

Kolekce v pythonu

(Tuple, List, Set, Dictionary, String)

Tuple

Je zachovano poradi ve kterem jsou prvky kolekce vkladany?

Ano, ale prvky se dají přidat pouze jednou a to poprvé když je Tuple vytvořena.

```
In [ ]: example_tuple = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
print(example_tuple)
```

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

Jsou hodnoty uloženy v seřazeném pořadí od nejmenšího po největší?

Pouze, pokud jsou při vytváření Tuple seřazeny.

```
In [ ]: example_tuple = (5,1,2,3)
sorted_tuple = (1,2,3,5)

print(example_tuple)
print(sorted_tuple)
```

(5, 1, 2, 3)

(1, 2, 3, 5)

Používá se pro přístup k hodnotám index? (Index je systémem vygenerované číslo 0,1,2..)

Ano, index se používá pro přístup k hodnotám.

```
In [ ]: example_tuple = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
print(example_tuple[0])
```

1

Je možné uložit dva stejné prvky?

Ano, je možné uložit dva stejné prvky.

```
In [ ]: example_tuple = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
print(example_tuple)
```

(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)

Je možné vložit hodnotu None?

Ano, je možné vložit hodnotu None.

```
In [ ]: example_tuple = (None, None, None)
        print(example_tuple)
```

```
(None, None, None)
```

Je možné kolekce vnořovat, tedy vložit jako prvek kolekce další kolekci s několika prvky?

Ano, je možné kolekce vnořovat.

```
In [ ]: example_tuple = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10), (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
        print(example_tuple[0])
```

```
(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

Je možné za běhu programu změnit velikost kolekce?

Ne, velikost kolekce nelze za běhu programu změnit.

Jakým způsobem zjistím počet prvků v kolekci?

Pomocí funkce len().

```
In [ ]: example_tuple = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
        print(len(example_tuple))
```

```
10
```

Jakými všemi metodami je možné vložit nový prvek do kolekce a jaké mají tyto metody vstupy?

U kolekce Tuple není možné vložit nový prvek.

Jakými všemi metodami je možné odebrat prvek z kolekce a jaké mají tyto metody vstupy?

U kolekce Tuple není možné odebrat prvek.

Jakými všemi metodami je možné upravit hodnotu

existujícího prvek v kolekci a jaké mají tyto metody vstupy? U kolekce Tuple není možné upravit hodnotu existujícího prvku.

Jakými všemi metodami je možné ověřit jestli kolekce obsahuje nějaký prvek?

Pro určení konkrétního prvku v kolekci se používá operátor in. Pro učení zda kolekce obsahuje vůbec nějaký prvek se používá funkce len().

```
In [ ]: example_tuple = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
print(1 in example_tuple)
print(len(example_tuple))
```

True
10

Jakými všemi metodami je možné získat hodnotu v

pořadí třetího prvku v kolekci? Pomocí indexu.

```
In [ ]: example_tuple = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
print(example_tuple[2])
```

3

List

Je zachováno pořadí ve kterém jsou prvky kolekce vkladány?

Ano, je zachováno pořadí ve kterém jsou prvky kolekce vkládány.

```
In [ ]: example_list = [5, 1, 2, 3]
print(example_list)
```

[5, 1, 2, 3]

Jsou hodnoty uloženy v seřazeném pořadí od nejmenšího po největší?

Ne, hodnoty nejsou uloženy v seřazeném pořadí.

```
In [ ]: example_list = [5, 1, 2, 3]
print(example_list)
```

[5, 1, 2, 3]

Používá se pro přístup k hodnotám index? (Index je systémem vygenerované číslo 0,1,2..)

Ano, index se používá pro přístup k hodnotám.

```
In [ ]: example_list = [5, 1, 2, 3]
print(example_list[0])
```

5

Je možné uložit dva stejné prvky?

Ano, je možné uložit dva stejné prvky.

```
In [ ]: example_list = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
print(example_list)
```

```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

Je možné vložit hodnotu None?

Ano, je možné vložit hodnotu None.

```
In [ ]: example_list = [None, None, None]
print(example_list)
```

```
[None, None, None]
```

Je možné kolekce vnořovat, tedy vložit jako prvek kolekce další kolekci s několika prvky?

Ano, je možné kolekce vnořovat.

```
In [ ]: example_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(example_list[0])
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Je možné za běhu programu změnit velikost kolekce?

Ano, je možné za běhu programu změnit velikost kolekce.

