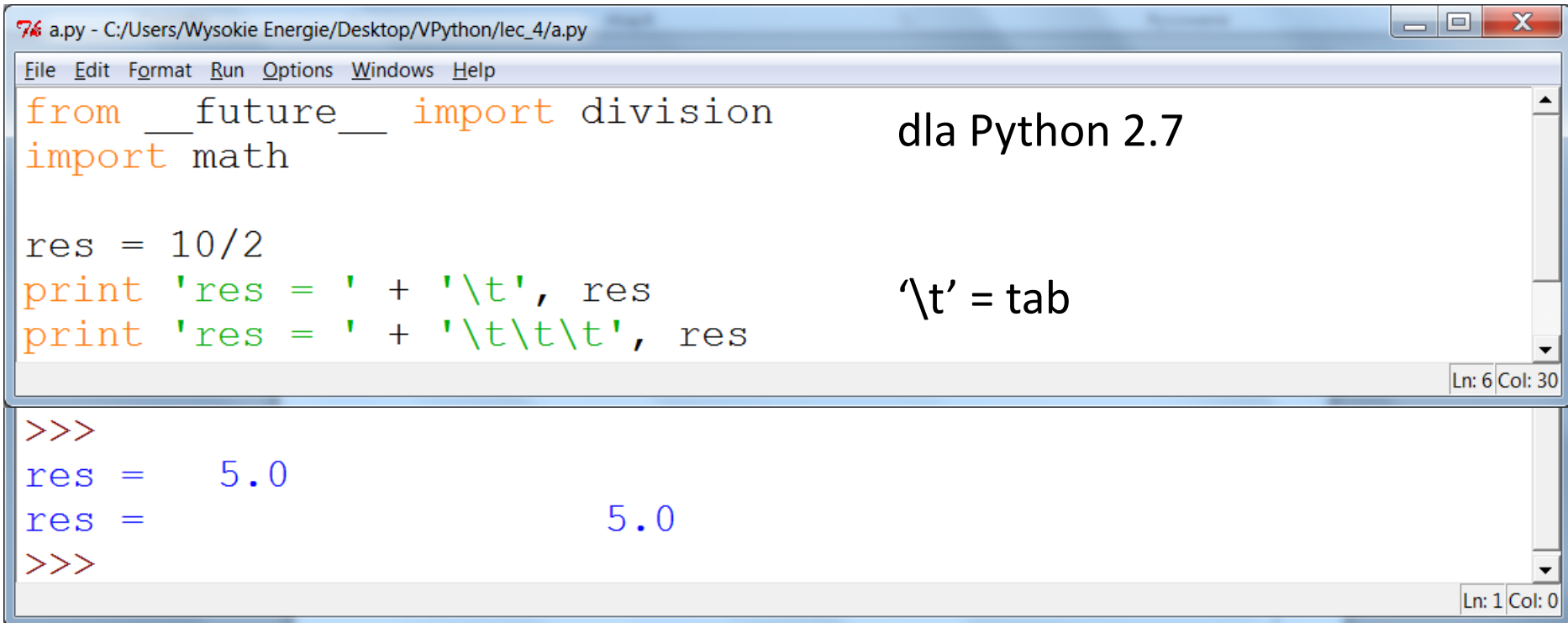


# VPython - symulacje fizyczne z grafiką 3D dla każdego

## wykład 4

Dr hab. Adam Bzdak

from \_\_future\_\_ import division musi być na samym początku



The screenshot shows a Python IDE window titled 'a.py - C:/Users/Wysokie Energie/Desktop/VPython/lec\_4/a.py'. The code in the editor is:

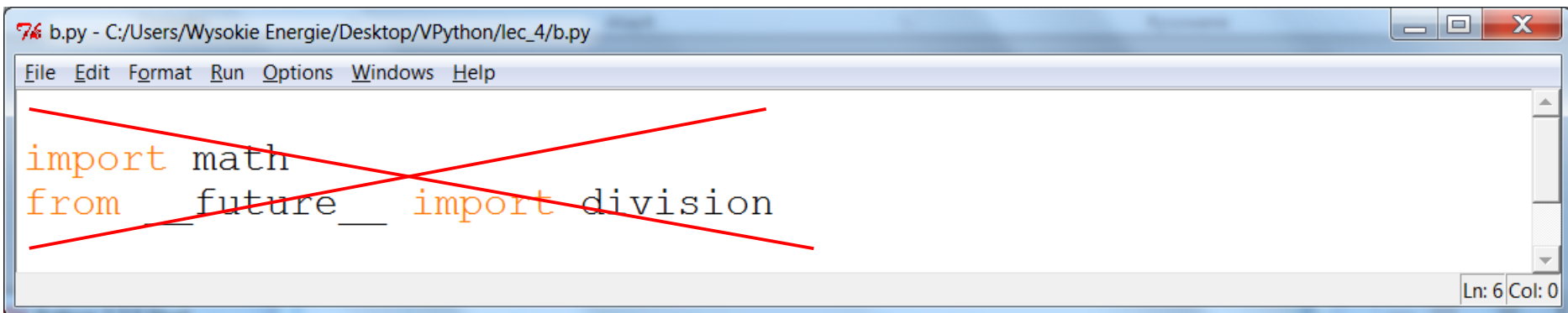
```
from __future__ import division
import math

res = 10/2
print 'res = ' + '\t', res
print 'res = ' + '\t\t\t', res
```

To the right of the code, the text 'dla Python 2.7' is displayed. Below the code, the text '\t' = tab is shown. The output of the code is displayed in the console:

```
>>>
res = 5.0
res = 5.0
>>>
```

The status bar at the bottom right of the IDE shows 'Ln: 6 Col: 30' for the code editor and 'Ln: 1 Col: 0' for the console.



