, 'x', ms=9)           # x
plt.plot(x, y*10, '+', ms=9)          # +

plt.axis([-1,11,0,11])
plt.show()
```

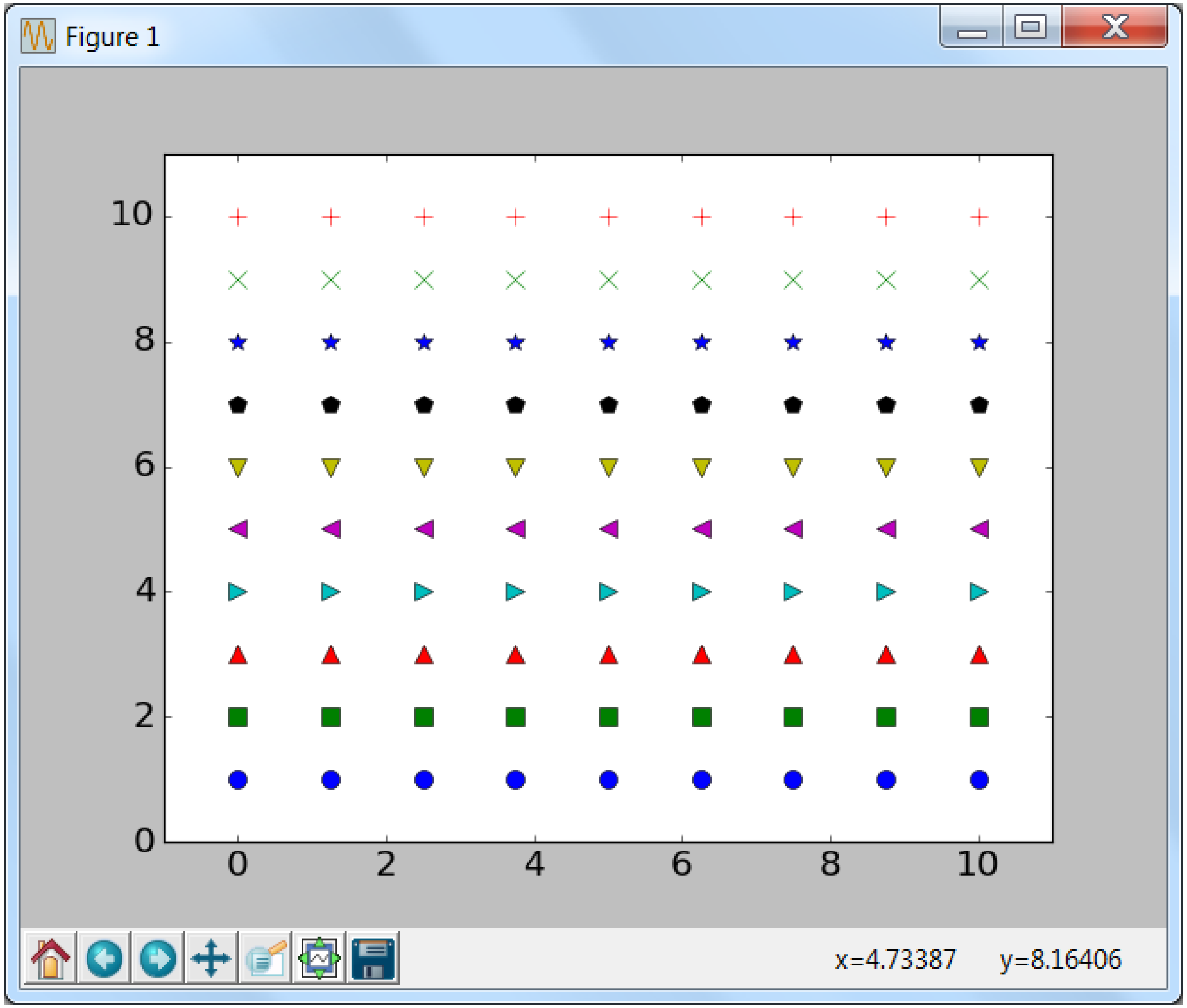
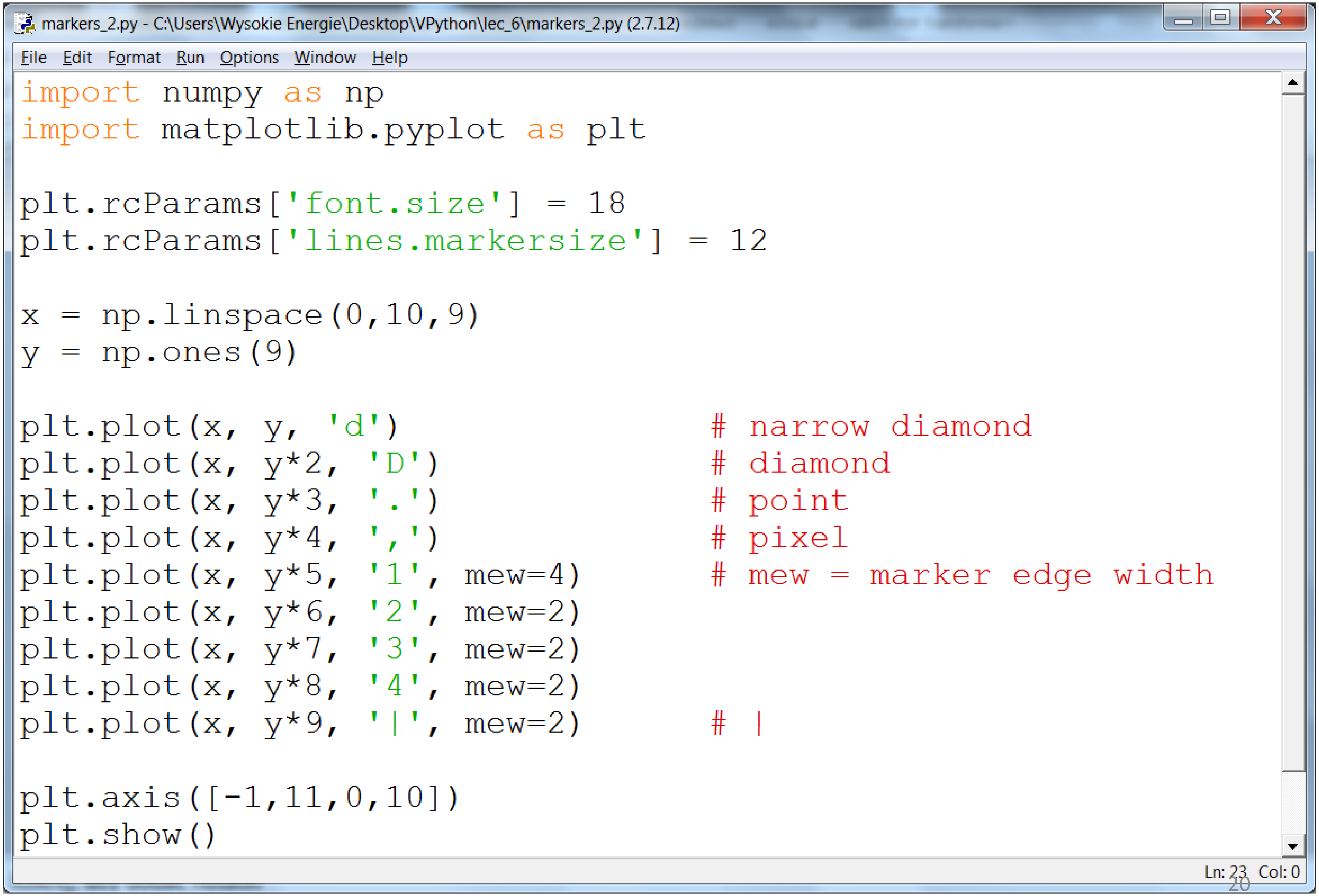
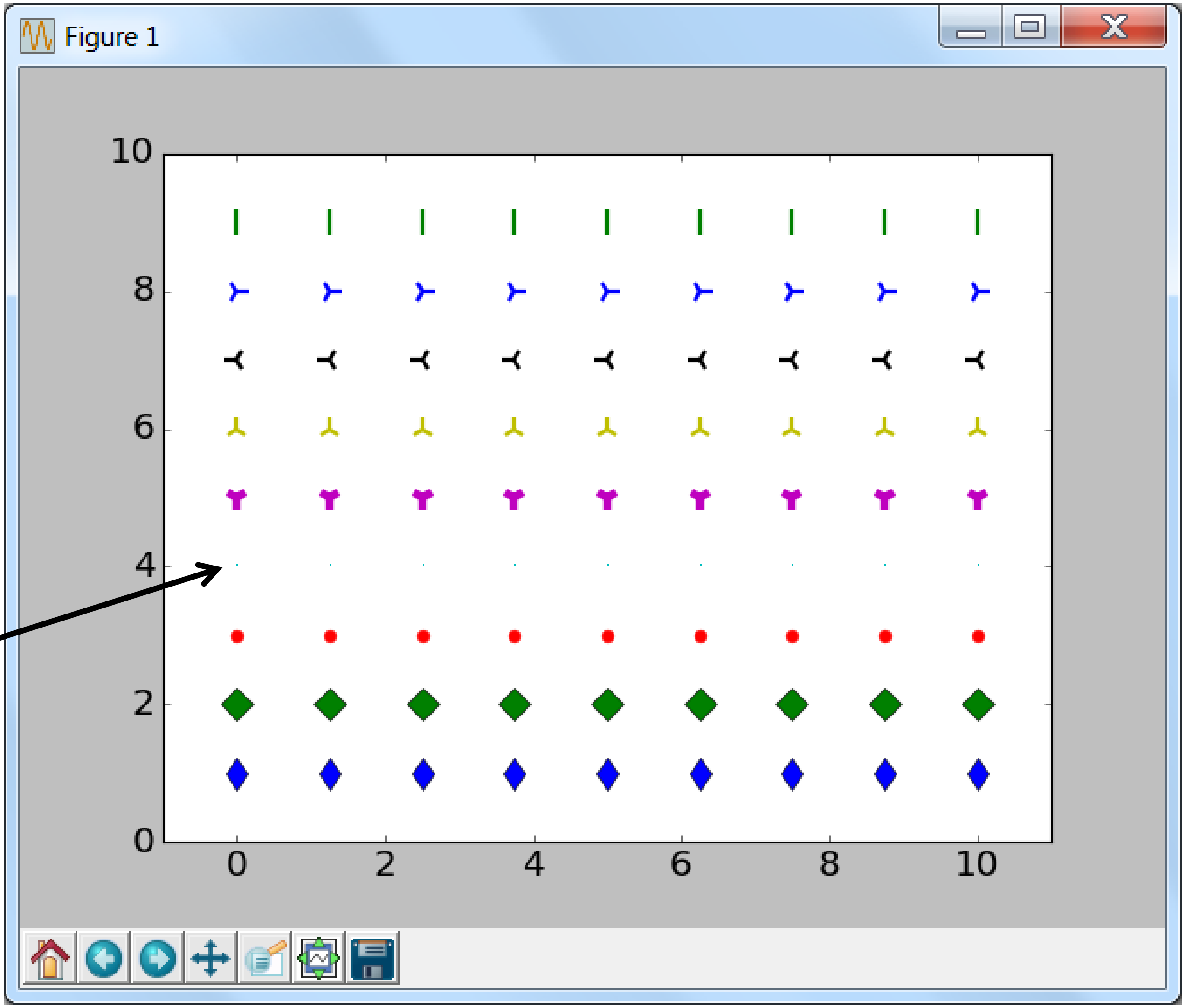


Figure 1
10
8
6
4
2
0
0
2
4
6
8
10
x=4.73387 y=8.16406

więcej

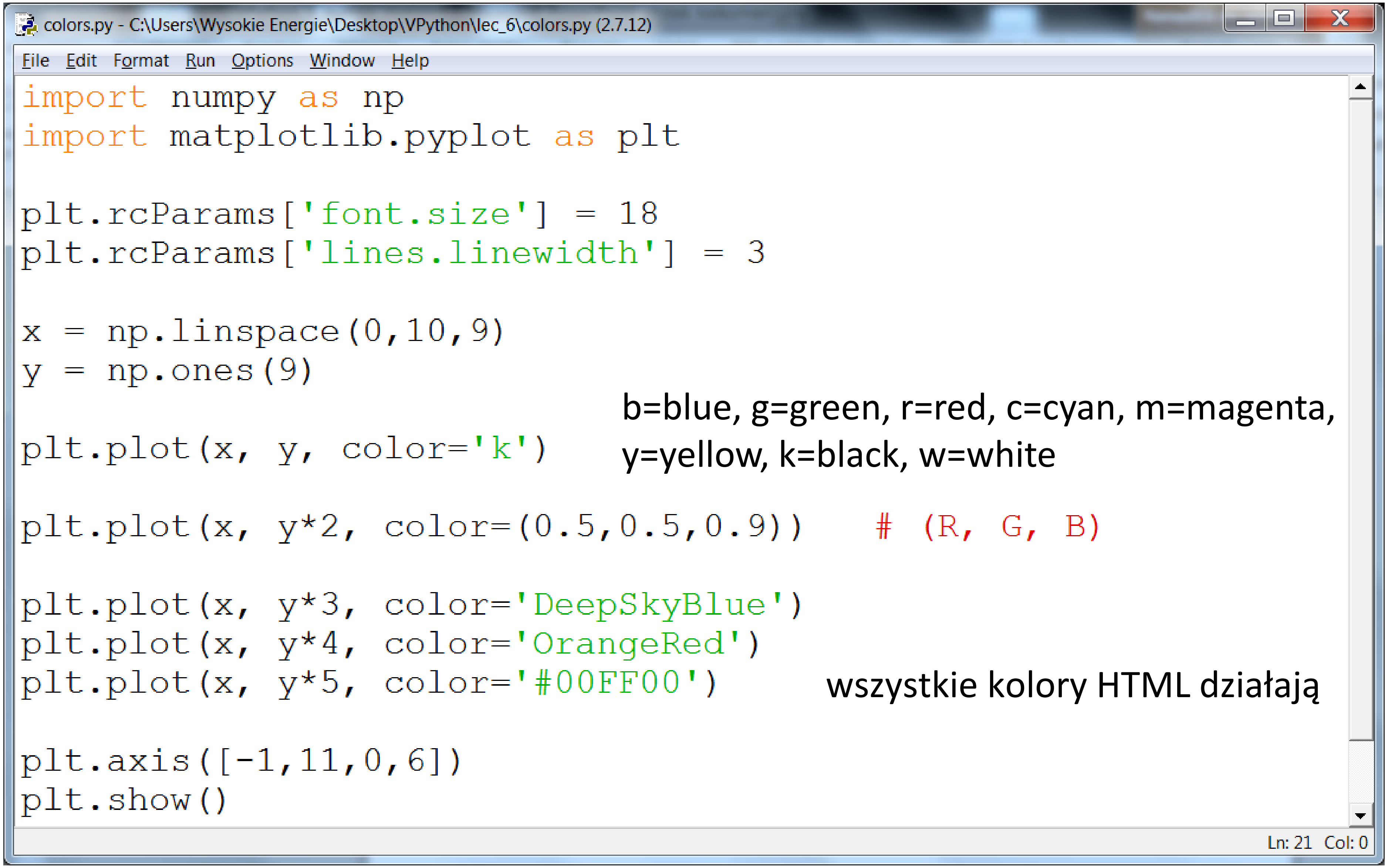

markers_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\markers_2.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 23 Col: 0

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 18
plt.rcParams['lines.markersize'] = 12

x = np.linspace(0,10,9)
y = np.ones(9)

plt.plot(x, y, 'd')                  # narrow diamond
plt.plot(x, y*2, 'D')                # diamond
plt.plot(x, y*3, '.')                # point
plt.plot(x, y*4, ',')                # pixel
plt.plot(x, y*5, '1', mew=4)         # mew = marker edge width
plt.plot(x, y*6, '2', mew=2)
plt.plot(x, y*7, '3', mew=2)
plt.plot(x, y*8, '4', mew=2)
plt.plot(x, y*9, '|', mew=2)         # |

plt.axis([-1,11,0,10])
plt.show()
```



piksele

colory


colors.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\colors.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 21 Col: 0

