

Języki i techniki programowania

Wykład 1

Maciej Rybczyński

Prof. dr hab. Maciej Rybczyński
maciej.rybczynski@ujk.edu.pl
Instytut Fizyki, pokój 306A

www.ujk.edu.pl/strony/mryb

Konsultacje:
Po wcześniejszym umówieniu przez e-mail

Wykład: wtorki, 11:30-12:15, Katedra Matematyki, sala 407

Laboratorium: wtorki, 12:30-14:45, Katedra Matematyki, sala 407

Proszę przynosić (i używać) własne laptopy – w miarę możliwości

Najważniejszą częścią tego kursu są zajęcia laboratoryjne.

Będą Państwo pisać programy oparte wyłącznie na materiale z wykładów.

Ocena z laboratorium na podstawie częściowych ocen z kolokwium.

Egzamin ustny.

Oceny częściowe: 2.0, 3.0, 3.5, 4.0, 4.5, 5.0

Aby ukończyć kurs trzeba uzyskać zaliczenie projektu oraz zdobyć minimum 3.0 z zajęć laboratoryjnych i 3.0 z egzaminu ustnego.

Moim celem jest nauczyć Państwa podstaw programowania.
Proszę mnie o wszystko pytać, nie ma głupich pytań.

Omawiane treści:

- Python
- NumPy
- Matplotlib

przykłady ...

Python

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



Donate



Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

Download the latest version for Windows

Download Python 3.10.7

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

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

Looking for Python 2.7? See below for specific releases



Wybieramy: Download Python...

Jeśli potrzebna starsza wersja:

Active Python Releases

For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.10	bugfix	2021-10-04	2026-10	PEP 619
3.9	security	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537
2.7	end-of-life	2010-07-03	2020-01-01	PEP 373

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more	
Python 3.7.14	Sept. 6, 2022	Download	Release Notes
Python 3.8.14	Sept. 6, 2022	Download	Release Notes
Python 3.9.14	Sept. 6, 2022	Download	Release Notes
Python 3.10.7	Sept. 6, 2022	Download	Release Notes
Python 3.10.6	Aug. 2, 2022	Download	Release Notes
Python 3.10.5	June 6, 2022	Download	Release Notes
Python 3.9.13	May 17, 2022	Download	Release Notes
Python 3.10.4	March 24, 2022	Download	Release Notes

[View older releases](#)

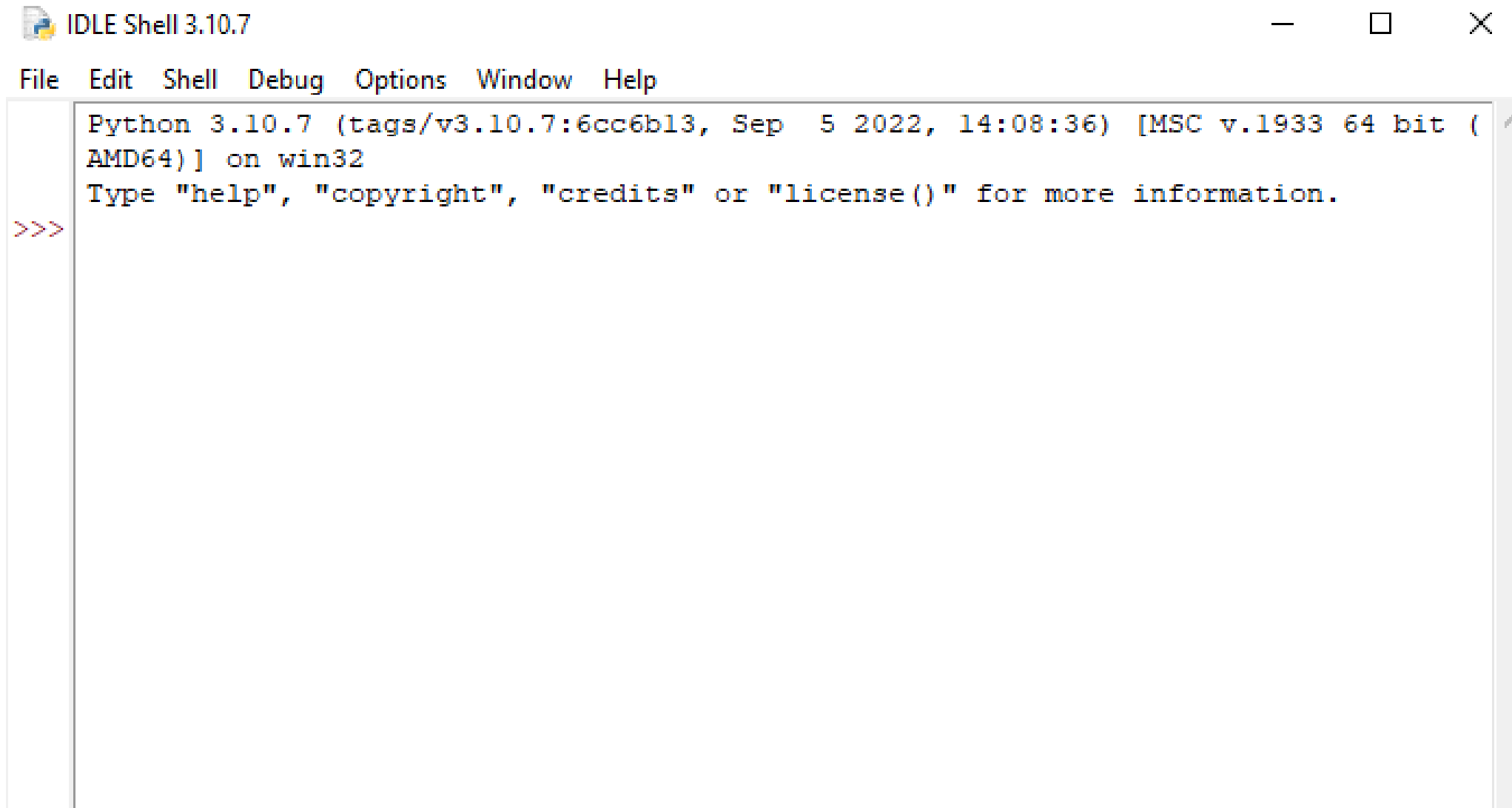
Proszę, pozwolić Pythonowi zainstalować się tam, gdzie chce!