The screenshot shows a Python IDE window titled 'b.py - C:/Users/Wysokie Energie/Desktop/VPython/lec\_4/b.py'. The code in the editor is:

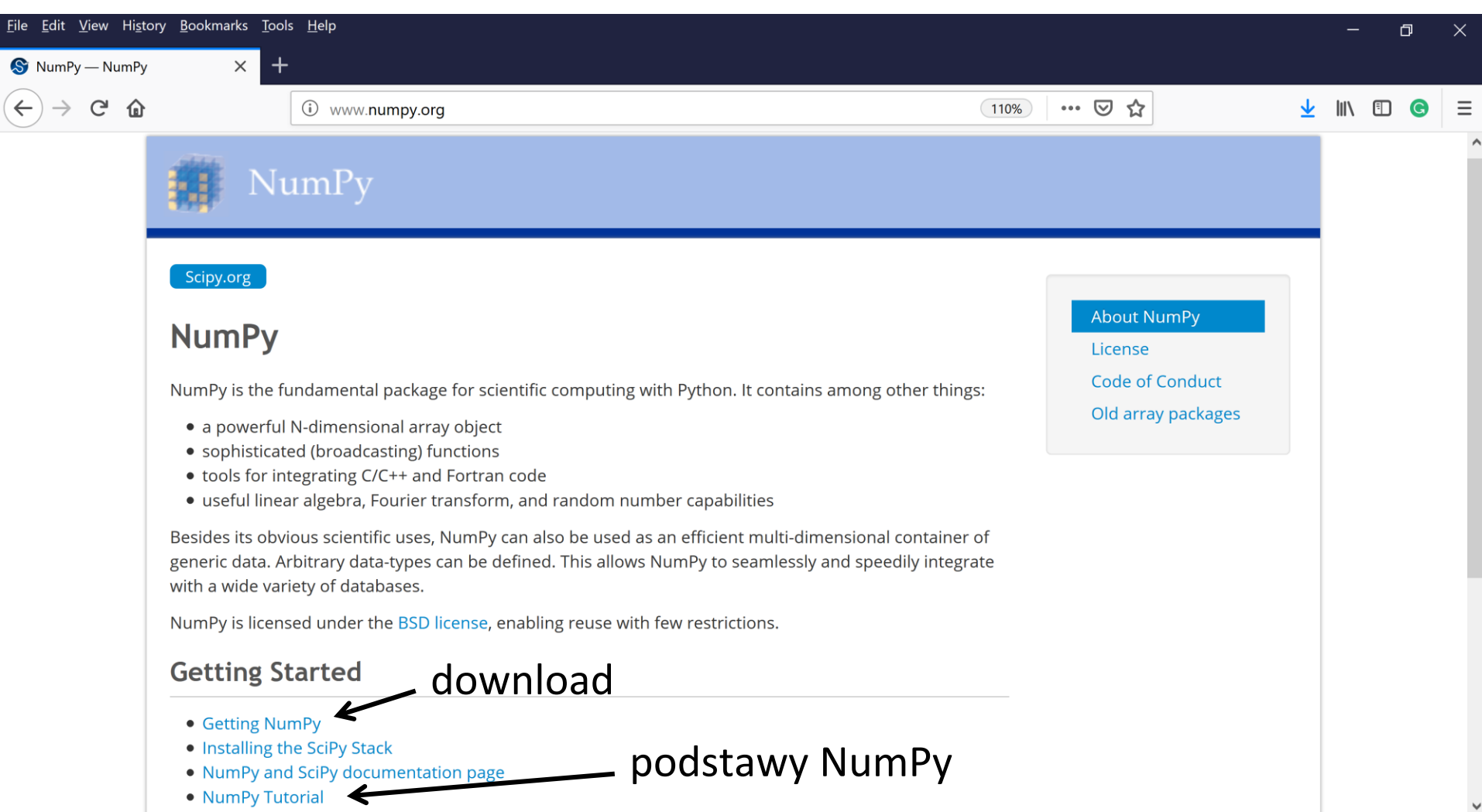
```
import math
from __future__ import division
```

The code is crossed out with a large red X. The status bar at the bottom right of the IDE shows 'Ln: 6 Col: 0'.

# NumPy

# NumPy (Numerical Python for scientific computing)

<http://www.numpy.org/>



The screenshot shows the NumPy website in a web browser. The browser's address bar displays 'www.numpy.org'. The website has a blue header with the NumPy logo. Below the header, there is a 'Scipy.org' button. The main content area features the title 'NumPy' and a description: 'NumPy is the fundamental package for scientific computing with Python. It contains among other things:'. A list of features follows: 'a powerful N-dimensional array object', 'sophisticated (broadcasting) functions', 'tools for integrating C/C++ and Fortran code', and 'useful linear algebra, Fourier transform, and random number capabilities'. Below this, a paragraph states: 'Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.' Another paragraph mentions: 'NumPy is licensed under the [BSD license](#), enabling reuse with few restrictions.' On the right side, there is a sidebar with links: 'About NumPy', 'License', 'Code of Conduct', and 'Old array packages'. At the bottom, there is a 'Getting Started' section with a list of links: 'Getting NumPy', 'Installing the SciPy Stack', 'NumPy and SciPy documentation page', and 'NumPy Tutorial'. Two arrows point to these links: one from the word 'download' to 'Getting NumPy', and another from the phrase 'podstawy NumPy' to 'NumPy Tutorial'.

