Web Development with Python Lesson 6

>>>

Ing. Patrik Dendis

STEP IT ACADEMY BRATISLAVA


YOU'RE SLOW
QUICK SORT
MERGE SORT
HEAP SORT
SELECTION SORT
MERGE SORT
QUICK SORT
HEAP SORT
SELECTION SORT
US TOO
SELECTION SORT
BUBBLE SORT
INSERTION SORT

OBSAH PREZENTÁCIE

- Opakovanie
- Tuples
- Sets
- Dictionaries
- Praktické použitie Dictionaries

OPAKOVANIE

- Čo je sorting?
- Aký je rozdiel medzi sort a sorted v Pythone?
- Povedzte ako funguje Bubble sort
- Aké ďalšie algoritmy na sortovanie poznáte?
- Ako funguje lineárne hľadanie?
- Na ako zozname môžeme použiť binárne hľadanie?
- Čo je to zložitosť pri algoritmoch a aké dva typy máme?

N-TICA TUPLE

structure for storing multiple values in one variable but **immutable**

main.py > ...

```
1  userTypes = ('admin','student','teacher','moderator')
2  print(userTypes)
3  print(userTypes[1])
```

Run

```
('admin', 'student', 'teacher', 'moderator')
student
```

METÓDY V TUPLE

python tuple methods

- **count()** - returns the number of times a specified value occurs in a tuple
- **index()** - searches the tuple for a specified value and returns the position of where it was found

PRIKLÁDY TUPLE

main.py > ...

```
1  userTypes = ('admin','student','teacher','moderator')
2  print(len(userTypes))
3  print(userTypes[1:3])
4  print(userTypes[-2])
5  print(type(userTypes))
6  print(list(userTypes))
7  userTypes[1] = 'basic_user'
8
```

ZADANIE TUPLE

Ako pridáte prvok do Tuple ak Tuple je nemenná dátová štruktúra?

ZADANIE TUPLE

Napíšte funkciu, ktorá Vám vráti najstaršieho študenta v zozname študentov.

```
# (meno, vek)
student1 = ("Jana", 10)
student2 = ("Lukáš", 11)
student3 = ("Ema", 40)
```

MNOŽINY SET

structure for storing multiple values in one variable but **without duplications and random order**

main.py > ...

```
1  mnozina = {1, "asd", 2, ("Jana", 10), False}
2  print(mnozina)
```

Run

```
{False, 1, 2, ('Jana', 10), 'asd'}
```

PRÍKLADY V SET

main.py > ...

```python
mena = {'Joe','Bob','Kate','Bob'}
print(mena)

first_set = {"apple", "banana", "cherry"}
first_set.add("orange")

second_set = {"google", "microsoft", "apple"}

first_set.update(second_set)
print(first_set)

first_set.remove("apple")
print(first_set)
first_set.discard("apple")
print(first_set)
first_set.remove("apple")
print(first_set)

```

METÓDY V SET

the **intersection()** method returns a set that contains the similarity between two or more sets. Logical operator **AND.**

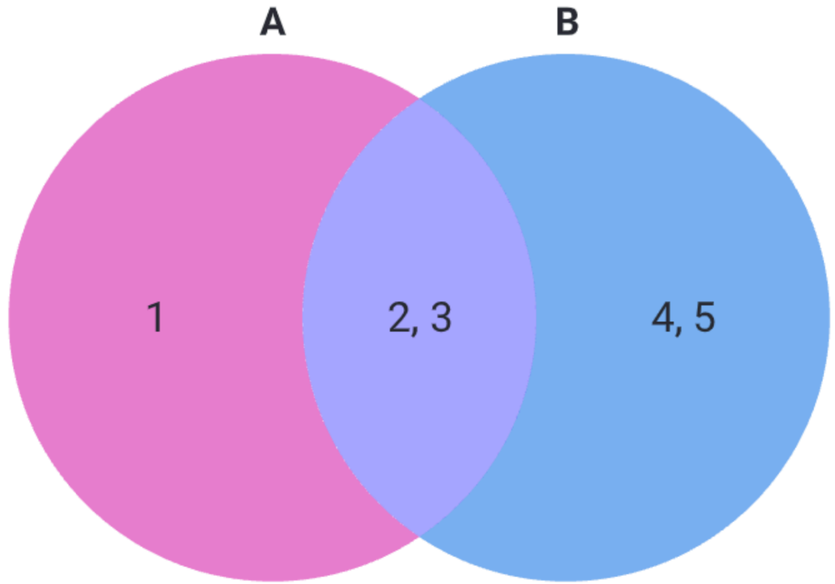
main.py > ...

```python
first_set = {"apple", "banana", "cherry"}
second_set = {"google", "microsoft", "apple"}

vzajomne = first_set.intersection(second_set)
print(vzajomne)
```