Wybieramy IDLE (Python...):



(link powinien być w menu start)

Interpreter Pythona jako kalkulator

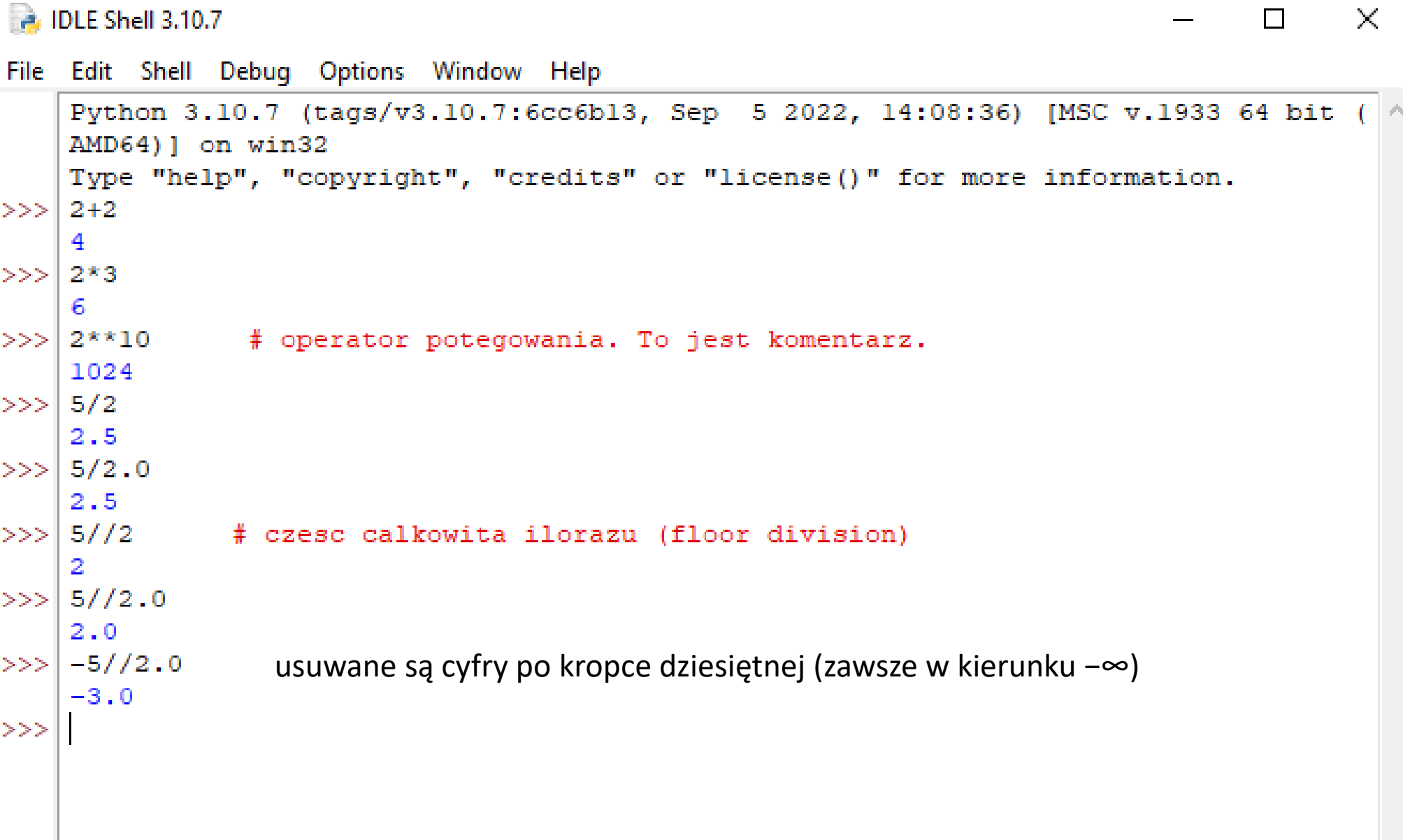


The screenshot shows the IDLE Shell 3.10.7 window. The title bar reads "IDLE Shell 3.10.7" with standard window controls (minimize, maximize, close). The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the Python 3.10.7 startup message: "Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32". Below this, it says "Type 'help', 'copyright', 'credits' or 'license()' for more information." The prompt ">>>" is visible on the left side of the text area.