File Edit View History Bookmarks Tools Help

NumPy — NumPy

www.numpy.org 110%

NumPy

Scipy.org

**NumPy**

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the [BSD license](#), enabling reuse with few restrictions.

**Getting Started**

- [Getting NumPy](#)
- [Installing the SciPy Stack](#)
- [NumPy and SciPy documentation page](#)
- [NumPy Tutorial](#)

download

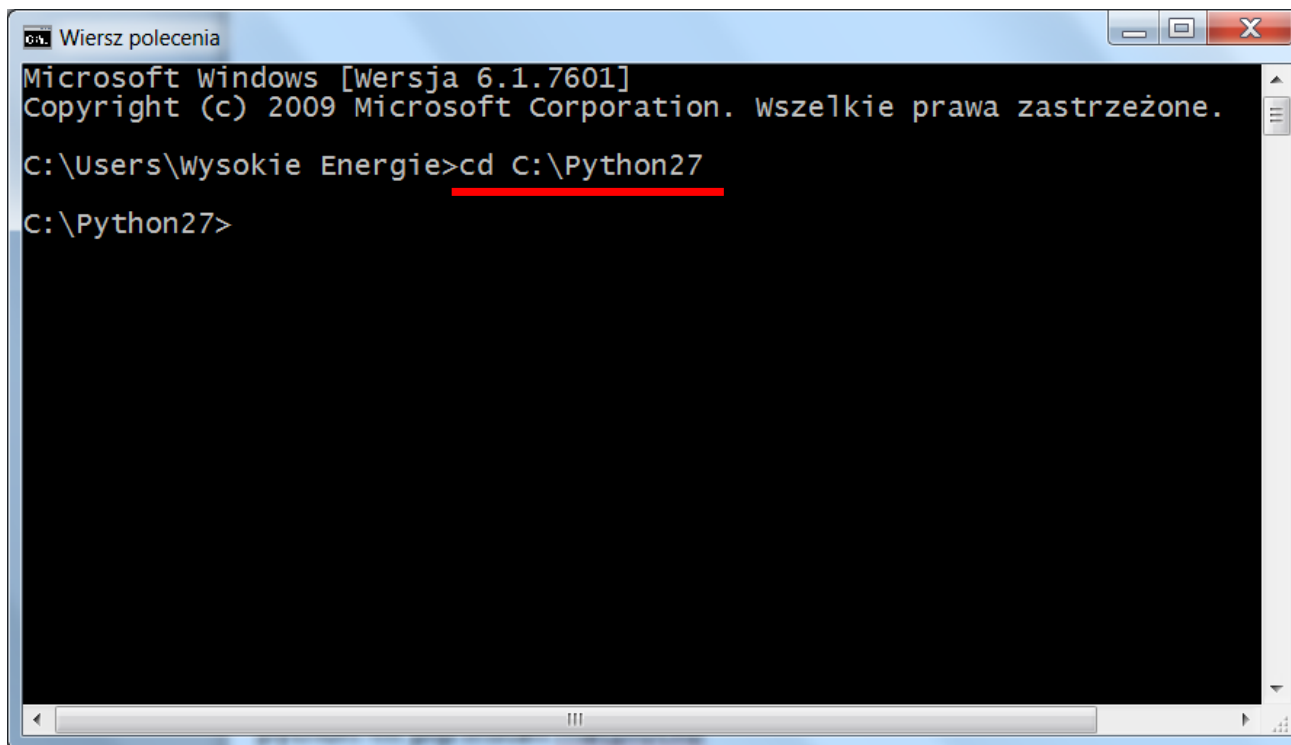
podstawy NumPy

About NumPy  
License  
Code of Conduct  
Old array packages

W wierszu poleceń wpisujemy

**cd C:\Python27**

jeśli tam jest zlokalizowany Python (lub PythonXX, np. Python 34, w zależności od wersji)



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

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

Następnie wpisujemy

**python -m pip install numpy**

lub

**python -m pip install --user numpy**

instaluje NumPy dla local user

## **pip ?**

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

Dla Python 2  $\leq$  2.7.8 i Python 3  $\leq$  3.3 należy zainstalować pip

<http://stackoverflow.com/questions/4750806/how-do-i-install-pip-on-windows>

# NumPy - array

```
numpy1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy1.py
File Edit Format Run Options Windows Help

import numpy as np

L = [1.0, 2.2, 3.1, 5.5]

a = np.array(L)

print a
print a.shape
print a.ndim
print a.size
print a.dtype.name
```

rozmiar array in każdym wymiarze  
liczba wymiarów  
całkowita liczba elementów  
typ elementów

```
>>>
[ 1.    2.2   3.1   5.5]
(4,)
1
4
float64
```

# NumPy, array

76 numpy2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_4\numpy2.py

File Edit Format Run Options Windows Help

```
import numpy as np
```

```
L = [[1,2,3], [4,5,6]]
```

```
a = np.array(L)
```

dwuwymiarowy array

```
print a
```

```
print a.shape
```

```
print a.ndim
```

```
print a.size
```

```
print a.dtype.name
```

