

Języki i techniki programowania

Wykład 2

Maciej Rybczyński

delete

lec_2a.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2a.py (3.10.7)

File Edit Format Run Options Window Help

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

```
del L[0]      usuwanie pierwszego elementu listy L
print(L)
```

```
del L[0:2]
print(L)
```

```
del L[:]
print(L)
```

```
del L      usuwanie listy L z pamięci
print(L)
```

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

Traceback (most recent call last):

File "D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2a.py", line 13, in <module>

print(L)

NameError: name 'L' is not defined

więcej o listach

lec_2b.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2b.py (3.10.7)

File Edit Format Run Options Window Help

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

```
L.append(137)
```

```
print(L)
```

```
L.insert(1, 500)
```

 umieszcza wartość 500 na ustalonej pozycji (1)

```
L.insert(-1, 500)
```

```
print(L)
```

```
L.remove(500)
```

 usuwa pierwszy element o podanej wartości (500)

```
print(L)
```

```
a = L.pop(2)
```

 usuwa element listy na podanej pozycji (2) i zwraca wartość elementu

```
print(L, a)
```

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

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

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

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

```
>>>
```

więcej o listach

lec_2c.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2c.py (3.10.7)

File Edit Format Run Options Window Help

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

```
a = L.index(15)      zwraca indeks pierwszego elementu, którego wartość = 15
print(a)
```

```
b = L.count(15)      zwraca liczbę wystąpień wartości 15
print(b)
```

```
L.sort()
print(L)
```

```
L.reverse()
print(L)
```

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

zbiory

lec_2d.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2d.py (3.10.7)



File Edit Format Run Options Window 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)      # w set1 ale nie w set2
print(set1 | set2)      # w set1 lub w set2
print(set1 & set2)      # w set1 i w set2
print(set1 ^ set2)      # albo w set1 albo w set2 ale nie w obu
```

```
{1, 2, 3, 4, 5}
```

```
{9, 10, 4, 5}
```

zbiory są kolekcjami **unikalnych** elementów

```
{1, 2, 3}
```

```
{1, 2, 3, 4, 5, 9, 10}
```

```
{4, 5}
```

```
{1, 2, 3, 9, 10}
```

```
>>>
```

zbiory

lec_2e.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wykklad_2/lec_2e.py (3.10.7)

File Edit Format Run Options Window Help

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

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

```
print(set1 & set2)
```

```
{'s', 'B', 'M', ' ', 'r', 'a', 't', 'n', 'e', 'i'}  
{'H', 'B', ' ', 'y', 'l', 'n', 'e', 'i'}  
  
{'B', ' ', 'n', 'e', 'i'}
```

słowniki

lec_2f.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2f.py (3.10.7)

File Edit Format Run Options Window 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))          # długość
```

```
print('Tom' in tel)
```

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2f.py =
```

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

