

VPython - symulacje fizyczne z grafiką 3D dla każdego

wykład 1

Dr hab. Adam Bzdak

Dr hab. Adam Bzdak

bzdak@fis.agh.edu.pl

pokój 112, budynek D-11

Konsultacje:

dowolny dzień, proszę o wsześniejszy e-mail

Wykłady będą pojawiać się na stronie

<https://sites.google.com/site/vpython2718pl/>

Laboratoria:

Poniedziałek, 14.40, pokój **207**, D-10

Piątek, 13:00, pokój **205**, D-10

W każdej grupie może być do 15 osób. Wyjątkowo może być troszkę więcej (np. odrabianie itp.) ale wtedy proszę przynieść swój komputer.

Każde laboratorium zaczynam od mini-wykładu gdzie omawiam zagadnienie. **Proszę być na czas.**

Wykład: czwartek, 9:40, sala 08, U-5

Wykłady są nieobowiązkowe (ale zaznajomienie się z materiałem z wykładu jest obowiązkowe).

Najważniejszą częścią kursu są laboratoria.

Będziemy pisać programy oparte tylko na materiale przerobionym na wykładzie.

Ocena końcowa oparta jest na ocenach cząstkowych za laboratoria.

Oceny cząstkowe: 2, 3, 3+, 4, 4+, 5.

Jak jestem zachwycony to 5+.

Jeśli ktoś nie odda programu na zajęciach to:

Jeśli zadanie jest w stanie zaawansowanym, tzn. kwalifikuje się na ocenę 4.0, to program (bez błędów) przysłany tego samego dnia do godziny 23.59 będę uważał za oddany na ćwiczeniach.

W tym przypadku ocena może zostać poprawiona na 5.0. Jeśli program nadejdzie później (do 4 dni) to maksymalna ocena 4.5.

Jeśli uznam, że zadanie kwalifikuje się na mniej niż 4.0 to traci się powyższy przywilej. Zadanie nadesłane do 4 dni podniesie ocenę o pół stopnia.

Proszę pamiętać, że jedyny cel tych zajęć to to aby się Państwo czegoś nauczyli.

Proszę się pytać o wszystko.

Własne komputery.

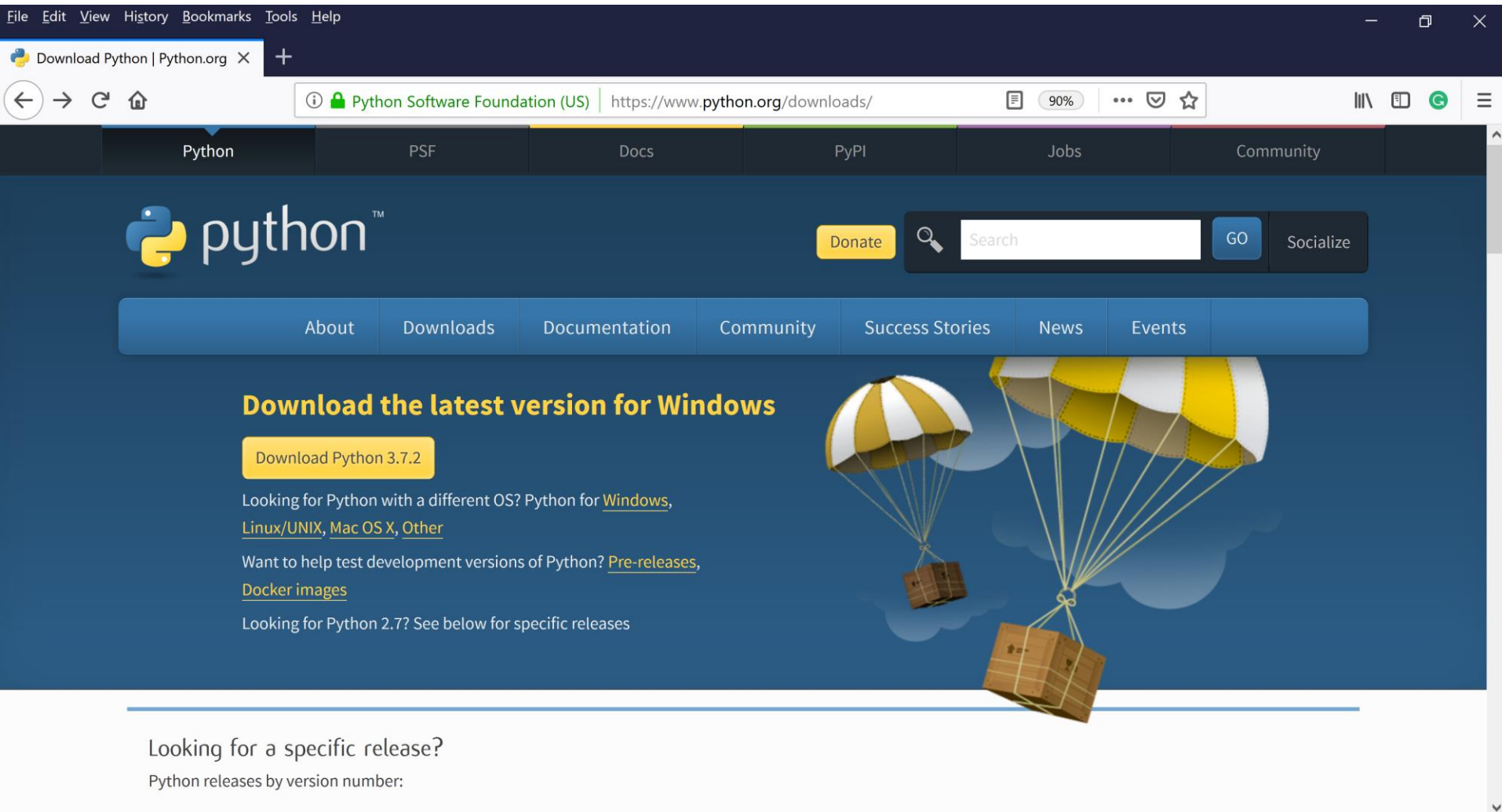
Plan:

- Python
- NumPy
- Matplotlib
- Vpython

przykłady ...

Python

<https://www.python.org/downloads/>



The screenshot shows a web browser window with the URL <https://www.python.org/downloads/>. The browser's address bar shows the Python Software Foundation (US) and the URL. The page features a dark blue header with the Python logo and navigation links: Python, PSF, Docs, PyPI, Jobs, and Community. Below the header is a search bar with a "Donate" button and a "Socialize" button. A horizontal menu contains links for About, Downloads, Documentation, Community, Success Stories, News, and Events. The main content area has a dark blue background with the text "Download the latest version for Windows" in yellow. Below this is a yellow button labeled "Download Python 3.7.2". To the right of the text is an illustration of two parachutes carrying boxes. The text continues: "Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)". It then says: "Want to help test development versions of Python? [Pre-releases](#), [Docker images](#)". Finally, it says: "Looking for Python 2.7? See below for specific releases". At the bottom, there is a link: "Looking for a specific release? Python releases by version number:".

Python

PSF

Docs

PyPI

Jobs

Community

python™

Donate

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

Download the latest version for Windows

Download Python 3.7.2

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

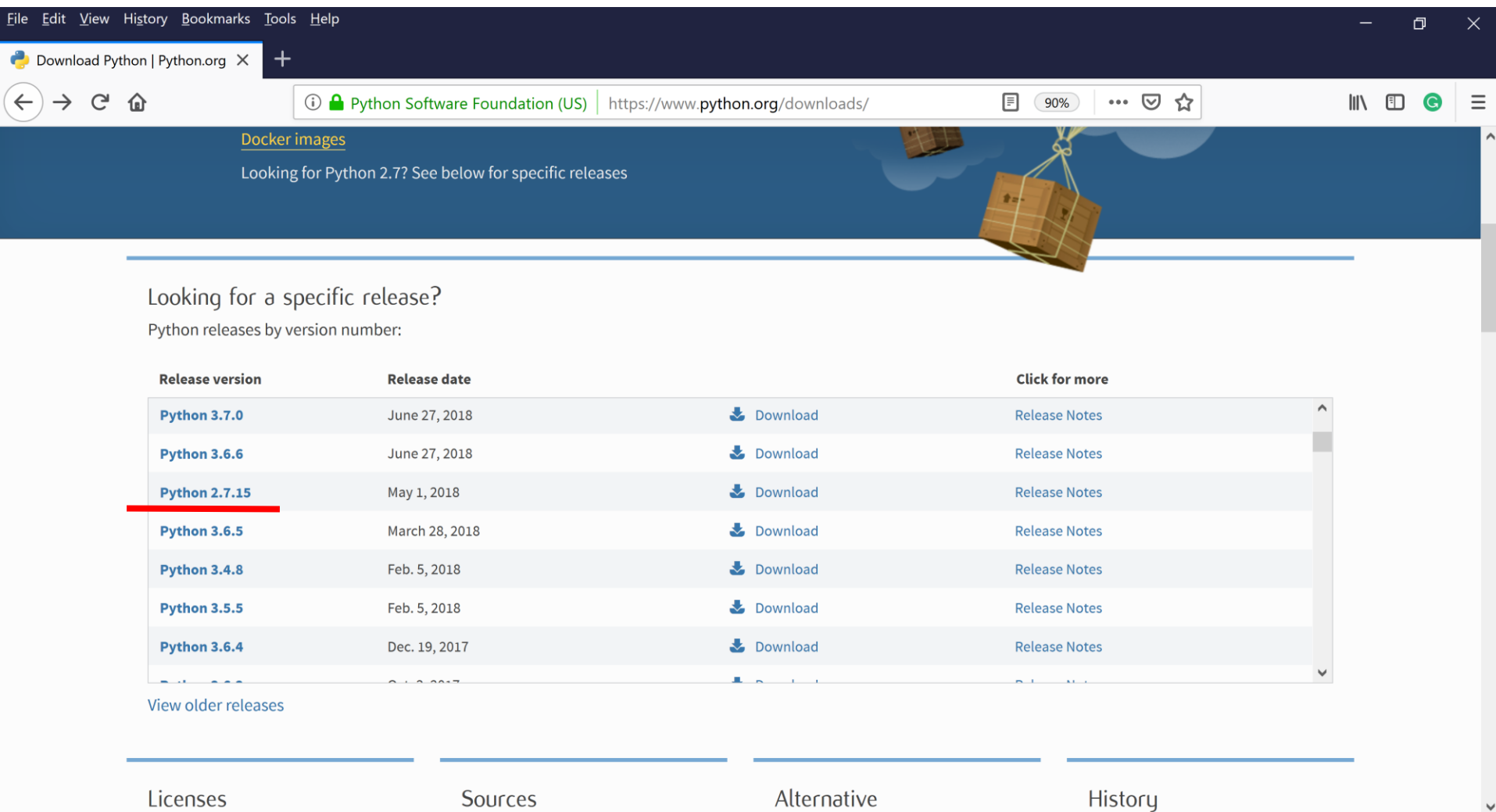
Want to help test development versions of Python? [Pre-releases](#), [Docker images](#)

