**Создание кортежа**

**Создать кортеж можно несколькими способами:**

**Используя круглые скобки:**

my\_tuple = (1, 2, 3, 4)

**Без скобок, просто перечислив элементы через запятую:**

**my\_tuple = 1, 2, 3, 4**

**С помощью функции tuple():**

my\_tuple = tuple([1, 2, 3, 4])

**Методы кортежа**

**Кортежи имеют гораздо меньше методов по сравнению со списками из-за их неизменяемости. Вот они:**

**tuple.count(x) - возвращает количество вхождений элемента x в кортеж:**

my\_tuple = (1, 2, 3, 1, 2, 1)

count = my\_tuple.count(1)

# count теперь: 3

**tuple.index(x) - возвращает индекс первого вхождения элемента x в кортеж:**

my\_tuple = (1, 2, 3, 1, 2, 1)

index = my\_tuple.index(2)

# index теперь: 1

**Функции, которые можно использовать с кортежами**

**Кортежи также поддерживают ряд встроенных функций Python:**

**len() - возвращает количество элементов в кортеже:**

my\_tuple = (1, 2, 3, 4)

length = len(my\_tuple)

# length теперь: 4

**max() - возвращает максимальный элемент в кортеже:**

my\_tuple = (1, 2, 3, 4)

max\_value = max(my\_tuple)

# max\_value теперь: 4

**min() - возвращает минимальный элемент в кортеже:**

my\_tuple = (1, 2, 3, 4)

min\_value = min(my\_tuple)

# min\_value теперь: 1

**sum() - возвращает сумму элементов кортежа:**

my\_tuple = (1, 2, 3, 4)

total = sum(my\_tuple)

# total теперь: 10

**sorted() - возвращает отсортированный список элементов кортежа (не изменяя сам кортеж):**

my\_tuple = (4, 3, 1, 2)

sorted\_list = sorted(my\_tuple)

# sorted\_list теперь: [1, 2, 3, 4]

**Индексация и срезы**

**Как и списки, кортежи поддерживают индексацию и срезы:**

my\_tuple = (1, 2, 3, 4)

# Индексация

print(my\_tuple[1]) # Выведет: 2

# Срезы

print(my\_tuple[1:3]) # Выведет: (2, 3)

**Конкатенация и умножение**

**Кортежи можно конкатенировать и умножать, создавая новые кортежи:**

tuple1 = (1, 2, 3)

tuple2 = (4, 5, 6)

# Конкатенация

new\_tuple = tuple1 + tuple2

# new\_tuple теперь: (1, 2, 3, 4, 5, 6)

# Умножение

repeated\_tuple = tuple1 \* 2

# repeated\_tuple теперь: (1, 2, 3, 1, 2, 3)

**Распаковка кортежей**

**Распаковка кортежей позволяет присваивать значения из кортежа переменным напрямую:**

my\_tuple = (1, 2, 3)

x, y, z = my\_tuple

# Теперь x = 1, y = 2, z = 3

**Вложенные кортежи**

**Кортежи могут содержать другие кортежи как элементы:**

nested\_tuple = ((1, 2), (3, 4), (5, 6))

# Доступ к элементам вложенного кортежа

print(nested\_tuple[1][0]) # Выведет: 3

**Преобразование списков и строк в кортежи**

**Вы можете преобразовать список или строку в кортеж с помощью функции tuple():**

# Преобразование списка в кортеж

my\_list = [1, 2, 3]

list\_to\_tuple = tuple(my\_list)

# list\_to\_tuple теперь: (1, 2, 3)

# Преобразование строки в кортеж

my\_string = "hello"

string\_to\_tuple = tuple(my\_string)

# string\_to\_tuple теперь: ('h', 'e', 'l', 'l', 'o')