```
In [ ]: example_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(len(example_list))
example_list.append(11)
print(len(example_list))
```

```
10
```

```
11
```

Jakým způsobem zjistím počet prvků v kolekci?

Pomocí funkce len().

```
In [ ]: example_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(len(example_list))
```

```
10
```

Jakými všemi metodami je možné vložit nový prvek do kolekce a jaké mají tyto metody vstupy?

Pomocí funkce append() a insert().

```
In [ ]: example_list = ["a", "b", "c", "d", "e"]
print(example_list)
example_list.append("f")
print(example_list)
```

```
example_list.insert(0, "z")
print(example_list)
```

```
['a', 'b', 'c', 'd', 'e']
['a', 'b', 'c', 'd', 'e', 'f']
['z', 'a', 'b', 'c', 'd', 'e', 'f']
```

Jakými všemi metodami je možné odebrat prvek z kolekce a jaké mají tyto metody vstupy?

Pomocí funkce pop() a remove().

```
In [ ]: example_list = ["a", "b", "c", "d", "e"]
print(example_list)
example_list.pop(0)
print(example_list)
example_list.remove("e")
print(example_list)
```

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

Jakými všemi metodami je možné upravit hodnotu existujícího prvku v kolekci a jaké mají tyto metody vstupy?

Pomocí indexu.

```
In [ ]: example_list = ["a", "b", "c", "d", "e"]
print(example_list)
example_list[0] = "z"
print(example_list)
```

```
['a', 'b', 'c', 'd', 'e']
['z', 'b', 'c', 'd', 'e']
```

Jakými všemi metodami je možné ověřit jestli kolekce obsahuje nějaký prvek?

Pro určení konkrétního prvku v kolekci se používá operátor in. Pro učení zda kolekce obsahuje vůbec nějaký prvek se používá funkce len().

```
In [ ]: example_list = ["a", "b", "c", "d", "e"]
print("a" in example_list)
print(len(example_list))
```

```
True
5
```

Jakými všemi metodami je možné získat hodnotu v pořadí třetího prvku v kolekci?

Pomocí indexu.

```
In [ ]: example_list = ["a", "b", "c", "d", "e"]
```

```
print(example_list[2])
```

c

Set

Je zachováno poradi ve kterem jsou prvky kolekce vkladany?

Ne, není zachováno pořadí ve kterém jsou prvky vkládány.

```
In [ ]: example_set = {5, 1, 2, 4 ,3}
print(example_set)
```

```
{1, 2, 3, 4, 5}
```

Jsou hodnoty ulozene v serazenem poradi od nejmensiho po největsi?

Ne, hodnoty nejsou uloženy v seřazeném pořadí.

```
In [ ]: example_set = {5, 1, 2, 4 ,3}
print(example_set)
```

```
{1, 2, 3, 4, 5}
```

Používá se pro přístup k hodnotám index? (Index je systémem vygenerované číslo 0,1,2..)

Ne, index se nepoužívá pro přístup k hodnotám.

Je mozne uložit dva stejné prvky?

Ne, není možné uložit dva stejné prvky.

```
In [ ]: example_set = {1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
print(example_set)
```

```
{1}
```

Je mozne vložit hodnotu None?

Ano, je možné vložit hodnotu None.

```
In [ ]: example_set = {None, 4, 5, 6, 7, 8, 9, 10}
print(example_set)
```

```
{4, 5, 6, 7, 8, 9, 10, None}
```

Je možné kolekce vnořovat, tedy vložit jako prvek kolekce další kolekci s několika prvky?

Ano, je možné kolekce vnořovat.

```
In [ ]: example_set = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
print(example_set)

({1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, {1, 2, 3, 4, 5, 6, 7, 8, 9, 10})
```

Je možné za běhu programu změnit velikost kolekce?

Ano, je možné za běhu programu změnit velikost kolekce.

```
In [ ]: example_set = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
print(len(example_set))
example_set.add(11)
print(len(example_set))
```

Jakým způsobem zjistím počet prvků v kolekci?

Pomocí funkce len().

```
In [ ]: example_set = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
print(len(example_set))
```

Jakými všemi metodami je možné vložit nový prvek do kolekce a jaké mají tyto metody vstupy?

Pomocí funkce add().

```
In [ ]: example_set = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
print(example_set)
example_set.add(11)
print(example_set)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}
```

Jakými všemi metodami je možné odebrat prvek z kolekce a jaké mají tyto metody vstupy?

Pomocí funkce remove().

