

Практическая работа №3_5

Основы Python, часть 2

Цель работы

Продолжение изучения базового синтаксиса Python.

Задачи работы

1. Изучить итеративные типы данных Python: списки, кортежи, словари.
2. Изучать основные операции с итеративными типами данных.

Перечень обеспечивающих средств

1. ПК.
2. Учебно-методическая литература.
3. Задания для самостоятельного выполнения.

Общие теоретические сведения

Итеративные типы данных Python

В дополнение к спискам, рассмотренным в предыдущей работе, имеет смысл рассмотреть ещё два важных итеративных типа данных: кортежи и словари.

Кортеж (tuple) — заключенная в круглые скобки последовательность значений любых типов, разделенных запятыми.

Примеры кортежей:

```
(1, 'текст', 2, -0.19)
((0, 1), 2, 'три')
```

Основное отличие кортежей и списков: кортежи нельзя изменять и дополнять, списки — можно.

Список можно преобразовать в кортеж с помощью метода tuple() и, наоборот, кортеж в список с помощью метода list().

Словарь (dict) — заключенная в фигурные скобки последовательность пар ключ-значение. Ключ и значение разделяются двоеточием, пары между собой — запятыми.

Примеры словарей:

```
{1 : 1, 2 : 'два', 'три' : 3, 'четыре' : 'четыре'}
{'Арсенал' : 'Тула', 'Зенит' : 'Санкт-Петербург', 'Спартак' : 'Москва'}
```

Ключи в словаре не могут повторяться, значения — могут.

Ключами могут выступать только неизменяемые значения, например, числа, строки, кортежи. Значения могут быть любыми, в том числе другими словарями.

Основные операции со списками, кортежами и словарями

Рассмотрим основные операции на следующих примерах:

```
my_list = [1, 2, 4, 8, 16]
my_tuple = (10, 'десять', 'X')
my_dict = {3 : 'три', 2 : 'два', 1 : 'один'}
```

Индексация:

Список	Кортеж	Словарь
Можно извлечь элемент по его номеру, начиная с нуля: my_list[1] равно 2. Либо по номеру с конца: my_list[-1] равно 16.	Можно извлечь элемент по его номеру, начиная с нуля: my_tuple[1] равно 'десять'. Либо по номеру с конца: my_tuple[-1] равно 'X'.	Можно извлечь значение по ключу: my_dict[1] равно 'один'.

Добавление элемента:

Список	Кортеж	Словарь
Можно добавить новый элемент в конец списка: my_list.append(32) Теперь my_list равен [1, 2, 4, 8, 16, 32] Можно добавить новый элемент в любую позицию списка: my_list.insert(1, 32) После этого my_list будет равен [1, 32, 2, 4, 8, 16]	Нельзя изменить.	Можно добавить новый элемент, указав пару ключ-значение: my_dict[0] = 'ноль' После этого my_dict будет равен {3 : 'три', 2 : 'два', 1 : 'один', 0 : 'ноль'}

Удаление элемента:

Список	Кортеж	Словарь
Метод pop() возвращает элемент с указанными индексом, удаляя его из списка. Если индекс не указан, то возвращается и удаляется последний элемент. Например, my_list.pop(2) вернет 4, а my_list после этого будет равен [1, 2, 8, 16].	Нельзя изменить.	Метод pop() возвращает значение с указанными ключом, удаляя пару ключ-значение из словаря. Например, my_dict.pop(3) вернет 'три', а my_dict после этого будет равен {2 : 'два', 1 : 'один'}.

Метод <code>remove()</code> удаляет из списка указанный элемент. Например, после выполнения <code>my_list.remove(8)</code> <code>my_list</code> будет равен <code>[1, 2, 4, 16]</code> .		
---	--	--

Поиск элемента:

Список	Кортеж	Словарь
Оператор <code>in</code> позволяет проверить, содержится ли элемент в списке. Например, <code>2 in my_list</code> вернёт <code>True</code> (истина), <code>5 in my_list</code> вернёт <code>False</code> (ложь).	Оператор <code>in</code> позволяет проверить, содержится ли элемент в кортеже. Например, <code>10 in my_tuple</code> вернёт <code>True</code> (истина), <code>'XI' in my_tuple</code> вернёт <code>False</code> (ложь).	Оператор <code>in</code> позволяет проверить, содержится ли ключ в словаре. Например, <code>3 in my_dict</code> вернёт <code>True</code> (истина), <code>'три' in my_dict</code> вернёт <code>False</code> (ложь).

Сортировка:

Список	Кортеж	Словарь
Метод <code>sort()</code> сортирует содержимое списка. Пример вызова: <code>my_list.sort()</code> . Метод <code>sorted()</code> возвращает новый отсортированный список. Пример вызова: <code>sorted(my_list)</code> .	Метод <code>sorted()</code> возвращает новый отсортированный кортеж. Пример вызова: <code>sorted(my_tuple)</code> .	Метод <code>sorted()</code> возвращает новый отсортированный список ключей словаря. Пример вызова: <code>sorted(my_dict)</code> .

