

Wizualizacja danych

- wykład 3

dr Piotr Jastrzębski

Wstęp do języka Python - cd.

Struktury danych w Pythonie

- ▶ listy
- ▶ zbiory
- ▶ krotki
- ▶ słowniki

{Ostatnia aktualizacja pliku: 2020-03-19 22:13:03.}

Listy

Listy w Pythonie mogą przechowywać elementy różnych typów.

```
list1 = ['raz', 'dwa', 5, 5];  
list2 = [1, 2, 3, 4, 5 ];  
list3 = ["a", "b", "c", "d"];  
print(list3)
```

```
## ['a', 'b', 'c', 'd']
```

```
list4 = ['s', 'ww', True, 5]
print(list4[3])
```

```
## 5
```

```
list4[1] = True
print(list4[1])
```

```
## True
```

```
print(list4[-1])
```

```
## 5
```

```
print(list4[2:])
```

```
## [True, 5]
```

```
print(len([2, 3, 4]))
```

```
## 3
```

```
print([1, 2, 3] + [4, 5, 6])
```

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

```
print(['Hi!'] * 4)
```

```
## ['Hi!', 'Hi!', 'Hi!', 'Hi!']
```

```
print(3 in [1, 2, 3])
```

```
## True
```

```
lista = ['a', 'b', 34, 5.6, True]
lista.append('5')
print(lista)
```

```
## ['a', 'b', 34, 5.6, True, '5']
```

```
lista.extend([4, 5, 6])
print(lista)
```

```
## ['a', 'b', 34, 5.6, True, '5', 4, 5, 6]
```

```
lista.insert(2, 'w')  
print(lista)
```

```
## ['a', 'b', 'w', 34, 5.6, True, '5', 4, 5, 6]
```

```
lista.remove(True)  
print(lista)
```

```
## ['a', 'b', 'w', 34, 5.6, '5', 4, 5, 6]
```



```
lista.pop()
```

```
## 6
```

```
print(lista)
```

```
## ['a', 'b', 'w', 34, 5.6, '5', 4, 5]
```

```
lista.pop(4)
```

```
## 5.6
```

```
print(lista)
```

```
## ['a', 'b', 'w', 34, '5', 4, 5]
```

```
lista.pop(-2)
```

```
## 4
```

```
print(lista)
```

```
## ['a', 'b', 'w', 34, '5', 5]
```

```
lista.pop(0)
```

```
## 'a'
```

```
print(lista)
```

```
## ['b', 'w', 34, '5', 5]
```

```
lista.clear()  
print(lista)
```

```
## []
```

Alternatywnie: `del lista[:]`.

```
lista2 = ['a', 'b', 5, 'A', 'a', 'b']  
print(lista2.index('a'))
```

```
## 0
```

```
print(lista2.index('a', 3))
```

```
## 4
```

```
print(lista2.index('a', 1, 4))
```

```
## ValueError: 'a' is not in list
```

```
print(lista2.index('a', 1, 5))
```

```
## 4
```

```
lista2.reverse()
```

```
print(lista2)
```

```
## ['b', 'a', 'A', 5, 'b', 'a']
```

```
lista3 = ['a', 'b', 'A', 'a', 'b']
```

```
lista3.sort()
```

```
print(lista3)
```

```
## ['A', 'a', 'a', 'b', 'b']
```

```
lista4 = lista3.copy()
print(lista4)
```

```
## ['A', 'a', 'a', 'b', 'b']
```

Lista jako stos

```
stack = [3, 4, 5, 8, 9]
stack.append(6)
stack.append(7)
print(stack)
```

```
## [3, 4, 5, 8, 9, 6, 7]
```

```
print(stack.pop())
```

```
## 7
```

```
print(stack)
```

```
## [3, 4, 5, 8, 9, 6]
```

Lista jako kolejka

```
from collections import deque
```

```
queue = deque(["aw", "tg", "kj"])  
queue.append("gg")  
print(queue)
```

```
## deque(['aw', 'tg', 'kj', 'gg'])
```

```
print(queue.popleft())
```

```
## aw
```

```
print(queue)
```

```
## deque(['tg', 'kj', 'gg'])
```


List Comprehensions

```
squares = []  
for x in range(5):  
    squares.append(x ** 2)  
  
print(squares)
```

```
## [0, 1, 4, 9, 16]
```

```
squares = [x**2 for x in range(5)]  
print(squares)
```

```
## [0, 1, 4, 9, 16]
```

Krotka - tuple

```
krotka = 123, 'abc', True  
print(krotka[2])
```

```
## True
```

```
krotka[0] = 1
```

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

Zbiór - set

```
cyfry = {'raz', 'dwa', 'raz', 'trzy', 'raz', 'osiem'}  
print(cyfry)
```

```
## {'dwa', 'trzy', 'raz', 'osiem'}
```

Słownik

```
tel = {'jack': 4098, 'sape': 4139}
tel['guido'] = 4127
print(tel)
```

```
## {'jack': 4098, 'sape': 4139, 'guido': 4127}
```

```
tel['jack']
```

```
## 4098
```

```
del tel['sape']
tel['irv'] = 4127
print(tel)
```

```
## {'jack': 4098, 'guido': 4127, 'irv': 4127}
```

```
print(list(tel))
```

```
## ['jack', 'guido', 'irv']
```

```
print(sorted(tel))
```

```
## ['guido', 'irv', 'jack']
```