Looking for Python 2.7? See below for specific releases

Looking for a specific release?
Python releases by version number:

i klikamy: Download Python 3.7.2

Python 2.7 jest ponízej strony



The screenshot shows the Python.org download page. The browser address bar displays 'https://www.python.org/downloads/'. The page header includes a link to 'Docker images' and a note: 'Looking for Python 2.7? See below for specific releases'. The main content area is titled 'Looking for a specific release?' and 'Python releases by version number:'. It contains a table with columns for 'Release version', 'Release date', and 'Click for more'. The row for 'Python 2.7.15' is highlighted with a red underline. Below the table is a link to 'View older releases'. At the bottom, there are four tabs: 'Licenses', 'Sources', 'Alternative', and 'History'.

Release version	Release date	Click for more
Python 3.7.0	June 27, 2018	Download Release Notes
Python 3.6.6	June 27, 2018	Download Release Notes
Python 2.7.15	May 1, 2018	Download Release Notes
Python 3.6.5	March 28, 2018	Download Release Notes
Python 3.4.8	Feb. 5, 2018	Download Release Notes
Python 3.5.5	Feb. 5, 2018	Download Release Notes
Python 3.6.4	Dec. 19, 2017	Download Release Notes

[View older releases](#)

[Licenses](#) [Sources](#) [Alternative](#) [History](#)

Python 2 lub 3?

<https://wiki.python.org/moin/Python2orPython3>

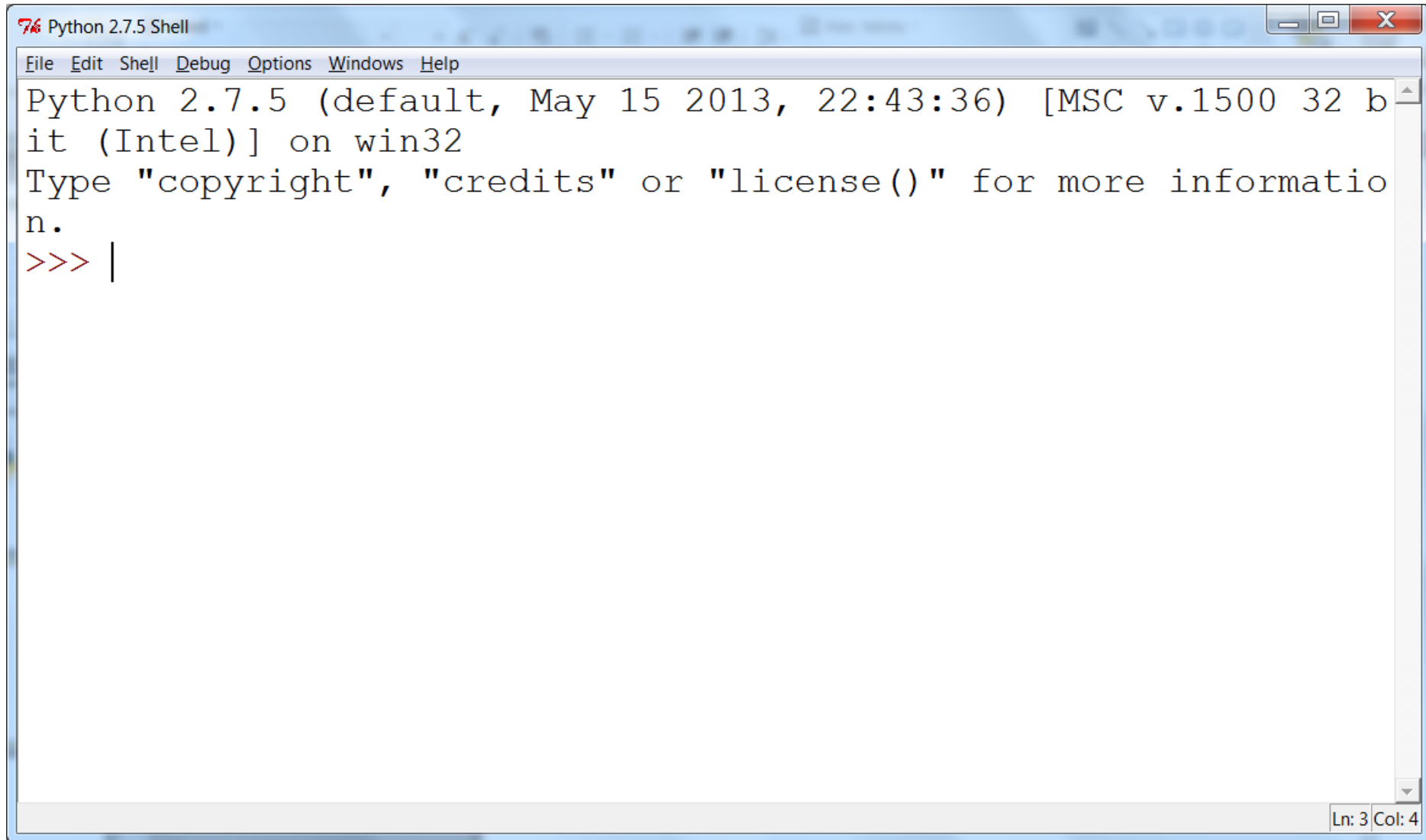
Niech Python zainstaluje się tam gdzie chce (proszę nie kombinować)

Na pulpicie pojawi się ikona



Klikamy i można bawić się Pythonem

Interpreter Pythona jako kalkulator



```
Python 2.7.5 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> |
```

Zamykamy przez **Control-D** lub piszemy **quit()** i enter

Proste operacje

```
Python 2.7.5 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> 2+2
4
>>> 2*3
6
>>> 2**10      # the power operator. This is a comment
1024
>>> 5/2
2
>>> 5/2.0
2.5
>>> 5//2
2
>>> 5//2.0
2.0
>>> -5//2.0
-3.0
>>>
```

5/2 = 2.5 w Python 3

// to jest floor division, liczby po kropce są usunięte (w kierunku $-\infty$)

Ln: 19 Col: 4

Liczby zespolone (complex numbers)

Python 2.7.5 Shell

File Edit Shell Debug Options Windows Help

Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32

Type "copyright", "credits" or "license()" for more information.

```
>>> a = 1 + 1j
```

```
>>> b = 2 + 5j
```

można używać J (duże j)

```
>>> c = a + b
```

```
>>> c
```

```
(3+6j)
```

```
>>> print c
```

print(c) w Python 3

```
(3+6j)
```

```
>>> c.real
```

```
3.0
```

część rzeczywista i urojona (real, imaginary)

```
>>> c.imag
```

```
6.0
```

```
>>> d = complex(1.5, -100)
```

```
>>> d
```

