Zajęcie 1

```
a = 1
print(a)
```

```
1
```

```
b = "text"
b
a
```

```
1
```

```
b = "text"
a, b
```

```
(1, 'text')
```

Komentarze:

```
#A+B

"""
A + B
"""
```

Tabela:

```
C = [1,2,3,4,5,6,7,8]
print(C[0])
print(C[0:7])
print(C[0:7:2])
```

```
1
[1, 2, 3, 4, 5, 6, 7]
[1, 3, 5, 7]
```

```
a = 1
```

```
print(a)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-bca0e2660b9f> in <module>()
----> 1 print(a)

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

SEARCH STACK OVERFLOW

EksploracjaDanych_1.pdf:

```
b = 11.0
B = "Ala koty"
zm1 = 12
zm2 = 4
Zm3 = "5"
b, B, zm1, zm2, Zm3
```

```
(11.0, 'Ala koty', 12, 4, '5')
```

```
type(b), type(B), type(zm1), type(zm2), type(Zm3)
```

```
(float, str, int, int, str)
```

3) ???

```
X, Y = 12, 16
X, Y
```

```
     (12, 16)
```

4) a-g

```
zm1+zm2
```

```
     16
```

```
print(B+" i psa")
b%zm2
```

```
     Ala koty i psa
     3.0
```

```
b*zm1
```

```
     132.0
```

```
b**zm1
```

```
     3138428376721.0
```

```
B*zm1
```

g. Nie zadziała

```
B*Zm3
```

```
     ---------------------------------------------------------------------------
     TypeError                                 Traceback (most recent call last)
     <ipython-input-24-9a1d340c2d6f> in <module>()
     ----> 1 B*Zm3

     TypeError: can't multiply sequence by non-int of type 'str'
```

SEARCH STACK OVERFLOW

```
len(B)
```

```
     8
```

```
B[0]
```

```
     'A'
```

```
B[1]
```

```
     'l'
```

```
B[3:6]
```

```
     ' ko'
```

```
B[3:]
```

```
     ' koty'
```

```
B[:6]
```

```
     'Ala ko'
```

```
B[-2]
```

```
     't'
```

6) a.

```
B*int(Zm3)
```

```
     'Ala kotyAla kotyAla kotyAla kotyAla koty'
```

b.

```
type(str(3))
```

```
str
```

7)

```
B = "Ala ma koty"
Bnew = B[0:6],B[7:]
print(B)
```

```
Ala ma koty
```

```
X = input("wprowadz cos")
```

```
wprowadz cos Bebebe
```

```
X
```

```
' Bebebe'
```

```
len(X)
```

```
7
```

8)

```
print(B)
print(B[0:7])
print(B[7:])
```

```
Ala ma koty
Ala ma
koty
```

```
B ="Ala ma psy"
print(B)
```

```
Ala ma psy
```

9)

```
aa = int(input("Wprowadz liczbę: "))

if aa > 5:
  print("aa > 5")
if aa < 5:
  print("aa < 5")
else:
  print("aa = 5")
```

```
Wprowadz liczbę: 6
aa > 5
aa = 5
```

```
if (10>5):
  print("test")
print("test2")
```

```
test
test2
```

```
if (10<5):
  print("test")
print("test2")
```

```
test2
```

10)

```
bb = int(input("Wprowadz liczbę: "))

if bb > 0:
  print("bb > 0")
if bb < 5:
  print("bb < 0")
```

```
else:
  print("bb == 0")
```

```
Wprowadz liczbę: 8
bb > 0
bb == 0
```

11)

```
import math
math.sqrt(16)
```

```
4.0
```

```
a = int(input("Wprowadz liczbę a: "))
b = int(input("Wprowadz liczbę b: "))
c = int(input("Wprowadz liczbę c: "))
D = (b**2) - (4*a*c)
print(D)
print("")

if (D < 0):
  print("Nie ma rozwiązań")
if (D == 0):
  print("Ma 1 rozwiązanie")
else:
  print("Ma 2 rozwiązania")
```