```
dict_keys(['John', 'Kate', 'Tom', 'Bob'])
```

```
dict_values([1234, 5678, 9000, 1000])
```

```
5678
```

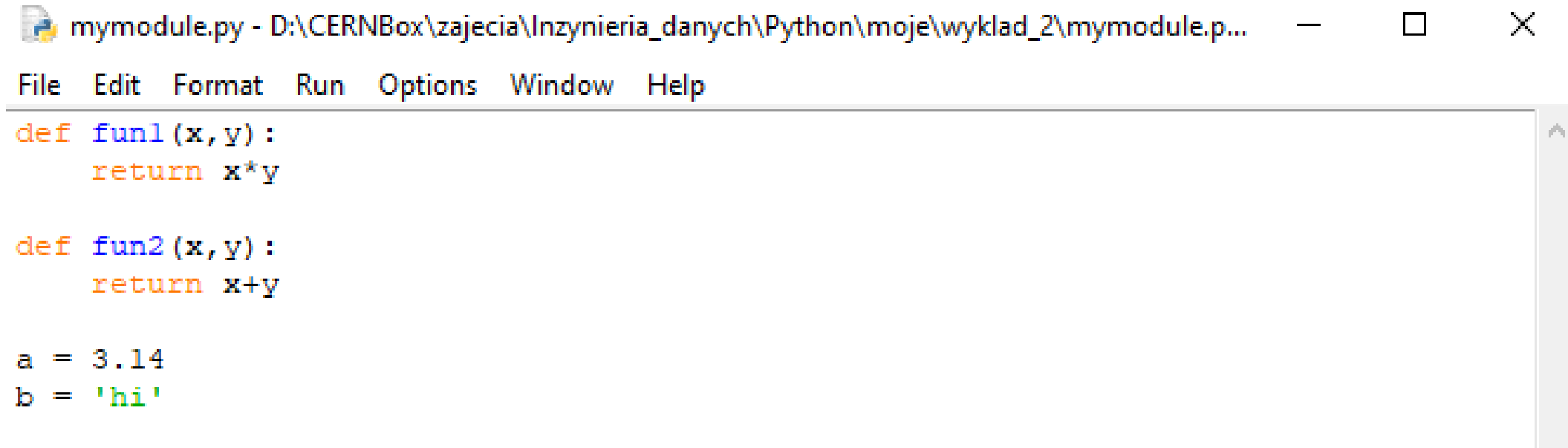
```
4
```

```
True
```

```
>>> |
```

moduły


Utwórzmy plik mymodule.py



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

Teraz w naszym głównym programie możemy używać obiektów zdefiniowanych w mymodule.py

moduły

 lec_2g.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2g.py (3.10.7) — □ ×

File Edit Format Run Options Window Help

```
import mymodule # importuje zawartość modułu
```

```
print(mymodule.fun1(2,3))
```

```
print(mymodule.fun2(2,10))
```

```
print(mymodule.a)
```

```
print(mymodule.b)
```

```
>>>
```

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2g.py =
```

```
6
```

```
12
```

```
3.14
```

```
hi
```

moduły

lec_2h.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2h.py (3.10.7) — □ ×

File Edit Format Run Options Window Help

```
import mymodule as mm # importuje zawartość modułu i przypisuje mu nazwę |
```

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

```
>>> = RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2h.py =  
6  
12  
3.14  
hi
```

moduły

lec_2i.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2i.py (3.10.7)

File Edit Format Run Options Window Help

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

```
#from mymodule import * # importuje wszystkie obiekty z modułu, nie rekomendowane
```

```
print(fun1(2,3))
```

```
print(fun2(2,10))
```

```
print(a)
```

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2i.py =
```

```
6
```

```
12
```

```
Traceback (most recent call last):
```

```
  File "D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2i.py", line 7, in <module>
```

```
    print(a)
```

```
NameError: name 'a' is not defined
```

moduł „random”

lec_2j.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2j.py (3.10.7)

File Edit Format Run Options Window Help

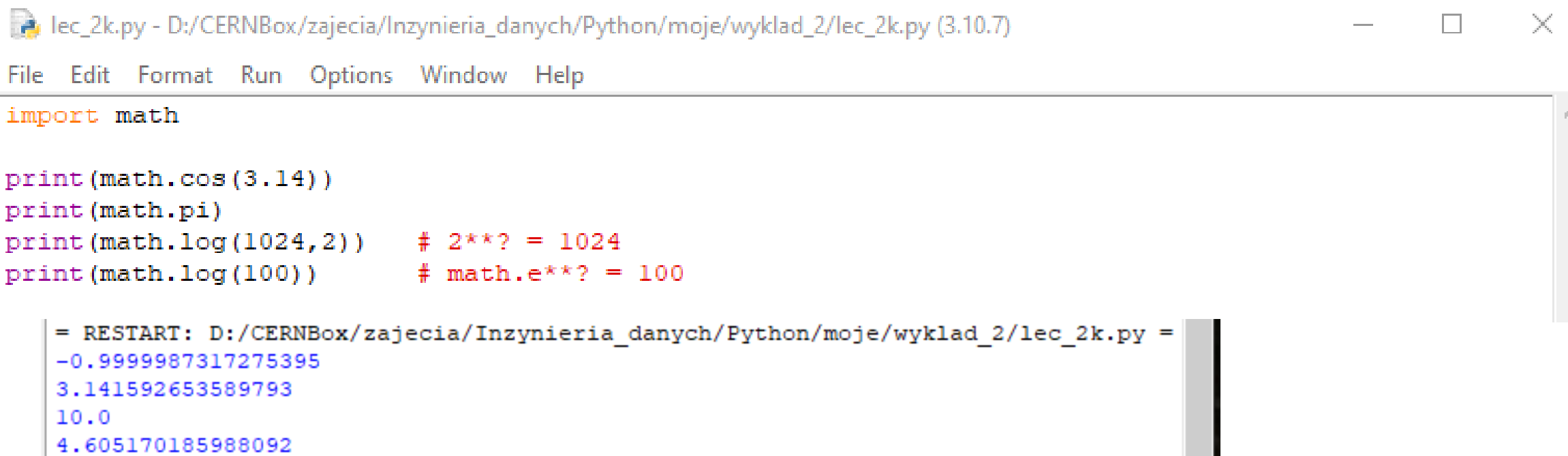
```
import random
```

