



Programování 1 (NPRG030)

Cvičení 5: Kolekce

Shrnutí přednášky: Tuple

```
1 # Inicializace n-tice (angl. tuple)  
2 tuple = (hodnota1, hodnota2, ...) # vytvoření n-tice s určenými hodnotami  
  
3 # Základní operace s n-ticí  
4 tuple[pozice]           # získání hodnoty n-tice na zadané pozici  
5 tuple.count(prvek)      # počet výskytů zadaného prvku v n-tici  
6 tuple.index(prvek)      # získání indexu prvního výskytu zadaného prvku v n-tici
```

Shrnutí přednášky: Množina

```
7 # Vytvoření množiny (angl. set)
8 mnozina = {hodnota1, hodnota2, ...} # vytvoření množiny s určenými hodnotami
9 mnozina = set(), set(hodnota), set([hodnota1, hodnota2]) # vytvoření množiny (z listu hodnot)

10 # Základní operace s množinou
11 len(mnozina) # Určení počtu prvků v množině
12 prvek in mnozina # Test, zda prvek je v množině (vrací bool)
13 prvek not in mnozina # Test, zda prvek není v množině (vrací bool)

14 # Vybrané metody množiny
15 mnozina.add(prvek) # Přidání prvku do množiny
16 mnozina.update(mnozina2) # Aktualizace množiny prvky z jiné množiny
17 mnozina.remove(prvek) # Odebrání prvku z množiny (způsobí chybu, pokud není prvek v množině)
18 mnozina.discard(prvek) # Odebrání prvku z množiny (bez chyby, pokud není prvek v množině)
19 mnozina.pop() # Odebrání a vrácení prvního prvku z množiny
20 mnozina.clear() # Vyprázdnění množiny
21 mnozina.copy() # Vytvoření (mělké) kopie množiny
22 mnozina.union(mnozina2) # Spojení dvou množin (bez duplicit), totéž jako mnozina | mnozina2
23 mnozina.intersection(mnozina2) # Průnik dvou množin, totéž jako mnozina & mnozina2
24 mnozina.difference(mnozina2) # Rozdíl dvou množin, totéž jako mnozina - mnozina2
25 mnozina.isdisjoint(mnozina2) # Test, zda jsou množiny nedisjunktní (tj. mají prázdný průnik)
26 mnozina.issubset(mnozina2) # Test, zda množina je podmnožinou jiné množiny
27 mnozina.issuperset(mnozina2) # Test, zda je množina nadmnožinou jiné množiny
```

Shrnutí přednášky: Slovník

```
28 # Inicializace slovníku (angl. dictionary)
29 slovník = { klic1 : hodnota1, klic2 : hodnota2 } # Vytvoření slovníku
30 slovník = dict([(klic1, hodnota1), (klic2, hodnota2)]) # Vytvoření slovníku z listu n-tic

31 # Základní operace se slovníkem
32 len(slovník)           # Vrátí počet prvků ve slovníku
33 slovník[klic]          # Získá hodnotu pro daný klíč (nebo KeyError, pokud klíč neobsahuje)
34 slovník[klic] = hodnota # Přidá nebo aktualizuje hodnotu pro daný klíč
35 del slovník[klic]      # Odstraní dvojici (klíč, hodnota) podle klíče (nebo způsobí KeyError)
36 for klic in slovník:   # Iterace slovníkem přes klíče
37     klic in slovník     # Test, zda klíč je ve slovníku (vrací bool)
38     klic not in slovník # Test, zda klíč není ve slovníku (vrací bool)
39 if slovník:            # Test, zda je slovník prázdný (vrací bool)

40 # Vybrané metody slovníku
41 slovník.update(slovník2) # Aktualizace slovníku s pomocí jiného slovníku
42 slovník.get(klic)        # Získá hodnotu pro daný klíč s kontrolou existence
43 slovník.pop(klic)        # Odstraní a vrátí hodnotu podle klíče
44 slovník.clear()          # Vymaže všechny položky ve slovníku
45 slovník.copy()           # Vytvoří (mělkou) kopii slovníku
46 slovník.items()          # Vrátí seznam dvojic (klíč, hodnota)
47 slovník.keys()           # Vrátí seznam klíčů, totéž jako list(slovník)
48 slovník.values()         # Vrátí seznam hodnot
```

Cvičení 5.3: Vzdálenost mezi body

- ❖ Napište *funkci* v jazyce Python, která *určí vzdálenost mezi dvěma body* (x, y) *v rovině*
 - ❖ Bod reprezentujte jako tuple (n-tici)
- ❖ Vzdálenost dvou bodů v rovině je určena následovně:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- ❖ Pro výpočet odmocniny můžete použít funkci `math.sqrt()` z balíčku `math` nebo vhodně použít operátor `**`

Cvičení 5.4: Unikátní slova

- ❖ Napište *funkci* v jazyce Python, která přijme textový řetězec a *vrátí množinu unikátních slov* obsažených v tomto řetězci.
 - ❖ Předpokládejte, že slova jsou *oddělena mezerami*
 - ❖ Nezapomeňte na interpunkci.
- ❖ Otázky:
 - ❖ Co se stane, když do množiny opakovaně vkládáme stejné slovo?
 - ❖ Tj. které slovo v množině zůstane po pokusu o vložení duplikátu?

Cvičení 5.5: Frekvence slov

- ❖ Napište *funkci* v jazyce Python, která *spočítá frekvenci slov* v daném textu
 - ❖ Předpokládejte, že *slova* jsou *oddělena mezerami*
 - ❖ Nezapomeňte na interpunkci

Jak psát kód: Množiny pro testování členství

- ❖ Pro efektivní testování členství použijte množiny (Set) namísto seznamu (List)

```
1 valid_colours = { "red", "green", "blue" }  
2  
3 if user_colour in valid_colours:  
4     print("Valid colour")  
5 else:  
6     print("Invalid colour")
```

Jak psát kód: Kvantifikátory v podmínce

- ❖ Použijte `any()` jako existenční kvantifikátor a `all()` jako univerzální kvantifikátor

```
1 values = [True, False, True, True]
2
3 if any(values):
4     print("At least one value is True")
5
6 if all(values):
7     print("All values are True")
```

Reference

N-tice (angl. tuple)

- ❖ https://www.w3schools.com/python/python_tuples.asp
- ❖ https://www.w3schools.com/python/python_tuples_methods.asp

Množina (angl. set)

- ❖ https://www.w3schools.com/python/python_sets.asp
- ❖ https://www.w3schools.com/python/python_sets_methods.asp

Slovník (angl. dictionary)

- ❖ https://www.w3schools.com/python/python_dictionaries.asp
- ❖ https://www.w3schools.com/python/python_dictionaries_nested.asp
- ❖ https://www.w3schools.com/python/python_dictionaries_methods.asp