Ln: 13 Col: 0

```
>>>
```

```
[[1 2 3]  
 [4 5 6]]
```

↕ 2

← 3 →

2 rzędy , 3 kolumny

```
(2, 3)
```

```
2
```

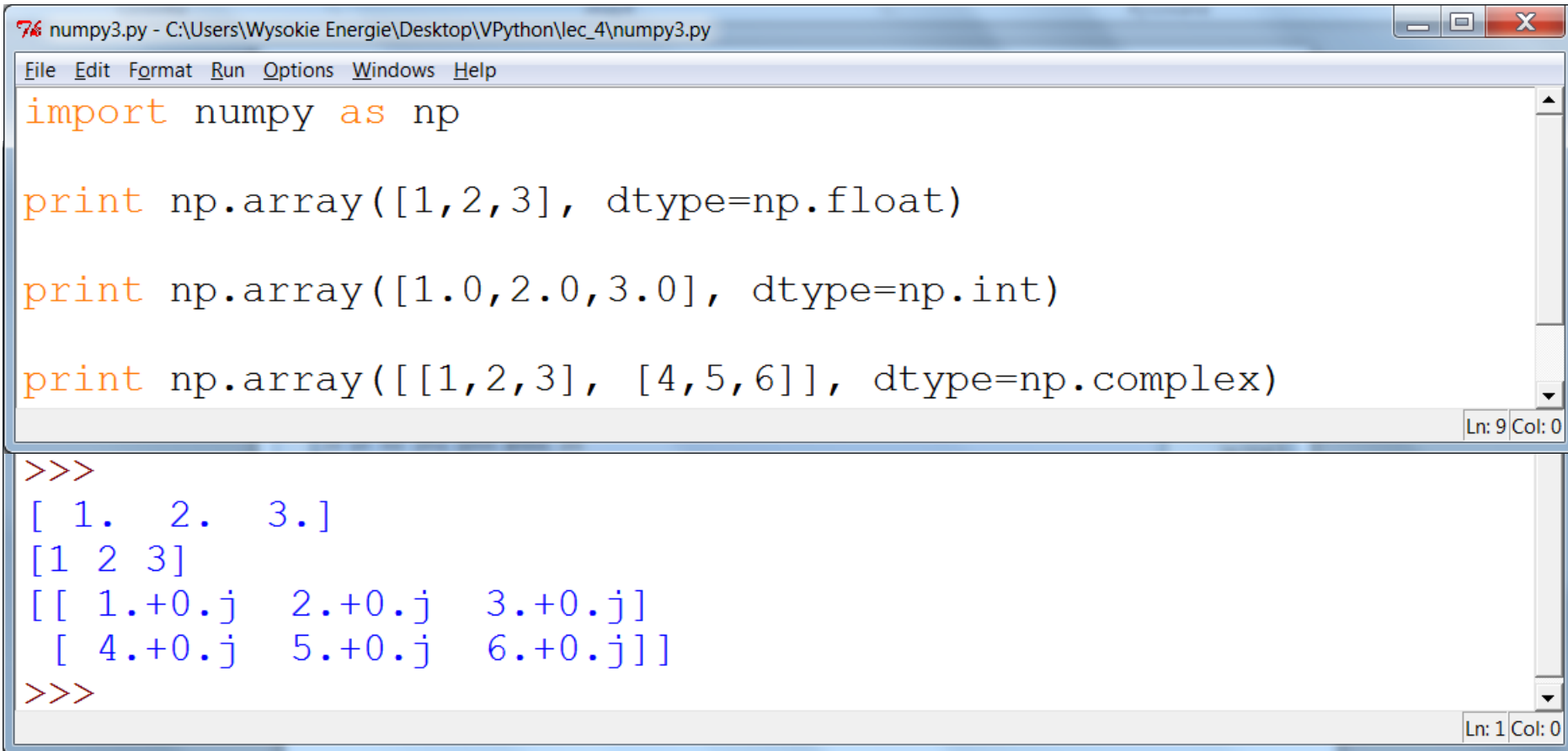
```
6
```

```
int32
```

Ln: 1 Col: 0



# definiowanie typu



The image shows a screenshot of a Python IDE window titled 'numpy3.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_4\numpy3.py'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The main text area contains the following Python code:

```
import numpy as np

print np.array([1,2,3], dtype=np.float)

print np.array([1.0,2.0,3.0], dtype=np.int)

print np.array([[1,2,3], [4,5,6]], dtype=np.complex)
```

The status bar at the bottom right of the editor shows 'Ln: 9 Col: 0'. Below the editor is a console window showing the output of the code:

```
>>>
[ 1.  2.  3.]
[1 2 3]
[[ 1.+0.j  2.+0.j  3.+0.j]
 [ 4.+0.j  5.+0.j  6.+0.j]]
>>>
```

The console window status bar at the bottom right shows 'Ln: 1 Col: 0'.