```
print(random.random())      # liczba rzeczywista z przedzialu [0,1) (rozklad rownomierny)
print(random.uniform(0,10)) # liczba rzeczywista z przedzialu [0,10) (rozklad rownomierny)
print(random.randint(2,100)) # liczba calkowita z przedzialu [2,100]
print(random.choice(['no','yes'])) # element z listy
print(random.randrange(1,100,2)) # z range(1,100,2)
```

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2j.py =
0.40781944091009836
5.23100060149937
7
no
3
```

Szczegółowy opis: <https://docs.python.org/2/library/random.html>

moduł „math”



The screenshot shows a Python IDE window titled "lec_2k.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2k.py (3.10.7)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor contains the following Python 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
```

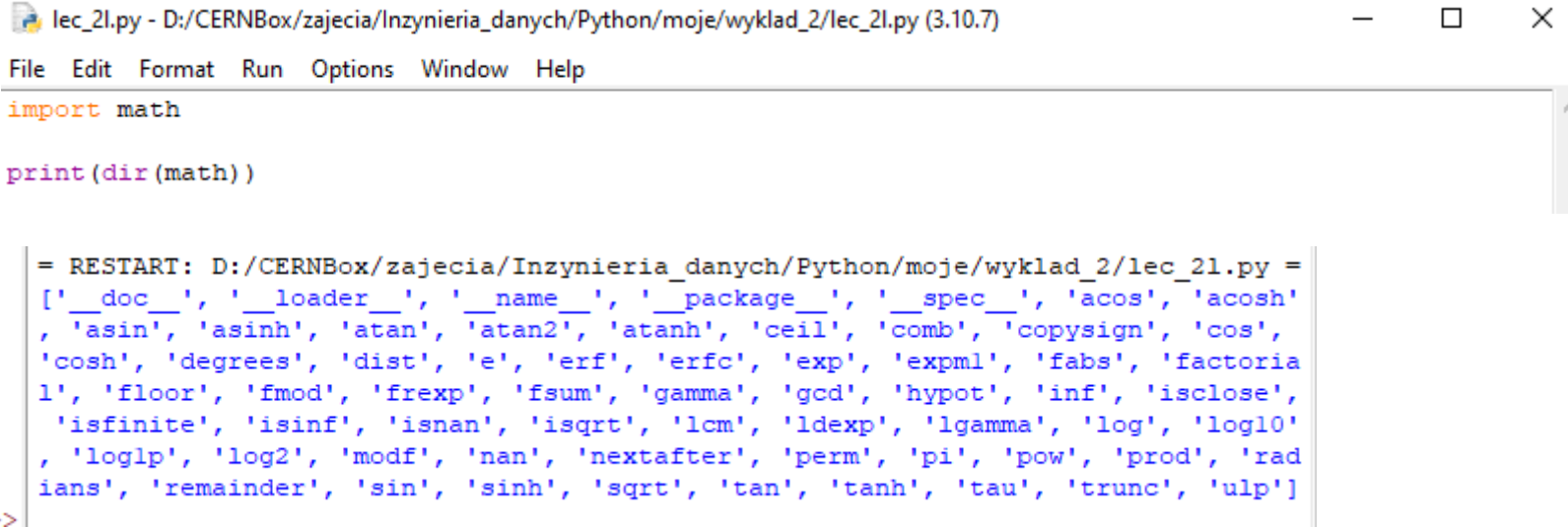
Below the code editor, the output of the script is displayed:

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2k.py =
-0.9999987317275395
3.141592653589793
10.0
4.605170185988092
```

Powyższe funkcje nie mogą być używane z liczbami zespolonymi – użyj **cmath**.

Szczegółowy opis: <https://docs.python.org/2/library/math.html>

co można znaleźć w module math?