Odpowiedź na pytanie z poprzedniego wykładu:

“Mutable” - zmienne typy::

- ▶ list
- ▶ dictionary
- ▶ set
- ▶ bytearray
- ▶ user defined classes

“Inmutable” - niezmiennie typy:

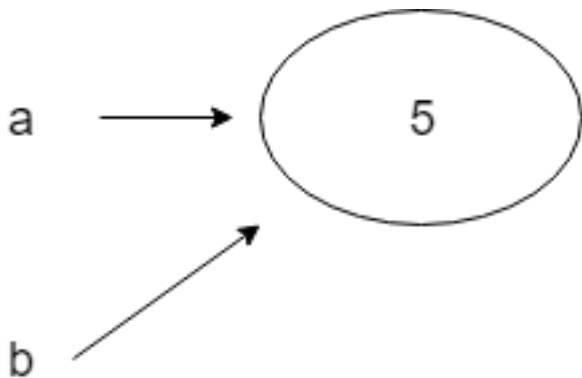
- ▶ int
- ▶ float
- ▶ decimal
- ▶ complex
- ▶ bool
- ▶ string
- ▶ tuple
- ▶ range
- ▶ frozenset
- ▶ bytes

```
a = 5  
b = a  
b += 2  
print(a)
```

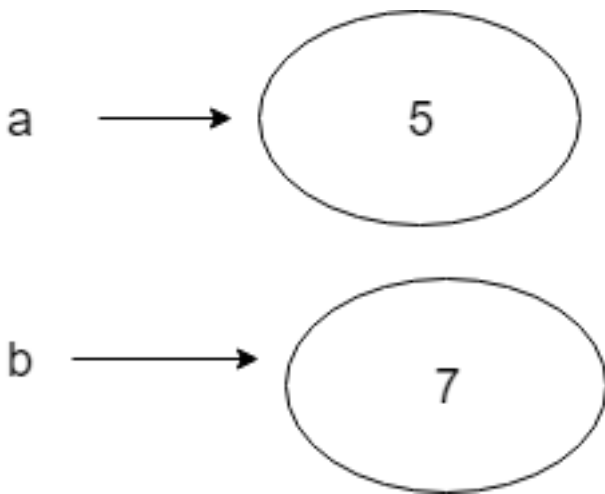
```
## 5
```

```
print(b)
```

```
## 7
```

Rysunek 1: Dwie pierwsze linijki



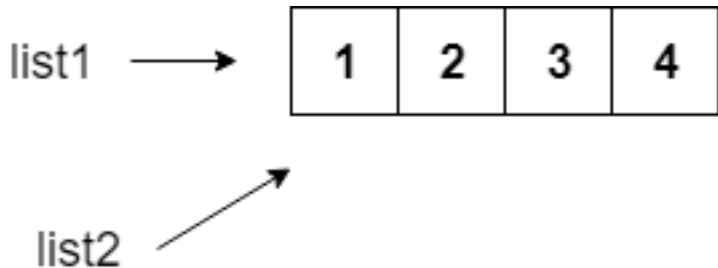
Rysunek 2: $b=+2$

```
list1 = [1, 2, 3, 4]
list2 = list1
list1[2] = 'a'
print(list1)
```

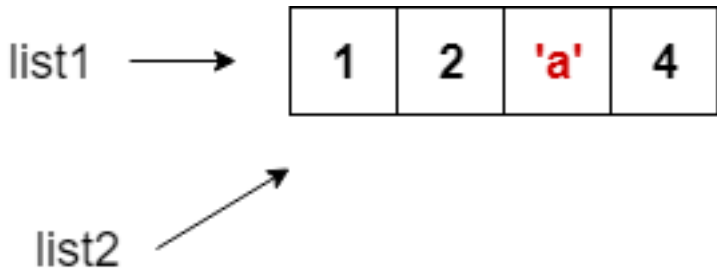
```
## [1, 2, 'a', 4]
```

```
print(list2)
```

```
## [1, 2, 'a', 4]
```



Rysunek 3: Dwie pierwsze linijki.



Rysunek 4: `list1[2] = 'a'`

Bibliografia

- ▶ <https://pl.wikipedia.org/wiki/Python>, dostęp online 12.02.2019.
- ▶ <https://bulldogjob.pl/news/264-java-php-ruby-jak-wlasciwie-wymawiac-nazwy-technologie>. dostęp online 12.02.2019.
- ▶ https://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html, dostęp online 14.02.2019.
- ▶ K. Ropiak, Wprowadzenie do języka Python, <http://wmii.uwm.edu.pl/~kropiak/wd/Wprowadzenie%20do%20j%C4%99zyka%20Python.pdf>, dostęp online 14.02.2019.
- ▶ B. Slatkin, Efektywny Python. 59 sposobów na lepszy kod, Helion 2015.

Bibliografia - cd2

- ▶ <https://docs.python.org/3/tutorial/datastructures.html>, dostęp online 1.03.2019.
- ▶ https://www.python-course.eu/python3_functions.php, dostęp online 2.03.2019.
- ▶ https://www.tutorialspoint.com/python3/python_functions.htm, dostęp online 2.03.2019.
- ▶ https://www.tutorialspoint.com/python3/python_classes_objects.htm, dostęp online 3.03.2019.
- ▶ <https://pl.wikipedia.org/wiki/Wizualizacja>