Run

```
{'apple'}
```



METÓDY V SET

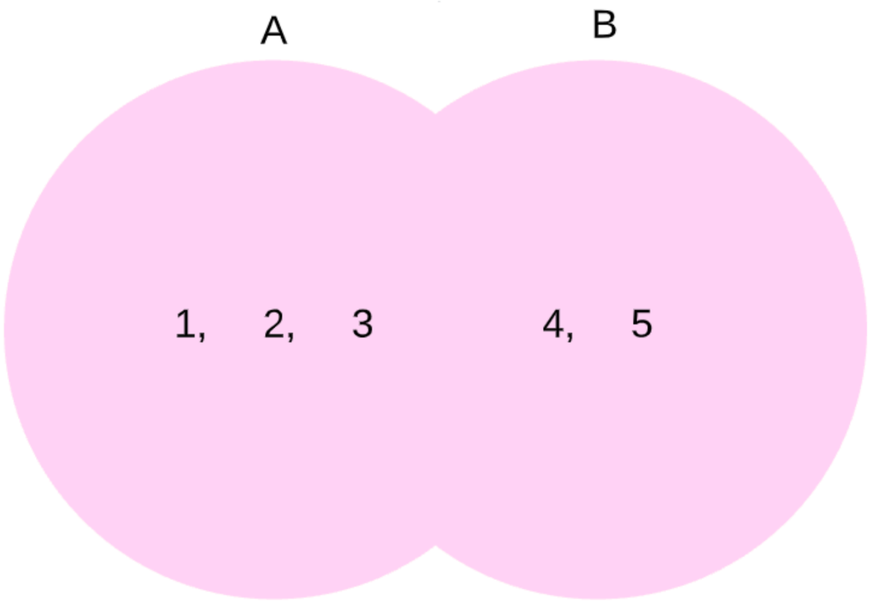
the **union()** method returns a set that contains all items from the original set, and all items from the specified set(s). Logical operator **OR.**

main.py > ...

```
1  first_set = {"apple", "banana", "cherry"}
2  second_set = {"google", "microsoft", "apple"}
3
4  dokopy = first_set.union(second_set)
5  print(dokopy)
```

Run

```
{'apple', 'cherry', 'microsoft', 'google', 'banana'}
```



METÓDY V SET

the **difference()** method returns a set that contains the difference between two sets

main.py > first_set

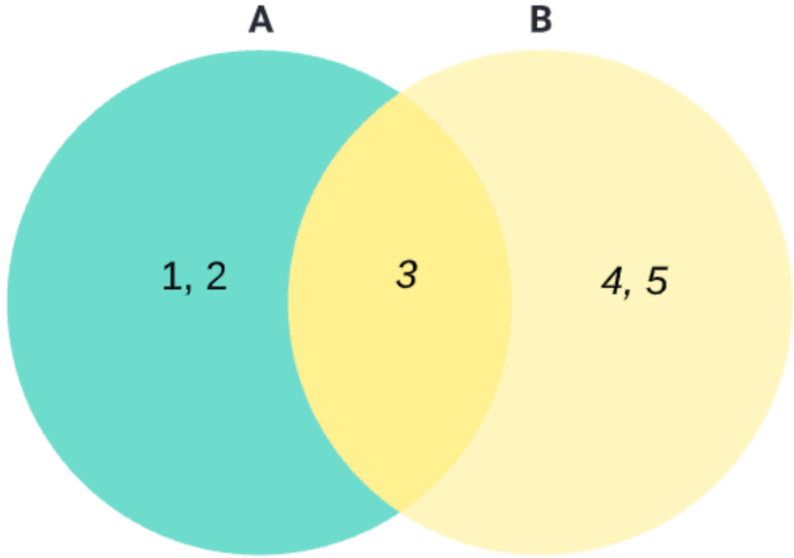
```python
first_set = {"apple", "banana", "cherry"}
second_set = {"google", "microsoft", "apple"}

rozdiely = first_set.difference(second_set)
print(rozdiely)