The screenshot shows a Python IDE window titled 'lec_21.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_21.py (3.10.7)'. The menu bar includes 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code editor contains the following Python code:

```
import math

print(dir(math))
```

The output of the code is displayed in the console area:

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_21.py =
['_doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh',
 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'comb', 'copysign', 'cos',
 'cosh', 'degrees', 'dist', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factoria
 1', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose',
 'isfinite', 'isinf', 'isnan', 'isqrt', 'lcm', 'ldexp', 'lgamma', 'log', 'log10'
 , 'loglp', 'log2', 'modf', 'nan', 'nextafter', 'perm', 'pi', 'pow', 'prod', 'rad
 ians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc', 'ulp']
```

Spróbuj `dir(random)`, etc...

Oczywiście najpierw trzeba zaimportować odpowiedni moduł.

dir() – wyświetla zawartość modułów, funkcje, etc. w programie

lec_2m.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2m.py (3.10.7)

File Edit Format Run Options Window Help

```
import math
```

```
my_number = 5
```

```
my_name = 'Jack'
```

```
def funk(x,y): return x*y
```

```
print(dir())
```

```
del my_number
```

```
print()
```

```
print(dir())
```

```
>>> = RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2m.py =  
['__annotations__', '__builtins__', '__doc__', '__file__', '__loader__', '__name__'  
__, '__package__', '__spec__', 'funk', 'math', 'my_name', 'my_number']  
  
['__annotations__', '__builtins__', '__doc__', '__file__', '__loader__', '__name__'  
__, '__package__', '__spec__', 'funk', 'math', 'my_name']  
>>>
```

kolejny sposób tworzenia listy

lec_2n.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2n.py (3.10.7)

File Edit Format Run Options Window 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)
```

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2n.py =
[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]
[0.8811949856439821, 0.5541745964801768]
```

zapisywanie danych do pliku

lec_2o.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2o.py (3.10.7) — □ ×

File Edit Format Run Options Window Help

```
f = open('first.txt', 'w')          # w = write (plik otwarty w trybie do zapisywania)
f.write('let us write something\n')  # \n = nowa linijka
f.write('oh yes')
f.close()
```

d:\CERNBox\zajecia\Inzynieria_danych\Python\moje\wyklad_2\first.txt - Notepad++

Plik Edycja Szukaj Widok Format Składnia Ustawienia Narzędzia Makra Uruc



first.txt

```
1 let us write something
2 oh yes
```

czytanie danych z pliku

lec_2p.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2p.py (3.10.7)

File Edit Format Run Options Window Help

```
f = open('first.txt', 'r')          # r = read (plik otwarty w trybie do odczytu)
print(f.read())
f.close()
```

```
# f.read(5) - czytanie 5 symboli z pliku
# f.readline() - czytanie jednej linijki
```

```
>>> = RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2p.py =
let us write something
oh yes
```

w write
r read
a append

zapisywanie danych do pliku

lec_2q.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2q.py (3.10.7)

File Edit Format Run Options Window 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()
```

d:\CERNBox\zajecia\Inzynieria_danych\Python\moje\wyklad_2\tel.txt - Notepad++

Plik Edycja Szukaj Widok Format Składnia Ustawienia Narzędzia Makra Uruchom Wtyczki Karty ?



first.txt x tel.txt x

```
1 0) Bob 123
```

```
2 1) Peter 456
```

```
3 2) Jane 789
```

```
4
```

str(x) – konwertuje x do ciągu znaków

czytanie danych z pliku

lec_2r.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2r.py (3.10.7)

File Edit Format Run Options Window Help

```
names = []
L = []

f = open('tel.txt', 'r')

for i in f:
    temp = i.split()      # split() dzieli łańcuch znaków na słowa
    print(temp)
    names.append(temp[1])
    L.append(int(temp[2])) # int(x) konwertuje x do liczby całkowitej
f.close()

print('\n')
print(names, L, end="")
```

```
= RESTART: D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2r.py =
['0)', 'Bob', '123']
['1)', 'Peter', '456']
['2)', 'Jane', '789']

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

append

lec_2s.py - D:/CERNBox/zajecia/Inzynieria_danych/Python/moje/wyklad_2/lec_2s.py (3.10.7)

File Edit Format Run Options Window Help

```
def wr_f(d1,d2,d3,d4):  
    f = open('results.txt', 'a')      # a = append  
    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])
```

d:\CERNBox\zajecia\Inzynieria_danych\Python\moje\wyklad_2\results.txt - Notepad++

Plik Edycja Szukaj Widok Format Składnia Ustawienia Narzędzia Makra Uruchom Wtyczki Karty ?



first.txt x tel.txt x results.txt x

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

PEP 8 - przewodnik po stylach dla kodu Pythona

- używaj 4 spacji na jedno wcięcie
- nie używaj tabulatora do robienia wcięć
- linijka kodu nie powinna przekraczać 79 znaków
- twórz opisy kodu w docstrings
- używaj spacji, np.: `a = f(1, 2) + g(3, 4)`
- nigdy nie stosuj `l` (małe el), `I` (duże i), `O` (duże o)

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