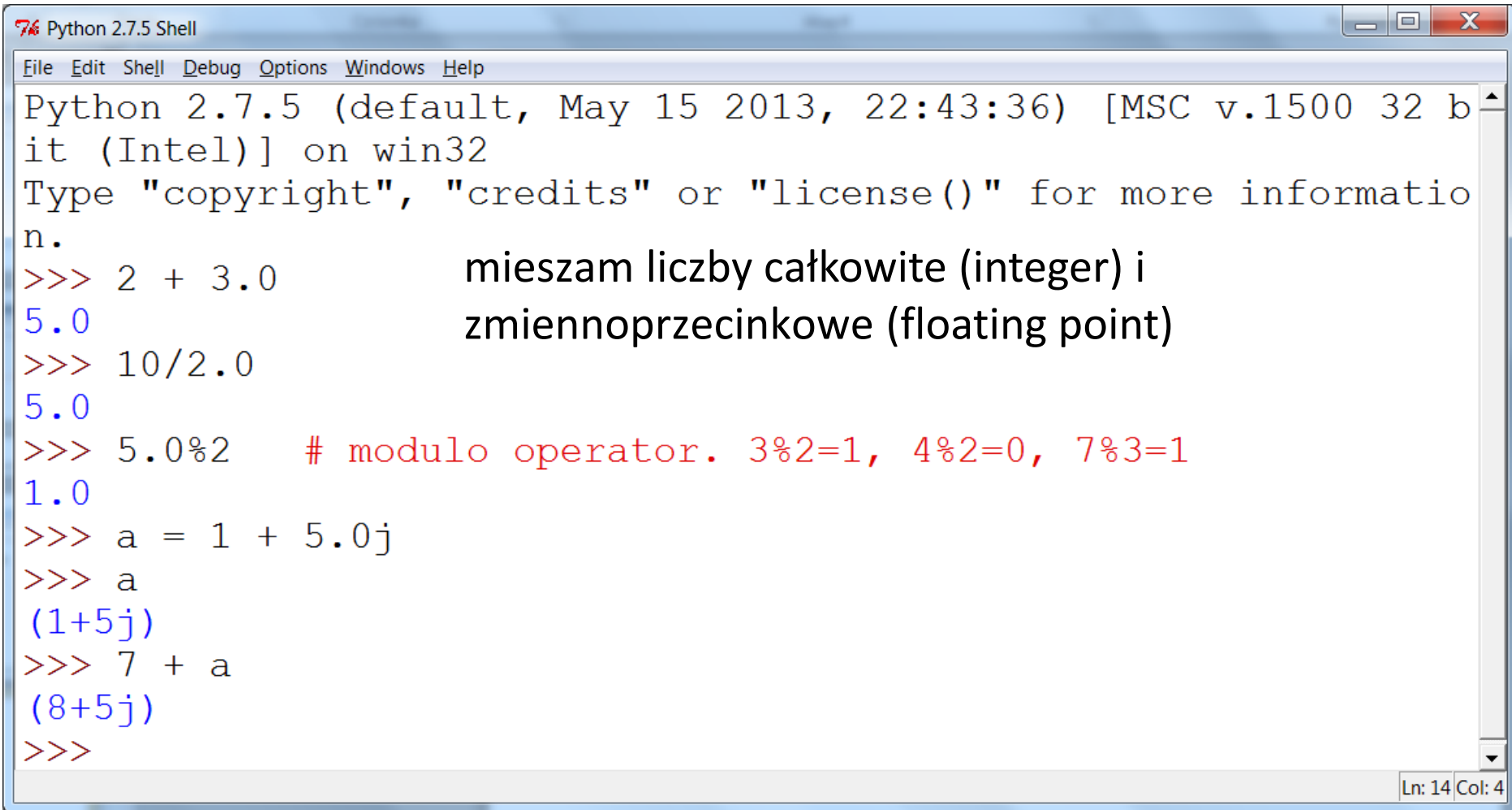
```
(1.5-100j)
```

```
>>> print 1j*1j
```

```
(-1+0j)
```

```
>>>
```

Możemy mieszać różne typy



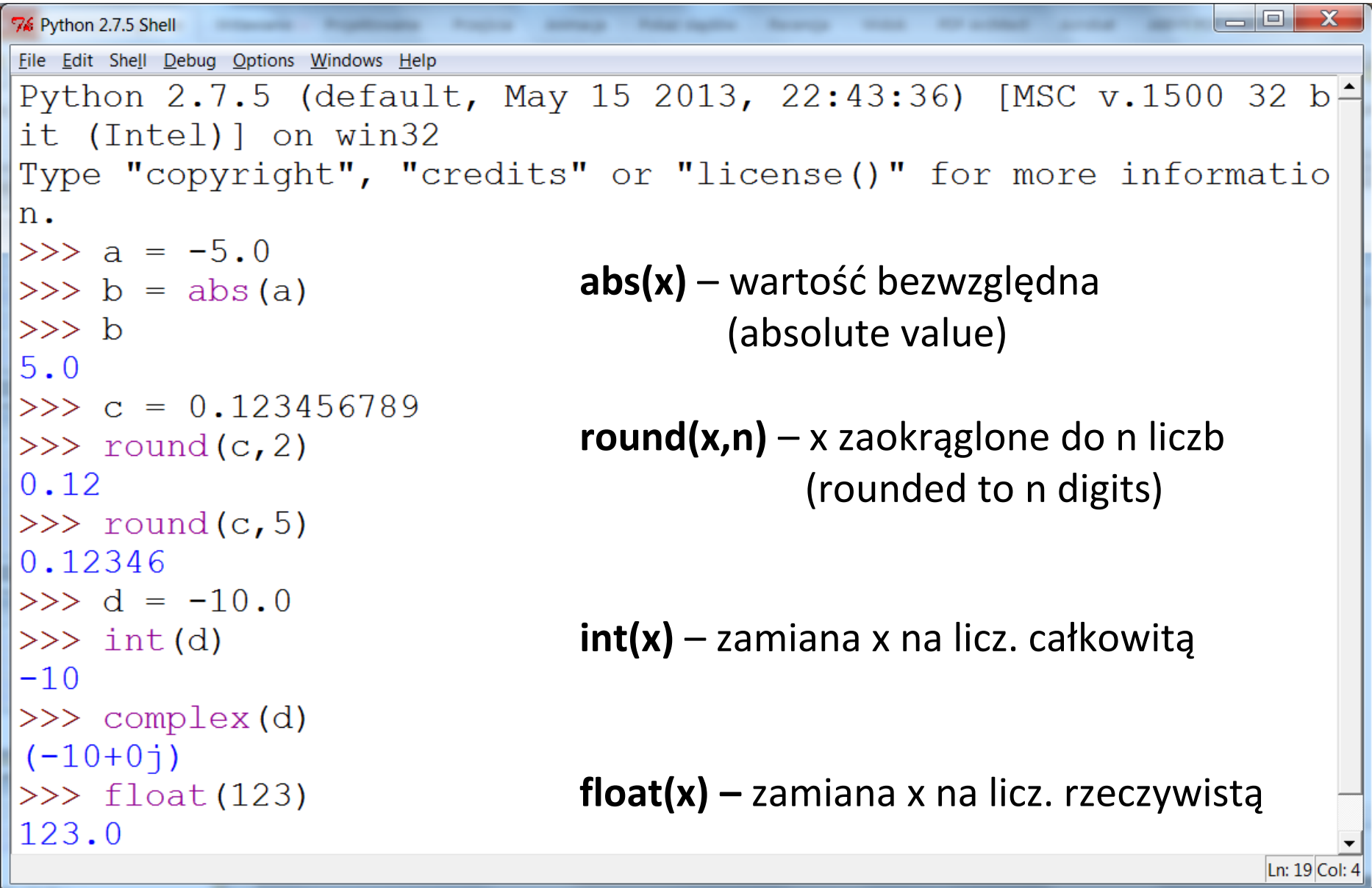
The screenshot shows a Python 2.7.5 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help) and a status bar (Ln: 14 Col: 4). The shell displays the following text:

```
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> 2 + 3.0
5.0
>>> 10/2.0
5.0
>>> 5.0%2      # modulo operator. 3%2=1, 4%2=0, 7%3=1
1.0
>>> a = 1 + 5.0j
>>> a
(1+5j)
>>> 7 + a
(8+5j)
>>>
```

Comments in Polish are placed to the right of the code:

- `mieszam liczby całkowite (integer) i` (I mix integer numbers and)
- `zmiennoprzecinkowe (floating point)` (floating point numbers)

Funkcje wbudowane (built-in)



The image shows a screenshot of a Python 2.7.5 Shell window. The window title is "Python 2.7.5 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area displays the following content:

```
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> a = -5.0
>>> b = abs(a)
>>> b
5.0
>>> c = 0.123456789
>>> round(c, 2)
0.12
>>> round(c, 5)
0.12346
>>> d = -10.0
>>> int(d)
-10
>>> complex(d)
(-10+0j)
>>> float(123)
123.0
```

Next to the code, the following descriptions are provided:

- abs(x)** – wartość bezwzględna (absolute value)
- round(x,n)** – x zaokrąglone do n liczb (rounded to n digits)
- int(x)** – zamiana x na licz. całkowitą
- float(x)** – zamiana x na licz. rzeczywistą

The status bar at the bottom right of the window shows "Ln: 19 Col: 4".