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

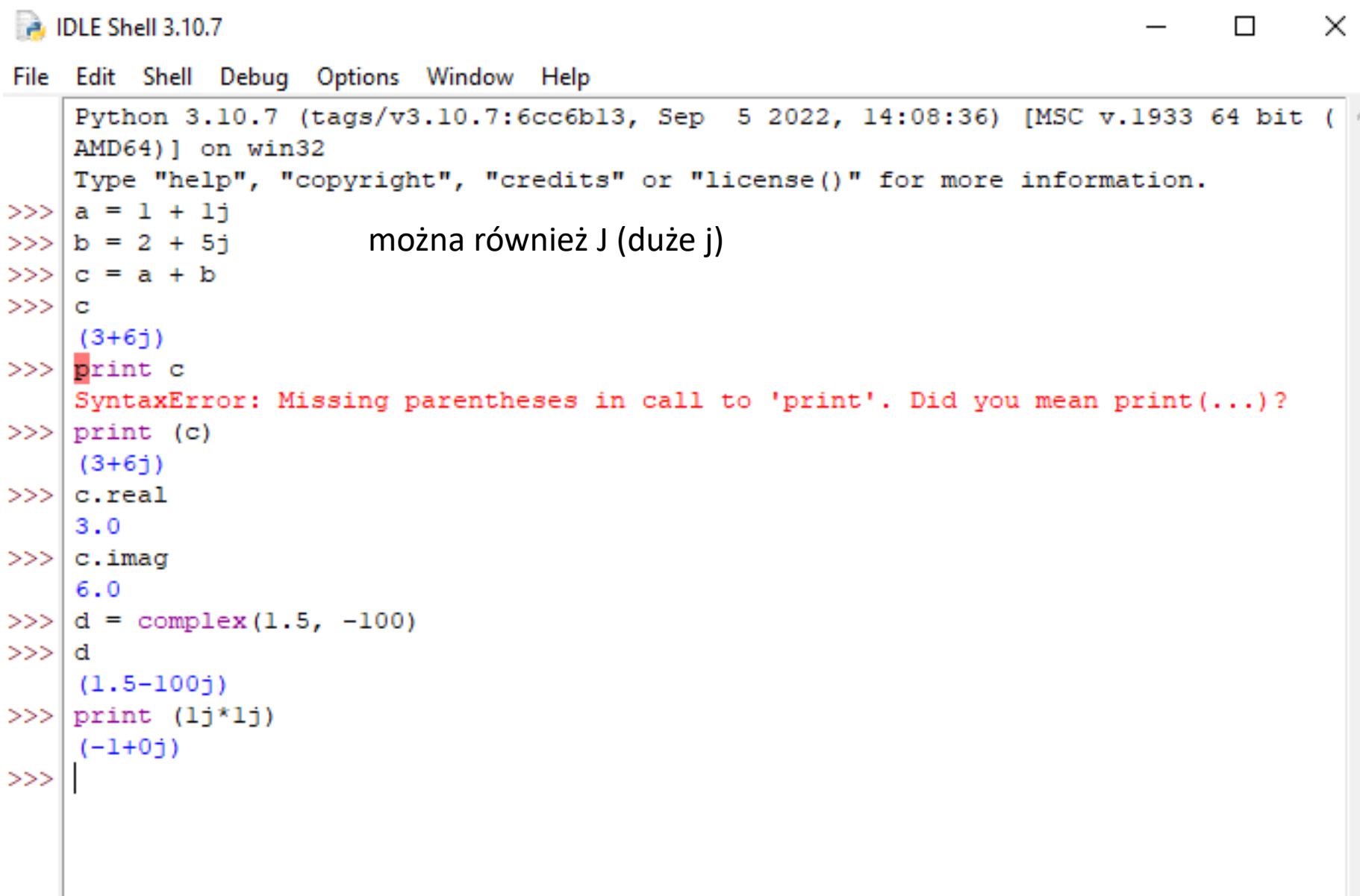
Zamknięcie: naciskamy **Ctrl-Q** lub piszemy **quit()**

„Proste” operacje



```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2+2
4
>>> 2*3
6
>>> 2**10      # operator potegowania. To jest komentarz.
1024
>>> 5/2
2.5
>>> 5/2.0
2.5
>>> 5//2      # czesc calkowita ilorazu (floor division)
2
>>> 5//2.0
2.0
>>> -5//2.0    usuwane są cyfry po kropce dziesiętnej (zawsze w kierunku -∞)
-3.0
>>> |
```

Liczby zespolone



The screenshot shows an IDLE Shell window titled "IDLE Shell 3.10.7". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The shell displays the following text:

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = 1 + 1j
>>> b = 2 + 5j
>>> c = a + b
>>> c
(3+6j)
>>> print c
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(...)?
>>> print (c)
(3+6j)
>>> c.real
3.0
>>> c.imag
6.0
>>> d = complex(1.5, -100)
>>> d
(1.5-100j)
>>> print (1j*1j)
(-1+0j)
>>> |
```

można również J (duże j)

Możemy mieszać różne typy zmiennych

IDLE Shell 3.10.7

File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2 + 3.0
5.0
>>> 10/2.0
5.0
>>> 5.0%2      # operator modulo (reszta z dzielenia).  3%2=1,  4%2=0,  7%3=1
1.0
>>> a = 1 + 5.0j
>>> a
(1+5j)
>>> 7 + a
(8+5j)
>>>
```