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 18
plt.rcParams['lines.linewidth'] = 3

x = np.linspace(0,10,9)
y = np.ones(9)

plt.plot(x, y, color='k')

plt.plot(x, y*2, color=(0.5,0.5,0.9))    # (R, G, B)

plt.plot(x, y*3, color='DeepSkyBlue')
plt.plot(x, y*4, color='OrangeRed')
plt.plot(x, y*5, color='#00FF00')

plt.axis([-1,11,0,6])
plt.show()
```

b=blue, g=green, r=red, c=cyan, m=magenta, y=yellow, k=black, w=white

wszystkie kolory HTML działają

http://www.w3schools.com/colors/colors_names.asp

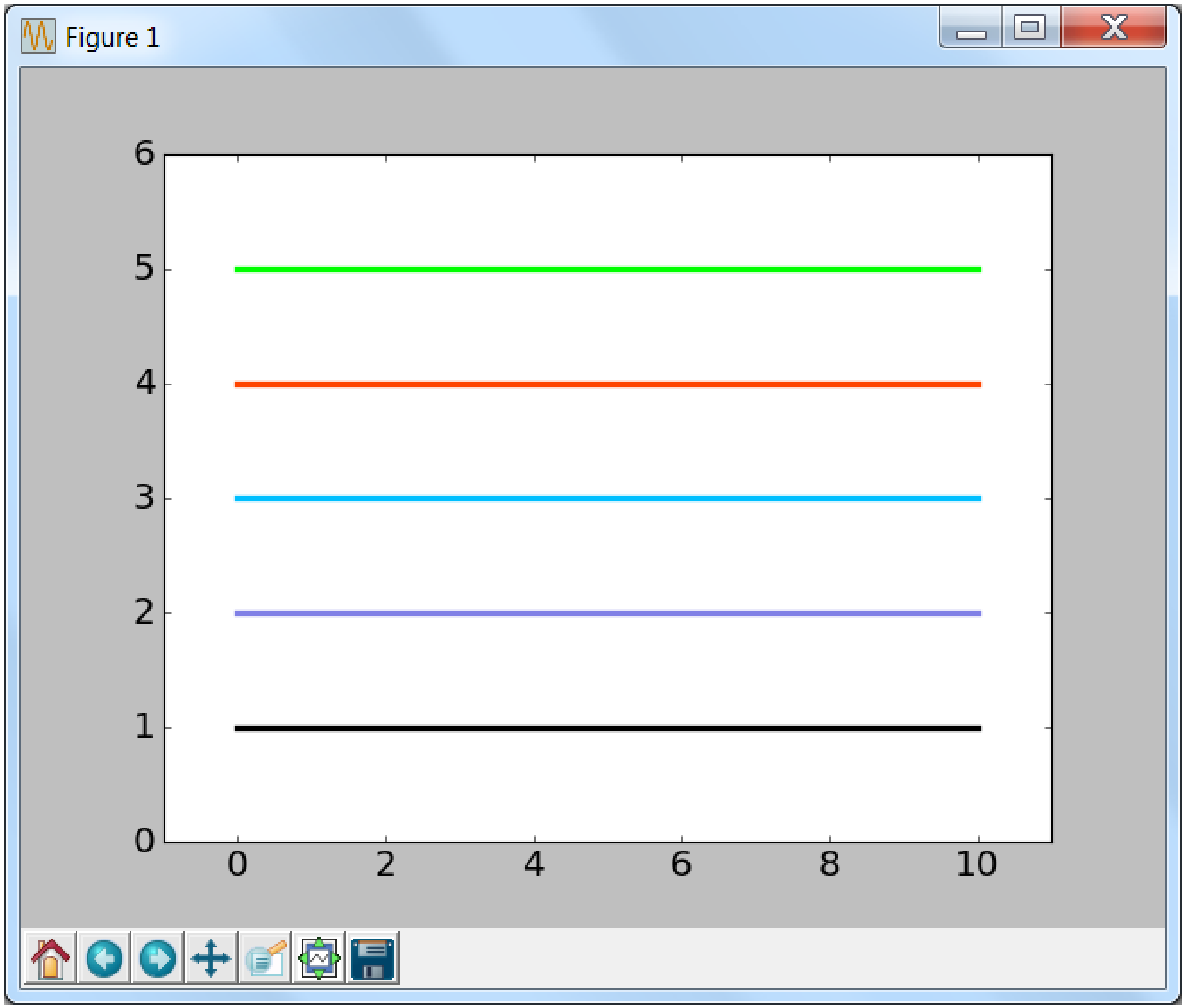
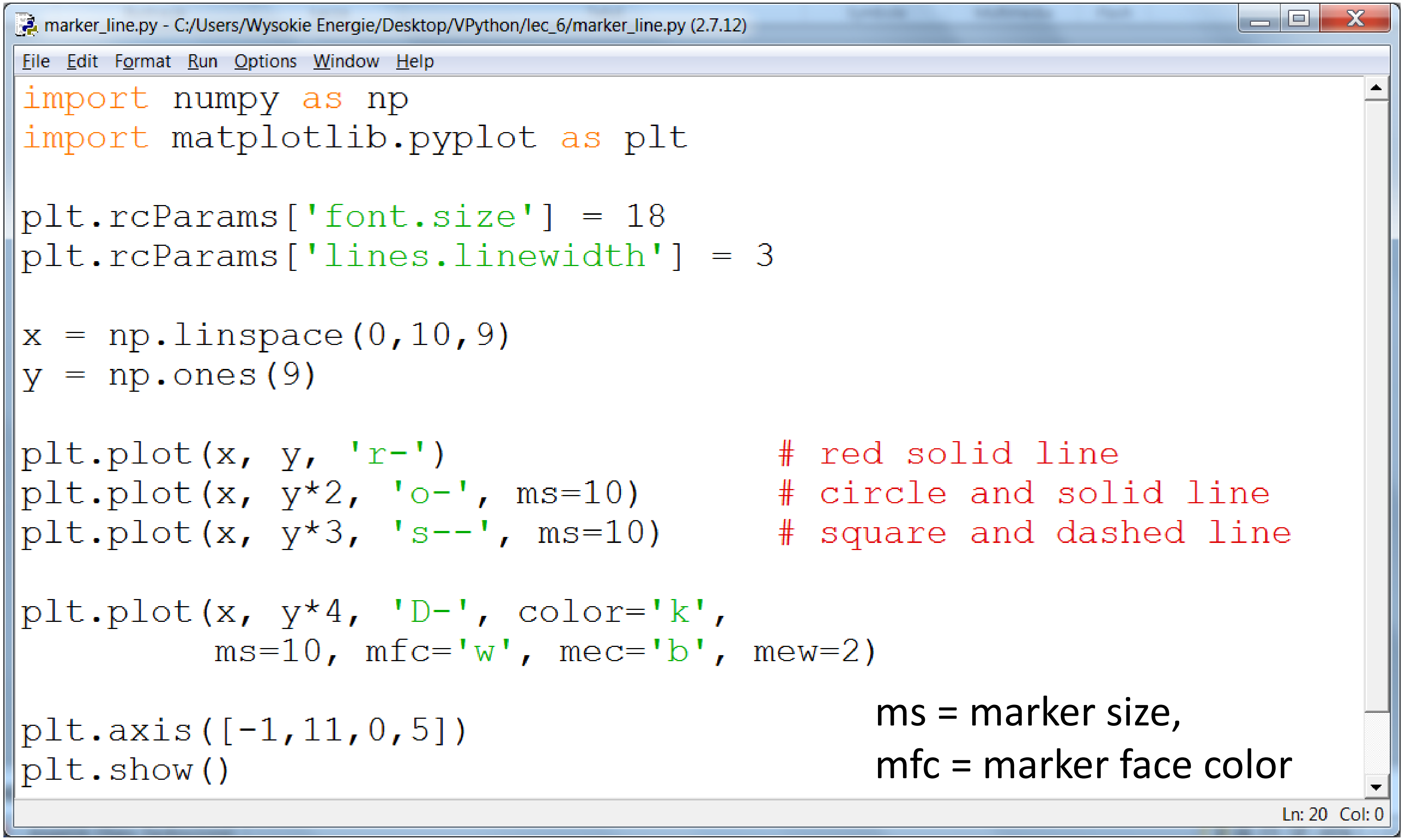


Figure 1
6
5
4
3
2
1
0
0
2
4
6
8
10

Symbole i linie


marker_line.py - C:/Users/Wysokie Energie/Desktop/VPython/lec_6/marker_line.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 20 Col: 0

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 18
plt.rcParams['lines.linewidth'] = 3

x = np.linspace(0,10,9)
y = np.ones(9)

plt.plot(x, y, 'r-')                # red solid line
plt.plot(x, y*2, 'o-', ms=10)       # circle and solid line
plt.plot(x, y*3, 's--', ms=10)      # square and dashed line

plt.plot(x, y*4, 'D-', color='k',
         ms=10, mfc='w', mec='b', mew=2)

plt.axis([-1,11,0,5])
plt.show()
```

ms = marker size,

mfc = marker face color

mec = marker edge color

mew = marker edge width

Proszę sprawdzić: mfc='None'