Stringi (strings)

Python 2.7.5 Shell

File Edit Shell Debug Options Windows Help

Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32

Type "copyright", "credits" or "license()" for more information.

```
>>> a = 'Hi'
```

```
>>> b = ' Bob'
```

```
>>> a + b
```

```
'Hi Bob'
```

```
>>> 'hi ' * 10
```

```
'hi hi hi hi hi hi hi hi hi hi '
```

```
>>> '-' * 30
```

```
'-----'
```

```
>>> s = 'abc'
```

```
>>> 'a' in s
```

```
True
```

```
>>> 'g' in s
```

```
False
```

```
>>> '.' * 30
```

```
'.....'
```

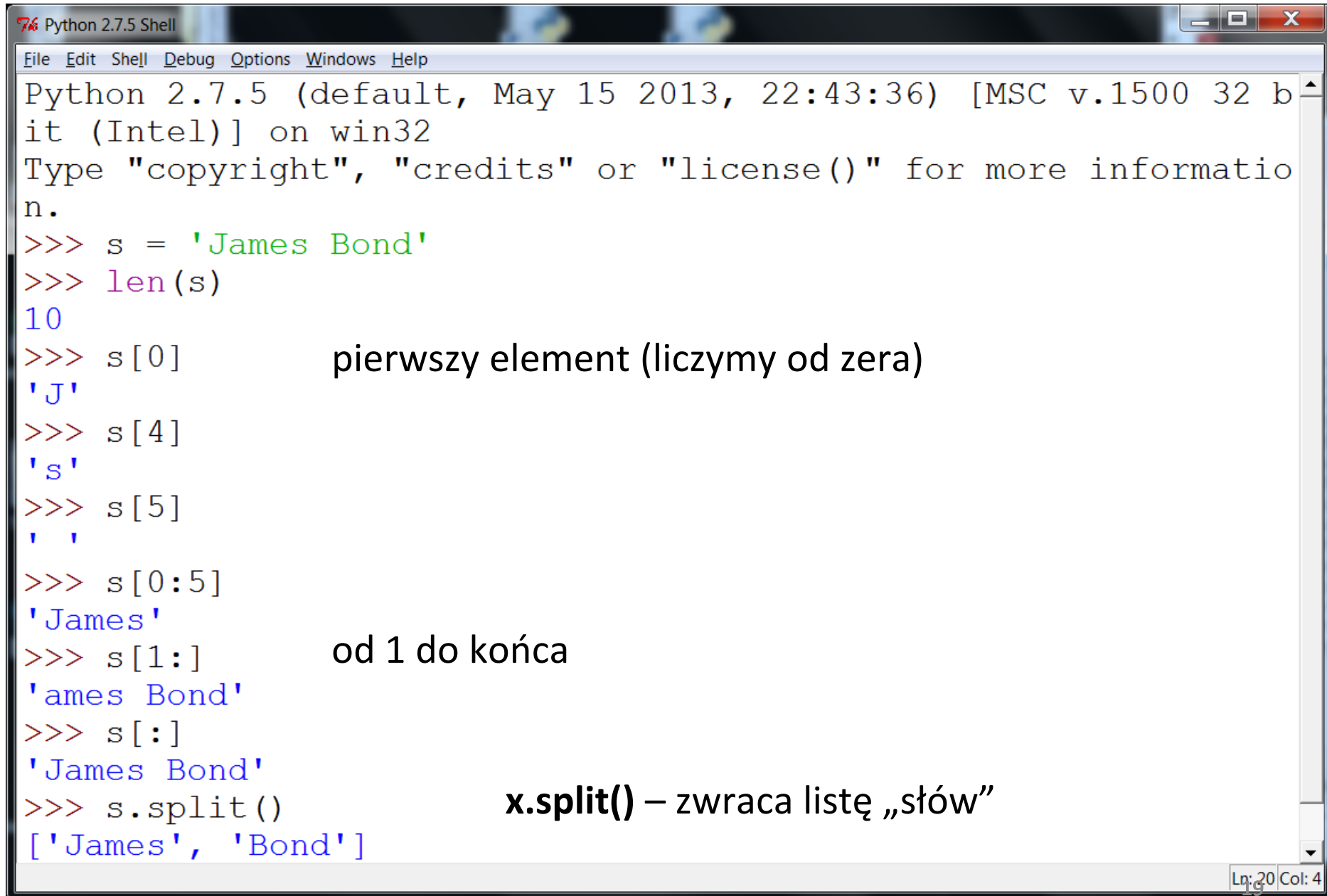
```
>>> len(s)
```

```
3
```

stringi można dodawać i mnożyć przez liczbę

len(x) – długość x (length)

Stringi



```
Python 2.7.5 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> s = 'James Bond'
>>> len(s)
10
>>> s[0]                pierwszy element (liczymy od zera)
'J'
>>> s[4]
's'
>>> s[5]
' '
>>> s[0:5]
'James'
>>> s[1:]               od 1 do końca
'ames Bond'
>>> s[:]
'James Bond'
>>> s.split()           x.split() – zwraca listę „słów”
['James', 'Bond']
```

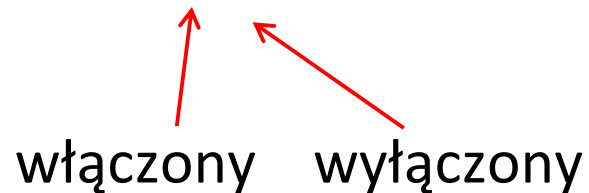
Ln: 20 Col: 4

Stringi

+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+
	P		y		t		h		o		n													
+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+	-	-	-	+
	0		1		2		3		4		5													
	-6		-5		-4		-3		-2		-1													

`s[2:5] = 'tho'`

włączony wyłączony



```
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 b
it (Intel)] on win32
```

```
Type "copyright", "credits" or "license()" for more informatio
n.
```

```
>>> s = 'Winnie-the-Pooh'
```

```
>>> len(s)
```

```
15
```

```
>>> s[0:2]
```

```
'Wi'
```

```
>>> s[2:]
```

```
'nnie-the-Pooh'
```

```
>>> s[0:2] + s[2:]
```

```
'Winnie-the-Pooh'
```

```
>>> s[-1]          ostatni element
```

```
'h'
```

```
>>> s[-2]          przedostatni element
```

```
'o'
```

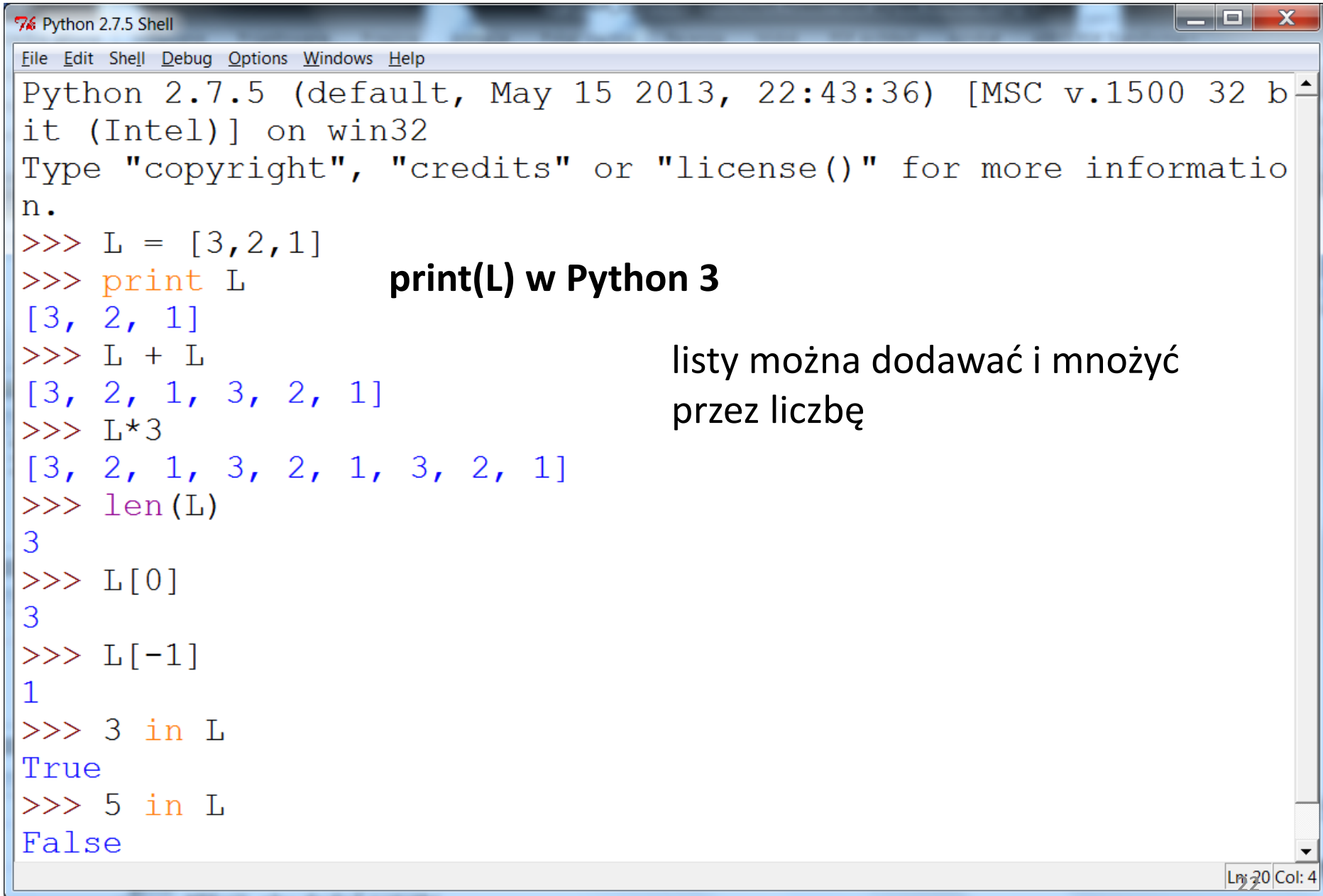
```
>>> s[-4:]
```

```
'Pooh'
```



$s = s[:i] + s[i:]$

Listy (lists)