# Dlaczego NumPy ?

numpy9.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_4\numpy9.py

File Edit Format Run Options Windows Help

```
from __future__ import division
import numpy as np
```

```
a = np.array([1,2,3])
```

```
print a*10
```

```
print a/2
```

```
print a**2
```

```
print a + 10.5
```

```
print np.cos(a)
```

```
print a>0
```

szybkie operacje na każdym elemencie

**np.cos(a)** – używamy funkcji z NumPy

Ln: 13 Col: 0

```
>>>
```

```
[10 20 30]
```

```
[ 0.5  1.   1.5]
```

```
[1 4 9]
```

```
[ 11.5  12.5  13.5]
```

```
[ 0.54030231 -0.41614684 -0.9899925 ]
```

```
[ True  True  True]
```

Ln: 1 Col: 0

# Szybkie obliczenia

```
speed.py - C:/Users/Wysokie Energie/Desktop/VPython_17_pl/lec_4/speed.py (2.7.12)
File Edit Format Run Options Window Help

import numpy as np
import time

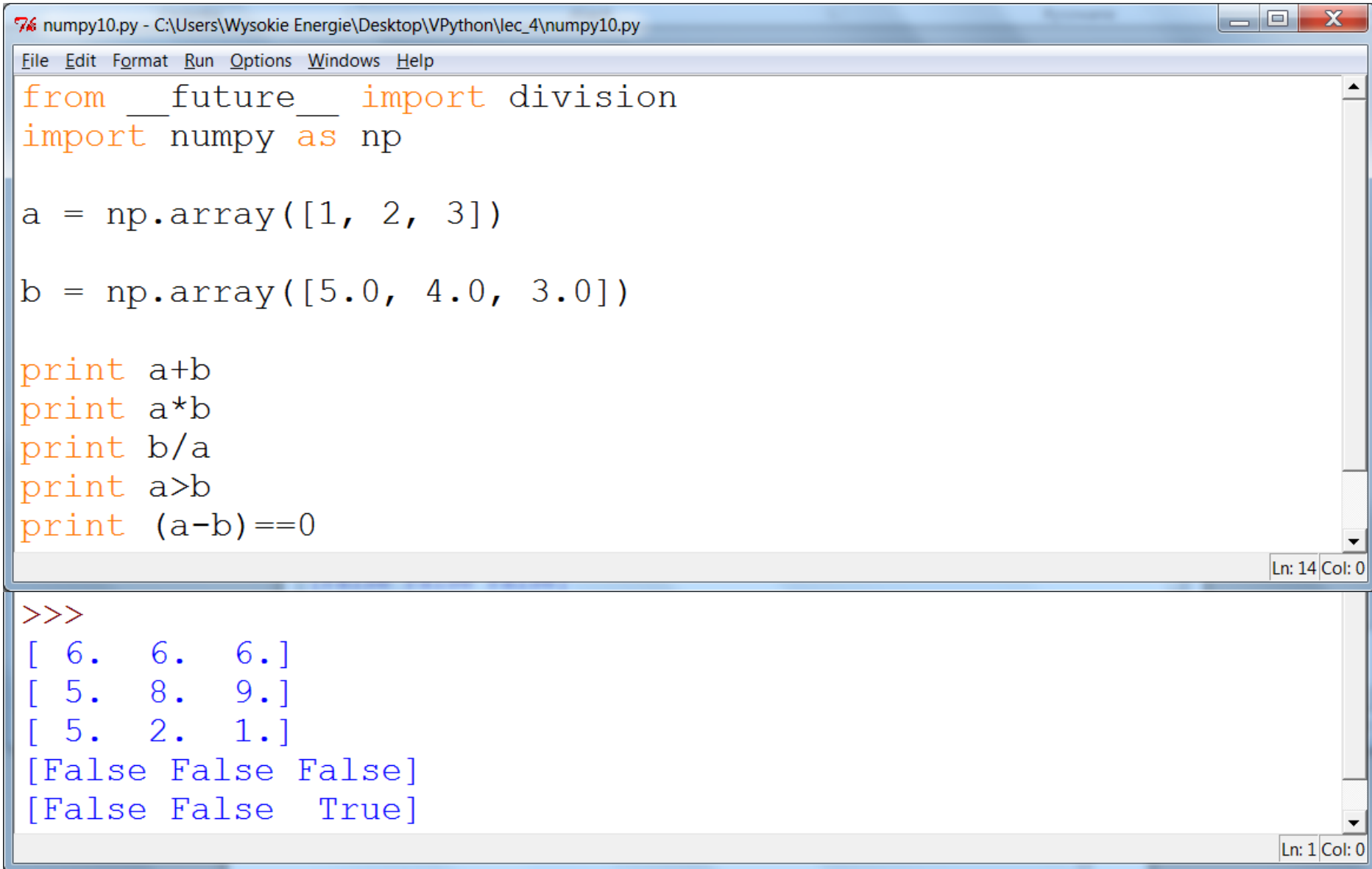
L = range(1,10**7)
L = np.array(L)
print L

start = time.clock()
G1 = []
for i in L:
    G1.append(i + 10)
print time.clock() - start

start = time.clock()
G2 = L + 10
print time.clock() - start

[      1      2      3 ..., 9999997 9999998 9999999]
3.47445617899
0.015455537019
>>>
```

# Dlaczego NumPy ?



The image shows a screenshot of a Python IDE window titled 'numpy10.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_4\numpy10.py'. The window contains a Python script and its execution output.

```
from __future__ import division
import numpy as np

a = np.array([1, 2, 3])

b = np.array([5.0, 4.0, 3.0])

print a+b
print a*b
print b/a
print a>b
print (a-b)==0
```

The output of the script is displayed in the console area below the editor:

```
>>>
[ 6.  6.  6.]
[ 5.  8.  9.]
[ 5.  2.  1.]
[False False False]
[False False  True]
```