```
In [ ]: example_set = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
print(example_set)
example_set.remove(10)
print(example_set)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

Jakými všemi metodami je možné upravit hodnotu existujícího prvku v kolekci a jaké mají tyto metody vstupy?

Upravit již přidáný prvek nelze. Jediné řešení je ho smazat a přidat znovu upravený.

Jakými všemi metodami je možné ověřit jestli kolekce obsahuje nějaký prvek?

Pro určení konkrétního prvku v kolekci se používá operátor `in`. Pro učení zda kolekce obsahuje vůbec nějaký prvek se používá funkce `len()`.

```
In [ ]: example_set = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
print(1 in example_set)
print(len(example_set))
```

```
True
10
```

Jakými všemi metodami je možné získat hodnotu v pořadí třetího prvku v kolekci?

Nelze

Dictionary

Je zachováno pořadí ve kterém jsou prvky kolekce vkladány?

Ne, není zachováno pořadí ve kterém jsou prvky vkládány.

Jsou hodnoty uloženy v seřazeném pořadí od nejmenšího po největší?

Ano, hodnoty jsou uloženy v seřazeném pořadí.

```
In [ ]: example_dict = {1: "a", 2: "b", 3: "c", 4: "d", 5: "e"}
print(example_dict)
```

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}
```

Používá se pro přístup k hodnotám index? (Index je systémem vygenerované číslo 0,1,2..)

Ne, index se nepoužívá pro přístup k hodnotám.

Je možné uložit dva stejné prvky?

Ne, není možné uložit dva stejné prvky.

Je možné vložit hodnotu None?

Ano, je možné vložit hodnotu None.

```
In [ ]: example_dict = {None: "a", 2: "b", 3: "c", 4: "d", 5: "e"}
print(example_dict)
```

```
{None: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}
```

Je možné kolekce vnořovat, tedy vložit jako prvek kolekce další kolekci s několika prvky?

Ano, je možné kolekce vnořovat.

```
In [ ]: example_dict = {"second_dict": {1: "a", 2: "b", 3: "c", 4: "d", 5: "e"}}
print(example_dict["second_dict"])
```

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}
```

Je možné za běhu programu změnit velikost kolekce?

Ano, je možné za běhu programu změnit velikost kolekce.

```
In [ ]: example_dict = {1: "a", 2: "b", 3: "c", 4: "d", 5: "e"}
print(len(example_dict))
example_dict[6] = "f"
print(len(example_dict))
```

```
5
```

```
6
```

Jakým způsobem zjistím počet prvků v kolekci?

Pomocí funkce len().

```
In [ ]: example_dict = {1: "a", 2: "b", 3: "c", 4: "d", 5: "e"}
print(len(example_dict))
```

```
5
```

Jakými všemi metodami je možné vložit nový prvek do kolekce a jaké mají tyto metody vstupy?

Skrze operátor [].

```
In [ ]: example_dict = {1: "a", 2: "b", 3: "c", 4: "d", 5: "e"}
print(example_dict)
example_dict[6] = "f"
print(example_dict)
```

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}
```

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e', 6: 'f'}
```

Jakými všemi metodami je možné odebrat prvek z kolekce a jaké mají tyto metody vstupy?

Pomocí funkce pop().

```
In [ ]: example_dict = {1: "a", 2: "b", 3: "c", 4: "d", 5: "e"}
print(example_dict)
example_dict.pop(5)
print(example_dict)
```

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}
```

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd'}
```

Jakými všemi metodami je možné upravit hodnotu existujícího prvek v kolekci a jaké mají tyto metody vstupy?

Pomocí operátoru [].

```
In [ ]: example_dict = {1: "a", 2: "b", 3: "c", 4: "d", 5: "e"}
print(example_dict)
example_dict[5] = "f"
print(example_dict)
```

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}
```

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'f'}
```

Jakými všemi metodami je možné ověřit jestli kolekce obsahuje nějaký prvek?

Pro určení konkrétního prvku v kolekci se používá operátor in. Pro učení zda kolekce obsahuje vůbec nějaký prvek se používá funkce len().

```
In [ ]: example_dict = {1: "a", 2: "b", 3: "c", 4: "d", 5: "e"}
print(1 in example_dict)
print(len(example_dict))
```

```
True
```

```
5
```

Jakými všemi metodami je možné získat hodnotu v pořadí třetího prvku v kolekci?

Nelze