



Date: 07/05/2024

Tochilkina Ksenya Stanislavovna



Кортежи

Что такое кортеж?

Кортеж - НЕИЗМЕНЯЕМЫЕ, упорядоченные объекты. Кортежи похожи на списки, но имеют несколько ключевых отличий. Вот основные характеристики кортежей:

1. Упорядоченность: Элементы кортежа упорядочены и доступ к ним осуществляется по индексу, подобно спискам.
2. Неизменяемость: Кортежи являются неизменяемыми, что означает, что после создания кортежа нельзя изменить его элементы. Однако, если элементами кортежа являются изменяемые объекты (например, списки), то можно изменять состояние этих объектов.
3. Разнообразие типов данных: Кортежи могут содержать элементы разных типов данных, таких как числа, строки, списки и другие кортежи.

Кортежи в Python обычно создаются с использованием круглых скобок(или без них). Например:

```
my_tuple = (1, 2, 3)
```

Основная причина использования кортежа вместо списка - безопасность хранения данных и защита их от изменения. Например, нам нужно забрать из БД данные для формирования статистики: мы точно знаем, что менять их не собираемся. В таком случае, уместно завернуть данные в кортеж, для защиты от случайного изменения.

Вторая причина - меньший размер, по сравнению со списками. Если вы создаете блог или новостной портал, то скорее всего не почувствуете разницу, однако как только вы захотите влиться в мир BigData или DataScience, то увидите, насколько часто там используются кортежи.

Для примера, давайте создадим список и кортеж с одинаковыми элементами и сравним их размеры:

```
my_list = ["python", "is", "awesome"]
my_tuple = ("python", "is", "awesome")

print(f"Размер списка: {my_list.__sizeof__()}")
print(f"Размер кортежа: {my_tuple.__sizeof__()}")
```

```
D:\pythonProject\venv\Scripts\python.exe D:\pythonProject\main.py
Размер списка: 72
Размер кортежа: 48

Process finished with exit code 0
```

Основные методы для работы с кортежами

Основные методы для работы с кортежами в языке программирования Python включают:

- Метод `count()`:
Возвращает количество вхождений указанного элемента в кортеже.

```
my_tuple = (1, 2, 2, 3, 4, 2)
count = my_tuple.count(2)
print(count) # 3
```

- Метод `index()`:

Возвращает индекс первого вхождения указанного элемента в кортеже.

```
my_tuple = (1, 2, 2, 3, 4, 2)
index = my_tuple.index(3)
print(index) # 3
```

- Оператор `in`:

Проверяет, содержит ли кортеж указанный элемент, и возвращает логическое значение.

```
my_tuple = (1, 2, 3, 4, 5)
contains = 3 in my_tuple
print(contains) # True
```

- Оператор `+`:

Создает новый кортеж, объединяя два или более кортежа.

```
tuple1 = (1, 2, 3)
tuple2 = (4, 5, 6)
new_tuple = tuple1 + tuple2
print(new_tuple) # (1, 2, 3, 4, 5, 6)
```

- Оператор `*`: Создает новый кортеж, повторяя элементы указанное количество раз.

```
my_tuple = (1, 2)
new_tuple = my_tuple * 3
print(new_tuple) # (1, 2, 1, 2, 1, 2)
```

- Метод `len()`: Возвращает количество элементов в кортеже.

```
my_tuple = (1, 2, 3, 4, 5)
length = len(my_tuple)
print(length) # 5
```

-
- Метод `sorted()`: Возвращает новый отсортированный список из элементов кортежа.

```
my_tuple = (3, 1, 4, 2, 5)
sorted_tuple = sorted(my_tuple)
print(sorted_tuple) # [1, 2, 3, 4, 5]
```

Это лишь некоторые примеры методов для работы с кортежами в Python. Кортежи являются неизменяемыми, поэтому большинство методов, связанных с изменением или модификацией, не применимы к кортежам.