The IDE window includes a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The status bar at the bottom right of the editor shows 'Ln: 14 Col: 0' and the status bar at the bottom right of the console shows 'Ln: 1 Col: 0'.

```
76 numpy10a.py - C:/Users/Wysokie Energie/Desktop/VPython/lec_4/numpy10a.py
File Edit Format Run Options Windows Help

from __future__ import division
import numpy as np

def fun(x, y):
    print "let's calculate"
    return np.sqrt(x*y)

a = np.array([4,2,3])
b = np.array([4,6,7.0])

print fun(a,b)

>>>
let's calculate
[ 4.          3.46410162  4.58257569]
>>>
```

## zeros, ones

76 numpy5.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_4\numpy5.py

File Edit Format Run Options Windows Help

```
import numpy as np

print np.zeros(10)
print np.zeros(10, dtype=np.int)

print np.ones(10)
print np.ones(10, dtype=np.int)
```

Ln: 9 Col: 0

```
>>>
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]      float64
[0 0 0 0 0 0 0 0 0 0]
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]
[1 1 1 1 1 1 1 1 1 1]
>>>
```

Ln: 1 Col: 0

# zeros, ones

numpy6.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_4\numpy6.py

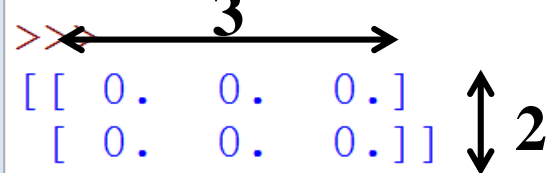
File Edit Format Run Options Windows Help

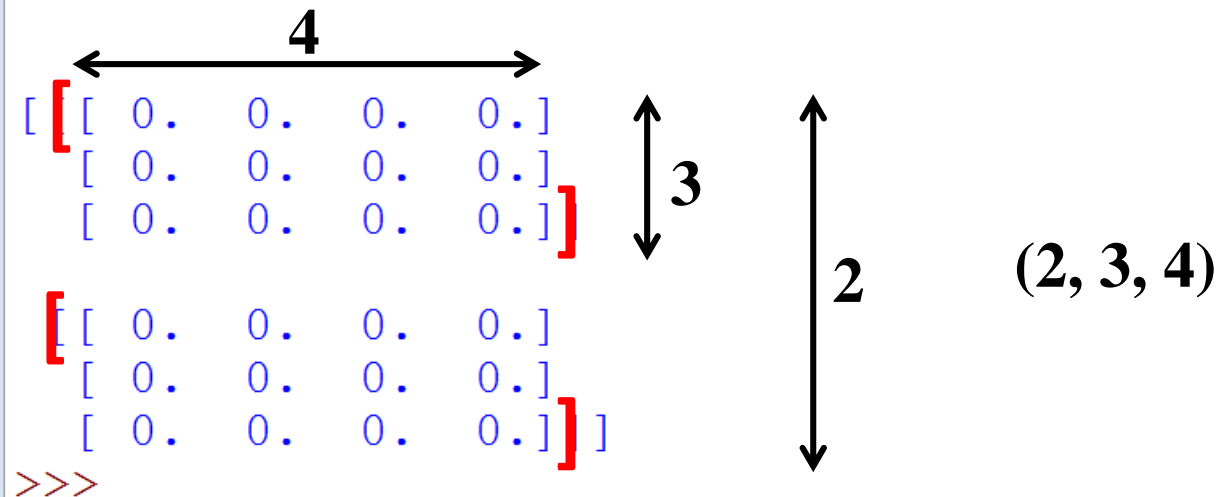
```
import numpy as np

print np.zeros((2,3))
print '\n'
print np.zeros((2,3,4))
```

