

VPython - symulacje fizyczne z grafiką 3D dla każdego

wykład 2

Dr hab. Adam Bzdak

Delete

```
*del_2a.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\del_2a.py*
File Edit Format Run Options Windows Help

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

del L[0]          usuwamy pierwszy element listy
print L

del L[0:2]
print L

del L[:]          usuwamy wszystkie elementy i mamy pustą listę
print L

del L             usuwamy listę z pamięci
print L
```

Ln: 16 Col: 0

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

Traceback (most recent call last):

File "C:\Users\Wysokie Energie\Desktop\VPython\lec_2\del_2a.py", line 13, in <module>

print L

NameError: name 'L' is not defined

Listy

```
76 x_4.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\x_4.py
File Edit Format Run Options Windows Help
L = [5, 10, 15]

L.append(137)
print L

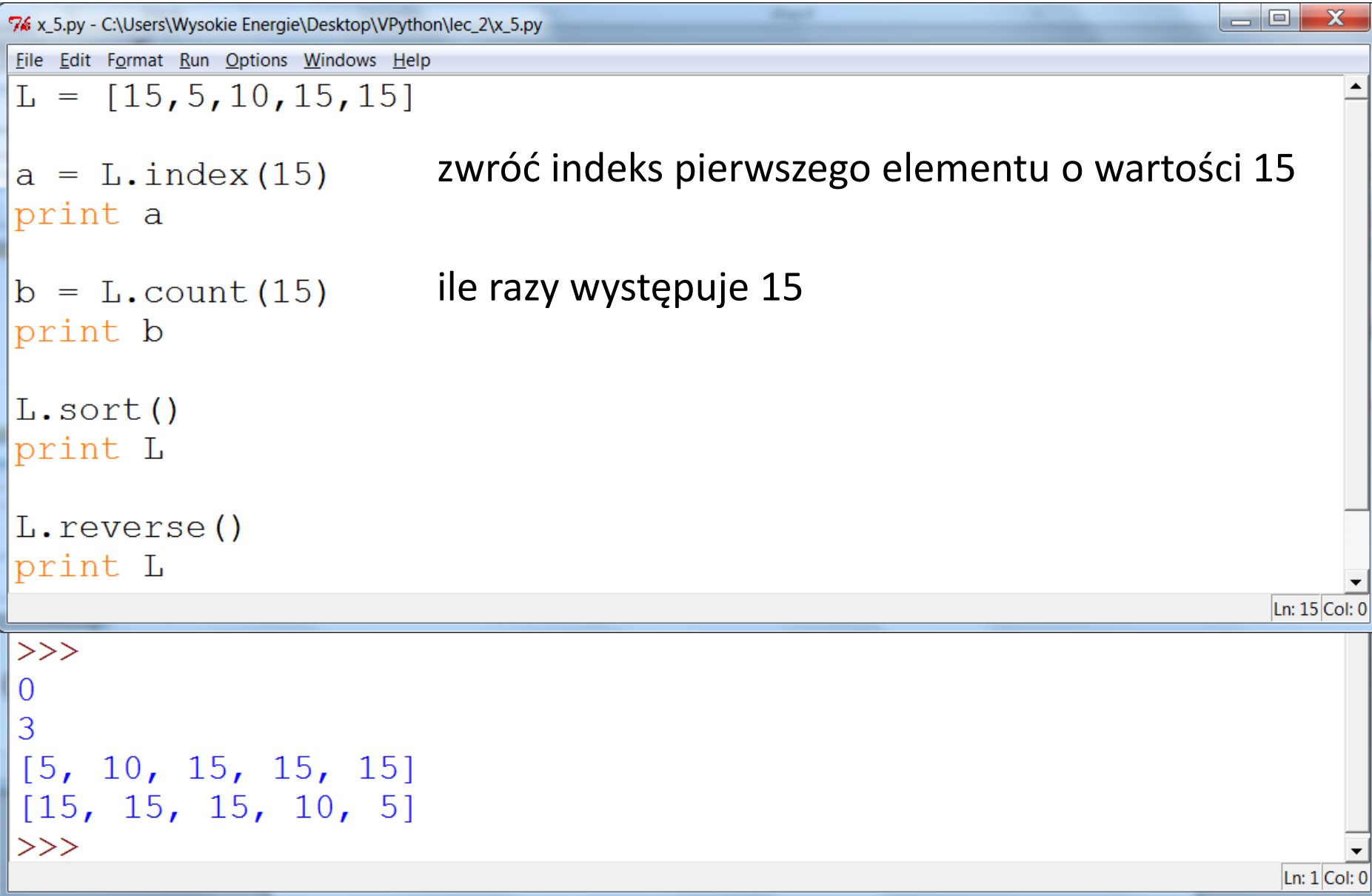
L.insert(1, 500)          wstaw 500 na pozycji 1
L.insert(-1, 500)
print L

L.remove(500)            usuń pierwszy element o wartości 500
print L

a = L.pop(2)             usuń drugi element i zwróć jego wartość
print L, a

>>>
[5, 10, 15, 137]
[5, 500, 10, 15, 500, 137]
[5, 10, 15, 500, 137]
[5, 10, 500, 137] 15
```

Listy



The image shows a screenshot of a Python IDE window titled 'x_5.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\x_5.py'. The window contains two panes. The top pane shows a Python script with the following code:

```
L = [15, 5, 10, 15, 15]

a = L.index(15)      zwróć indeks pierwszego elementu o wartości 15
print a

b = L.count(15)      ile razy występuje 15
print b

L.sort()
print L

L.reverse()
print L
```

The bottom pane shows the output of the script in a REPL environment:

```
>>>
0
3
[5, 10, 15, 15, 15]
[15, 15, 15, 10, 5]
>>>
```

The status bar at the bottom of the top pane indicates 'Ln: 15 Col: 0', and the status bar at the bottom of the bottom pane indicates 'Ln: 1 Col: 0'.

Zbiory (sets)

```
76 x_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\x_2.py
File Edit Format Run Options Windows Help
L = [1,2,3,4,5]
set1 = set(L)
set2 = set([4,5,9,10,10,10])
print set1
print set2, '\n'

print set1 - set2      # in set1 but not in set2
print set1 | set2      # in set1 or set2
print set1 & set2      # in set1 and in set2
print set1 ^ set2      # either in set1 or set2 but not in both
Ln: 13 Col: 0
```

```
>>>
set([1, 2, 3, 4, 5])
set([9, 10, 4, 5])      w zbiorach elementy się nie powtarzają

set([1, 2, 3])
set([1, 2, 3, 4, 5, 9, 10])
set([4, 5])
set([1, 2, 3, 9, 10])
Ln: 1 Col: 0
```

Zbiory (sets)

x_3.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\x_3.py

File Edit Format Run Options Windows Help

```
set1 = set("Mister Bean")
set2 = set("Benny Hill")
```

```
print set1
print set2, '\n'

print set1 & set2
```

Ln: 9 Col: 0

```
>>>
set(['a', ' ', 'B', 'e', 'i', 'M', 'n', 's', 'r', 't'])
set([' ', 'B', 'e', 'i', 'H', 'l', 'n', 'y'])

set(['i', ' ', 'B', 'e', 'n'])
>>>
```