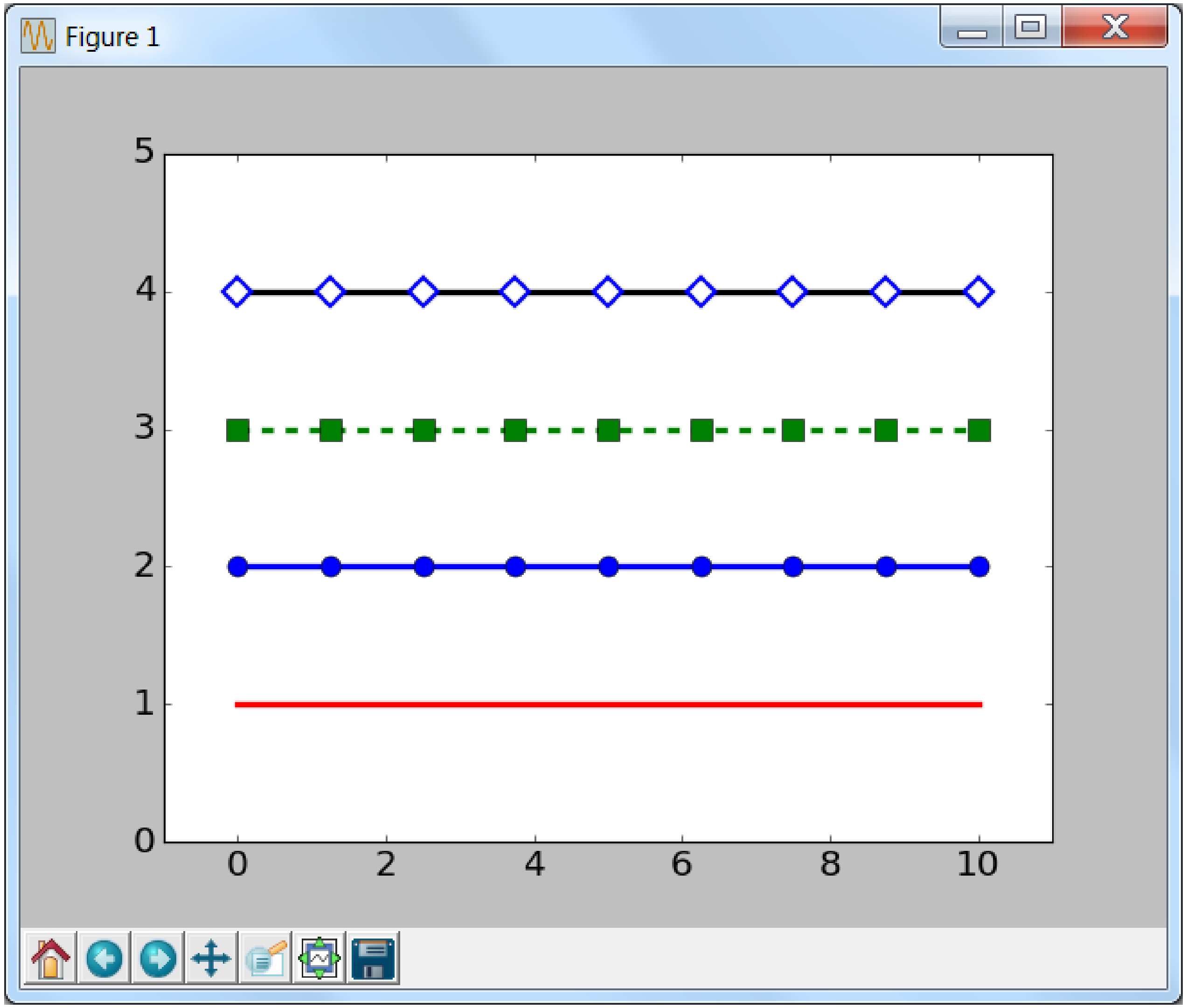
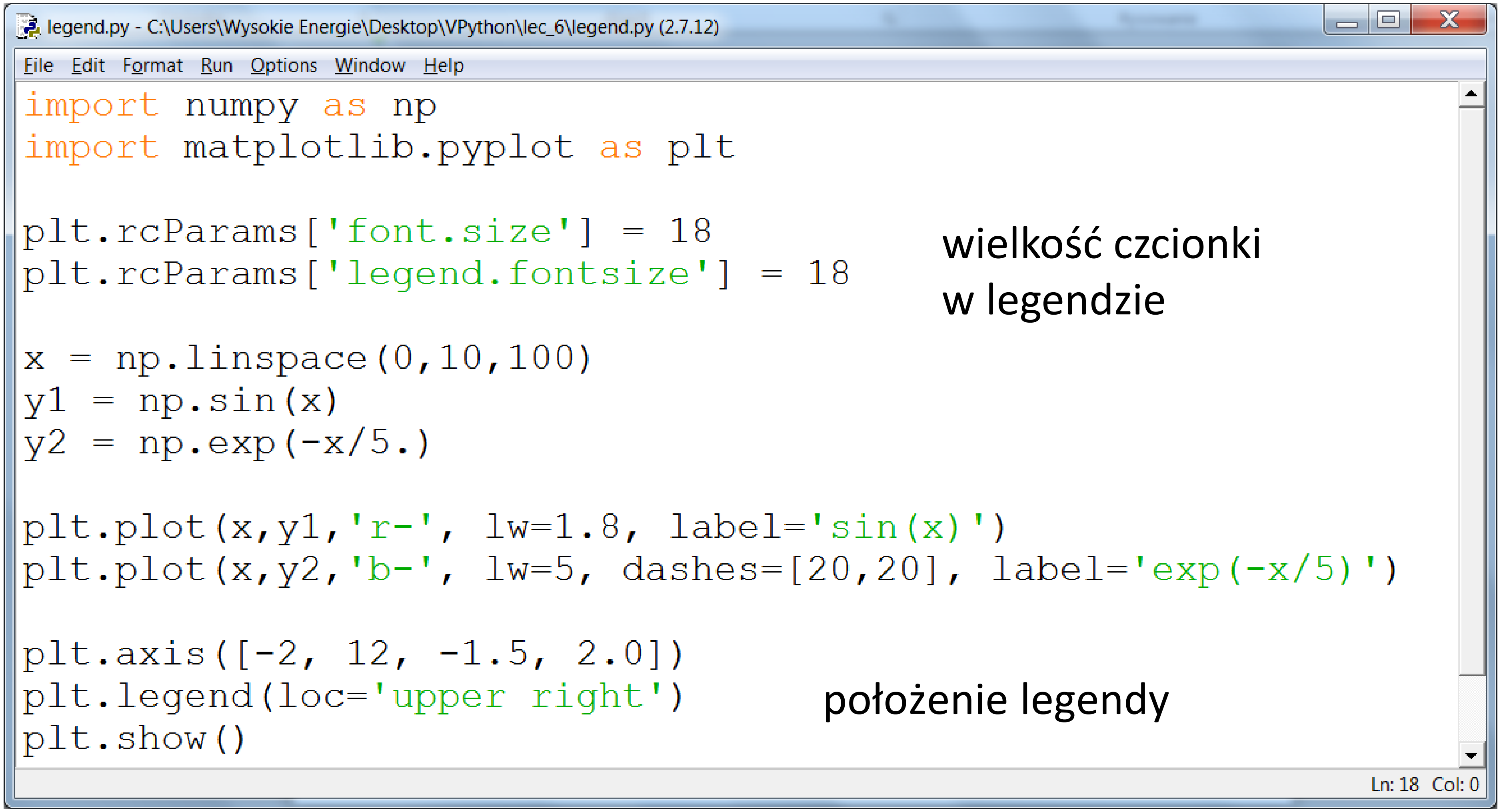