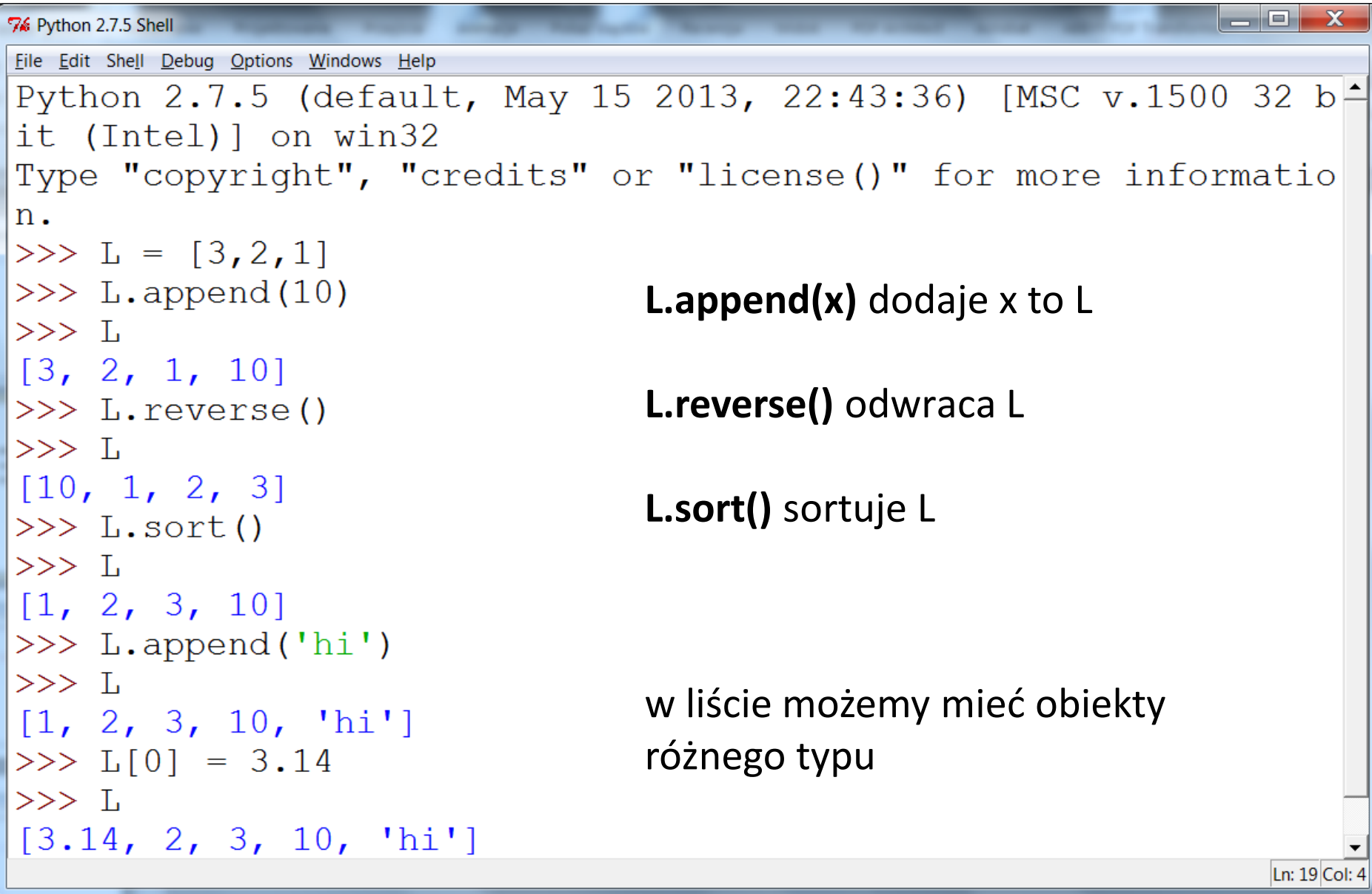
```
Python 2.7.5 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> L = [3,2,1]
>>> print L
[3, 2, 1]
>>> L + L
[3, 2, 1, 3, 2, 1]
>>> L*3
[3, 2, 1, 3, 2, 1, 3, 2, 1]
>>> len(L)
3
>>> L[0]
3
>>> L[-1]
1
>>> 3 in L
True
>>> 5 in L
False
```

print(L) w Python 3

listy można dodawać i mnożyć przez liczbę

Ln: 20 Col: 4

Listy

A screenshot of a Python 2.7.5 Shell window. The window has a title bar with the text 'Python 2.7.5 Shell' and standard Windows window controls (minimize, maximize, close). Below the title bar is a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Windows', and 'Help'. The main area of the window contains a series of Python commands and their outputs. The commands are: 'Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32', 'Type "copyright", "credits" or "license()" for more information.', '>>> L = [3,2,1]', '>>> L.append(10)', '>>> L', '[3, 2, 1, 10]', '>>> L.reverse()', '>>> L', '[10, 1, 2, 3]', '>>> L.sort()', '>>> L', '[1, 2, 3, 10]', '>>> L.append('hi')', '>>> L', '[1, 2, 3, 10, 'hi']', '>>> L[0] = 3.14', '>>> L', '[3.14, 2, 3, 10, 'hi']'. The status bar at the bottom right shows 'Ln: 19 Col: 4'.

```
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> L = [3,2,1]
>>> L.append(10)
>>> L
[3, 2, 1, 10]
>>> L.reverse()
>>> L
[10, 1, 2, 3]
>>> L.sort()
>>> L
[1, 2, 3, 10]
>>> L.append('hi')
>>> L
[1, 2, 3, 10, 'hi']
>>> L[0] = 3.14
>>> L
[3.14, 2, 3, 10, 'hi']
```

L.append(x) dodaje x to L

L.reverse() odwraca L

L.sort() sortuje L

w liście możemy mieć obiekty
różnego typu

Listy

```
Python 2.7.5 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> L = [7,5,3,1]
>>> L[1:3]
[5, 3]
>>> L[:]
[7, 5, 3, 1]
>>> Lnew = L[:]      powstaje nowy obiekt
>>> Lnew.append(22)
>>> L
[7, 5, 3, 1]
>>> Lnew
[7, 5, 3, 1, 22]
>>> Lnew[0:2] = []
>>> Lnew
[3, 1, 22]
```