Ln: 1 Col: 0

Słownik (dictionary)

lec_2c.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2c.py

File Edit Format Run Options Windows Help

```
tel = {'John': 1234, 'Kate': 5678}
```

```
tel['Tom'] = 9000
```

```
tel['Bob'] = 1000
```

```
print tel
```

```
print tel.keys()
```

```
print tel.values()
```

```
print tel['Kate']
```

```
print len(tel)          # length
```

```
print 'Tom' in tel
```

Ln: 14 Col: 0

```
>>> {'Bob': 1000, 'John': 1234, 'Kate': 5678, 'Tom': 9000}
```

```
['Bob', 'John', 'Kate', 'Tom']
```

```
[1000, 1234, 5678, 9000]
```

```
5678
```

```
4
```

```
True
```

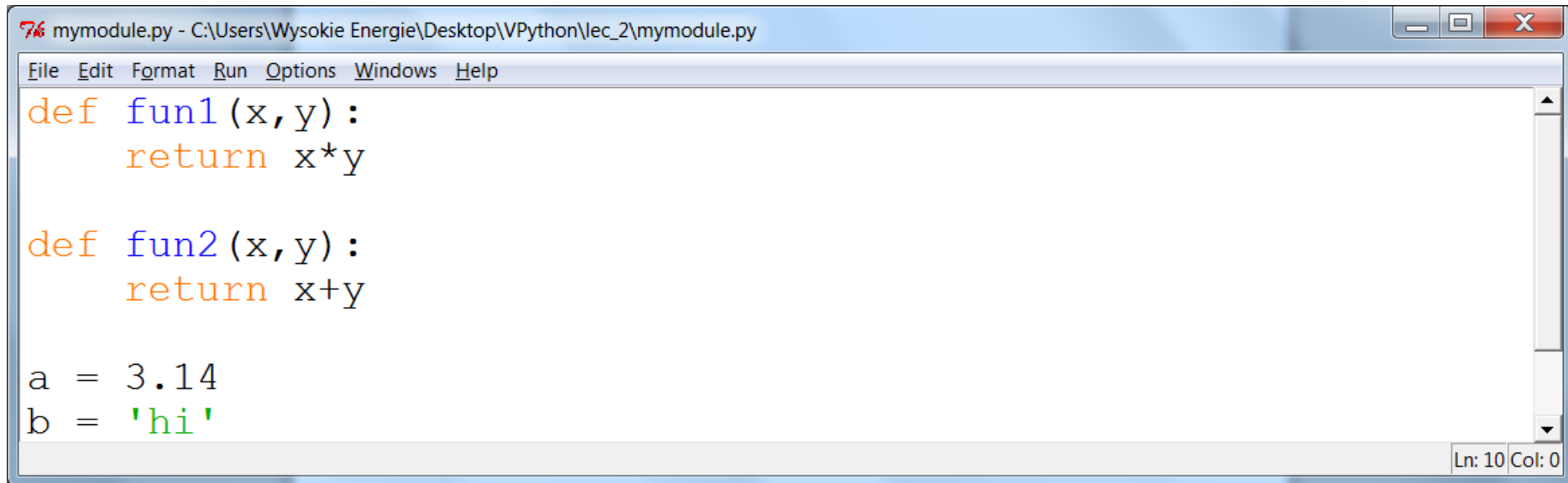
Ln: 1 Col: 0

list(tel.keys()) w Python 3

list(tel.values()) w Python 3

Moduły (modules)

Tworzymy plik *mymodule.py*

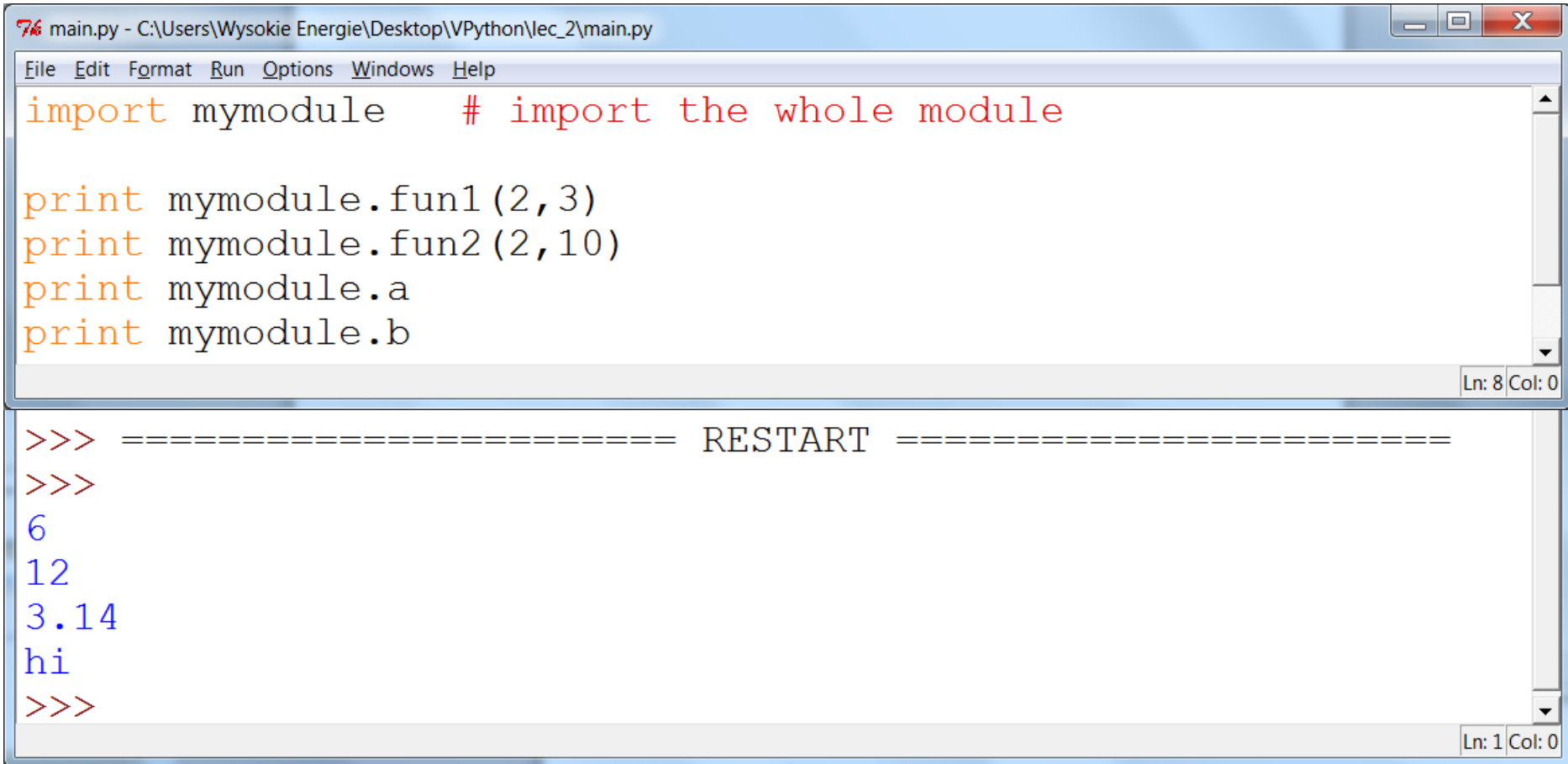
A screenshot of a Python IDE window titled "mymodule.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\mymodule.py". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The main text area contains the following Python code:

```
def fun1(x, y):  
    return x*y  
  
def fun2(x, y):  
    return x+y  
  
a = 3.14  
b = 'hi'
```

The code is color-coded: "def" is orange, "fun1" and "fun2" are blue, "return" is orange, "x" and "y" are black, and "a" and "b" are black. The status bar at the bottom right shows "Ln: 10 Col: 0".

