

Struktura danych - tupla

Co to jest tupla?

W Pythonie tuple (pol. krotki) jest wbudowanym typem danych, który umożliwia tworzenie **niezmiennych sekwencji** wartości. Wartości lub elementy w krotce mogą być dowolnego typu. To sprawia, że krotki są całkiem przydatne w sytuacjach, gdy trzeba przechowywać heterogeniczne dane, na przykład w rekordzie bazy danych, do przedstawienia współrzędnych kartezjańskich(x,y), kolorów RGB(red, green, blue) i wielu innych sekwencji wartości.

Uwaga: w Pythonie krotki obsługują kilka operacji wspólnych dla innych typów sekwencji, takich jak listy, ciągi znaków i zakresy. Operacje te nazywane są operacjami na sekwencji wspólnej. To dowiesz się o kilku operacjach należących do tej kategorii.

Do najważniejszych cech obiektu tupla należą:

- **Uporządkowane:** zawierają elementy ułożone sekwencyjnie zgodnie z określoną kolejnością wstawiania;
- **Lekkie:** zużywają stosunkowo niewielką ilość pamięci w porównaniu do innych sekwencji, takich jak listy;
- **Indeksowalne:** indeks liczony od zera, umożliwiając dostęp do ich elementów za pomocą indeksów całkowitych rozpoczynających się od zera;
- **Niezmiennie:** nie obsługują mutacji lokalnych ani zmian w zawartych w nich elementach. Nie obsługują operacji zwiększania ani zmniejszania krotek;
- **Heterogeniczne:** mogą przechowywać obiekty o różnych typach danych, w tym obiekty;
- **Zagnieżdżane:** mogą zawierać inne krotki, więc możesz mieć krotki krotek;
- **Iterowalne:** obsługują iterację, więc możesz je przeglądać za pomocą pętli lub iteratora;
- **Sliceable:** obsługują operacje wycinania, co oznacza, że można wyodrębnić serię elementów z krotki i przekazać dalej;
- **Możliwość łączenia:** obsługują operacje łączenia, dzięki czemu można połączyć dwie lub więcej krotek za pomocą operatorów łączenia, co spowoduje utworzenie nowej krotki.
- **Hashowalny:** Mogą działać jako klucze w słownikach, gdy wszystkie elementy krotki są niezmiennie. Hashowalny obiekt musi posiadać metodę `__hash__()`, która zwraca liczbę całkowitą. Ta liczba jest używana jako identyfikator hasza.

Definiowanie tupli (krotki):

W Pythonie tuple tworzy się przez umieszczenie elementów w **nawiasach okrągłych ()** i oddzielenie ich przecinkami. Oto przykład tworzenia tupli:

```
my_tuple = (1, 2, 3, 'a', 'b', 'c')
```

Tupla z pojedynczym elementem musi zawierać na końcu przecinek.

```
my_single_tuple = (5, )
```

Możesz również tworzyć tuple bez użycia nawiasów, używając po prostu przecinków.

Na przykład:

```
tuple_bez_nawiasow = 1, 2, 3
print(tuple_bez_nawiasow)
```

Konstruktor klasy tuple() można używać do tworzenia obiektów krotek z obiektu iterowalnego, takiego jak lista, słownik lub ciąg znaków.

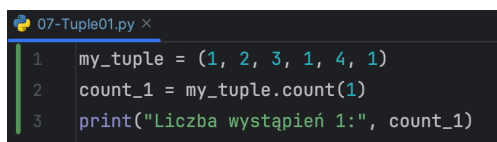
tuple([obiekt_iterowalny])

Przykład:

```
tuple(["Jane Doe", 25, 1.75, "Canada"])  
('Jane Doe', 25, 1.75, 'Canada')
```

```
tuple("Programista")  
('P', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 's', 't', 'a')
```

count() - Metoda ta zwraca liczbę wystąpień określonego elementu w tupli.



```
07-Tuple01.py x  
1 my_tuple = (1, 2, 3, 1, 4, 1)  
2 count_1 = my_tuple.count(1)  
3 print("Liczba wystąpień 1:", count_1)
```

index() - Metoda ta zwraca indeks pierwszego wystąpienia określonego elementu w tupli.



```
07-Tuple01.py •  
1 my_tuple = ('a', 'b', 'c', 'a', 'd')  
2 index_a = my_tuple.index('a')  
3 print("Indeks pierwszego wystąpienia 'a':", index_a)  
4
```