Ln: 7 Col: 0

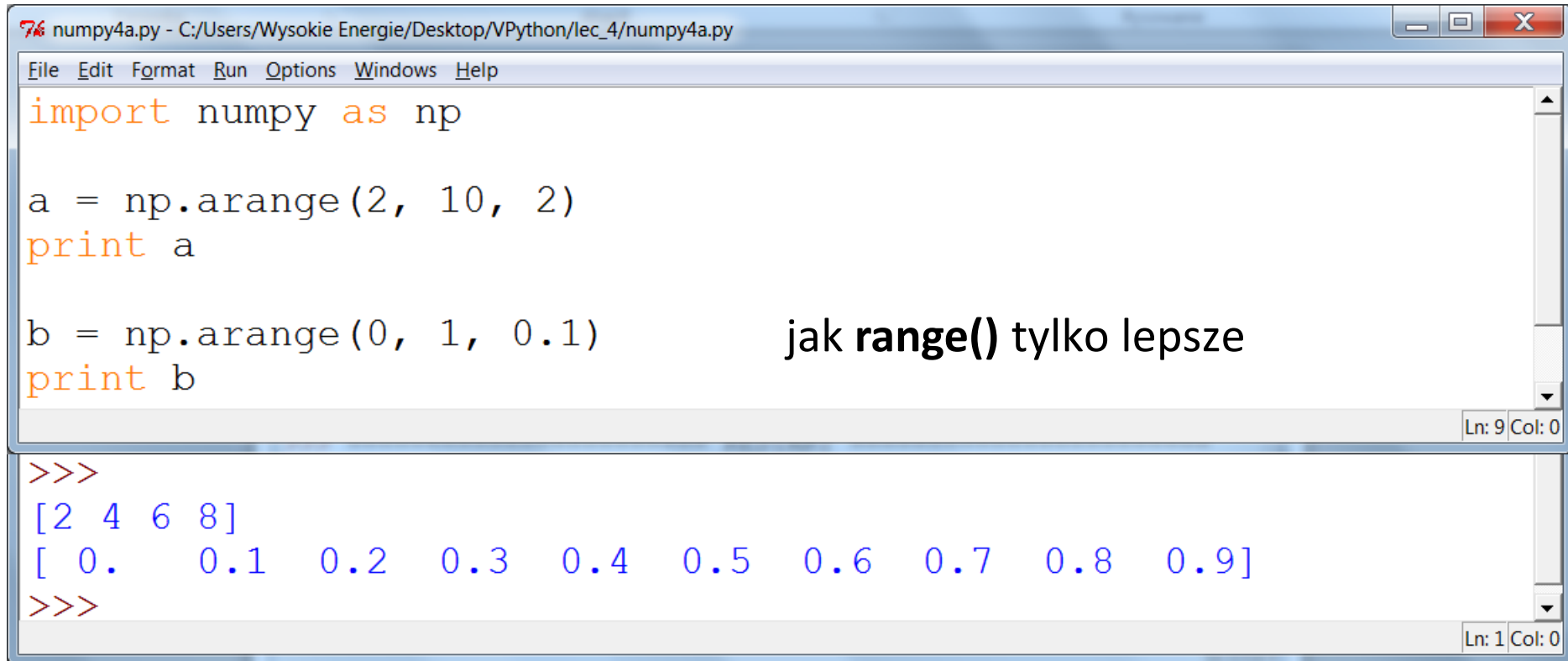
>>> ===== RESTART =====

>>> 



Ln: 2 Col: 29

# arange



The screenshot shows a Python IDE window titled 'numpy4a.py - C:/Users/Wysokie Energie/Desktop/VPython/lec\_4/numpy4a.py'. The code editor contains the following Python code:

```
import numpy as np

a = np.arange(2, 10, 2)
print a

b = np.arange(0, 1, 0.1)
print b
```

To the right of the code, the text 'jak range() tylko lepsze' is written. The output console at the bottom shows the results of the code execution:

```
>>>
[2 4 6 8]
[ 0.  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9]
>>>
```

b można uzyskać poprzez  
[i/10 for i in range(0,10)]



# reshape

76 numpy4.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_4\numpy4.py

File Edit Format Run Options Windows Help

```
import numpy as np
```

```
a = np.arange(0, 10, 1)
```

```
print a, '\n'
```

```
b = a.reshape(5, 2)
```

5 rzędów, 2 kolumny

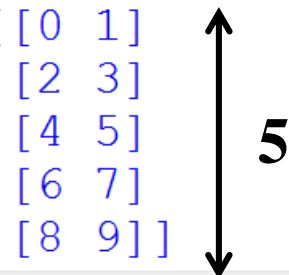
```
print b
```

Ln: 9 Col: 0

```
>>>
```

```
[0 1 2 3 4 5 6 7 8 9]
```

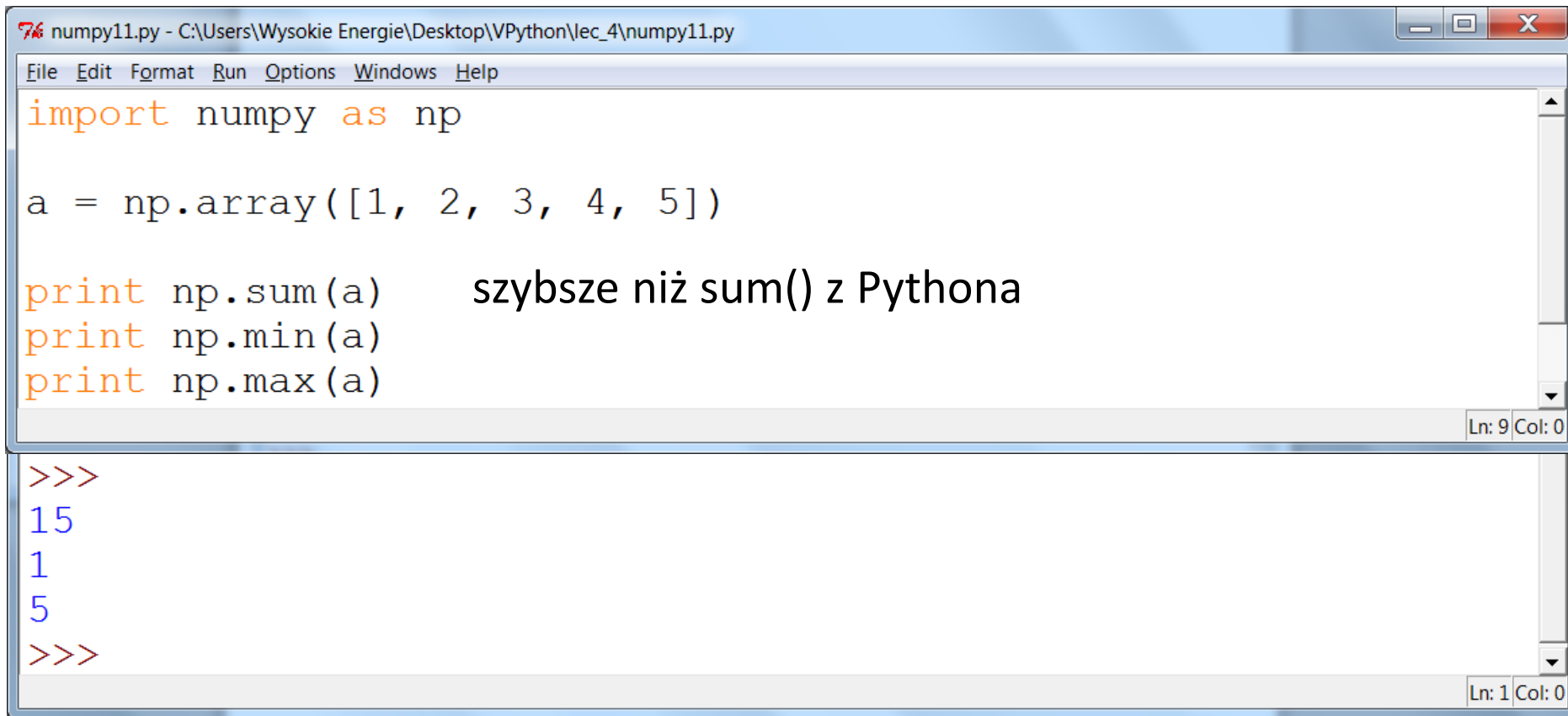
```
[[0 1]
 [2 3]
 [4 5]
 [6 7]
 [8 9]]
```