Lnew = L to nie jest **Lnew = L[:]**

↑
dwie nazwy odnoszą się do tego samego obiektu

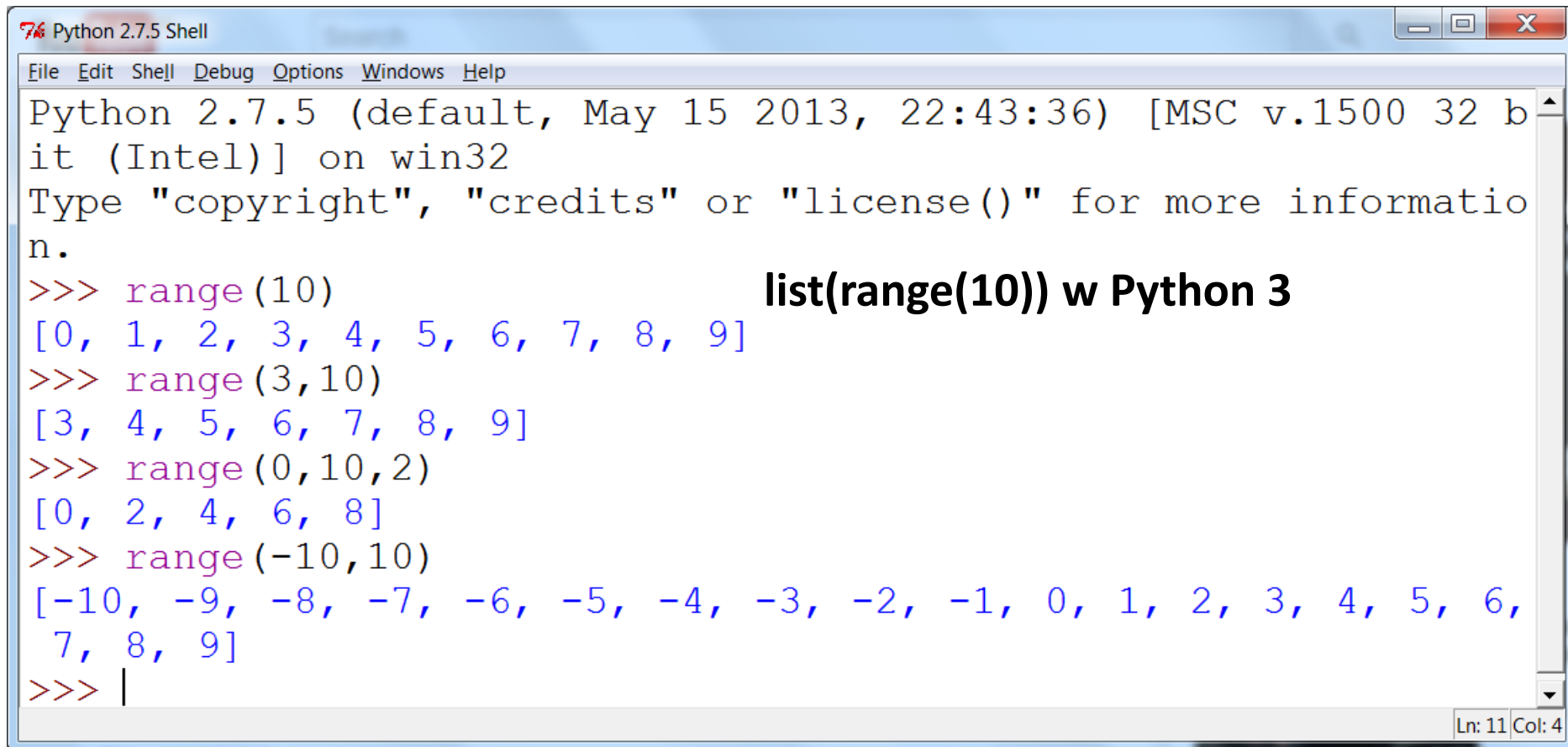
Ln: 17 Col: 4

Listy wielowymiarowe (multi-dimensional lists)

```
Python 2.7.5 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> L1 = [1,2,3]
>>> L2 = [10,20,30]
>>> L3 = ['a','b']
>>> F = [L1,L2,L3]
>>> F
[[1, 2, 3], [10, 20, 30], ['a', 'b']]
>>> F[0]
[1, 2, 3]
>>> F[2][0]
'a'
>>> F[0] = 'hi'
>>> F
['hi', [10, 20, 30], ['a', 'b']]
>>>
```

Ln: 16 Col: 4

range()



The screenshot shows a Python 2.7.5 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help) and a status bar (Ln: 11, Col: 4). The shell displays the Python version and system information, followed by instructions on how to get help. Below this, several examples of the range() function are shown, each followed by its output list. The examples include: range(10), range(3, 10), range(0, 10, 2), and range(-10, 10). The output for range(-10, 10) is truncated in the image.

```
Python 2.7.5 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.5 (default, May 15 2013, 22:43:36) [MSC v.1500 32 b
it (Intel)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> range(3, 10)
[3, 4, 5, 6, 7, 8, 9]
>>> range(0, 10, 2)
[0, 2, 4, 6, 8]
>>> range(-10, 10)
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6,
 7, 8, 9]
>>> |
```

list(range(10)) w Python 3

Interpreter jest OK dla prostych rzeczy

Do pisania programów otwieramy okno

File → New Window

lub

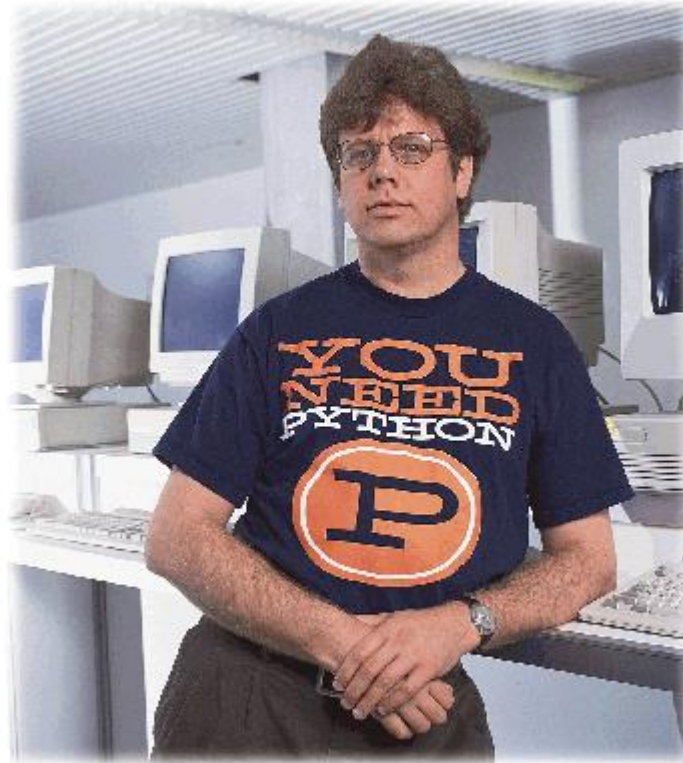
Ctrl+N

i zapisujemy plik z rozszerzeniem **.py**

Na przykład `calkowanie.py`

Aby uruchomić program naciskamy **F5**

Twórca Pythona: **Guido van Rossum**



https://en.wikipedia.org/wiki/Guido_van_Rossum

FileEditViewHistoryBookmarksToolsHelp

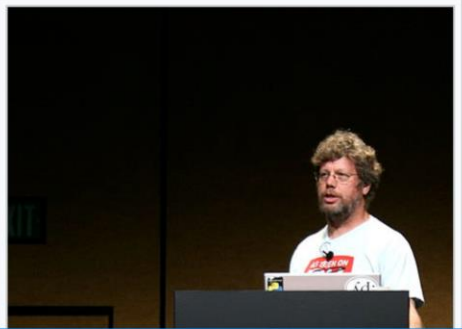
W Guido van Rossum - Wikipedia

←→↺🏠

https://en.wikipedia.org/wiki/Guido_van_Rossum133%

Python [edit]

In December 1989, Van Rossum had been looking for a "'hobby' programming project that would keep [him] occupied during the week around Christmas" as his office was closed when he decided to write an [interpreter](#) for a "new scripting language [he] had been thinking about lately: a descendant of [ABC](#) that would appeal to [Unix/C](#) hackers". He attributes choosing the name "Python" to "being in a slightly irreverent mood (and a big fan of [Monty Python's Flying Circus](#))".^[18]




←→↺🏠

https://en.wikipedia.org/wiki/Guido_van_Rossum133%

- An easy and intuitive language just as powerful as major competitors
- [Open source](#), so anyone can contribute to its development
- Code that is as understandable as plain English
- Suitability for everyday tasks, allowing for short development times

Python has grown to become a popular programming language. As of October 2017, it was the second most popular language on [GitHub](#), a social coding website, behind [JavaScript](#) and ahead of [Java](#).^[21] According to a programming language popularity survey^[22] it is consistently amongst the top 10 most mentioned languages in job postings. Furthermore, Python is



29

Raw_input, type, if, else, elif

lec_1a.py - C:\Users\Wysokie Energie\Desktop\Vpython\lec_1a.py

File Edit Format Run Options Windows Help

```
x = raw_input('Type a number: ')    input() w Python 3
print x
print type(x)                      sprawdzamy typ
```

```
x = int(x)                         zamieniamy string na liczbę całkowitą
print type(x)
```

```
if x==1:
    print 'x = 1'
elif x>1:    # else if
    print 'x > 1'
else:
    print 'x < 1'
```

wcięcia (indentation) !