Figure 1
5
4
3
2
1
0
0
2
4
6
8
10

Legenda (legend)


legend.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\legend.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 18 Col: 0

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.size'] = 18
plt.rcParams['legend.fontsize'] = 18

x = np.linspace(0,10,100)
y1 = np.sin(x)
y2 = np.exp(-x/5.)

plt.plot(x,y1,'r-', lw=1.8, label='sin(x)')
plt.plot(x,y2,'b-', lw=5, dashes=[20,20], label='exp(-x/5)')

plt.axis([-2, 12, -1.5, 2.0])
plt.legend(loc='upper right')
plt.show()
```

wielkość czcionki w legendzie

położenie legendy

loc = best, upper right, upper left, lower right, lower left
center left, center right, lower center, upper center, center

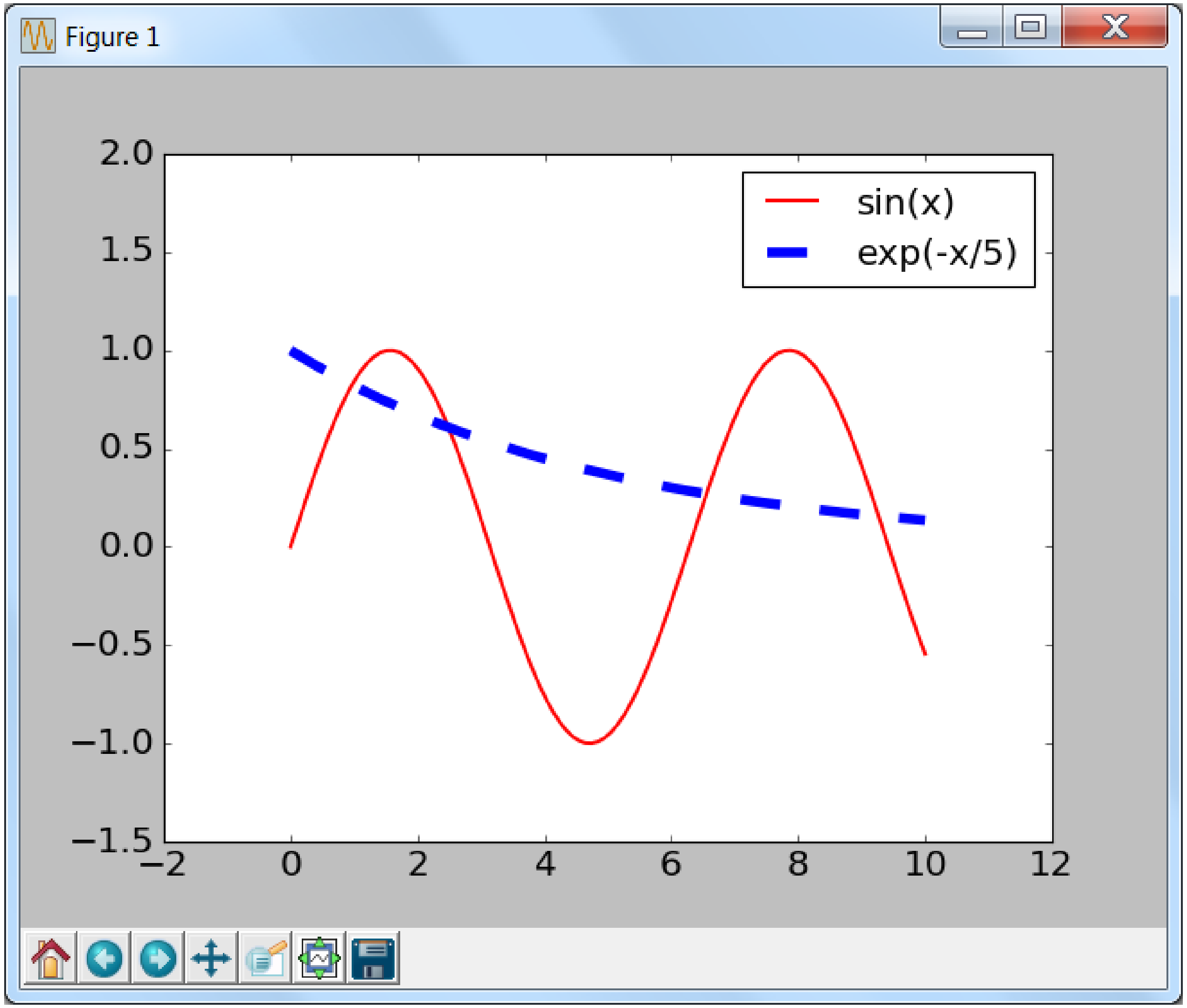
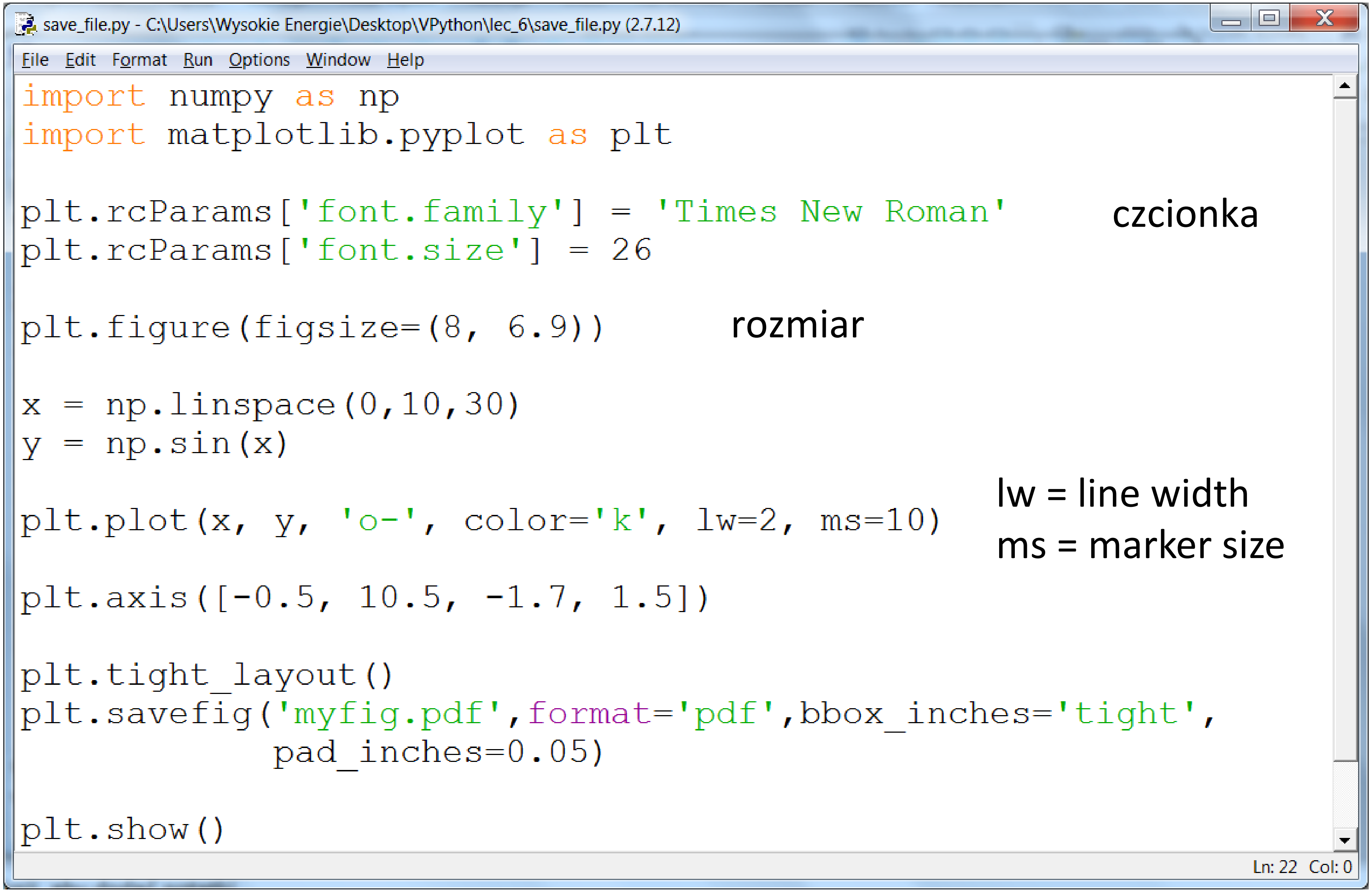


Figure 1
sin(x)
exp(-x/5)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2
0
2
4
6
8
10
12

Rozmiar wykresu i zapisywanie do pliku


save_file.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\save_file.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 22 Col: 0

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.family'] = 'Times New Roman'
plt.rcParams['font.size'] = 26

plt.figure(figsize=(8, 6.9))

x = np.linspace(0,10,30)
y = np.sin(x)

plt.plot(x, y, 'o-', color='k', lw=2, ms=10)

plt.axis([-0.5, 10.5, -1.7, 1.5])

plt.tight_layout()
plt.savefig('myfig.pdf',format='pdf',bbox_inches='tight',
            pad_inches=0.05)

plt.show()
```