Pewne użyteczne (wbudowane) funkcje

Python 3.10.7

File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a = -5.0
>>> b = abs(a)
>>> b
5.0
>>> c = 0.123456789
>>> round(c,2)
0.12
>>> d = -10.0
>>> int(d)
-10
>>> complex(d)
(-10+0j)
>>> float(123)
123.0
>>> |
```

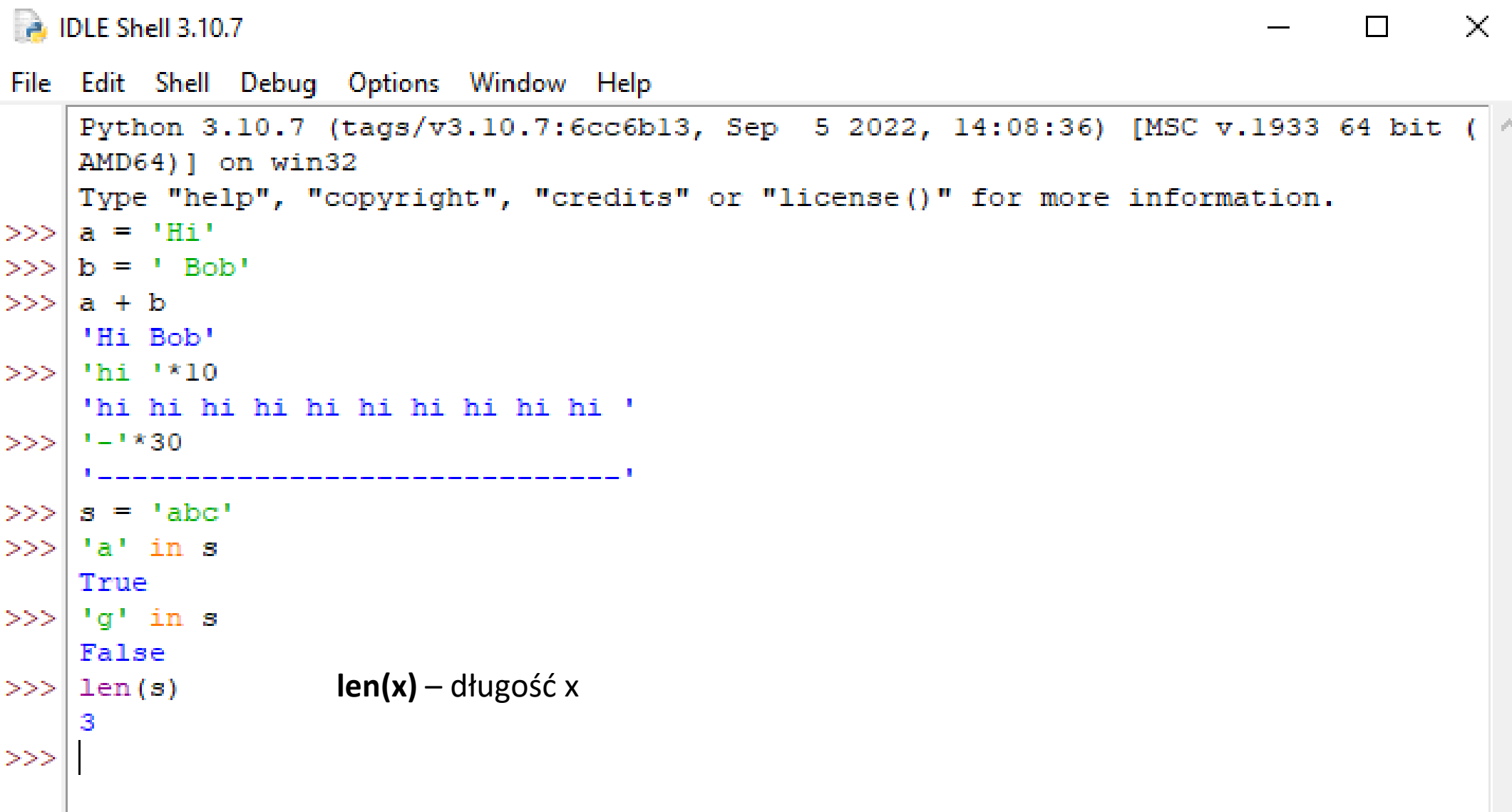
abs(x) – wartość bezwzględna x

round(x,n) – x zaokrąglone do n cyfr

int(x) – całkowita wartość x

float(x) – rzeczywista wartość x

łańcuchy znaków



```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "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
>>> len(s)
3
```

len(x) – długość x

łańcuchy znaków



IDLE Shell 3.10.7



File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> s = 'James Bond'
>>> len(s)
10
>>> s[0]          pierwszy element, zliczanie zaczyna się 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()      s.split() – zwraca listę słów
['James', 'Bond']
>>>
```

łańcuchy znaków

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

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

↑
włączony

↑
wyłączony

łańcuchy znaków



IDLE Shell 3.10.7



File Edit Shell Debug Options Window Help

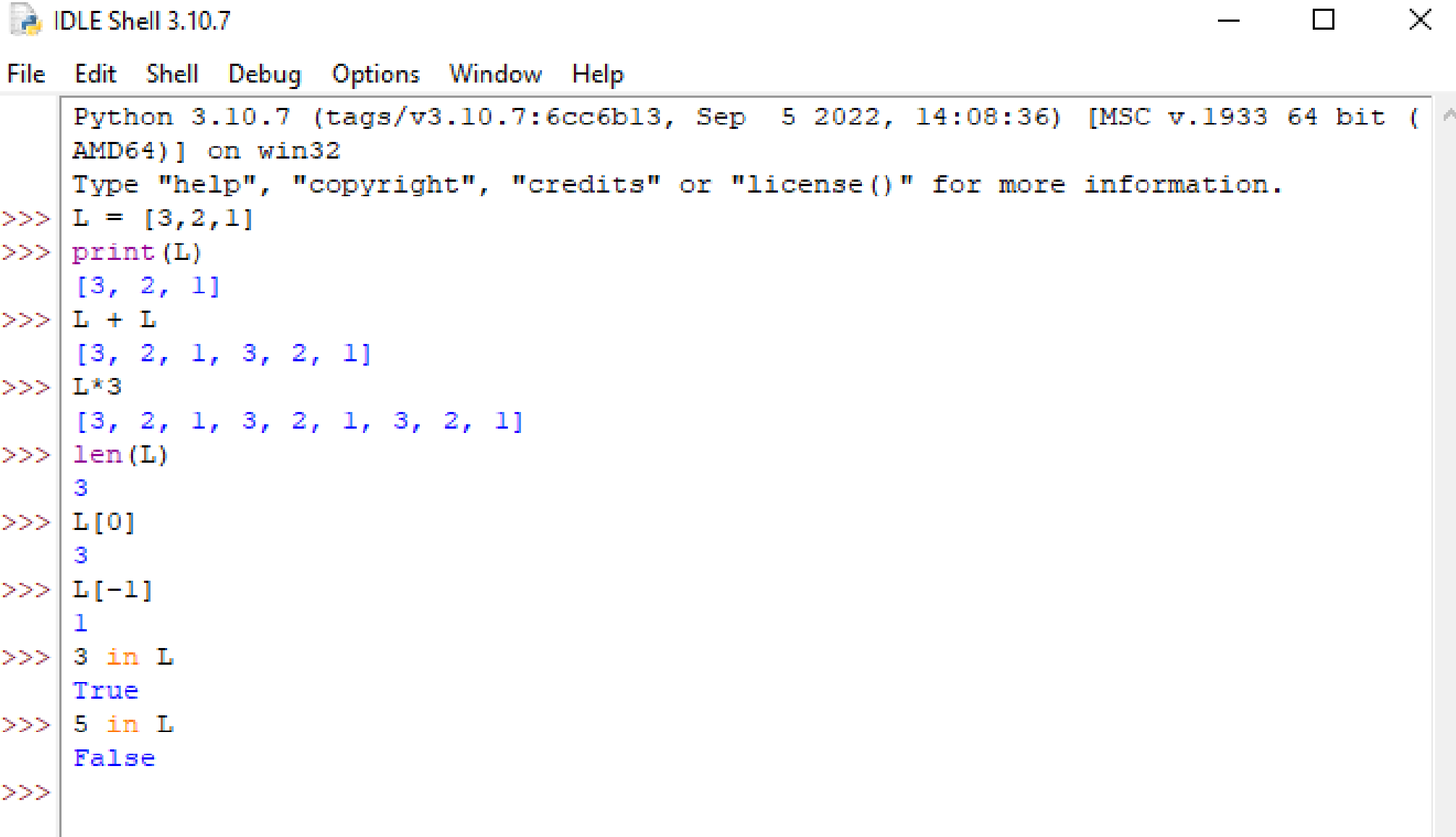
```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> s = 'Winnie-the-Pooh'
>>> len(s)
15
>>> s[0:2]
'Wi'
>>> s[2:]
'nnie-the-Pooh'
>>> s[0:2] + s[2:]
'Winnie-the-Pooh'
>>> s[-1]      # ostatni element
'h'
>>> s[-2]
'o'
>>> s[-4:]
'Pooh'
>>>
```