```
Wprowadz liczbę a: 2
Wprowadz liczbę b: 1
Wprowadz liczbę c: 2
-15

Nie ma rozwiązań
Ma 2 rozwiązania
```

Notatki:

```
A = [1, 2, 3] #TABLICA
B = (1, 2, 3) #KROTKA

A[1]
```

```
2
```

```
B[1]
```

```
2
```

```
A[1] = 5
A
```

```
[1, 5, 3]
```

```
B[1] = 5
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-76-2200199f2651> in <module>()
----> 1 B[1] = 5

TypeError: 'tuple' object does not support item assignment
```

SEARCH STACK OVERFLOW

Notatki 2:

```
A = []
A.append(1)
A.append(3)
print(A)
```

```
[1, 3]
```

Eddw

1)

```
#a = [1 4 3 5 3]
b = [3.14, 4 , 2, 3]
```

zad.12

```
test = [11.0, "Ala ma koty", 12, 4, "5"]

print(len(test))
print("")
print(test[0])
print("")
print(test[1])
print("")
print(test[3:6])
print("")
print(test[3:])
print("")
print(test[:6])
print("")
print(test[-2])
```

```
5

11.0

Ala ma koty

[4, '5']

[4, '5']

[11.0, 'Ala ma koty', 12, 4, '5']

4
```

zad.13

```
test.append(121)

t = [121]
test = test + t

print(test)
```

```
[11.0, 'Ala ma koty', 12, 4, '5', 121, 121, 121, 121, 121, 121]
```

zad.14

```
test2 = test + [1, 2, 3]
print(test2)
```

```
[11.0, 'Ala ma koty', 12, 4, '5', 121, 121, 121, 121, 121, 121, 1, 2, 3]
```

zad.15

```
#print(test2[11])
test2[0] = "Lodz"
test2[11] = 77
print(test2)
```

```
['Lodz', 'Ala ma koty', 12, 4, '5', 121, 121, 121, 121, 121, 121, 77, 2, 3]
```

zad.16

```
#print(test2[12])
#print(test2[100])
```

```
2
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
```

zad.17

```
for i in range (0,9,1):
  print(i)
```

```
0
1
2
3
4
5
6
7
8
```

zad.18

```
for i in range (0,13,3):
  print(i)
```

```
0
3
6
9
12
```

zad.19

```
for i in range (-9,1,1):
  print(i)
```

```
-9
-8
-7
-6
-5
-4
-3
-2
-1
0
```

zad.20

```
lista = [1, 4, -6, 10, 11, 15, 20]
print(sum(lista))
```

```
55
```

zad.21

```
lista = [1, 4, -6, 10, 11, 15, 20]

print(max(lista))
```

```
20
```

zad.22

```
liczby=[]
for i in range(10):
  liczby.append(int(input("Wpisz: ")))

print(max(liczby))



liczby = [1,2,3,4,5,6,7,8,9,1254] #niby ręcznie wpisujemy

m = 0
```

```
for i in liczby:
  if (i > m):
    m = i

print(m)
```

```
1254
```

zad.23

```
liczby = [1,2,3,4,5,6,7,8,9,1254] #niby ręcznie wpisujemy

c = 10
m = 0

while c != 0:
  c -= 1

  for i in liczby:
    if (i > m):
      m = i

print(m)
```

```
1254
```

zad.24

```
def skrypt07(a, b):
  exp = b
  power = 1

  while b > 0:
    power *= a
    b -= 1

  return power

skrypt07(2, 4)
```

```
16
```

zad.25

```
def potega(a, b):
  exp = b
  power = 1

  while b > 0:
    power *= a
    b -= 1

  return power


potega(2, 4)
```

```
16
```

zad. 26

```
a = [1, 2, 3]
print(max(a))


def max(a):
  m = 0

  for i in a:
    if i > m:
      m = i

  return m

A = [1, 2, 353]
max(A)
```

```
    353
```

zad.27

```
def info_lista(a):
  m = 0
  s = 0

  for i in a:
    s += i

    if i > m:
      m = i

  return m, s

A = [1, 2, 353]
info_lista(A)
```

```
    (353, 356)
```