czcionka

rozmiar

lw = line width

ms = marker size

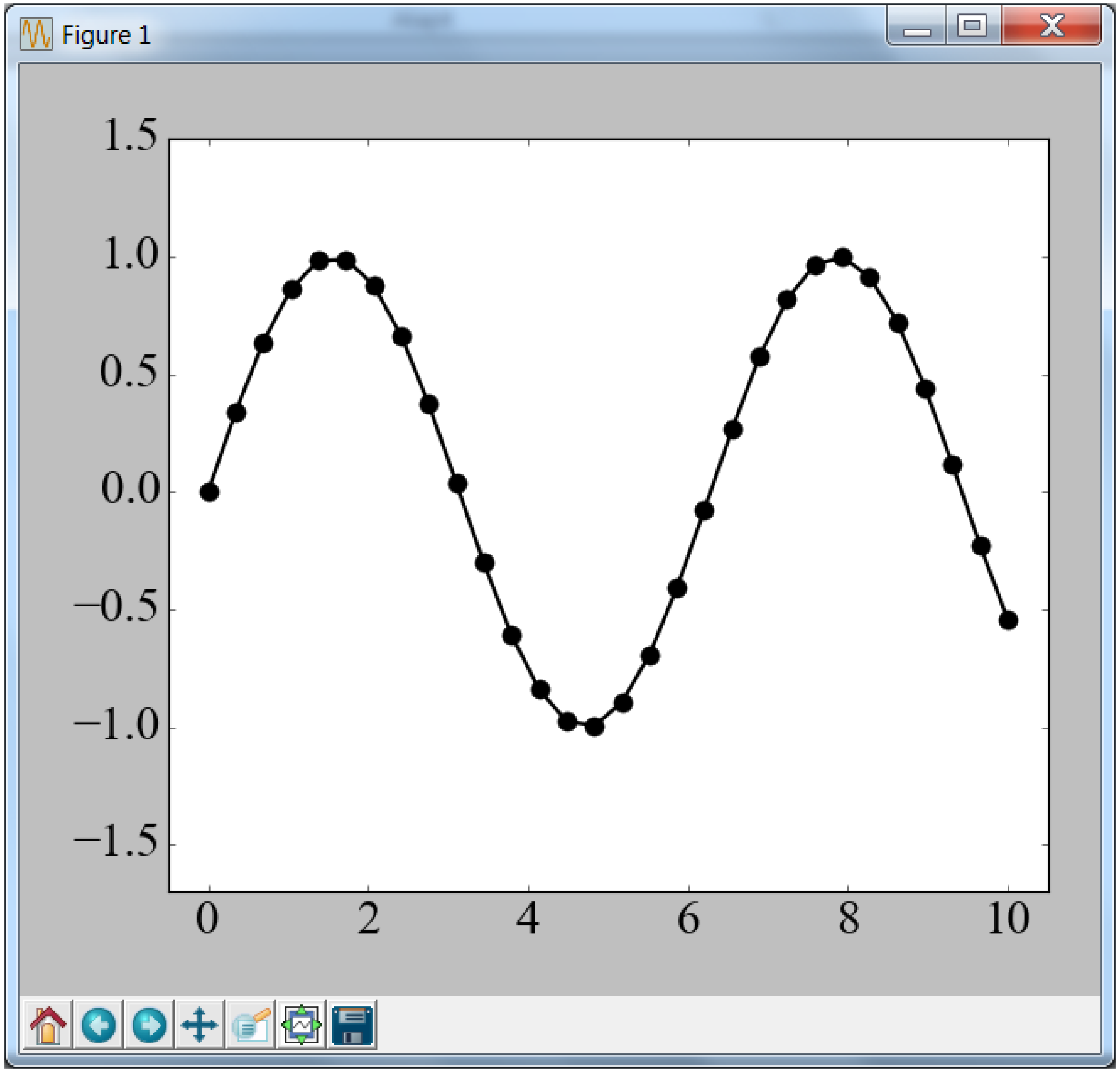


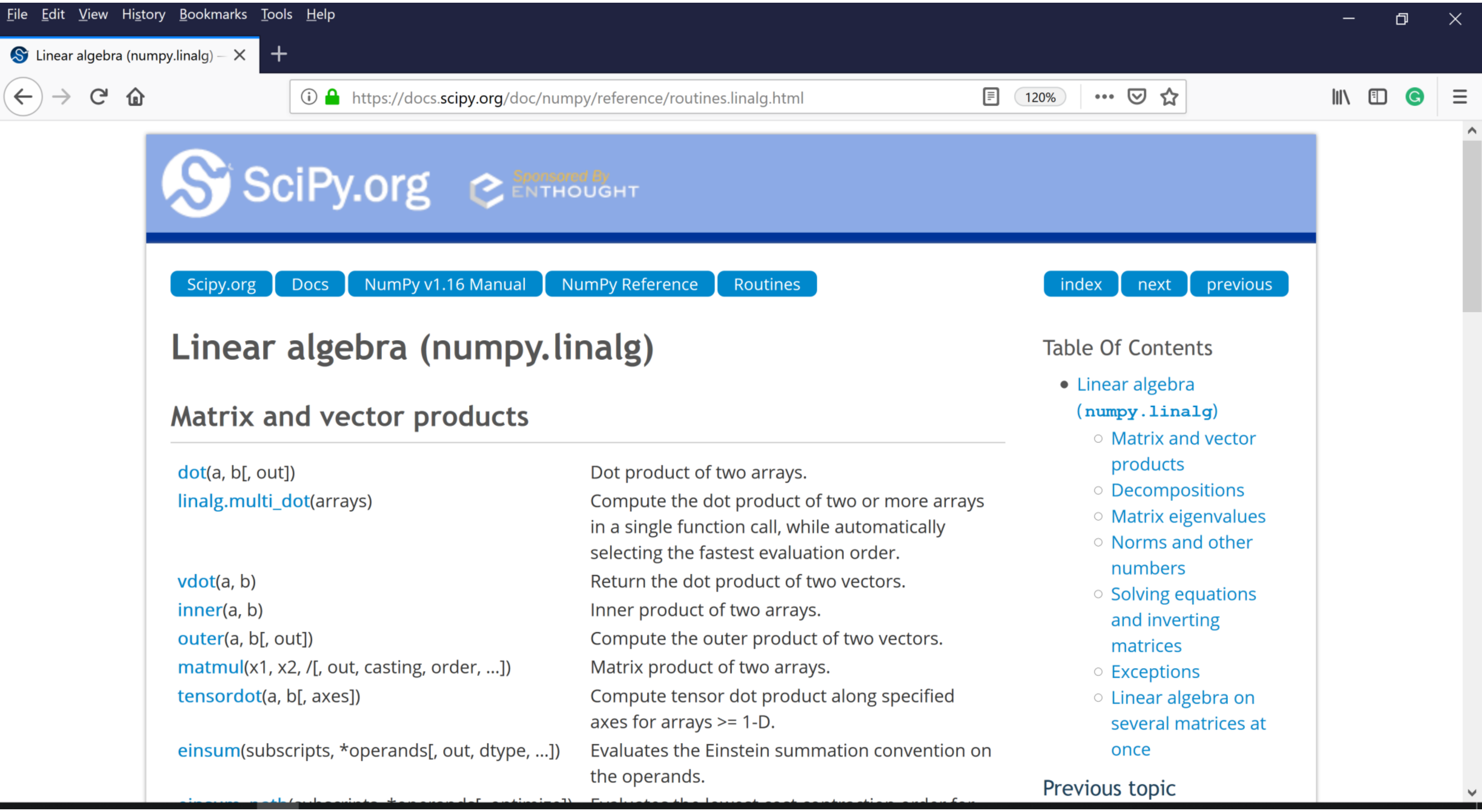
Figure 1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
2
4
6
8
10

