
Кортежи (Tuple)

Что такое кортежи?

Кортежи в Python представляют собой структуру данных, похожую на списки, но с одним основным отличием: они неизменяемы. Кортежи используются для хранения коллекций элементов, которые не могут быть изменены после создания.

Их основные свойства:

Неизменяемость: Основное свойство кортежей состоит в том, что они не могут быть изменены после создания. Это означает, что после того, как кортеж создан, вы не можете изменить его содержимое, добавить новые элементы или удалить существующие.

Упорядоченность: Элементы в кортеже хранятся в определенном порядке, который определяется при создании кортежа. Порядок элементов сохраняется при доступе к ним.

Зачем используются кортежи?

Кортежи предоставляют стабильный и неизменяемый набор данных, который может использоваться в качестве ключей в словарях или как элементы в множествах.

Они также могут использоваться для представления структур данных, которые не должны изменяться или модифицироваться в процессе выполнения программы.

Важно: Поскольку кортежи неизменяемы, они более эффективны с точки зрения памяти и времени выполнения по сравнению со списками, особенно при работе с большими объемами данных. Однако, если требуется изменять элементы или дополнять структуру данных в процессе работы программы, лучше использовать списки.

Создание кортежей:

1. Создание пустого кортежа: Пустой кортеж может быть создан с использованием пустых круглых скобок (). Это наименее распространенный способ создания кортежа.
2. Создание кортежа с одним элементом: Если вы хотите создать кортеж с одним элементом, необходимо включить запятую после этого элемента, даже если он один.
3. Создание кортежа с несколькими элементами: Кортеж с несколькими элементами создается с помощью запятых для разделения элементов внутри круглых скобок.
4. Использование функции `tuple()`: Вы также можете создать кортеж, используя встроенную функцию `tuple()`, передавая ей другой итерируемый объект, такой как список или строка.

```
empty_tuple = ()           # Создание пустого кортежа
single_tuple = (1,)        # Создание кортежа с одним элементом
my_tuple = (1, 2, 'a', 'b', 3.14) # Создание кортежа с несколькими элементами
tuple_from_list = tuple([1, 2, 3]) # Создание кортежа из списка
```

Операции с кортежами:

Длина кортежа: Длина кортежа определяется с помощью функции `len()`. Функция `len()` возвращает количество элементов в кортеже.

Проверка наличия элемента в кортеже: Для проверки наличия определенного элемента в кортеже можно использовать оператор `in`. Оператор `in` возвращает `True`, если элемент присутствует в кортеже, и `False` в противном случае.

```
my_tuple = (1, 2, 3, 'a', 'b', 'c')
is_present = 'a' in my_tuple # is_present = True
```

Методы кортежей:

Кортежи являются неизменяемыми, поэтому у них нет методов, которые изменяют их содержимое, таких как `append()`, `insert()`, `remove()` и `pop()`.

Однако у кортежей есть некоторые встроенные методы для работы с ними.

`count()`: Возвращает количество вхождений заданного элемента в кортеже.

`index()`: Возвращает индекс первого вхождения заданного элемента в кортеже.

```
my_tuple = (1, 2, 3, 'a', 'b', 'c', 3, 3)
count_a = my_tuple.count('a') # count_a = 1
index_b = my_tuple.index('b') # index_b = 4
```

Практические задания

1. Напишите программу, которая преобразует список в кортеж и кортеж обратно в список.
2. Напишите программу, которая находит минимальный и максимальный элементы в кортеже.
3. Напишите программу, которая подсчитывает количество определенного элемента в кортеже.
4. Напишите программу, которая меняет порядок элементов в кортеже на обратный.
5. Напишите программу, которая сортирует кортеж в порядке возрастания.
6. Напишите программу, которая извлекает подкортеж из исходного кортежа.
7. Напишите программу, которая объединяет несколько кортежей в один.
8. Напишите программу, которая проверяет, присутствует ли определенный элемент в кортеже.
9. Напишите программу, которая преобразует кортеж в строку и наоборот.
10. Напишите программу, которая разделяет кортеж на два кортежа в заданной позиции.