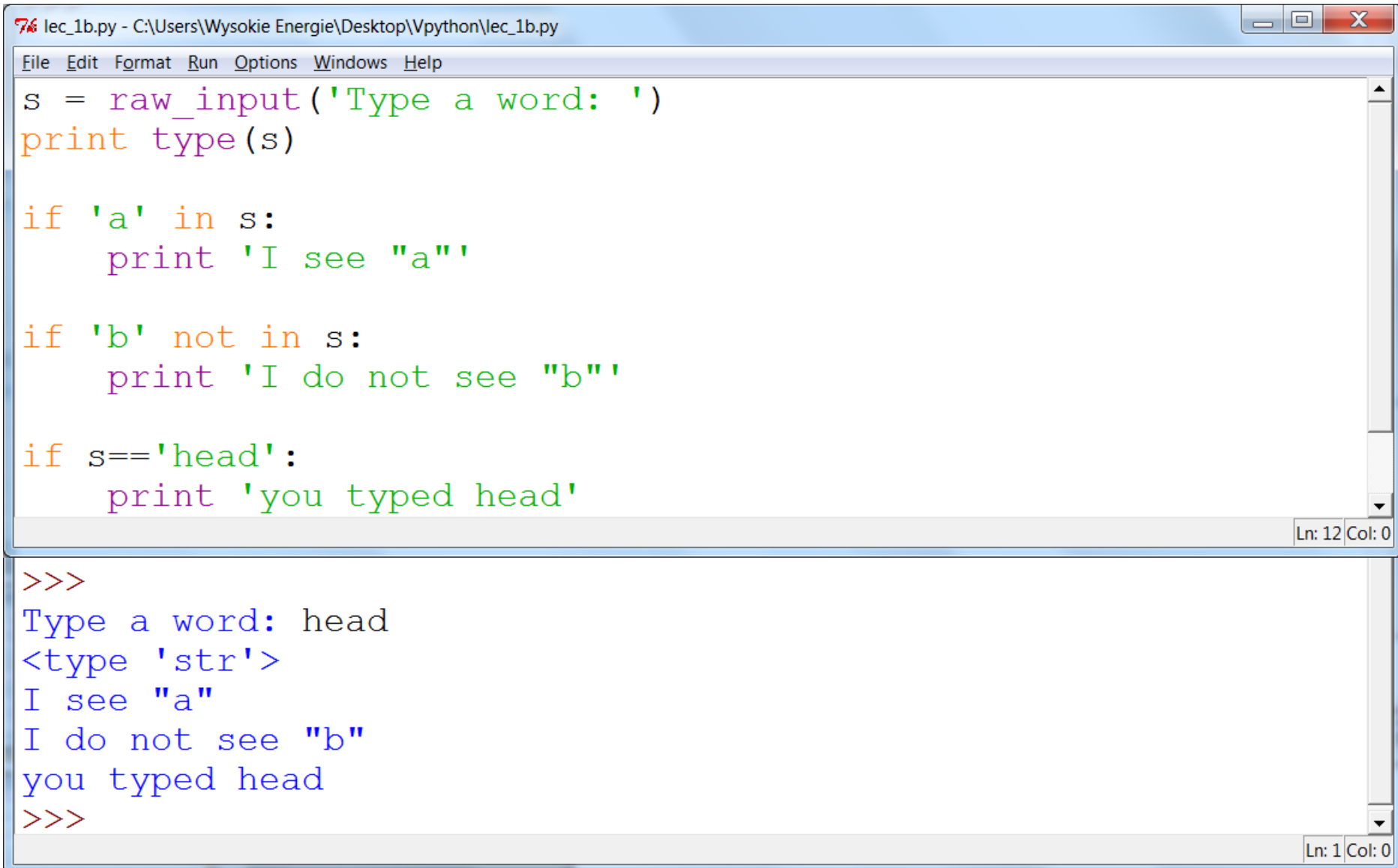


Ln: 15 Col: 0

```
Type a number: 5
5
<type 'str'>
<type 'int'>
x > 1
>>>
```

Ln: 1 Col: 0

If



The image shows a screenshot of a Python IDE window. The title bar reads 'lec_1b.py - C:\Users\Wysokie Energie\Desktop\Vpython\lec_1b.py'. The menu bar includes 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The main text area contains the following Python code:

```
s = raw_input('Type a word: ')
print type(s)

if 'a' in s:
    print 'I see "a"'

if 'b' not in s:
    print 'I do not see "b"'

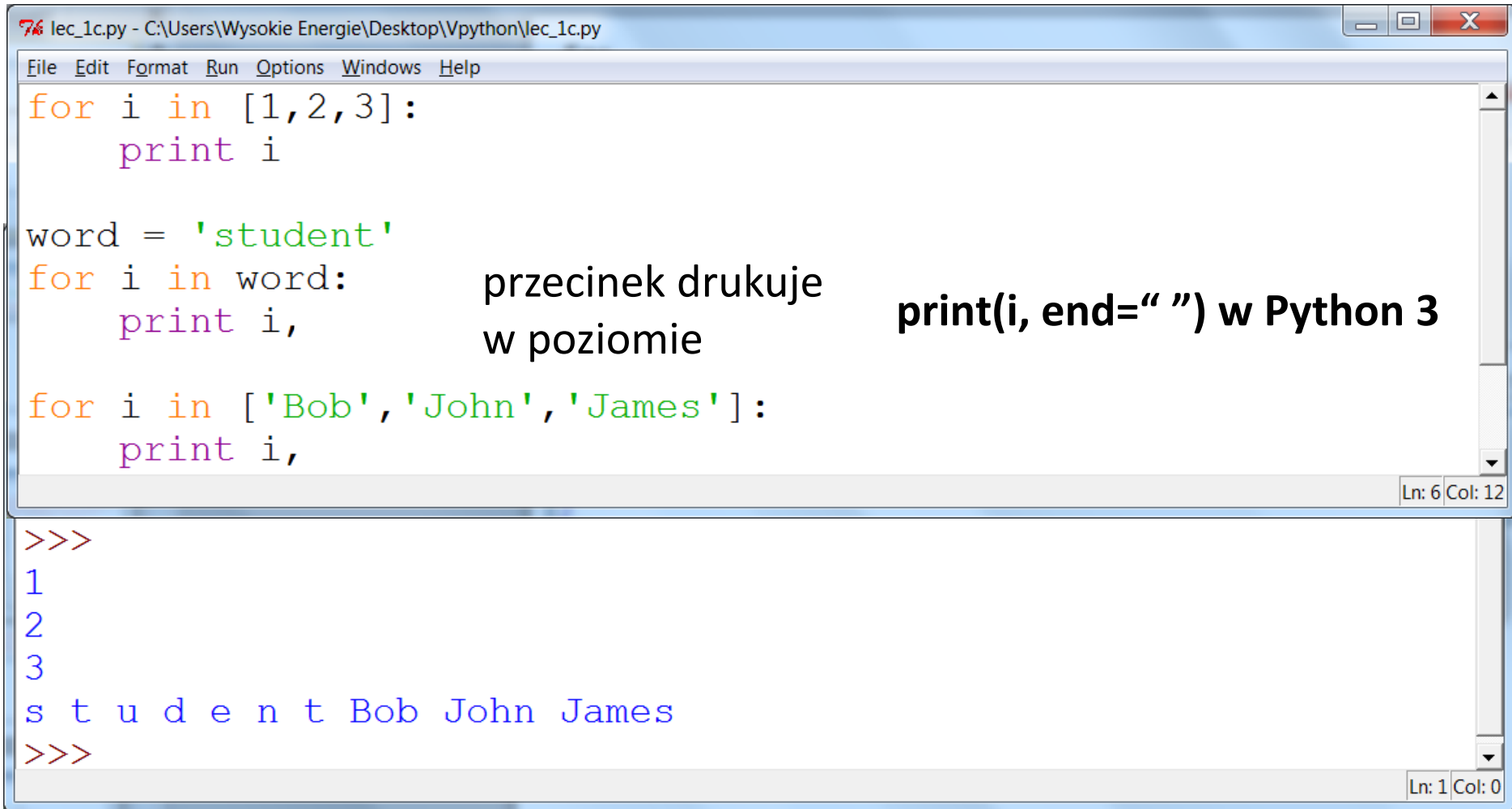
if s=='head':
    print 'you typed head'
```

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

```
>>>
Type a word: head
<type 'str'>
I see "a"
I do not see "b"
you typed head
>>>
```

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

For



The image shows a screenshot of a Python IDE window titled 'lec_1c.py - C:\Users\Wysokie Energie\Desktop\Vpython\lec_1c.py'. The window contains three code snippets demonstrating the 'for' loop. The first snippet iterates over a list [1, 2, 3] and prints each element on a new line. The second snippet iterates over the string 'student' and prints each character followed by a comma. The third snippet iterates over a list of names ['Bob', 'John', 'James'] and prints each name followed by a comma. To the right of the second snippet, there is a text annotation in Polish: 'przecinek drukuje w poziomie' (comma prints horizontally). To the right of the third snippet, there is a text annotation in Polish: 'print(i, end=" ") w Python 3' (print(i, end=" ") in Python 3). Below the code editor, there is a console window showing the output of the code: three numbers (1, 2, 3) on separate lines, followed by the string 'student' with each character on a new line, and then the names 'Bob', 'John', and 'James' on separate lines. The console window has a status bar at the bottom right showing 'Ln: 1 Col: 0'.

```
lec_1c.py - C:\Users\Wysokie Energie\Desktop\Vpython\lec_1c.py
File Edit Format Run Options Windows Help

for i in [1,2,3]:
    print i

word = 'student'
for i in word:
    print i,

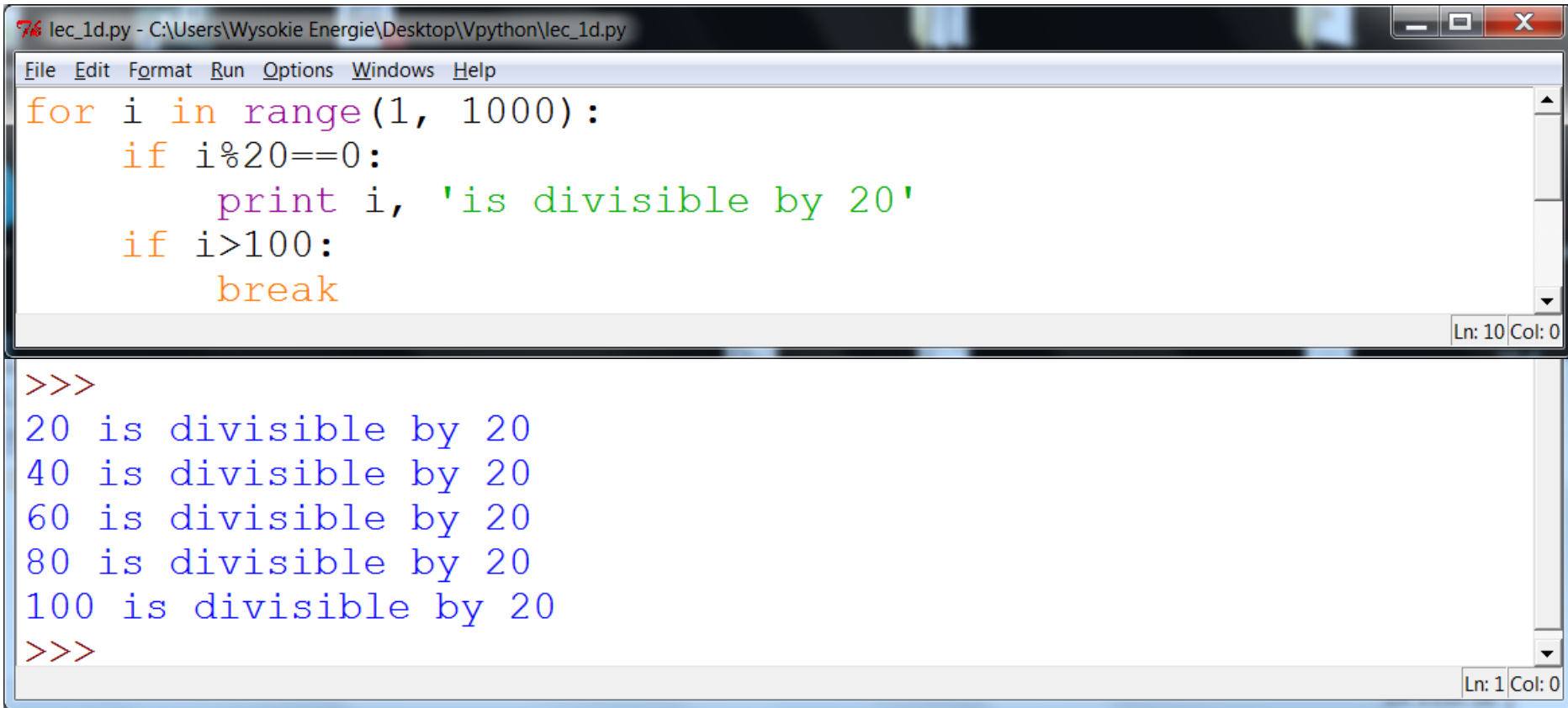
for i in ['Bob', 'John', 'James']:
    print i,
```