Ln: 1 Col: 0

2

## sum, min, max



The screenshot shows a Python IDE window titled 'numpy11.py - C:\Users\Wysokie Energie\Desktop\VPython\lec\_4\numpy11.py'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The main text area contains the following code:

```
import numpy as np

a = np.array([1, 2, 3, 4, 5])

print np.sum(a)      szybsze niż sum() z Pythona
print np.min(a)
print np.max(a)
```

The status bar at the bottom right of the text area shows 'Ln: 9 Col: 0'. Below the text area is a console window showing the output of the script:

```
>>>
15
1
5
>>>
```

The status bar at the bottom right of the console window shows 'Ln: 1 Col: 0'.

inny sposób

`a.sum(), a.min(), a.max()`

# NumPy, liczby przypadkowe

```
numpy12b.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy12b.py (2.7.12)
File Edit Format Run Options Window Help
import numpy as np

print np.random.random(3)

print np.random.uniform(0, 10, 5)

print np.random.randint(0, 6, (2,21))

>>>
[ 0.4189823  0.77655661  0.06715871]
[ 2.39741697  3.44915269  9.47332416  9.52902428  2.9812941 ]
[[0 2 2 1 2 1 5 3 4 0 5 3 1 2 5 5 5 0 3 2 1]
 [0 2 3 3 5 2 3 4 0 0 5 5 3 1 2 1 2 3 4 5 0]]
>>>
```

3 liczby float z  $[0,1)$

5 liczb float z przedziału  $[0,10)$

array (2,21) liczb całkowitych z  $[0,6)$ , bez 6

Proszę sprawdzić

<https://docs.scipy.org/doc/numpy/reference/routines.random.html>

## Wiele generatorów liczb losowych

FileEditViewHistoryBookmarksToolsHelp

Random sampling (numpy.rand X

←→↺↻🏠

🔒 https://docs.scipy.org/doc/numpy/reference/routines.random.html

⋮🔒🌟

⬇️🔍📄🔄

⋮

Scipy.orgDocsNumPy v1.16 ManualNumPy ReferenceRoutines

indexnextprevious

### Random sampling (numpy.random)

#### Simple random data

<code>rand(d0, d1, ..., dn)</code>	Random values in a given shape.
<code>randn(d0, d1, ..., dn)</code>	Return a sample (or samples) from the "standard normal" distribution.
<code>randint(low[, high, size, dtype])</code>	Return random integers from <i>low</i> (inclusive) to <i>high</i> (exclusive).
<code>random_integers(low[, high, size])</code>	Random integers of type np.int between <i>low</i> and <i>high</i> , inclusive.
<code>random_sample([size])</code>	Return random floats in the half-open interval [0.0, 1.0).
<code>random([size])</code>	Return random floats in the half-open interval [0.0, 1.0).
<code>randf([size])</code>	Return random floats in the half-open interval [0.0, 1.0).
<code>sample([size])</code>	Return random floats in the half-open interval [0.0, 1.0).
<code>choice(a[, size, replace, p])</code>	Generates a random sample from a given 1-D array
<code>bytes(length)</code>	Return random bytes.

#### Permutations

<code>shuffle(x)</code>	Modify a sequence in-place by shuffling its contents.
<code>permutation(x)</code>	Randomly permute a sequence, or return a permuted range.

#### Distributions

<code>beta(a, b[, size])</code>	Draw samples from a Beta distribution.
<code>binomial(n, p[, size])</code>	Draw samples from a binomial distribution.
<code>chisquare(df[, size])</code>	Draw samples from a chi-square distribution.

#### Table Of Contents

- Random sampling (`numpy.random`)
  - Simple random data
  - Permutations
  - Distributions
  - Random generator

#### Previous topic

[numpy.RankWarning](#)

#### Next topic

[numpy.random.rand](#)

#### Quick search

search