Dodatek

(dla zainteresowanych)

algebra w NumPy

http://docs.scipy.org/doc/numpy/reference/routines.linalg.html


File Edit View History Bookmarks Tools Help
Linear algebra (numpy.linalg)
https://docs.scipy.org/doc/numpy/reference/routines.linalg.html
120%
SciPy.org
Sponsored By ENTHOUGHT
Scipy.org
Docs
NumPy v1.16 Manual
NumPy Reference
Routines
index
next
previous

Linear algebra (numpy.linalg)

Matrix and vector products

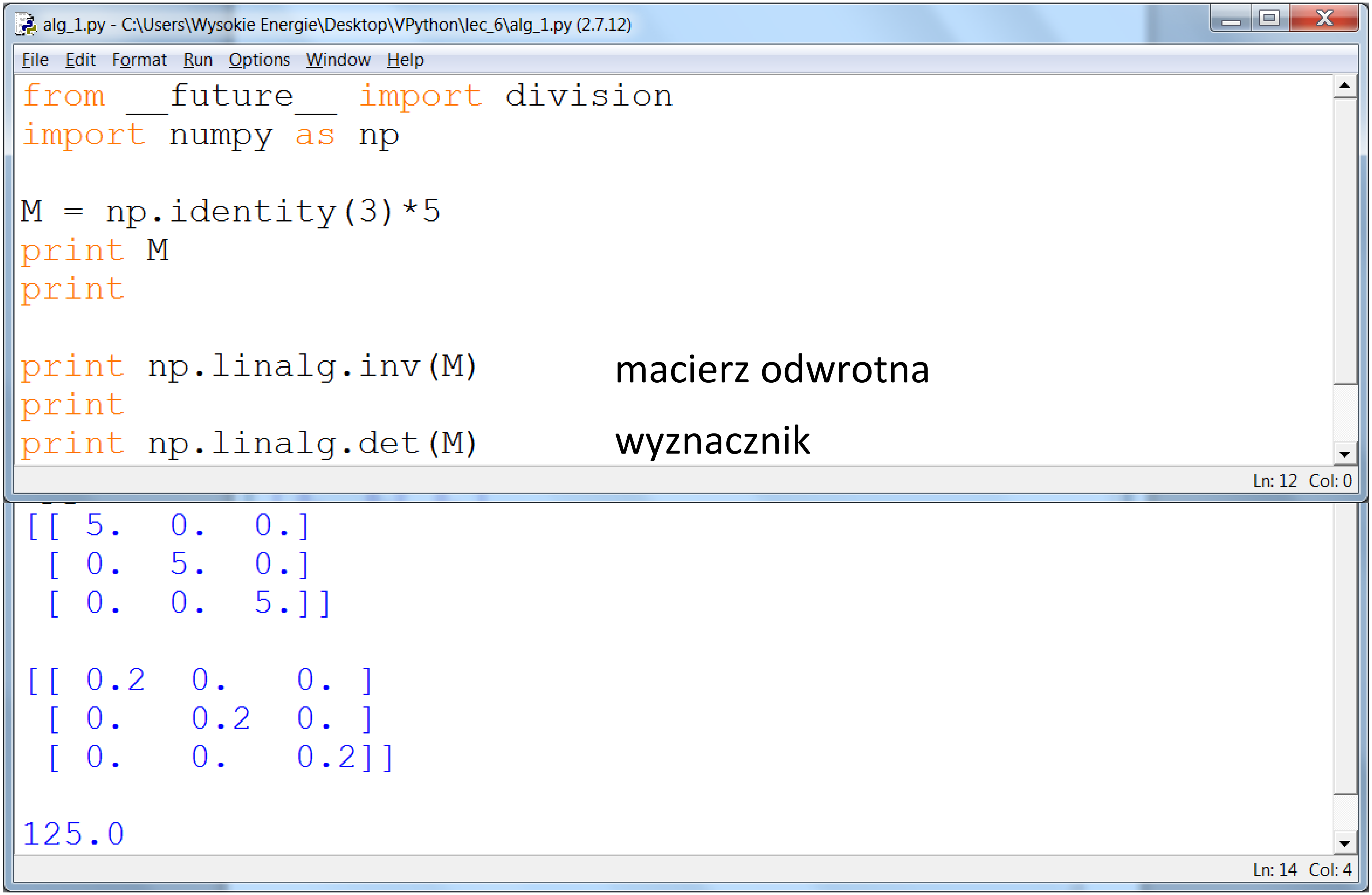
dot(a, b[, out])	Dot product of two arrays.
linalg.multi_dot(arrays)	Compute the dot product of two or more arrays in a single function call, while automatically selecting the fastest evaluation order.
vdot(a, b)	Return the dot product of two vectors.
inner(a, b)	Inner product of two arrays.
outer(a, b[, out])	Compute the outer product of two vectors.
matmul(x1, x2, /[, out, casting, order, ...])	Matrix product of two arrays.
tensordot(a, b[, axes])	Compute tensor dot product along specified axes for arrays >= 1-D.
einsum(subscripts, *operands[, out, dtype, ...])	Evaluates the Einstein summation convention on the operands.

Table Of Contents

Previous topic

Rozwiązywanie równań, wartości i wektory własne itd.

na przykład:


alg_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\alg_1.py (2.7.12)

```
from __future__ import division
import numpy as np

M = np.identity(3)*5
print M
print

print np.linalg.inv(M)
print
print np.linalg.det(M)
```

