Фоксфорд

Кружки

Кружок по программированию на Python

Занятие №11

Фоксфорд
Кружки

Кружок по программированию на Python

Кортежи, множества и словари

- Кортежи в Python
- Множества в Python
- Словари (ассоциативные массивы) в Python
- Хеширование и хеш-функция

Фоксфорд
Кружки

Введение

Что такое массив?

- Список -- изменяемая последовательность элементов произвольных типов.

- Однако:
- В языке Python есть и другие типы массивов, схожие по структуре, но принципиально отличные от списков.

Фоксфорд
Кружки

Введение

Кортеж (tuple)

- Кортеж (tuple) – это неизменяемая структура данных, которая по своей структуре очень похожа на список.

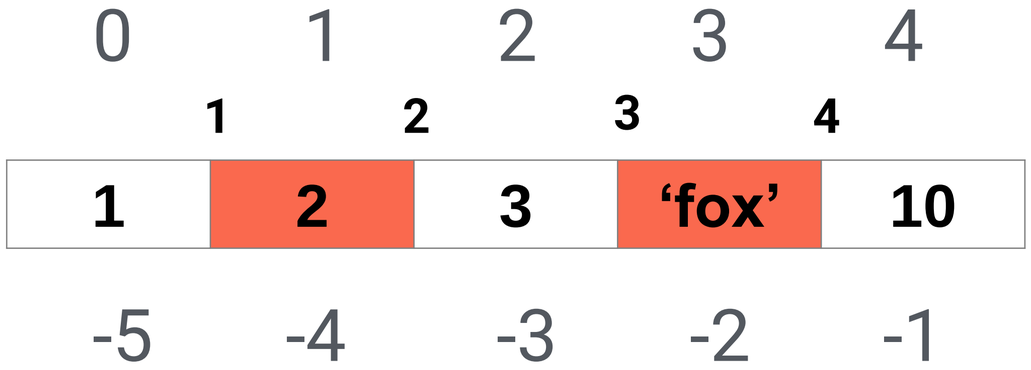
- A = (1, 2, 3)
- A = ()
- A = tuple()

Но:
A = (4,) в случае одного элемента!

Индексация

Срезы

- a = (1, 2, 3, ‘fox’, 10)
- a [1::2] - ?



a[::3] -> (1, ‘fox’)

a[1:3:2] -> (2)

a[1:3:-2] -> ()

- a[1::2] -> (2, ‘fox’)
- Однако недопустима операция присваивания в отношении элементов кортежа: ~~a[2] = 5~~

Кортежи в Python

Функции

По прежнему работают те функции и методы, характерные для списков, которые не изменяют кортеж:

- len(list): возвращает длину кортежа
- min(list): возвращает наименьший элемент кортежа
- max(list): возвращает наибольший элемент кортежа

Кортежи в Python

Методы

- index(x): возвращает индекс элемента x. Если элемент не найден, генерирует исключение ValueError;
- count(x): возвращает количество вхождений элемента x в список;

Используемые методы не должны менять сам список, иначе генерируется исключение AttributeError

Хеширование

- Хеширование (англ. hashing) —метод поиска, идея которого состоит в вычислении хеш-кода, однозначно определяемого каждым элементом с помощью хеш-функции, и использовании его как индекса (индексирование в памяти по хеш-коду выполняется за O(1))
- H(x) = x%8

0	1	2	3	4	5	6	7
16	1	26			45		63
	9						

Фоксфорд
Кружки

Введение

Множество (set)

- Множество - структура, содержащая неповторяющиеся элементы в случайном порядке.

- A=set()
- A=set(‘foxford’)

>> {'d', 'f', 'o', 'r', 'x’}

- A = {4, 5, 6}
- ~~A = {}~~

Множества в Python

Функции

По прежнему работают :

- len(set): возвращает количество элементов множества
- min(set): возвращает наименьший элемент множества
- max(set): возвращает наибольший элемент множества
- x in set: проверяет принадлежность множеству
- set.clear(): очищает множество
- set.pop(): удаляет первый элемент из множества. Так как множества не упорядочены, нельзя точно сказать, какой элемент будет первым.

Фоксфорд
Кружки

Задача №1

Дано два списка.
Узнать:

1. количество уникальных элементов списка;
2. максимальный из них;
3. минимальный из них ;
4. вывести общие элементы для двух списков

Задача №1

```
A = set(input().split())
B = set(input().split())
print(len(A), len(B))
print(max(max(A),max(B)))
print(min(min(A),min(B)))
print(A.intersection(B))
```

Множества в Python

A | B	A.union(B)	Возвращает множество, являющееся объединением множеств A и B.
A & B	A.intersection(B)	Возвращает множество, являющееся пересечением множеств A и B.
A - B	A.difference(B)	Возвращает разность множеств A и B.
A ^ B	A.symmetric_difference(B)	Возвращает симметрическую разность множеств A и B.
A < B		Возвращает true, если A является подмножеством B и не равно B.
A > B		Возвращает true, если A является подмножеством B и не равно B.

Множества в Python

A |= B	A.update(B)	Добавляет в множество A все элементы множества B.
A &= B	A.intersection_update(B)	Оставляет в множестве A только элементы множества B.
A -= B	A.difference_update(B)	Удаляет из множества A все элементы, входящие в B.
A ^= B	A.symmetric_difference_update(B)	Записывает в A симметрическую разность множеств A и B.
A <= B	A.issubset(B)	Возвращает true, если A является подмножеством B.
A>=B	A.issuperset(B)	Возвращает true, если A является подмножеством B.

Задача №2

- Во входном файле записан текст. Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки.

- Определите, сколько различных слов содержится в тексте.

Задача №2

```
a=set()
f = open('test.txt')
for w in f.read().split():
    a.add(w)
print(len(a))
```

Или:

```
print(len(set(w for w in open('test.txt').read().split())))
```

Введение

Словари (dict)

- Словари в Python - неупорядоченные коллекции произвольных объектов с доступом по ключу. Иногда их называют ассоциативными массивами.

- A=dict()
- A = {}
- A = {‘Ivan’: 12, ‘Pavel’: 19}

Словари в Python

- dict.clear() - очищает словарь.
- dict.copy() - возвращает копию.
- dict.get(key[, default]) - возвращает значение ключа, но если его нет, возвращает None.
- dict.items() - возвращает пары (ключ, значение).
- dict.keys() - возвращает ключи в словаре.
- dict.values() - возвращает значения словаря
- dict.pop(key[, default]) - удаляет ключ и возвращает значение. Если ключа нет, возвращает исключение
- dict.update([x]) - обновляет словарь, добавляя пары (ключ, значение) из x.

Задача №3

Дан англо-русский словарь. Сделать из него русско-английский, вывести в файл.

Задача №3

```
for line in f:
    words  = line.strip().split(' - ')
    key = words[0]
    val = words [1].split(',')
    for k in val:
        if k in d:
            d[k].append(key)
        else:
            d[k] = [key]
```

Фоксфорд
Кружки

Задача №4

Посчитать частоту появления слов в текстовом файле.

Задача №4

Посчитать частоту появления слов в текстовом файле.

```
words = {}
strip = string.whitespace + string.punctuation + string.digits + "\""
for line in open('shakespeare.txt'):
    for word in line.lower().split():
        word = word.strip(strip)
        if len(word) > 2:
            words[word] = words.get(word, 0) + 1
```

Спасибо за внимание!