```

Run

```
{'banana', 'cherry'}
```



METÓDY V SET

python set methods

https://www.w3schools.com/python/python_ref_set.asp

FROZENSET

Immutable set

```
nemenny_zoznam = frozenset({"Milan", "Peter", "Patrik", "Milan"})
print(nemenny_zoznam)
nemenny_zoznam.add("Jozo")
```

ZADANIE SET

the **set()** method is used to convert any of the iterable to sequence of iterable elements with distinct elements

You have three tuples of integers.
Find elements present in all tuples.

SLOVNÍK DICTIONARY

structure for storing multiple values identified by key-value

main.py > ...

```python
student = {
    "meno": "Jozko Mrkvicka",
    "vek": 23,
    "zamestnanie": "lekar",
    "aktivny_student": True
}

print(student["meno"])
```

METÓDY V DICTIONARY

main.py > ...

```python
student = {
  "meno": "Jozko Mrkvicka",
  "vek": 23,
  "zamestnanie": "lekar",
  "aktivny_student": True
}

student["vyska"] = 180
print(student.keys())
print(student.items())
print(student.values())
print(student.get("hmotnost"))
print(student.get("hmotnost", 80))
```

Run

```
dict_keys(['meno', 'vek', 'zamestnanie', 'aktivny_student', 'vyska'])
dict_items([('meno', 'Jozko Mrkvicka'), ('vek', 23), ('zamestnanie', 'lekar'), ('aktivny_student', True), ('vyska', 180)])
dict_values(['Jozko Mrkvicka', 23, 'lekar', True, 180])
None
80
```

METÓDY V DICTIONARY

python dictionary methods

https://www.w3schools.com/python/python_ref_dictionary.asp

PRÍKLADY S DICTIONARY

```python
def find_user(users, name):
    for user in users:
        if user["name"] == name:
            return user
    return {}

users = [
    {'name': 'Hanna', 'age': 10, 'login':'user56'},
    {'name': 'Mark', 'age': 15, 'login':'usER111'},
    {'name': 'Jane', 'age': 17, 'login':'superGirl'},
    {'name': 'Jack', 'age': 7, 'login':'userJack'}
]

finding_name = input("Zadaj meno pouzivatela: ")
finded_user = find_user(users, finding_name)
print(finded_user.get("age", "Nenaslo sa"))
```

PRÍKLADY S DICTIONARY

```python
def sort_users_by_age(users, min_age):
    return list(filter(lambda user: user.get("age", 0) > min_age, users))

users = [
    {'name': 'Hanna', 'age': 10, 'login':'user56'},
    {'name': 'Mark', 'age': 15, 'login':'usER111'},
    {'name': 'Jane', 'age': 17, 'login':'superGirl'},
    {'name': 'Jack', 'age': 7, 'login':'userJack'}
]

print(sort_users_by_age(users, 10))
```

ZADANIE

You have a tuple of integers. Display statistics on the number of digits in tuple elements.
For instance:
One digit — 3 elements
Two digits — 4 elements
Three digits — 5 elements

ZADANIE

Create an app Book Collection. Store the following information about books: author, title, genre, year of release, publisher. Provide the possibility to add, delete, search, and replace data. Use a dictionary to store information.

ĎAKUJEM ZA POZORNOSŤ