W innym programie możemy użyć obiekty z *mymodule.py*

Moduły (modules)



The screenshot shows a Python IDE window titled "main.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\main.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 mymodule      # import the whole module

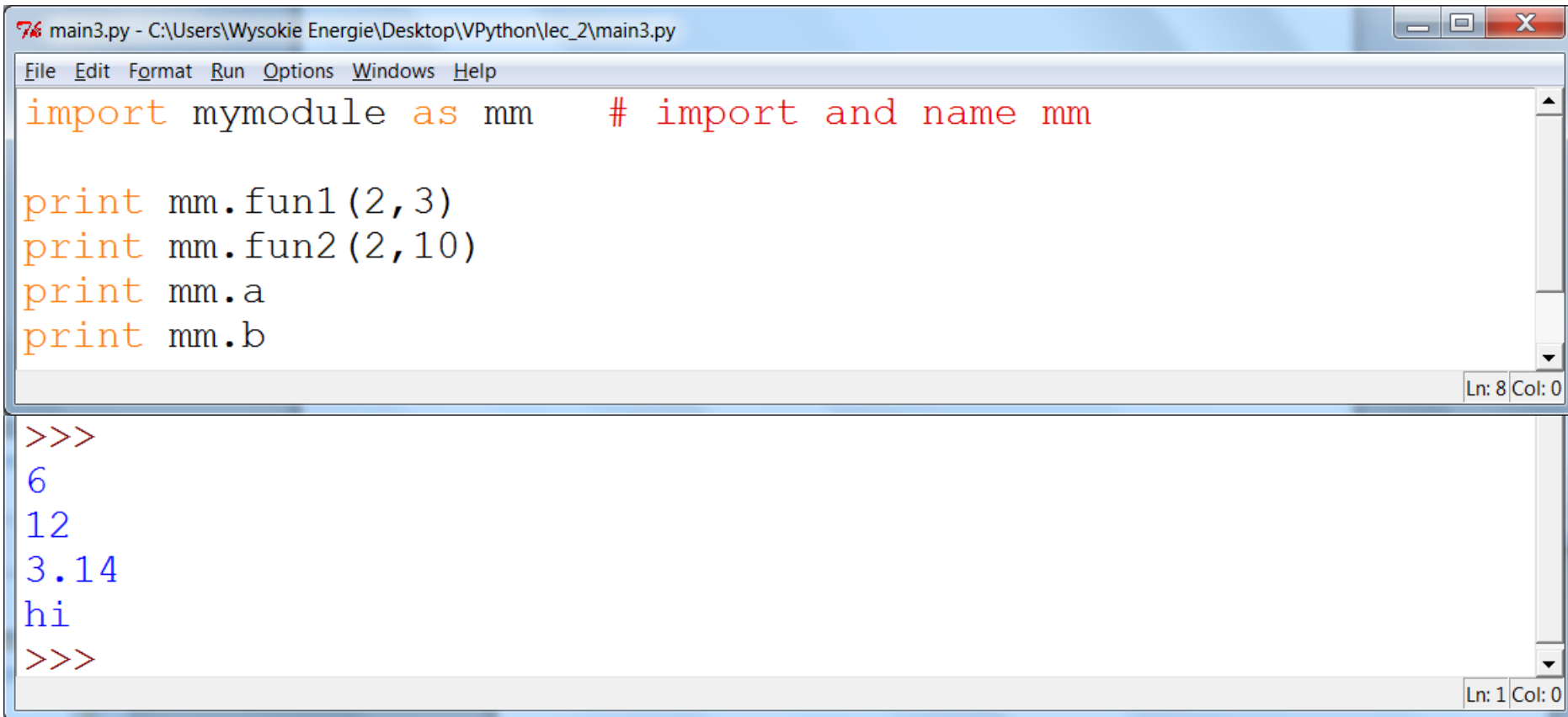
print mymodule.fun1(2,3)
print mymodule.fun2(2,10)
print mymodule.a
print mymodule.b
```

The status bar at the bottom right of the text area indicates "Ln: 8 Col: 0". Below the text area is a console window showing the output of the script:

```
>>> ===== RESTART =====
>>>
6
12
3.14
hi
>>>
```

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

import i zmiana nazwy



The image shows a screenshot of a Python IDE window titled "main3.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\main3.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 mymodule as mm      # import and name mm

print mm.fun1(2,3)
print mm.fun2(2,10)
print mm.a
print mm.b
```

The status bar at the bottom right of the text area indicates "Ln: 8 Col: 0". Below the text area is a console window showing the output of the script:

```
>>>
6
12
3.14
hi
>>>
```

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

`from mymodule import fun1, fun2` importuje tylko fun1 i fun2

`# from mymodule import *` importuje wszystkie obiekty

```
print fun1(2,3)
print fun2(2,10)
print a
```

Ln: 11 Col: 0

>>>

6

12

Traceback (most recent call last):