"What day is it,?" asked Pooh.
"It's today," squeaked Piglet.
"My favorite day," said Pooh.

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

Listy

A screenshot of the Python IDLE Shell 3.10.7 window. The window has a title bar with a file icon, the text "IDLE Shell 3.10.7", and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main area is a text editor showing a Python shell session. The session starts with the Python version and environment information, followed by a prompt to type "help", "copyright", "credits", or "license()". The user then enters several commands: creating a list L, printing it, concatenating it with itself, multiplying it by 3, checking its length, and accessing its elements. The output shows the list [3, 2, 1] and the results of the operations.

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 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
>>>
```

Listy

IDLE Shell 3.10.7

File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
```

```
Type "help", "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) – dołącza x do listy L

L.reverse() – odwrotny porządek elementów L

L.sort() – sortuje L (rosnąco)

Listy

IDLE Shell 3.10.7

File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> L = [7,5,3,1]
>>> L[1:3]
[5, 3]
>>> L[:]
[7, 5, 3, 1]
>>> Lnew = L[:] # utworzenie nowego obiektu
>>> 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 to samo co **Lnew = L[:]**



Dwie nazwy odnoszące się
do tego samego obiektu

Listy „wielowymiarowe”



IDLE Shell 3.10.7



File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 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']]
>>> |
```

Listy, range(), list()



IDLE Shell 3.10.7



File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> range(10)
range(0, 10)
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> list(range(3,10))
[3, 4, 5, 6, 7, 8, 9]
>>> list(range(0,10,2))
[0, 2, 4, 6, 8]
>>> list(range(-10,10))
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
```

Interpreter jest OK dla „prostych” rzeczy.

W przypadku bardziej skomplikowanych:

File -> New File

i zapisz treść w pliku z rozszerzeniem .py, np.: simulation.py

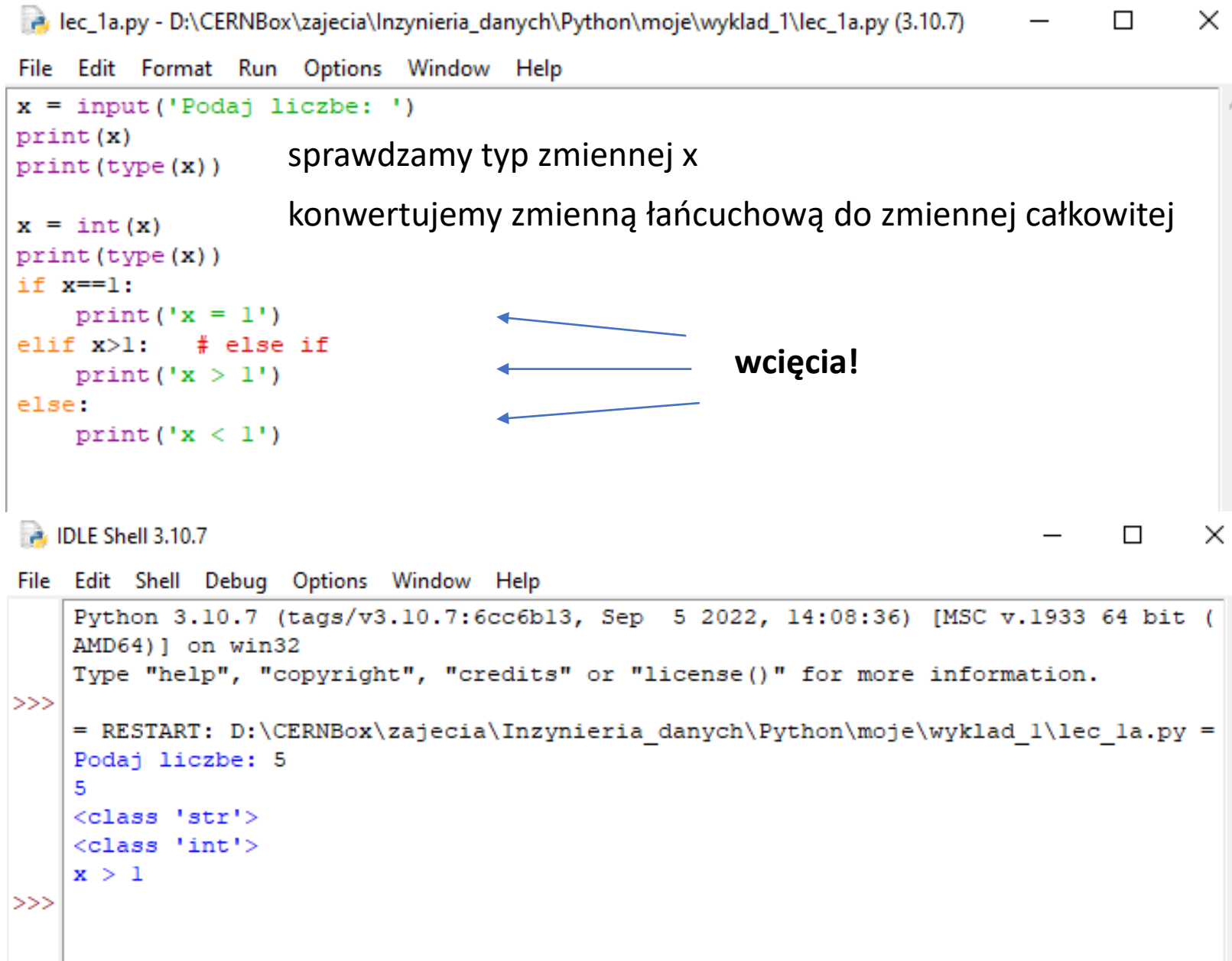
lub

File -> Open

i otwórz istniejący plik .py

Aby uruchomić program naciśnij **F5**

input(), type(), if, else, elif



The image shows a screenshot of the Python IDLE 3.10.7 environment. The top window, titled 'lec_1a.py', contains the following Python code:

```
x = input('Podaj liczbę: ')
print(x)
print(type(x))

x = int(x)
print(type(x))
if x==1:
    print('x = 1')
elif x>1: # else if
    print('x > 1')
else:
    print('x < 1')
```

Annotations on the right side of the code window:

- 'sprawdzamy typ zmiennej x' points to the `print(type(x))` line.
- 'konwertujemy zmienną łańcuchową do zmiennej całkowitej' points to the `x = int(x)` line.
- 'wcięcia!' (indentations!) points to the indented lines within the `if`, `elif`, and `else` blocks.

The bottom window, titled 'IDLE Shell 3.10.7', shows the execution output:

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: D:\CERNBox\zajecia\Inzynieria_danych\Python\moje\wyklad_1\lec_1a.py =
Podaj liczbę: 5
5
<class 'str'>
<class 'int'>
x > 1
>>>
```

if

lec_1b.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1b.py (3.10.7) — □ ×

File Edit Format Run Options Window Help

```
s = input('Wpisz słowo: ')
print(type(s))

if 'a' in s:
    print('Widzę "a"')

if 'b' not in s:
    print('Nie widzę "b"')

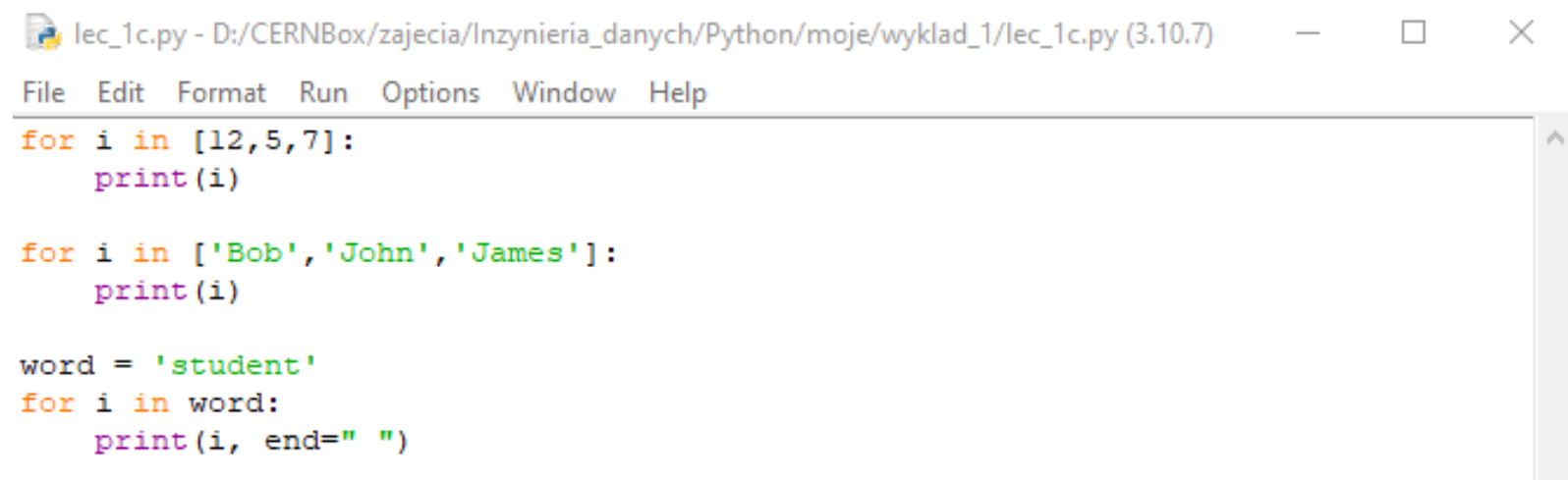
if s=='head':
    print('Wpisano "head"')
```