przecinek drukuje
w poziomie

print(i, end=" ") w Python 3

```
>>>
1
2
3
s t u d e n t Bob John James
>>>
```


break, for, if



The image shows a screenshot of a Python IDE window titled 'lec_1d.py - C:\Users\Wysokie Energie\Desktop\Vpython\lec_1d.py'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The main editor area contains the following Python code:

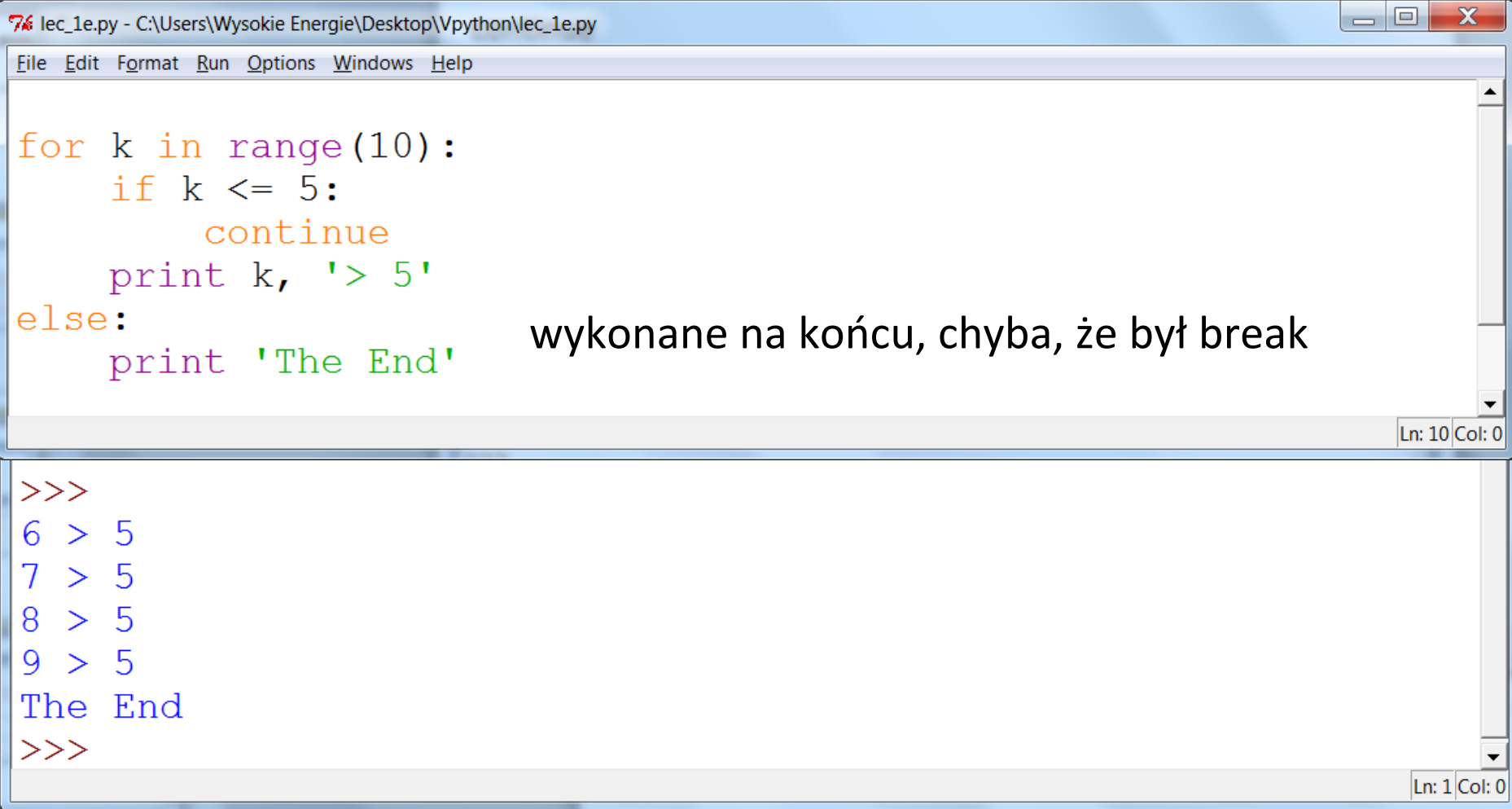
```
for i in range(1, 1000):  
    if i%20==0:  
        print i, 'is divisible by 20'  
    if i>100:  
        break
```

The status bar at the bottom right of the editor shows 'Ln: 10 Col: 0'. Below the editor is a console window showing the output of the script:

```
>>>  
20 is divisible by 20  
40 is divisible by 20  
60 is divisible by 20  
80 is divisible by 20  
100 is divisible by 20  
>>>
```

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

continue, for, else



The image shows a screenshot of a Python IDE window titled "lec_1e.py - C:\Users\Wysokie Energie\Desktop\Vpython\lec_1e.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:

```
for k in range(10):  
    if k <= 5:  
        continue  
    print k, '> 5'  
else:  
    print 'The End'
```

To the right of the code, there is a text annotation: "wykonane na końcu, chyba, że był break".

The bottom panel of the IDE shows the execution output:

```
>>>  
6 > 5  
7 > 5  
8 > 5  
9 > 5  
The End  
>>>
```

The status bar at the bottom right of the IDE shows "Ln: 10 Col: 0" for the script editor and "Ln: 1 Col: 0" for the output console.

while

lec_1f.py - C:\Users\Wysokie Energie\Desktop\Vpython\lec_1f.py

File Edit Format Run Options Windows Help

```
i = 0
while i<10:
    a = int(raw_input('Type a number: '))

    if a == 0:
        print 'This is 0'
        break
    else:
        print 'This is not zero'

    i = i+1    # i += 1
```

zamiana na int

Ln: 14 Col: 0

```
>>>
Type a number: 1
This is not zero
Type a number: 0
This is 0
>>>
```

Ln: 1 Col: 0

Funkcje (functions)

lec_1g.py - C:\Users\Wysokie Energie\Desktop\Vpython\lec_1g.py

File Edit Format Run Options Windows Help

```
def func(x,y):  
    return x*y  
  
print func(2, 3.0)  
print func(4, 'ok '  
print func(3, [1,2])  
#print func('hi','ok')  
  
f = func  
print f(5, 'a')
```

Ln: 14 Col: 0

```
>>>  
6.0  
ok ok ok ok  
[1, 2, 1, 2, 1, 2]  
aaaaa  
>>>
```

Ln: 1 Col: 0

Funkcje

lec_1h.py - C:\Users\Wysokie Energie\Desktop\Vpython\lec_1h.py

File Edit Format Run Options Windows Help

```
def func(x, y):  
    """func multiplies two objects
```

```
    It does not work when two objects cannot be multiplied,  
    for example two strings.  
    """
```

```
    return x*y
```

```
print func.__doc__
```

docstring (Python documentation string)

Ln: 13 Col: 0

```
>>>  
func multiplies two objects
```

```
    It does not work when two objects cannot be multiplied,  
    for example two strings.
```

```
>>>
```

Ln: 1 Col: 0