

Wizualizacja danych

- wykład 4

dr Piotr Jastrzębski

Funkcje

```
def functionname( parameters ):  
    "function_docstring"  
    function_suite  
    return [expression]
```

```
def printme(str):  
    """Funkcja wyświetlająca string"""  
    print(str)  
    return
```

```
printme("abc")
```

```
## abc
```

```
print(printme.__doc__)
```

```
## Funkcja wyświetlająca string
```

Przekazywanie przez referencję

```
def changeme(lista):  
    print("Przed zmianą: ", lista)  
    lista[2] = 50  
    print("Po zmianie: ", lista)  
    return
```

```
mylist = [10, 20, 30]  
changeme(mylist)
```

```
## Przed zmianą: [10, 20, 30]
```

```
## Po zmianie: [10, 20, 50]
```

```
print("Poza funkcją: ", mylist)
```

```
## Poza funkcją: [10, 20, 50]
```

```
def changeme(lista):  
    lista = [2, 3, 4]  
    print("Wewnątrz funkcji: ", lista)  
    return
```

```
lista = [10, 20, 30]  
changeme(lista)
```

```
## Wewnątrz funkcji: [2, 3, 4]
```

```
print("Poza funkcją: ", lista)
```

```
## Poza funkcją: [10, 20, 30]
```

```
def changeme():  
    global lista  
    lista = [2, 3, 4]  
    print("Wewnątrz funkcji: ", lista)  
    return
```

```
changeme()
```

```
## Wewnątrz funkcji: [2, 3, 4]
```

```
print("Poza funkcją: ", lista)
```

```
## Poza funkcją: [2, 3, 4]
```

Obowiązkowy argument

```
def printme(str):  
    print(str)  
    return
```

```
printme()
```

```
## TypeError: printme() missing 1 required  
positional argument: 'str'
```

Keyword argument

```
def kwadrat(a):  
    return a*a
```

```
print(kwadrat(a=4))
```

```
## 16
```


Domyślny argument

```
def sumsub(a, b, c=0, d=0):  
    return a - b + c - d
```

```
print(sumsub(12, 4))
```

```
## 8
```

```
print(sumsub(3, 4, 5, 7))
```

```
## -3
```

```
def srednia(first, *values):  
    return (first + sum(values)) / (1 + len(values))
```

```
print(srednia(2, 3, 4, 6))
```

```
## 3.75
```

```
print(srednia(45))
```

```
## 45.0
```

```
def f(**kwargs):  
    print(kwargs)
```

```
f()
```

```
## {}
```

```
f(pl="Polish", en="English")
```

```
## {'pl': 'Polish', 'en': 'English'}
```

Funkcje matematyczne

Link do dokumentacji <https://docs.python.org/3/library/math.html>

```
import math
```

```
a=0
```

```
b=math.sin(2*math.pi)
```

```
print(b)
```

```
## -2.4492935982947064e-16
```

```
print(math.isclose(a,b, rel_tol=1e-09, abs_tol=1e-09))
```

```
## True
```

Programowanie obiektowe w Pythonie



Rysunek 1: Lego jako model programowanie obiektowego

```
class Employee:
    """Common base class for all employees"""
    empCount = 0

    def __init__(self, name, salary):
        self.name = name
        self.salary = salary
        Employee.empCount += 1

    def displayCount(self):
        print("Total Employee %d" % Employee.empCount)

    def displayEmployee(self):
        print("Name : ", self.name, ", Salary: ", self.salary)
```

```
emp1 = Employee("John", 2000)
emp2 = Employee("Anna", 5000)
emp1.displayEmployee()
```

```
## Name : John , Salary: 2000
```

```
emp2.displayEmployee()
```

```
## Name : Anna , Salary: 5000
```

Wizualizacja danych

Czym zajmuje się wizualizacja danych?

Wizualizacja – ogólna nazwa graficznych metod tworzenia, analizy i przekazywania informacji. Za pomocą środków wizualnych ludzie wymieniają się zarówno ideami abstrakcyjnymi, jak i komunikatami mającymi bezpośrednie oparcie w rzeczywistości. W dzisiejszych czasach wizualizacja wpływa na sposób prowadzenia badań naukowych, jest rutynowo wykorzystywana w dyscyplinach technicznych i medycynie, służy celom dydaktycznym, a także bywa pojmowana jako środek wyrazu artystycznego.

Wizualizacja danych to zagadnienie ich obrazowego przedstawienia. Dane są rozumiane jako „informacje, które zostały zestawione w pewnej schematycznej formie, np. zmiennych lub współrzędnych”. Według Friedmana jej głównym celem jest skuteczny i zrozumiały przekaz zawartych w nich treści. Jednym z najczęściej popełnianych błędów bywa przykładanie zbytnej uwagi do formy komunikatu, który przestaje spełniać swoje zasadnicze zadanie. Odmienny pogląd na sens tej dziedziny wyrażają Fernanda Viegas i Martin M. Wattenberg, akcentując rolę pozyskania uwagi potencjalnego odbiorcy. Odpowiedni sposób przedstawienia danych pozwala na poprawne i szybkie zrozumienie zależności opisanych przez dane. Nieodpowiedni sposób prezentacji prowadzi do powstawania celowych lub przypadkowych zniekształceń w postrzeganiu zależności obecnych w danych.

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>