

Python. Laboratorium 7. Tuple(Krotki)

Tuple (krotka) – jest to kolekcja(zbiór) danych, podobna do listy z tą różnicą że, **użytkownik lub program w trakcie pracy nie może jej edytować**. Zawartość krotki określamy **tylko podczas jej tworzenia**. Jeśli chcielibyśmy używać danych "zabezpieczonych przed zapisem" (np. po to, żeby program był bezpieczniejszy), wykorzystajmy do tego krotki. **Korzystając z krotek, zamiast z list, mamy pewność, że dane w nich zawarte nie zostaną nigdzie zmienione**. Krotki są rzadziej używane niż listy ale działają od nich szybciej.

Przykład:

```
krotka = („pierwszy”, 2, 3, „cztery”)
print(krotka)
```

Wypisanie elementów krotki po kolei:

for element in krotka:

```
print(element)
```

Podobnie jak w listach, elementy w krotce mają określony porządek. Są one indeksowane od 0, więc pierwszym elementem w niepustej krotce jest zawsze t[0].

Przykład wypisania konkretnego elementu krotki po numerze indeksu:

```
tup1 = ('Fizyka', 'Chemia', 1997, 2000);
tup2 = (1, 2, 3, 4, 5, 6, 7 );
print (" tup1[0]"); //program wypisze „Fizyka”
print ("tup[-1]”) //program wypisze 2000
print ("tup2[1:5]”) //program wypisze wartości [2, 3, 4, 5]
```

Aby sprawdzić długość krotki korzystamy z wyrażenia len(len):

```
tuple2 = (123, 'reserved', 'zara')
tuple3 = (456, 'abc')
print (len(tuple2))
print (len(tuple3))
```

Można dodać następny element do krotki. Ale tak na prawdę tworzona jest w tym momencie nowa krotka o takiej samej nazwie. Taka nowa krotka zajmując już inny adres w pamięci.

Krotki nie posiadają metod typu append, czy też extend. Nie można usuwać elementów z krotki. Nie posiadają one ani metody remove ani metody pop.

Aby dodać tylko jeden element należy za nim umieścić przecinek.

Przykład dodania elementu do krotki:

```
mojaKrotka = (1, 2, 3)
print(mojaKrotka)
print(id(mojaKrotka)) //wyświetlenie miejsca w pamięci, które zajmują krotka
mojaKrotka = mojaKrotka + („cztery”, ) //dodanie jednego elementu
print(mojaKrotka)
print(id(mojaKrotka)) // wyświetlenie miejsca w pamięci, które zajmują krotka
mojaKrotka = mojaKrotka + (5, „sześć”) //dodanie kilku elementów
print(mojaKrotka) print(id(mojaKrotka)) //wyświetlenie miejsca w pamięci, które zajmują krotka
```

Konkatenacja krotek to łączenie poszczególnych krotek ze sobą:

Przykład:

```
tuple1 = (1,2,3,4,5)
tuple2 = (6,7,8,9,10)
tup = tuple1 + tuple2 print(tup) // (1,2, 3, 4, 5, 6, 7, 8, 9, 10)
```

W celu powtórzenia krotki należy po prostu użyć operatora *. Jeśli chcemy trzykrotnie powtórzyć krotkę tuple1, używamy poniższego polecenia

```
tup=tuple1 * 3
print(tup) // (1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5)
```

Aby sprawdzić, czy dany element znajduje się w krotce, używamy polecenia in:

```
7 in tuple1 //wartością zwrótną tego polecenia będzie False, ponieważ 7 nie znajduje się w krotce tuple1.
```

Aby wskazać, gdzie w krotce znajduje się konkretny element, używamy metody index. Na przykład, jeśli chcesz określić położenie elementu 5 w krotce tuple1, użyj poniższego polecenia.

```
tuple1.index(5) // wartością zwrótną będzie liczba 4, która jest indeksem elementu 5.
```

Zliczanie, ile razy dany element występuje w krotce:

```
tuple3 = (65,67,5,67,34,76,67,231,98,67)
tuple3.count(67) //sprawdzamy liczbę wystąpień elementu 67 w krotce tuple3. Wynikiem tego polecenia powinna być liczba 4
```

Lista zadań:

1. Zadeklaruj krotkę składającą się z 6 elementów znakowych i liczbowych. Wyznacz jej długość. Sprawdź jej miejsce w pamięci. Dodaj dwa dowolne elementy. Sprawdź po raz drugi jej miejsce w pamięci. Jakie widzisz zmiany? Skonwertuj krotkę na listę. Jakie widzisz zmiany?
2. Zadeklaruj krotkę 100 elementową, wypełnioną losowymi liczbami z przedziału 0–10 wykorzystując funkcję random. Sprawdź ile razy wystąpiła liczba 0 oraz 1. Sprawdź ile razy w krotce znajduje się liczba 2. Wypisz elementy parzyste i nie parzyste za pomocą pętli for. Zastanów się jak możesz dopisać dodatkowe elementy do krotki.
3. Zadeklaruj krotkę A = 100 elementową, wypełnioną tylko liczbami parzystymi oraz krotkę B = 100 elementową wypełnioną tylko liczbami nieparzystymi. Wykonaj konkatencję. Sprawdź jej długość. Sprawdź czy krotka zawiera liczby 0 oraz 100. Sprawdź jak zmieniło się ich miejsce w pamięci.
4. Zadeklaruj krotkę składającą się z 6 elementów znakowych. Sprawdź za pomocą polecenia in czy w krotce znajduje się słowo, które zostało zadeklarowane. Wypisz jego indeks jeśli słowo zostało znalezione. Spróbuj do krotki wprowadzić dłuższy tekst i sprawdź czy słowo wpisane przez użytkownika znajduje się w tym tekście i ile razy w nim wystąpiło.
5. Utwórz dwie krotki w których zadeklarujesz dwa wyrazy. Sprawdź czy są one anagramami – czy dwa słowa składają się z tych samych liter. Zaproponuj przykładowe wyrazy, które będą prawidłowymi rozwiązaniami zadania.