Слияние:

Список	Кортеж	Словарь
Метод <code>extend()</code> добавляет к первому списку элементы второго списка. Например, для <code>l1 = [0,1]</code> и <code>l2 = [2, 3]</code> после выполнения <code>l1.extend(l2)</code> <code>l1</code> равен <code>[0,1,2,3]</code> . Оператор <code>+</code> создает новый список, состоящий из элементов первого и второго списков. Например, для <code>l1 = [0,1]</code> и <code>l2 = [2, 3]</code> <code>l1+l2</code> равно <code>[0,1,2,3]</code> . Сами <code>l1</code> и <code>l2</code> при этом не изменяются.	Оператор <code>+</code> создает новый кортеж, состоящий из элементов первого и второго кортежей. Например, для <code>t1 = (0, 1)</code> и <code>t2 = (2, 3)</code> <code>t1+t2</code> равно <code>(0,1,2,3)</code> . Сами <code>t1</code> и <code>t2</code> при этом не изменяются.	Метод <code>update()</code> добавляет переданные пары ключ-значение к словарю. Например, для <code>d1 = {0:0, 1:1}</code> и <code>d2 = {2:2, 3:3}</code> после выполнения <code>d1.update(d2)</code> <code>d1</code> будет равен <code>{0:0, 1:1, 2:2, 3:3}</code> .

Получение длины:

Список	Кортеж	Словарь
Метод <code>len</code> возвращает количество элементов в списке. Например, <code>len(my_list)</code> вернет 5.	Метод <code>len</code> возвращает количество элементов в кортеже. Например, <code>len(my_tuple)</code> вернет 3.	Метод <code>len</code> возвращает количество элементов в словаре. Например, <code>len(my_dict)</code> вернет 3.

Копирование:

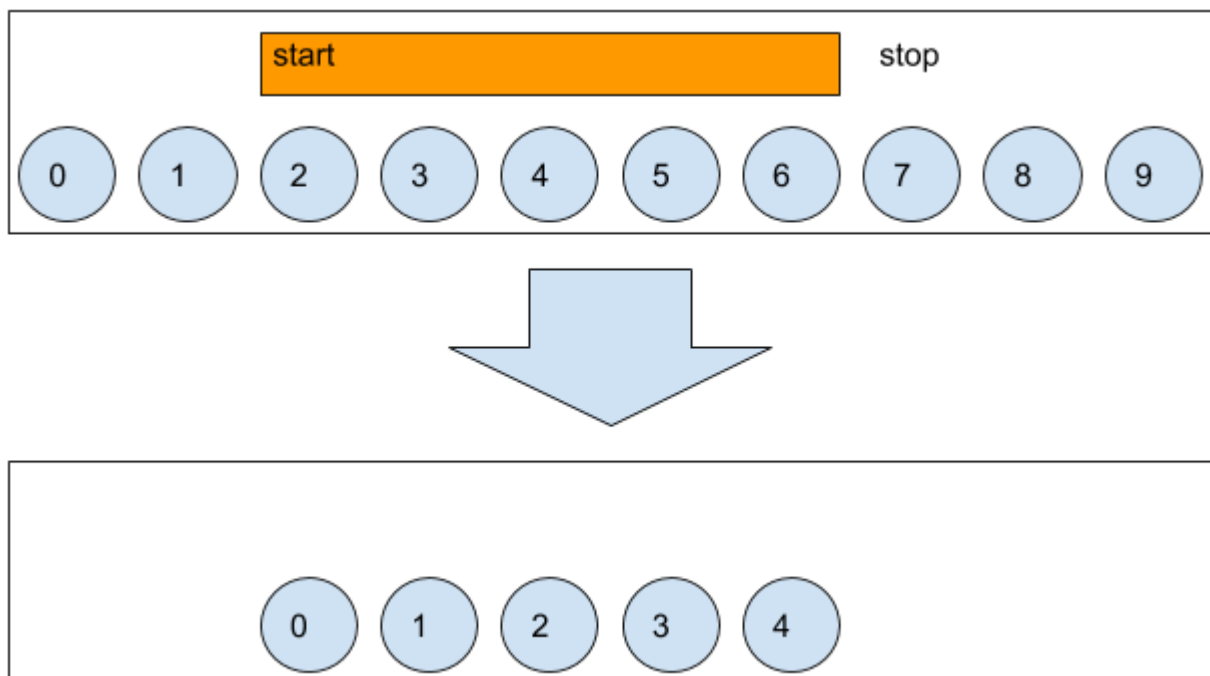
Список	Кортеж	Словарь
Метод <code>copy()</code> создает копию списка. Пример вызова: <code>my_list.copy()</code> .	Не имеет особого смысла, т. к. кортеж неизменяем. Два кортежа с одними и теми же значениями — по сути один и тот же кортеж.	Метод <code>copy()</code> создает копию словаря. Пример вызова: <code>my_dict.copy()</code> .

Слайсинг:

Слайсинг — это более мощный аналог индексирования. Слайсинг применим к спискам, кортежам и строкам.

Нотация слайсинга аналогична нотации функции `range()`:

`start : stop : step` — начало : остановка : шаг.