IDLE Shell 3.10.7 — □ ×

File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1b.py =
Wpisz słowo: head
<class 'str'>
Widzę "a"
Nie widzę "b"
Wpisano "head"
>>> |
```

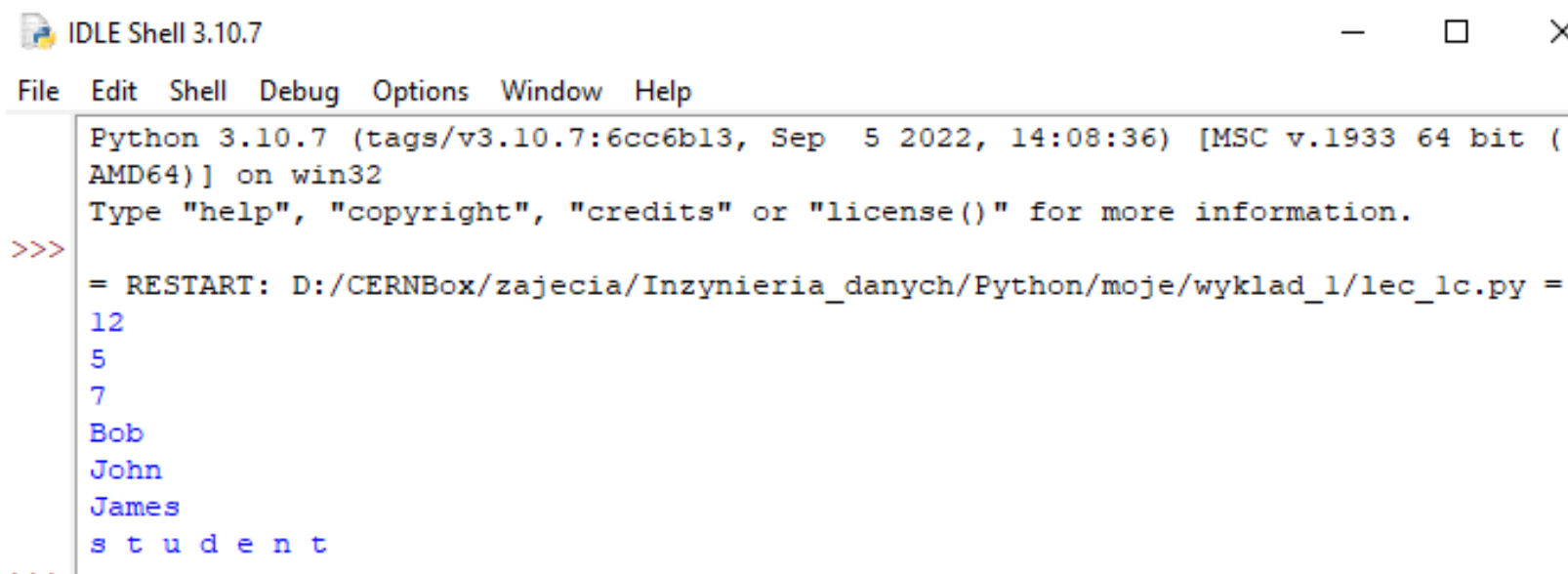
for



```
lec_1c.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1c.py (3.10.7)
File Edit Format Run Options Window Help
for i in [12,5,7]:
    print(i)

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

word = 'student'
for i in word:
    print(i, end=" ")
```



```
IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1c.py =
12
5
7
Bob
John
James
s t u d e n t
>>>
```

for, if, break

lec_1d.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1d.py (3.10.7)

File Edit Format Run Options Window Help

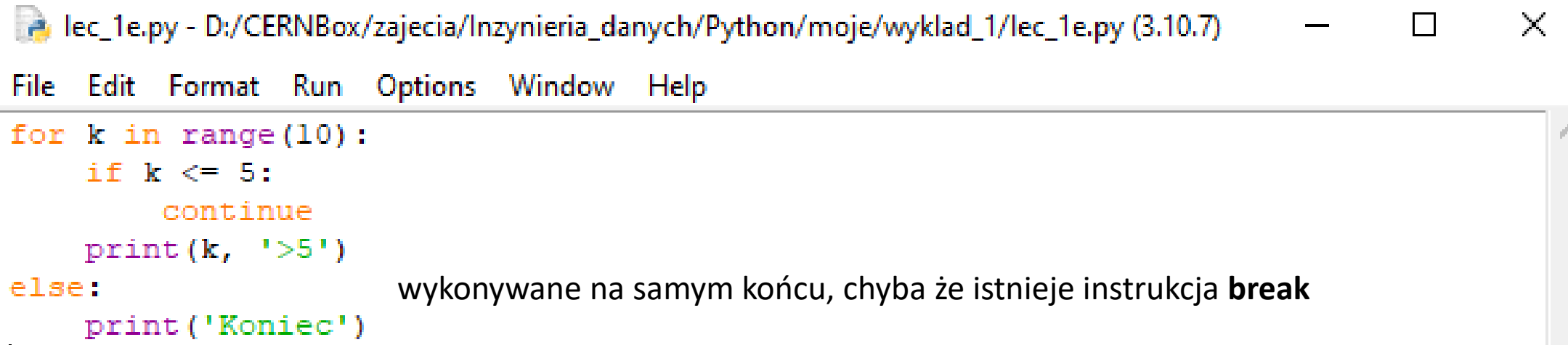
```
for i in range(1, 1000):  
    if i%20==0:  
        print(i, 'dzieli się przez 20')  
    if i>100:  
        break
```

IDLE Shell 3.10.7

File Edit Shell Debug Options Window Help

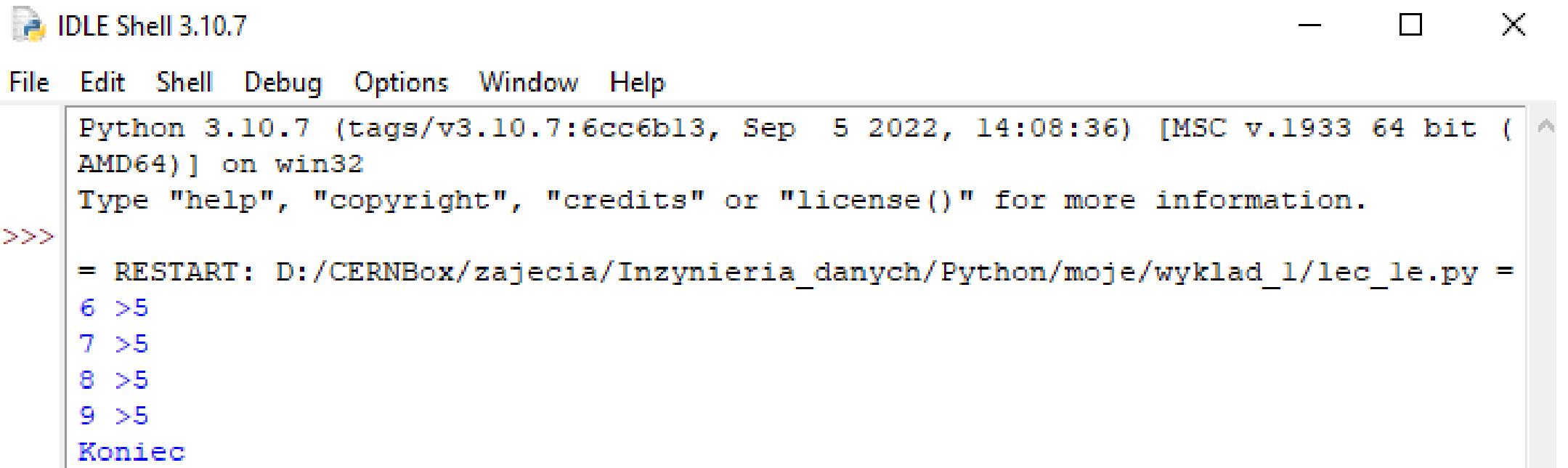
```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1d.py =  
20 dzieli się przez 20  
40 dzieli się przez 20  
60 dzieli się przez 20  
80 dzieli się przez 20  
100 dzieli się przez 20  
>>> |
```

for, else, continue