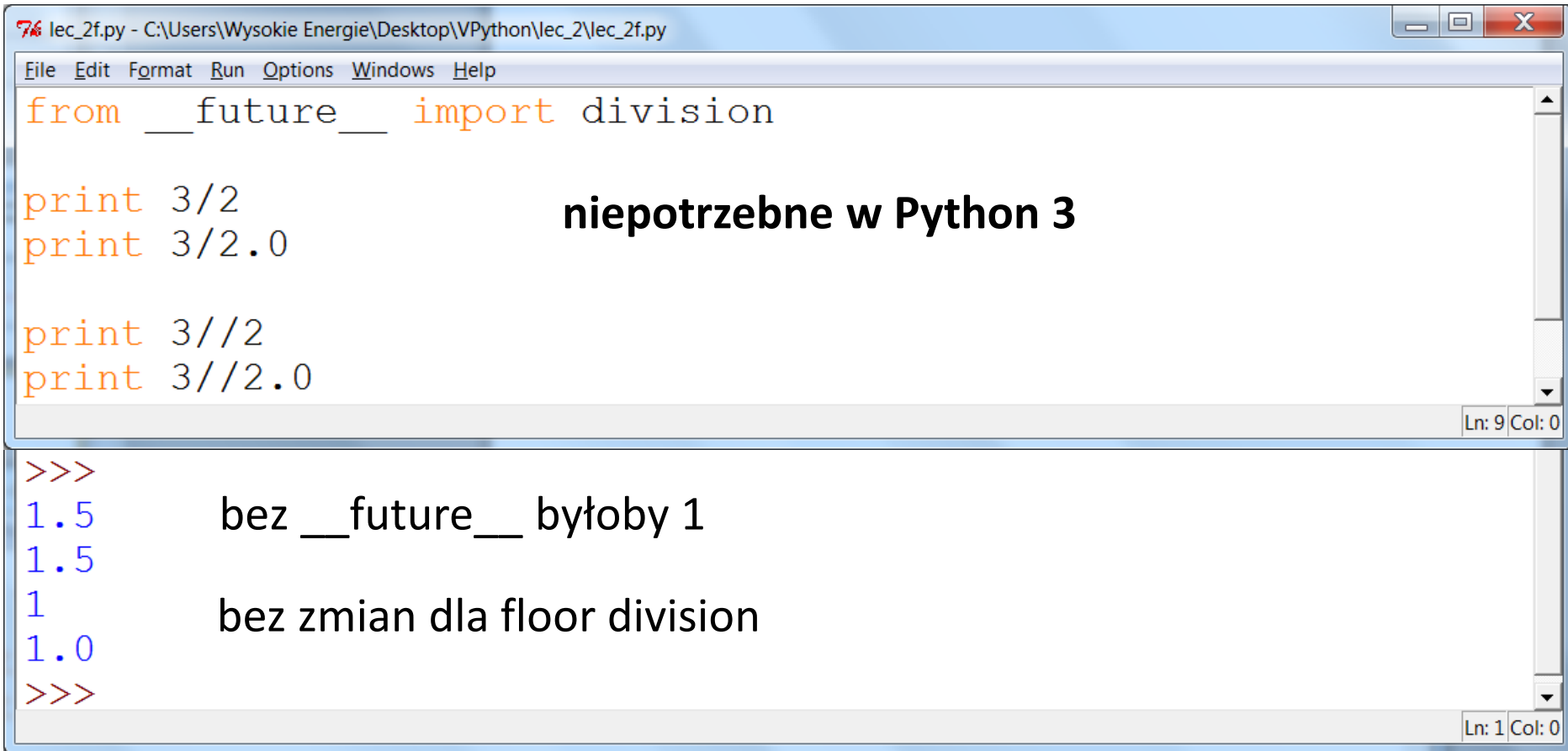
File "C:\Users\Wysokie Energie\Desktop\VPython\lec_2\main2.py", line 8, in <module>

print a

NameError: name 'a' is not defined

Ln: 12 Col: 4

moduł `__future__` wygodne w Python 2.7



The screenshot shows a Python 2.7 IDE window titled 'lec_2f.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2f.py'. The code editor contains the following code:

```
from __future__ import division

print 3/2
print 3/2.0

print 3//2
print 3//2.0
```

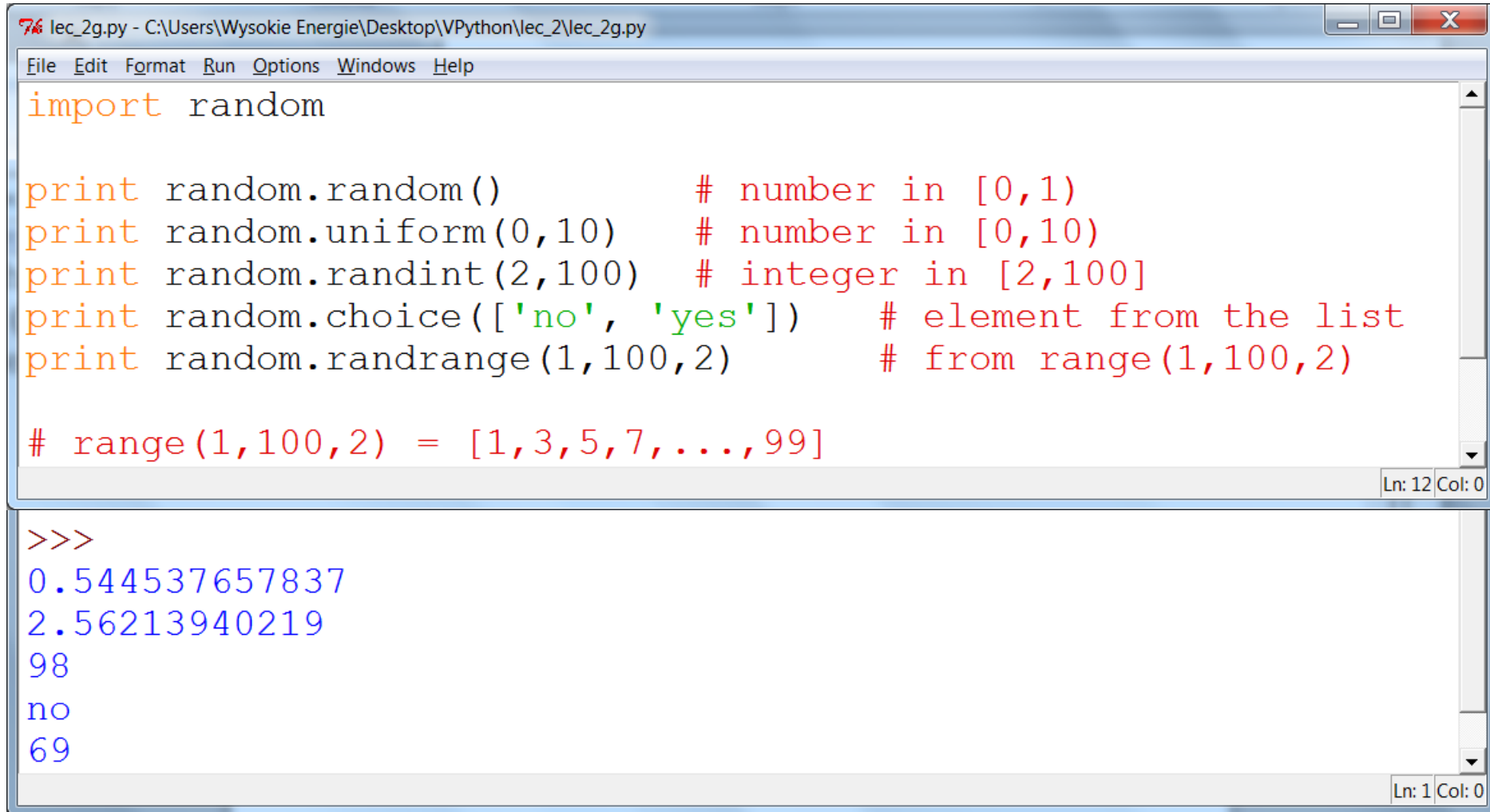
Below the code editor, the output of the program is shown in a console window:

```
>>>
1.5      bez __future__ byłoby 1
1.5
1        bez zmian dla floor division
1.0
>>>
```

The text 'niepotrzebne w Python 3' (not needed in Python 3) is written in the background of the code editor.

mając Python 2.7 zaczynam każdy program od importowania `__future__`. Musi to być pierwsza linia w programie.

