

Programowanie obiektowe

2021L

Co było ostatnio?
(04.03.2021)

Co było ostatnio?

- PVM, Moduły, Pakiety;
- typy danych (dynamiczne typowanie, „rzutowanie”);
- Struktury danych (mutable; immutable):
 - lista (list);
 - krotka (tuple);
 - słownik (dict);
 - zbiór (set, frozenset);
- Zaawansowane listy (list comprehension).

Python... użyteczne funkcje i metody

Python... użyteczne funkcje i metody

sort, sorted

Sorted – funkcja zwracająca nową, posortowaną listę z elementami sortowanej kolekcji.

```
In [1]: numbers = [5, 1, 4, 2, 3]
In [2]: sort_numbers = sorted(numbers)
In [3]: sort_numbers
Out[3]: [1, 2, 3, 4, 5]
In [4]: numbers
Out[4]: [5, 1, 4, 2, 3]
```

Sort – metoda sortująca listę w miejscu (nadpisuje).

```
In [1]: numbers = [5, 1, 4, 2, 3]
In [2]: numbers.sort()

In [3]: numbers
Out[3]: [1, 2, 3, 4, 5]
```

Python... użyteczne funkcje i metody

sort, sorted
f. anonimowe

- Czym jest funkcja anonimowa?
- Dlaczego używamy?
 - Zależna od stylu programowania;
 - funkcje wyższego rzędu oczekują, jako argument wejściowy, innej funkcji;
 - do krótkich „jednorazowych” wywołań (tu i teraz);
 - do operacji zmniejszania liczby argumentów wywołania.

```
In [1]: lambda x: x**2  
Out[1]: <function __main__.<lambda>(x)>
```

```
In [2]: f = lambda x: x**2  
In [3]: f(4)  
Out[3]: 16
```

```
In [4]: (lambda x: x**2)(4)  
Out[4]: 16
```

Python... użyteczne funkcje i metody

sort, sorted
f. anonimowe
f. wyższego
rzędu

Do najpopularniejszych należą: **filter**, **map**, reduce, np.:

- filtrowanie (**filter**)

```
In [1]: num = [1, 2, 3, 4, -3]
In [2]: filter(lambda x: x > 2, num)
Out[2]: <filter at 0x7fe6878f0550>

In [3]: list(filter(lambda x: x > 2, num))
Out[3]: [3, 4]
```

- mapowanie (**map**)

```
In [4]: list(map(lambda x: x > 2, num))
Out[4]: [False, False, True, True, False]

In [5]: set(map(lambda x: x > 2, num))
Out[5]: {False, True}
```

Python... użyteczne funkcje i metody

sort, sorted
f. anonimowe
f. wyższego
rzędu

- mapowanie (**map**)

```
names = ["Jacek", "Robert", "Bartek", "Wiktor",  
         "Paweł"]  
order = [i for i in range(5)]  
random.shuffle(order)  
  
order  
[3, 4, 0, 2, 1]  
names  
['Jacek', 'Robert', 'Bartek', 'Wiktor', 'Paweł']  
order_names = list(map(lambda order_id:  
                        names[order_id], order)  
order_names  
['Wiktor', 'Paweł', 'Jacek', 'Bartek', 'Robert']
```


Zagadka

Zagadka – co robi ten kod?

I dlaczego nie należy tak pisać?

```
(lambda _: list(map(lambda _: _ // 2, _)))([1,2,3,4,5,6,7,8,9,10])
```

```
(lambda _: set(map(lambda _: _ % 2, _)))([1,2,3,4,5,6,7,8,9,10])
```

Covid-19

Live

Programowanie obiektowe

(wreszcie!)

Programowanie obiektowe

Paradygmat programowania obiektowego

- klasa;
- obiekt;
- konstruktor (metoda specjalna);
- pole klasy / obiektu;
- metoda klasy / obiektu.

```
stud_1 = Student("Paweł", 18,  
12345)
```

```
class Student:  
# ...  
    def to_dict(self):  
        return {"name": self.name,  
                "age": self.age,  
                "id": self.id}
```

Dziękuję za uwagę.