```
lec_1e.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1e.py (3.10.7)
File Edit Format Run Options Window Help
for k in range(10):
    if k <= 5:
        continue
    print(k, '>5')
else:
    print('Koniec')
```

wykonywane na samym końcu, chyba że istnieje instrukcja **break**



```
IDLE Shell 3.10.7
File Edit Shell Debug Options Window Help
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1e.py =
6 >5
7 >5
8 >5
9 >5
Koniec
```

while

lec_1f.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1f.py (3.10.7)

File Edit Format Run Options Window Help

```
i=0
while i<10:
    a = int(input('Podaj liczbę: '))

    if a == 0:
        print('Wprowadziłeś 0')
        break
    else:
        print('Wprowadziłeś inną liczbę niż zero')

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

IDLE Shell 3.10.7

File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep  5 2022, 14:08:36) [MSC v.1933 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1f.py =
Podaj liczbę: 1
Wprowadziłeś inną liczbę niż zero
Podaj liczbę: 0
Wprowadziłeś 0
>>> |
```

Funkcja (własna)

lec_1g.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1g.py (3.10.7) — □ ×

File Edit Format Run Options Window 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'))
```

IDLE Shell 3.10.7 — □ ×

File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1g.py =  
6.0  
ok ok ok ok  
[1, 2, 1, 2, 1, 2]  
aaaaa  
>>> |
```

Funkcja (własna)

lec_1h.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1h.py (3.10.7)

File Edit Format Run Options Window Help

```
def func(x,y):  
    """  
    func mnoży dwa obiekty  
  
    Nie będzie działać, gdy dwa obiekty nie mogą być pomnożone,  
    na przykład dwa łańcuchy znaków.  
    """  
    return x*y
```

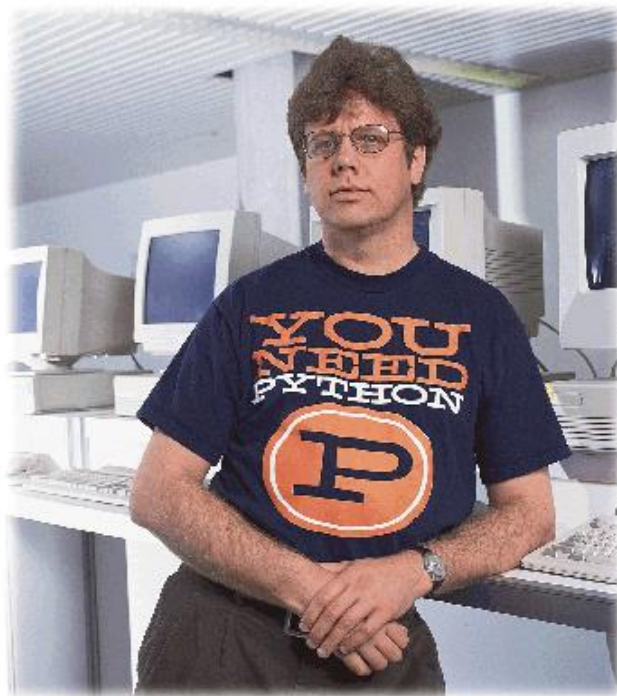
`print(func.__doc__)` dostęp do docstring (ciąg dokumentacji Pythona)

IDLE Shell 3.10.7

File Edit Shell Debug Options Window Help

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_1/lec_1h.py =  
  
func mnoży dwa obiekty  
  
Nie będzie działać, gdy dwa obiekty nie mogą być pomnożone,  
na przykład dwa łańcuchy znaków.
```

Twórca Pythona: **Guido van Rossum**



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

FileEditViewHistoryBookmarksToolsHelp

W Guido van Rossum - Wikip... x +

←

🔒

📄

🔄

🔍 Search

☆

📁

📌

⬇️

🏠

🌐

☰

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

Bahasa Indonesia

Italiano

Magyar

Malagasy

മലയാളം

Bahasa Melayu

Nederlands

日本語

Polski

Português

Русский

Slovenčina

Српски / srpski

Suomi

Svenska

தமிழ்

Türkçe

Українська

中文

✎ Edit links

About the origin of [Python](#), Van Rossum wrote in 1996:

Over six years ago, in December 1989, I was looking for a "hobby" programming project that would keep me occupied during the week around Christmas. My office ... would be closed, but I had a home computer, and not much else on my hands. I decided to write an interpreter for the new scripting language I had been thinking about lately: a descendant of ABC that would appeal to Unix/C hackers. I chose Python as a working title for the project, being in a slightly irreverent mood (and a big fan of [Monty Python's Flying Circus](#)).^[15]

In 2000 he further wrote:

Python's predecessor, [ABC](#), was inspired by [SETL](#) – [Lambert Meertens](#) spent a year with the SETL group at [NYU](#) before coming up with the final ABC design!^[16]


Computer Programming for Everybody

[edit]

In 1999, Van Rossum submitted a funding proposal to [DARPA](#) called [Computer Programming for Everybody](#), in which he further defined his goals for Python:

- 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 November 2011, it was the 3rd most popular language on the [GitHub](#) social



Van Rossum at the 2008 [Google I/O](#) Developer's Conference.

35