Software Engineering from Innopolis

Vladimir Maximov

26 июня 2023 г.

1 Итеритуемые объекты в Python

1.1 Цель

Изучить коллекции в Python и работу с ними.

1.2 Задачи

- Ознакомиться со списками в Python.
- Рассмотреть популярные операции, используемые над списками slice, filter, map, reduce.
- Ознакомиться с кортежем в Python.
- Разобрать различные подходы создания коллекций.
- Ознакомиться с множествами в Python.
- Понять, чем руководствоваться при выборе коллекции для хранения объектов.

1.3 Список

Список - упорядоченная и изменяемая коллекция объектов:

```
1   my_list = [1, 2, 3, None, [0], "word"]
```

Слайс (срез) списка - создание нового списка, используя элементы уже существующего:

```
1   my_list[2:4:1]
```

Берем со 2 элемента включительно по 4 элемент невключительно с шагом 1.

Краткая форма создания списка называется List comprehention (генератор списка):

```
1   [x for x in range(10)]
```

1.4 Кортеж

Кортеж - упорядоченная и неизменяемая коллекция:

```
1   (1, 2, 3, 4, 5)
```

1.5 Множество

Множество - неупоярдоченная коллекция значений, в которой не допускаются повторения и не может содержать не хешируемых объектов:

```
1   {1, 2, 3, "some string"}
```

1.6 Словари

Словарь - изменяемая коллекция объектов, которые обладают ключевыми словами. В Python словарь можно описать указанием ключей и значений элементов или генерирующим выражением:

```
1   {"key1" : "value1",
2   "key2" : "value2",
3   "key3" : "value3"}
```

1.7 Использование памяти

Для хранения коллекций выделяется чуть больше памяти, чем фактически необходимо. Делается это для того, чтобы интерпретатор Python при добавлении элементов выделял память реже.

В Python в коллекции можно хранить разные объекты, каждый из которых может быть непредвиданного размера. Поэтому в Python в списке хранятся указатели на нужные объекты.

1.8 Itertools

itertools - модуль, который предоставляет различные функции для работы с итерируемыми объектами. Его можно использовать для упрощения записи операций над итерируемыми объектами. Примеры методов:

itertools.combinations - метод для поиска подмножеств итерируемого объекта, возвращает генератор:

```
1   import itertools
2   itertools.combinations("ABCD", 2)
3   print(list(itertools.combinations("ABCD", 2)))
4   # Вывод: [(A, B), (A, C), (A, D), (B, C), (B, D), (C, D)]
```

itertools.compress - метод, который выбирает из исходного итерируемого объекта элементы, согласно селектору (маске):

```
1   import itertools
2   itertools.compress([1,2,3,4], [1,0,1,0])
3   print(list(itertools.compress([1,2,3,4], [1,0,1,0])))
4   # Вывод: [1,3]
```