Liczby losowe (random numbers)



The image shows a screenshot of a Python IDE window titled 'lec_2g.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2g.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 random

print random.random()          # number in [0,1)
print random.uniform(0,10)     # number in [0,10)
print random.randint(2,100)    # integer in [2,100]
print random.choice(['no', 'yes']) # element from the list
print random.randrange(1,100,2) # from range(1,100,2)

# range(1,100,2) = [1,3,5,7,...,99]
```

The status bar at the bottom right of the text area shows 'Ln: 12 Col: 0'. Below the text area is a console window showing the output of the code:

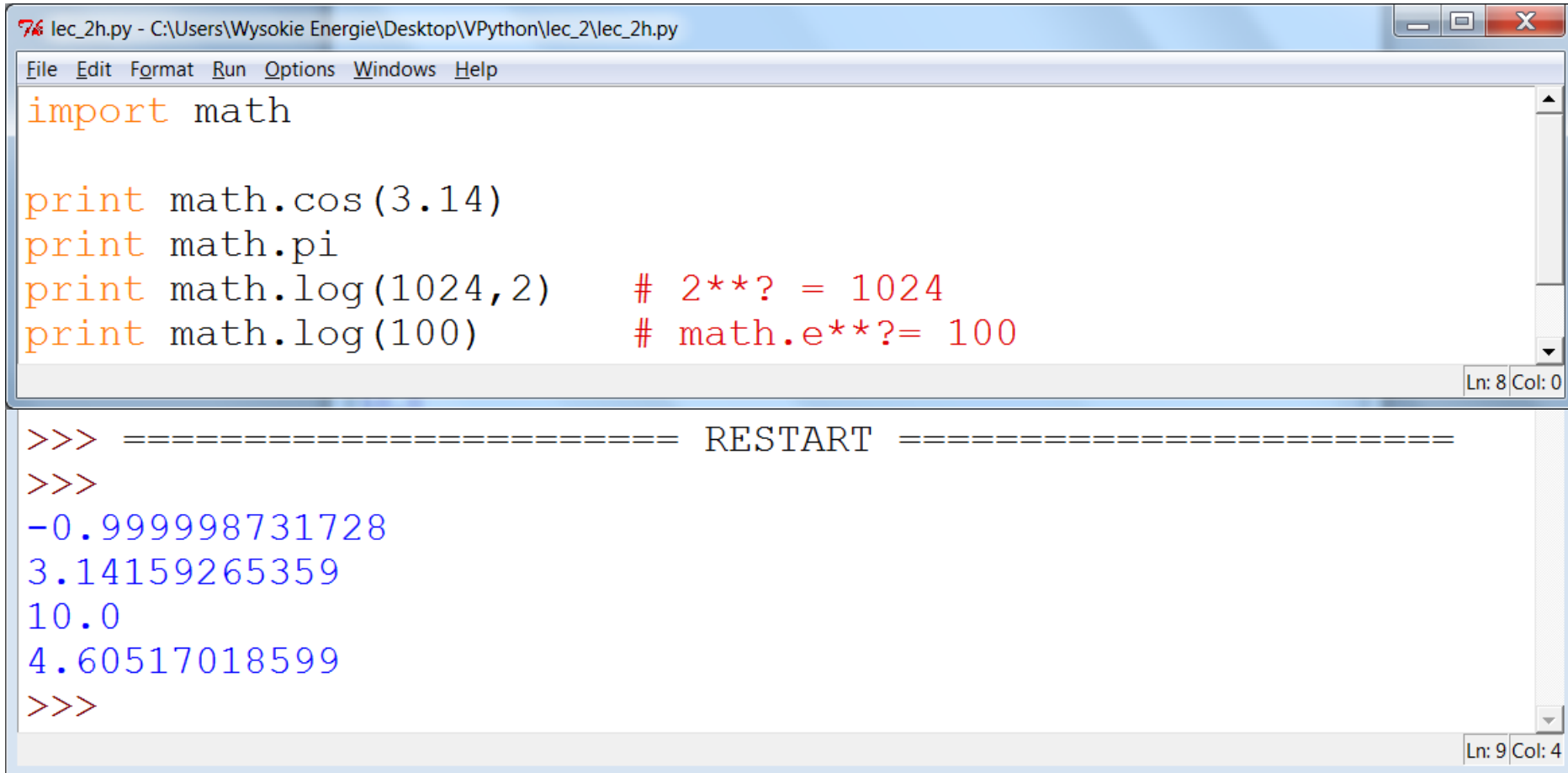
```
>>>
0.544537657837
2.56213940219
98
no
69
```

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

Proszę poczytać <https://docs.python.org/2/library/random.html>

Więcej rozkładów, Mersenne Twister, itp.

Moduł math



The screenshot shows a Python IDE window titled 'lec_2h.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2h.py'. The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The editor contains the following code:

```
import math

print math.cos(3.14)
print math.pi
print math.log(1024, 2)    # 2**? = 1024
print math.log(100)       # math.e**?= 100
```

The status bar at the bottom right of the editor shows 'Ln: 8 Col: 0'. Below the editor is a console window showing the execution output:

```
>>> ===== RESTART =====
>>>
-0.999998731728
3.14159265359
10.0
4.60517018599
>>>
```

The status bar at the bottom right of the console shows 'Ln: 9 Col: 4'.

Dla funkcji na liczbach zespolonych, patrz **cmath**.

Więcej informacji <https://docs.python.org/2/library/math.html>

Co jest w math module?



```
lec_2j.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2j.py
File Edit Format Run Options Windows Help

import math

print dir(math)

