

Krotki w Pythonie

M@я3k Pцđ€£kØ

Programowanie w Pythonie

Spis treści

- Krotka
- Przykład krotki
- Mutowalność krotki
- Krotka 0 lub 1-elementowa
- Pakowanie i rozpakowywanie krotki
- Właściwości krotek

Krotka

- Krotka (**tuple**) w Pythonie to uporządkowany, niezmienny (*niemutowalny*) zbiór elementów.
 - Tworzona za pomocą nawiasów okrągłych ()
 - Składa się z kilku wartości rozdzielonych przecinkami
 - Pozwala na przechowywanie elementów różnych typów.
- Po utworzeniu krotki nie można już zmieniać jej zawartości (dodawać, usuwać, modyfikować elementów).
 - Idealna do przechowywania danych, które mają pozostać stałe

Przykład krotki

```
krotka = 12345, 54321, 'dzień dobry!'
print (krotka)           #(12345, 54321, 'dzień dobry!')
print (krotka[1])        #54321
```

- krotki mogą być zagnieżdżane:

```
krotka1 = (123, 456)
krotka2 = ('abc', 'def')
krotka = krotka1, krotka2
print (krotka)           #((123, 456), ('abc', 'def'))
```

Mutowalność krotki

- krotki są niemutowalne (niemodyfikowalne)

```
krotka[1] = 8910
```

```
Traceback (most recent call last):  
  File "/home/main.py", line 9, in <module>  
    krotka[1] = 8910  
    ~~~~~^^^
```

```
TypeError: 'tuple' object does not support  
item assignment
```

- Krotki mogą zawierać mutowalne obiekty:

```
lista = ([7, 8, 9], [9, 8, 7])  
krotka1 = (123, 456)  
krotka = (krotka1, lista)  
print (krotka)
```

Krotka 0 lub 1-elementowa

- Specjalnym problemem jest konstrukcja krotek zawierających 0 lub 1 element: składnia przewiduje na to kilka sposobów.

- Krotka pusta (0 elementów)

- Puste krotki można konstruować pustą parą nawiasów

```
krotka_pusta = ()  
print (krotka_pusta)           # ()  
print (len(krotka_pusta))      # 0
```

- krotkę z jednym elementem można skonstruować umieszczając przecinek za wartością (nie wystarczy otoczyć pojedynczej wartości nawiasami). Przecinek jest konieczny, by odróżnić od wyrażenia w nawiasach.

```
krotka_pojedyncza = ('cześć',)  
print (krotka_pojedyncza)      # ('cześć',)  
print (len(krotka_pojedyncza)) # 1
```

Pakowanie i rozpakowywanie krotki

- **Pakowanie krotki**

- Polega na „spakowaniu” kilku zmiennych w krotkę.
- Pakowanie to po prostu utworzenie krotki z wartościami

```
krotka = 12345, 54321, 'hello!', True, 3.14
```

- **Rozpakowanie krotki**

- Polega na przypisaniu zawartości krotki do kilku zmiennych.
- Musi być ich po lewej stronie znaku równości dokładnie tyle, ile jest wartości w krotce.

```
x, y, tekst, Czy_jest, Pi = krotka
```

- Jest równoważne:

```
x = krotka[0]
```

```
y = krotka[1]
```

```
tekst = krotka[2]
```

```
Czy_jest = krotka[3]
```

```
Pi = krotka[4]
```

```
#x=12345,y=54321,tekst='hello!',Czy_jest=True,Pi=3.14 7
```

Właściwości krotek

- Na wyjściu krotki zawsze są otoczone nawiasami, tak aby zagnieżdżone krotki były poprawnie interpretowane.
 - Przy deklaracji można wpisać je można z lub bez otaczających nawiasów, ale często się je zostawia dla czytelności programu
- Nie da się przypisać wartości do pojedynczych elementów krotki, ale da się stworzyć krotki, które zawierają mutowalne obiekty, takie jak listy.
- Mają określony porządek i można się do nich odwoływać za pomocą indeksów (od 0).
- Może przechowywać różne typy danych (np. liczby, tekst, inne krotki).
- Często szybsza i bardziej efektywna pamięciowo niż listy, gdy dane nie muszą być zmieniane.
- Krotki są niemutowalne i zazwyczaj zawierają heterogeniczne sekwencje elementów, do których dostęp uzyskuje się przez rozpakowywanie lub indeksowanie (lub nawet przez atrybut w przypadku namedtuples).
- Listy są mutowalne i ich elementy są zazwyczaj homogeniczne i dostęp do nich uzyskuje się przez iterowanie po liście.
- Mimo że krotki mogą wydawać się podobne do list, często są używane w innych sytuacjach i do innych celów.
- Używa się ich:
 - Gdy trzeba zwrócić wiele wartości z funkcji (np. return x, y, z).
 - Do przechowywania stałych, niezmiennych zestawów danych (np. współrzędne, dane konfiguracyjne).
 - Jako klucze w słownikach (ponieważ są hashowalne, a listy nie).

Ćwiczenia 1

1. Napisz program, który:
 - a) Tworzy krotkę z 7 nazwami kolorów,
 - b) Pobiera od użytkownika nazwę koloru,
 - c) Sprawdza, czy podany kolor znajduje się w krotce
 - d) Wypisz każdy element krotki w osobnej linii używając pętli for.
2. Dana jest krotka: `dane = (4, 7, 1, 9, 2)`
 - a) Zamień krotkę na listę,
 - b) Posortuj listę rosnąco,
 - c) Zamień listę z powrotem na krotkę,
 - d) Wypisz wynik.
3. Dana jest krotka:
`uczniowie = ("Anna", 5), ("Jan", 4), ("Kasia", 6)`
 - a) Wypisz imiona uczniów, którzy mają ocenę większą lub równą 5.

Ćwiczenia 2

4. Dana jest krotka: `punkt = (3, 7)`

- a) Rozpakuj krotkę do zmiennych `x` i `y`,
- b) Oblicz odległość punktu od początku układu współrzędnych.

5. Napisz funkcję, która:

- a) przyjmuje dwie liczby,
- b) zwraca krotkę zawierającą sumę, różnicę i iloczyn tych liczb.

6. Dana jest krotka :

`liczby = (1, 2, 3, 2, 4, 2, 5)`.

- a) Policz, ile razy liczba 2 występuje w krotce.

Powtórzenie

1. Co to jest krotka?
2. Czym różni się krotka od listy?
3. Jak tworzymy krotkę?
4. Jakie problemy wiążą się z krotką 0 lub 1 elementową?
5. Na czym polega pakowanie i rozpakowywanie krotki?
6. Czy krotki można modyfikować?