Start (начало):

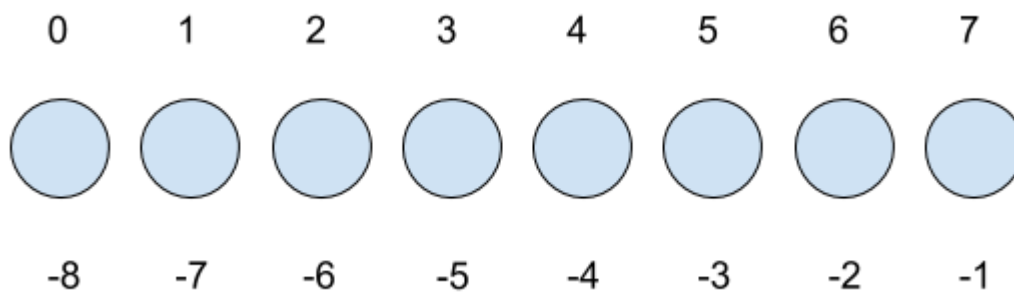
Если не указать, то считается равным 0.

Если указать отрицательное число, то элемент берется с конца списка или кортежа.

Stop (остановка):

Если не указать, то считается равным длине списка или кортежа.

Если указать отрицательное число, то элемент берется с конца списка или кортежа.



Step (шаг):

Число, на которое увеличивается индекс при проходе по списку или кортежу.

Если не указать, то считается равным 1.

Если указать отрицательное число, то отсчёт идет от конца к началу.

Слайсы можно использовать в большинстве методов или операторов, требующих указания индекса.

Задание

Часть 1

- Сделайте форк репозитория <https://github.com/mosalov/EmptyJupyterNotebookForBinder>
- Откройте сайт Binder: <https://mybinder.org/>.
- В поле «GitHub repository name or URL» укажите ссылку на свой репозиторий. Нажмите кнопку «launch», дождитесь открытия репозитория.
- Откройте (кликните) файл «empty_notebook.ipynb».
- Напишите код, который создаст и выведет список квадратов натуральных чисел от 1 до 1000.
- Сохраните файл Jupyter notebook с названием «*Фамилия_Задание 3_5_1.ipynb*» и загрузите его в созданный репозиторий.

Часть 2

- Вернитесь к файлу «empty_notebook.ipynb», открытому в Binder.
- Создайте кортеж из натуральных чисел от 1 до 20.
- С помощью слайсинга создайте кортеж, содержащий числа с 5 по 15 включительно.
- С помощью слайсинга создайте кортеж, содержащий последние 7 элементов первоначального кортежа.
- С помощью слайсинга создайте кортеж (7, 9, 11, 13).
- Сохраните файл Jupyter notebook с названием «*Фамилия_Задание 3_5_2.ipynb*» и загрузите его в созданный репозиторий.

Часть 3

- Вернитесь к файлу «empty_notebook.ipynb», открытому в Binder.
- Создайте список, состоящий из букв вашего имени (с помощью метода `list('строка')` или любым другим удобным способом).
- Создайте копию списка (второй список).
- Отсортируйте второй список.
- Создайте третий список, который состоит из элементов первого и второго списков.
- Из третьего списка удалите первые 3 элемента.
- Выведите получившиеся первый, второй и третий списки.
- Сохраните файл Jupyter notebook с названием «*Фамилия_Задание 3_5_3.ipynb*» и загрузите его в созданный репозиторий.

Часть 4

- Вернитесь к файлу «empty_notebook.ipynb», открытому в Binder.
- Создайте пустой словарь.

- Добавьте в словарь элементы «1 : 10», «'abc' : 'def'», «0 : [0, 1, 2]», «(0, 1) : 'a'».
- Прибавьте 5 к значению 10.
- Добавьте к строке 'def' ее копию (должно получиться 'defdef').
- В список [0, 1, 2] добавьте элемент 3, а затем удалите элемент 0.
- Удалите элемент со значением 'a'.
- Выведите получившийся словарь.
- Сохраните файл Jupyter notebook с названием «Фамилия_Задание 3_5_4.ipynb» и загрузите его в созданный репозиторий.

Контрольные вопросы

1. Какой тип данных: список, кортеж или словарь — вы считаете правильным использовать для указанных ниже данных и почему:
 - а) список студентов в группе,
 - б) список оценок студента,
 - в) данные студента: ФИО, адрес, дата рождения.
2. Напишите функцию, которая принимает на вход список натуральных чисел и возвращает словарь, в котором ключами являются элементы входного списка, а значениями — кортежи простых чисел, меньших или равных ключу.

Требования к отчету

Все файлы загрузите в свой репозиторий, созданный в практическом задании №1, по пути: «Notebook_For_AI_Main/2021 Осенний семестр/Практическое задание 3_5/» и сделайте пул-реквест.

Литература

1. <https://pythonworld.ru/samouchitel-python>
2. <https://www.coursera.org/learn/diving-in-python>
3. <https://stepik.org/course/67/promo>
4. <http://pythontutor.ru/>