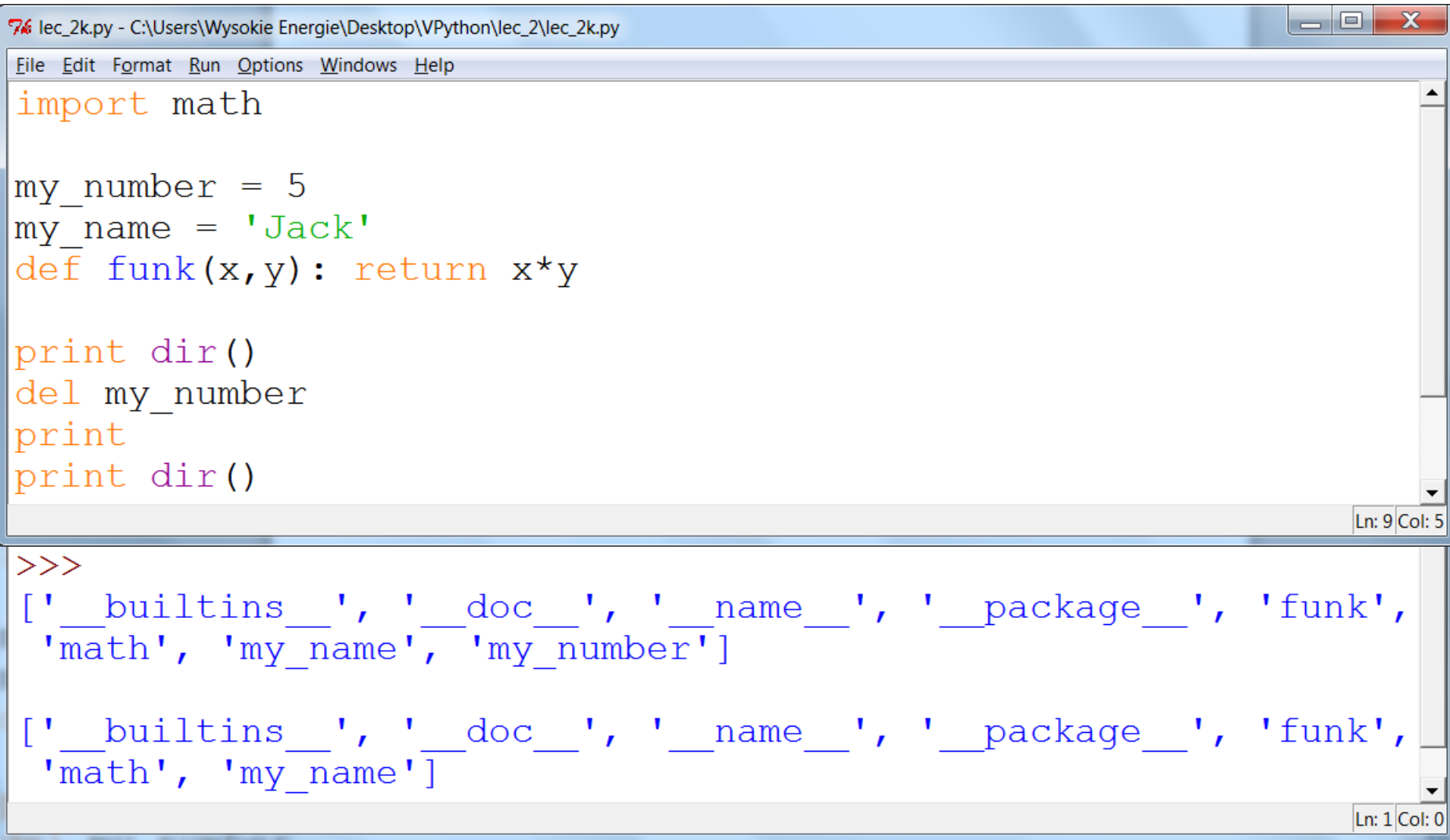
Ln: 6 Col: 0

>>> ===== RESTART =====
>>>
['__doc__', '__name__', '__package__', 'acos', 'acosh', 'asin',
 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos',
 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs',
 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'hypot',
 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'loglp',
 'modf', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan',
 'tanh', 'trunc']
>>>

Ln: 1 Col: 0
```

Proszę spróbować `dir(random)`, `dir(__future__)`, najpierw musimy je zaimportować

dir() – moduły, funkcje itp. w programie



The image shows a screenshot of a Python IDE window. The top pane contains a Python script named 'lec_2k.py' located at 'C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2k.py'. The script defines a function 'funk', assigns a value to 'my_number', and prints the directory of the current namespace before and after deleting 'my_number'. The bottom pane shows the execution output, which is a list of objects in the namespace before and after the deletion.

```
lec_2k.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2k.py
File Edit Format Run Options Windows Help

import math

my_number = 5
my_name = 'Jack'
def funk(x,y): return x*y

print dir()
del my_number
print
print dir()

>>>
['__builtins__', '__doc__', '__name__', '__package__', 'funk',
'math', 'my_name', 'my_number']

['__builtins__', '__doc__', '__name__', '__package__', 'funk',
'math', 'my_name']
```


List comprehensions

lec_2i.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2i.py

File Edit Format Run Options Windows Help

```
import math
import random

print [x/10. for x in range(0,10)]

print [[x,y] for x in range(5) for y in range(5) if x>y
        and x>=2 and y>=2]

print [math.log(x) for x in range(1,4)]

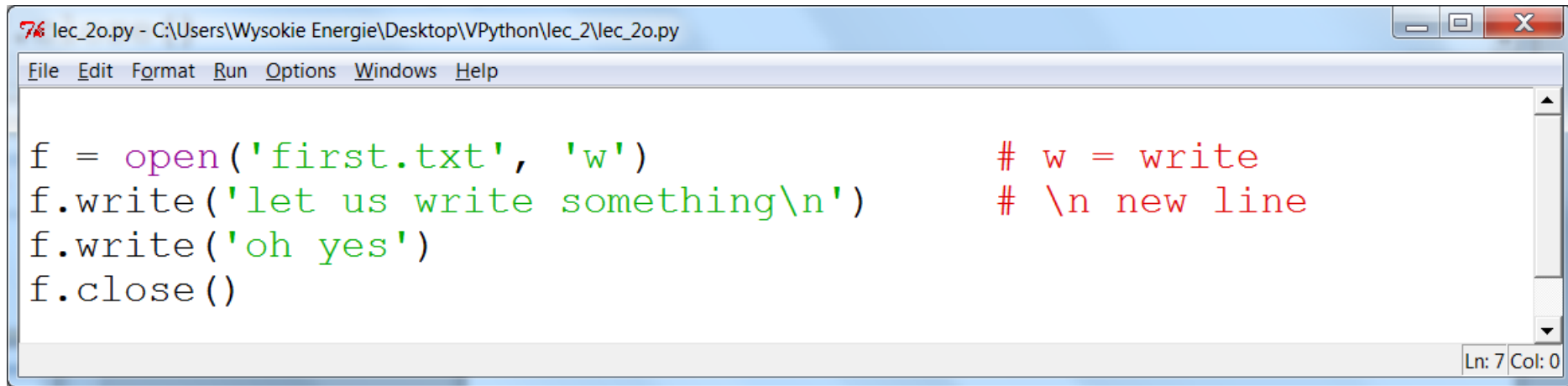
L = [random.random() for i in range(2)]
print L
```

Ln: 14 Col: 0

```
>>>
[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
[[3, 2], [4, 2], [4, 3]]
[0.0, 0.6931471805599453, 1.0986122886681098]
[0.8050451543490876, 0.8453338705541324]
```

Ln: 1 Col: 0

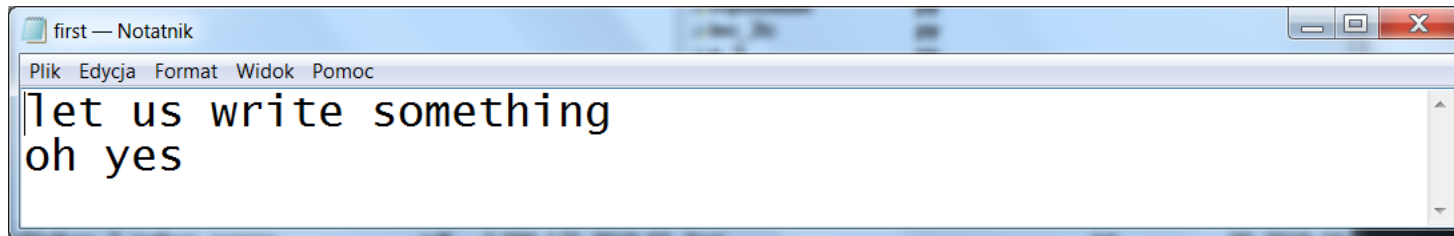
Zapisywanie plików



The screenshot shows a window titled "lec_2o.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2o.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code in the editor is as follows:

```
f = open('first.txt', 'w')  
f.write('let us write something\n')  
f.write('oh yes')  
f.close()
```

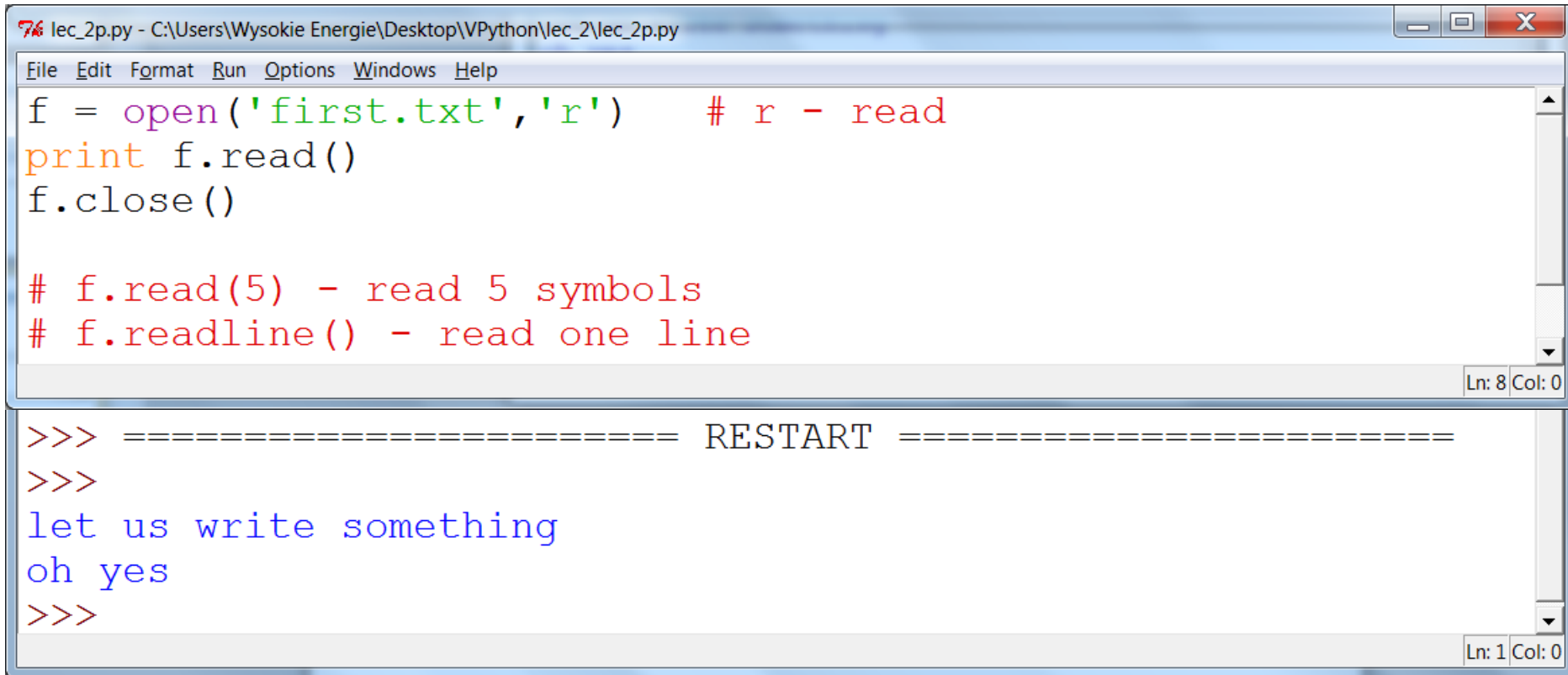
Comments on the right side of the code explain the parameters: "# w = write" and "# \n new line". The status bar at the bottom right indicates "Ln: 7 Col: 0".



The screenshot shows a window titled "first — Notatnik". The menu bar includes Plik, Edycja, Format, Widok, and Pomoc. The text in the editor is:

```
let us write something  
oh yes
```

Czytanie plików



The screenshot shows a Python IDE window titled 'lec_2p.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2p.py'. The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The editor contains the following Python code:

```
f = open('first.txt', 'r')    # r - read
print f.read()
f.close()

# f.read(5) - read 5 symbols
# f.readline() - read one line
```

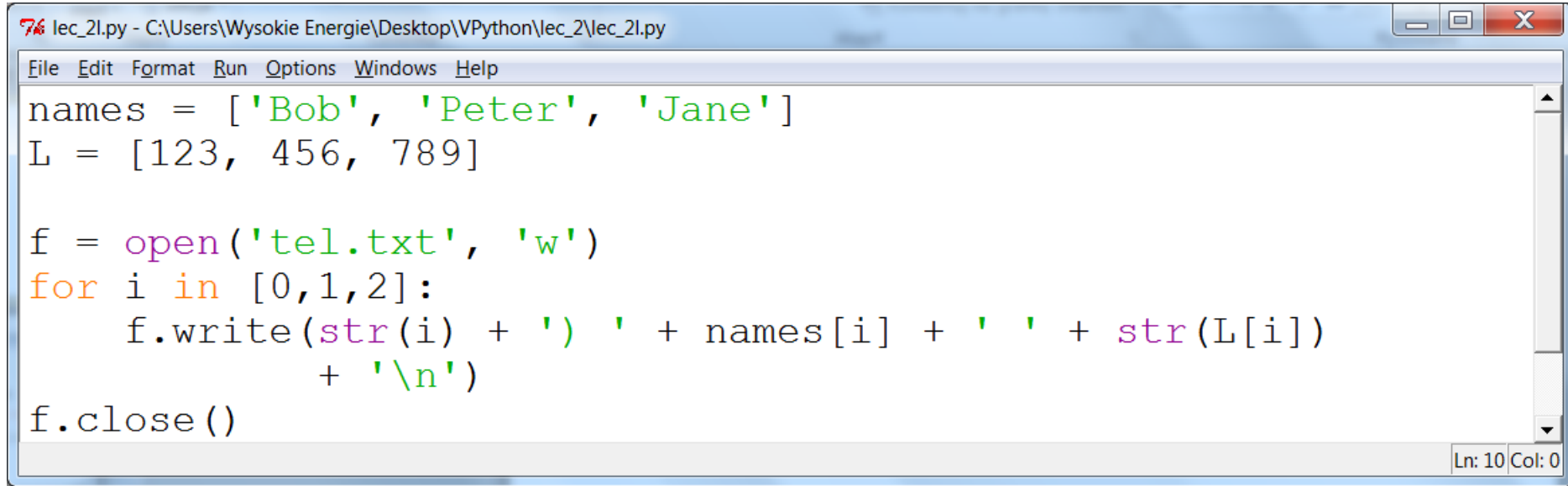
The status bar at the bottom right of the editor shows 'Ln: 8 Col: 0'. Below the editor is a REPL window with the following text:

```
>>> ===== RESTART =====
>>>
let us write something
oh yes
>>>
```

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

w write
r read
a append

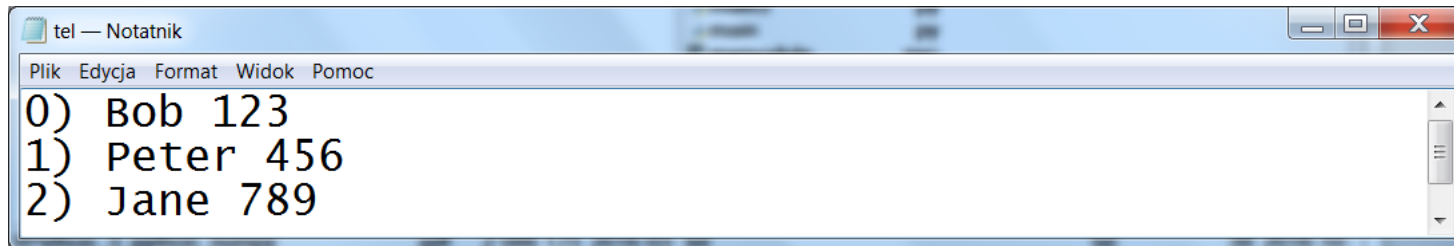
przykład



```
lec_2l.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2l.py
File Edit Format Run Options Windows Help
names = ['Bob', 'Peter', 'Jane']
L = [123, 456, 789]

f = open('tel.txt', 'w')
for i in [0,1,2]:
    f.write(str(i) + ' ' + names[i] + ' ' + str(L[i])
            + '\n')
f.close()
```

Ln: 10 Col: 0



```
tel — Notatnik
Plik Edycja Format Widok Pomoc
0) Bob 123
1) Peter 456
2) Jane 789
```

str(x) zamienia x na string

przykład

```
lec_2n.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2n.py
File Edit Format Run Options Windows Help

names = []
L = []

f = open('tel.txt', 'r')
for i in f:
    temp = i.split()
    print temp
    names.append(temp[1])
    L.append(int(temp[2]))
f.close()

print '\n', names, L
```

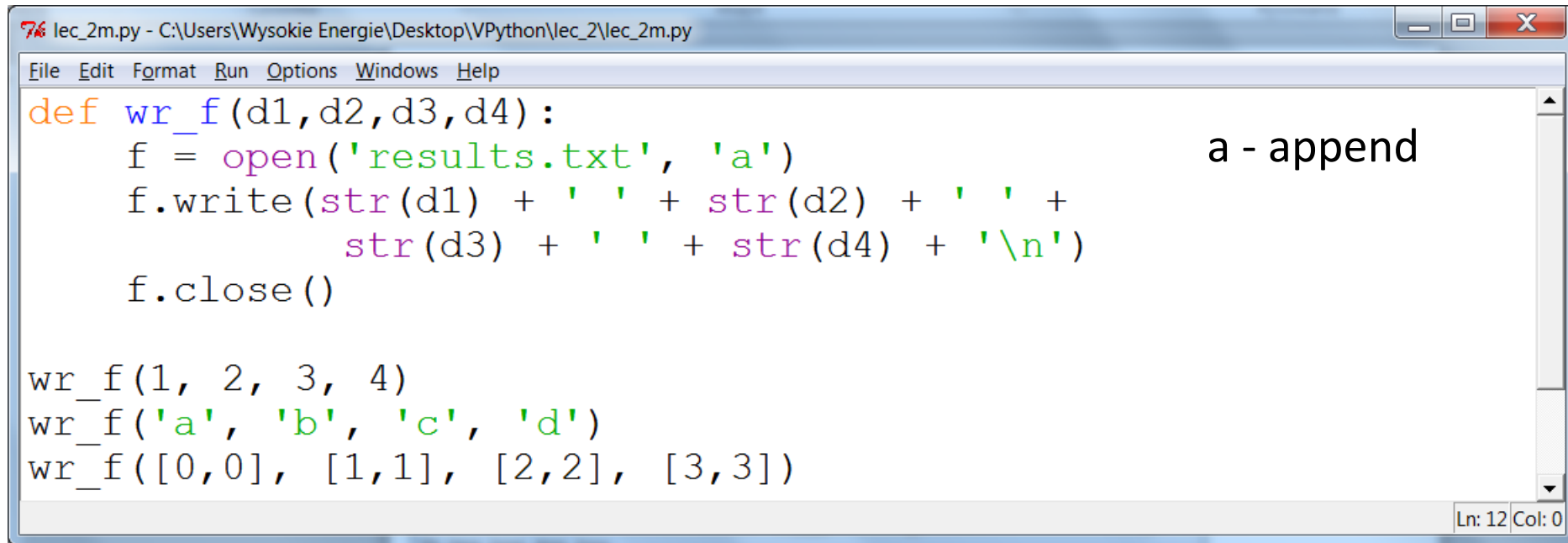
split dzieli na części

int(x) zamienia x na liczbę całkowitą

```
>>>
['0)', 'Bob', '123']
['1)', 'Peter', '456']
['2)', 'Jane', '789']

['Bob', 'Peter', 'Jane'] [123, 456, 789]
```

append



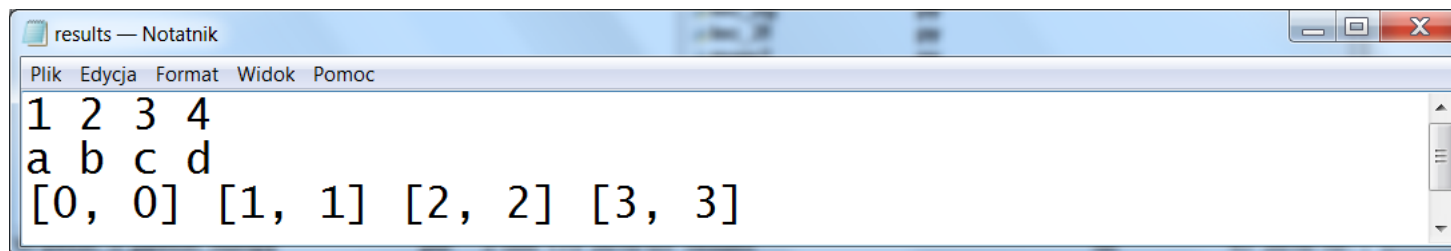
```
lec_2m.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_2\lec_2m.py
File Edit Format Run Options Windows Help

def wr_f(d1,d2,d3,d4):
    f = open('results.txt', 'a')
    f.write(str(d1) + ' ' + str(d2) + ' ' +
            str(d3) + ' ' + str(d4) + '\n')
    f.close()

wr_f(1, 2, 3, 4)
wr_f('a', 'b', 'c', 'd')
wr_f([0,0], [1,1], [2,2], [3,3])
```

a - append

Ln: 12 Col: 0



```
results — Notatnik
Plik Edycja Format Widok Pomoc

1 2 3 4
a b c d
[0, 0] [1, 1] [2, 2] [3, 3]
```

PEP 8 – Style Guide for Python Code

- 4 spacje na wcięcie
- Spacje są lepsze niż tabs
- Linie nie dłuższe niż 79 symboli
- docstrings
- function_names(x)
- spacje, e.g., `a = f(1, 2) + g(3, 4)`
- Nie używać `l` (małe L), `I` (duże i) and `O` (duże o)

Więcej informacji

<https://www.python.org/dev/peps/pep-0008/>