

Wstęp do Informatyki dla Bioinformatyków

Wykład 2.
Jak to się robi w pythonie?

Bartek Wilczyński

12. X 2015

Liczby w pythonie

```
>>> 2 + 2
```

```
4
```

```
>>> 2 / 3
```

```
0
```

```
>>> 2 / 3.0
```

```
0.6666666666666666
```

```
>>> 2 % 3
```

```
2
```

```
>>> 2 ** 3
```

```
8
```

```
>>> 2 ** 1000
```

```
107150860718626732094842504906000181056140481170553360744375038837035105112493611
```

```
224931983788156958581275946729175531468251871452856923140435984577574698574803931
```

```
6456777482423098542107460506237114187795418215304647498358194126739876755916554391
```

```
646077062914571196477686542167660429831652624386837205668069376L
```

Inne rodzaje wartości

- None
- Napisy
- Wartości logiczne
- Funkcje
- Listy
- Słowniki
-

```
>>> print None
None
>>> 'Alfabet ' + "Morse'a"
"Alfabet Morse'a"
>>> 3 == 4
False
>>> 4 >= 3
True
>>> 2 != 1
True
>>> len
<built-in function len>
>>> len("Ala ma kota")
11
_
```

Coś o listach i ich zachowaniu

- Listy można dodawać
- Wyjmować z nich elementy
- Wybierać zakresy
- Przechowywać na nich elementy różnych rodzajów (typów)

```
>>> [1,2,3]
[1, 2, 3]
>>> [1,2,3] + [4,5,6]
[1, 2, 3, 4, 5, 6]
>>> len([1,2,3,4])
4
>>> [1,2,3,4][1]
2
>>> [1,2,3,4][-1]
4
>>> [1,2,3,4][1:3]
[2, 3]
>>> [1,2,3,4][1:4:2]
[2, 4]
>>> [1,[1,2,3],[4],[ ]]
[1, [1, 2, 3], [4], []]
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Zmienne, czyli o zapamiętywaniu

```
>>> import math
>>> print math.sqrt(4*4+3*3)
5.0
>>>
>>> a=3
>>> b=4
>>> c=math.sqrt(a*a+b*b)
>>> print c
5.0
```

```
>>> l=[1,2,3]
>>> l += [4,5,6]
>>> l.append(7)
>>> print l
[1, 2, 3, 4, 5, 6, 7]
>>> l[2]=2
>>> l
[1, 2, 2, 4, 5, 6, 7]
>>> l[3],l[4] = l[4],l[3]
>>> l
[1, 2, 2, 5, 4, 6, 7]
```

Instrukcje warunkowe, czyli czasem trzeba podjąć decyzję

```
>>> kwota = 50000
>>> prog_1 = 35000
>>> prog_2 = 80000
>>> stawka_1= 0.19
>>> stawka_2 = 0.33
>>> if kwota<prog_1: # zwolniony z podatku
...     podatek=0.0
... elif kwota< prog_2: # prog_1 <= kwota < prog_2
...     podatek=(kwota-prog_1)*stawka_1
... else: # kwota >=prog_2
...     podatek = (kwota-prog_1)*stawka_1 + (kwota-prog_2)* (stawka_2-stawka_1)
...
>>> print podatek
2850.0
```


Iterowanie, czyli powtarzanie wg listy

```
>>> l=[1,2,3,4,5,6,7]
>>> i=0
>>> while i<len(l):
...     l[i]= l[i]*l[i]
...     i+=1
...
>>> l
[1, 4, 9, 16, 25, 36, 49]
>>> l=[1,2,3,4,5,6,7]
>>> l2=[]
>>> for x in l:
...     l2.append(x*x)
...
>>> l2
[1, 4, 9, 16, 25, 36, 49]
```

Różne sposoby
obliczania kwadratów
liczb z listy

```
>>> l=[1,2,3,4,5,6,7]
>>> for (i,x) in enumerate(l):
...     l[i]=x*x
...
>>> l
[1, 4, 9, 16, 25, 36, 49]
>>>
```


Funkcje, czyli jak zapamiętać metodę

```
>>> def podatek(kwota, prog_1=35000, stawka_1=0.19, prog_2= 80000, stawka_2 = 0.33 ):
...     if kwota<prog_1: #zwolnienie
...         podatek=0.0
...     elif kwota<prog_2: # prog_1 <= kwota < prog_2
...         podatek = (kwota-prog_1) * stawka_1
...     else: # prog_2 <= kwota
...         podatek = (kwota - prog_1) * stawka_1 + (kwota - prog_2) * (stawka_2 - stawka_1)
...     return podatek
...
>>> podatek(60000)
4750.0
>>> podatek(50000)
2850.0
>>> podatek(120000)
21750.0
>>> podatek(120000,prog_1=45000)
19850.0
```

Inne przykłady funkcji

```
>>> def kwadraty(l):
...     l2=[]
...     for x in l:
...         l2.append(x*x)
...     return l2
...
>>> kwadraty([1,2,3,4])
[1, 4, 9, 16]
>>> kwadraty(kwadraty([1,2,3,4]))
[1, 16, 81, 256]
```

```
>>> def dwa_razy(x):
...     return 2*x
...
>>> dwa_razy(3)
6
>>> dwa_razy("Lala")
'LalaLala'
>>> dwa_razy([1,2,3])
[[1, 2, 3, 1, 2, 3]]
```

Parę sztuczek z listami

- Wyrażenia listowe (ang. List comprehension)

```
>>> l=[1,2,3,4,5,6,7]
>>> [x*x for x in l]
[1, 4, 9, 16, 25, 36, 49]
>>> [x*x for x in l if x % 2 == 0]
[4, 16, 36]
```

- Sprawdzanie zawartości listy

```
>>> 3 in range(10)[::3]
True
>>> 3 in range(10)[1:7:3]
False
>>> 3 in range(10)[1:7:2]
True
```

Liczby pierwsze

- Liczba naturalna jest pierwsza, gdy ma dokładnie dwa dzielniki naturalne
- Zadanie: Znajdź liczby pierwsze z przedziału od 2 do N
- Obserwacja: jeśli liczba ma dzielniki naturalne, to ma też dzielniki pierwsze
- Rozwiązanie: Sito Eratostenesa

Dominanta

- Zadanie: Mamy tablicę N liczb. Czy istnieje wartość powtarzająca się na tej liście więcej niż $N/2$ razy (dominująca listę)?
- Obserwacja 0: naiwne metody są powolne
- Obserwacja 1: można posortować tablicę
- Obserwacja 2: Jeśli mamy kandydata, to sprawdzić czy jest on dominantą można łatwo
- Obserwacja 3: Jeśli x jest dominantą, to pojawia się w tablicy “gęsto”
- Rozwiązanie ?