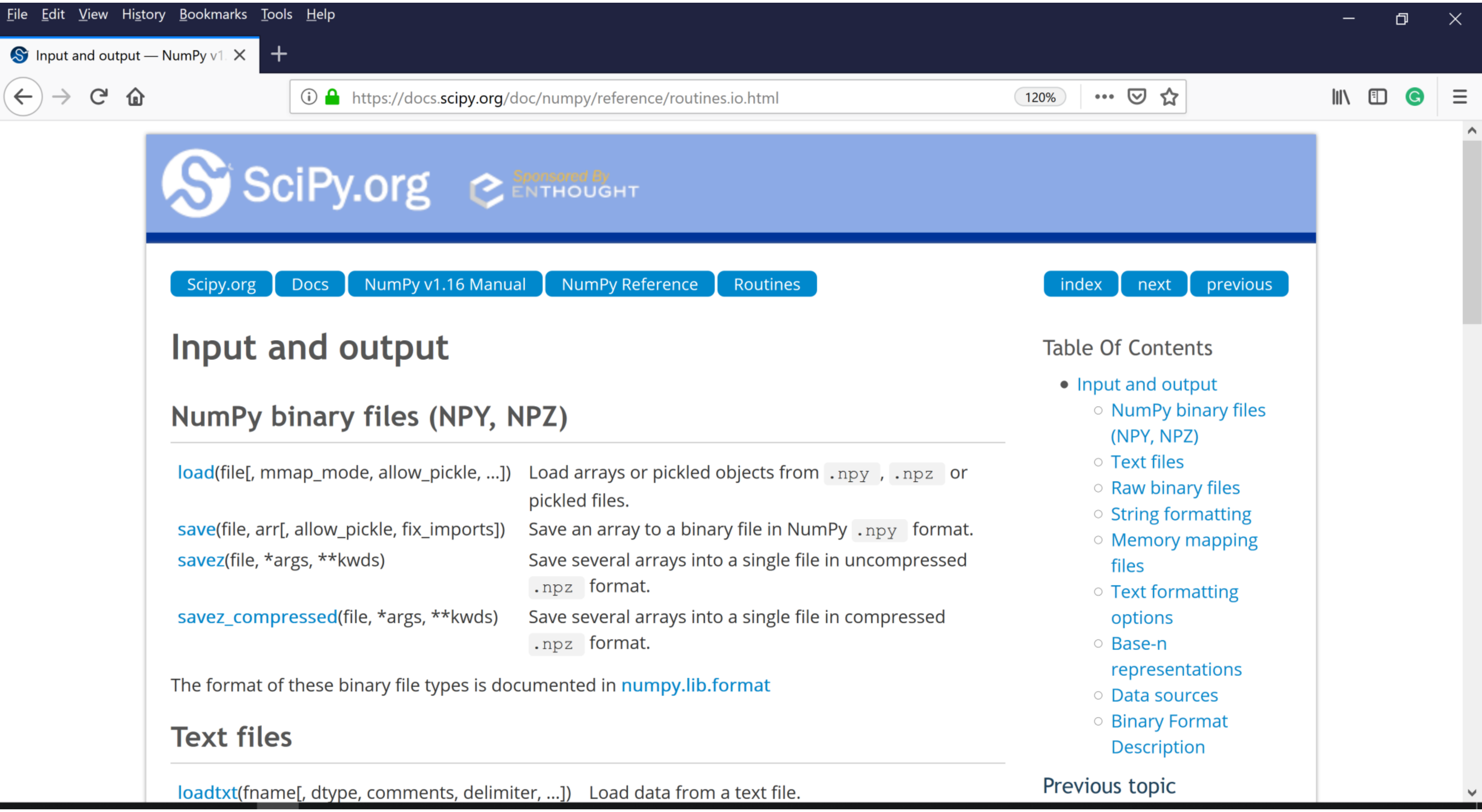
macierz odwrotna

wyznacznik

```
[[ 5.   0.   0.]
 [ 0.   5.   0.]
 [ 0.   0.   5.]]

[[ 0.2  0.   0. ]
 [ 0.   0.2  0. ]
 [ 0.   0.   0.2]]

125.0
```

Input, output w NumPy

http://docs.scipy.org/doc/numpy/reference/routines.io.html


File Edit View History Bookmarks Tools Help
Input and output — NumPy v1
https://docs.scipy.org/doc/numpy/reference/routines.io.html
120%
SciPy.org
Sponsored By ENTHOUGHT
Scipy.org
Docs
NumPy v1.16 Manual
NumPy Reference
Routines
index
next
previous

Input and output

NumPy binary files (NPY, NPZ)

load(file[, mmap_mode, allow_pickle, …])	Load arrays or pickled objects from `.npy`, `.npz` or pickled files.
save(file, arr[, allow_pickle, fix_imports])	Save an array to a binary file in NumPy `.npy` format.
savez(file, *args, **kwds)	Save several arrays into a single file in uncompressed `.npz` format.
savez_compressed(file, *args, **kwds)	Save several arrays into a single file in compressed `.npz` format.

The format of these binary file types is documented in numpy.lib.format

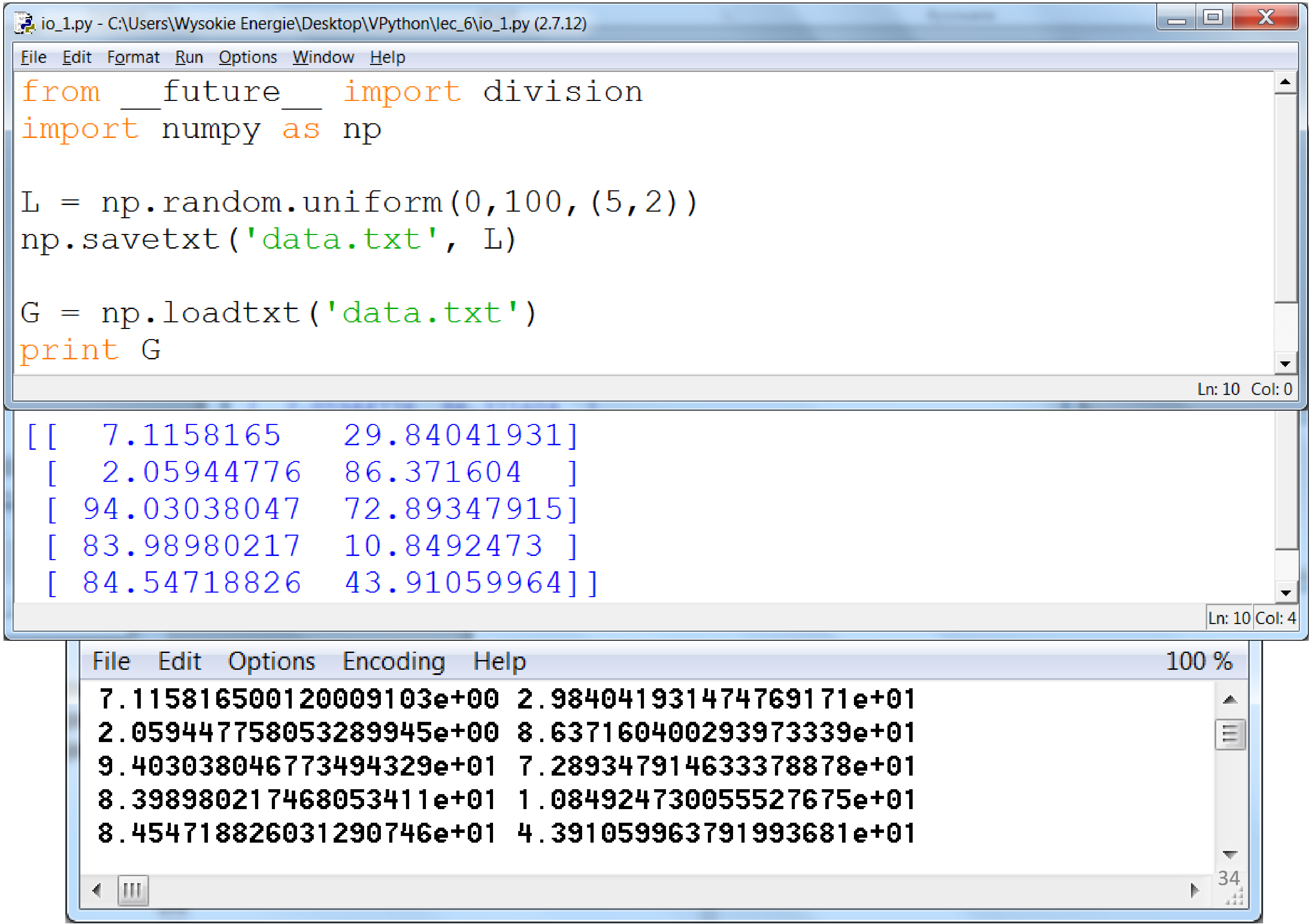
Text files

loadtxt(fname[, dtype, comments, delimiter, …])	Load data from a text file.

Table Of Contents

Previous topic

na przykład:


io_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_6\io_1.py (2.7.12)
File Edit Format Run Options Window Help
Ln: 10 Col: 0

```
from __future__ import division
import numpy as np

L = np.random.uniform(0,100,(5,2))
np.savetxt('data.txt', L)

G = np.loadtxt('data.txt')
print G
```

```
[[  7.1158165   29.84041931]
 [  2.05944776  86.371604  ]
 [ 94.03038047  72.89347915]
 [ 83.98980217  10.8492473 ]
 [ 84.54718826  43.91059964]]
```

Ln: 10 Col: 4
File Edit Options Encoding Help 100 %

```
7.115816500120009103e+00 2.984041931474769171e+01
2.059447758053289945e+00 8.637160400293973339e+01
9.403038046773494329e+01 7.289347914633378878e+01
8.398980217468053411e+01 1.084924730055527675e+01
8.454718826031290746e+01 4.391059963791993681e+01
```