

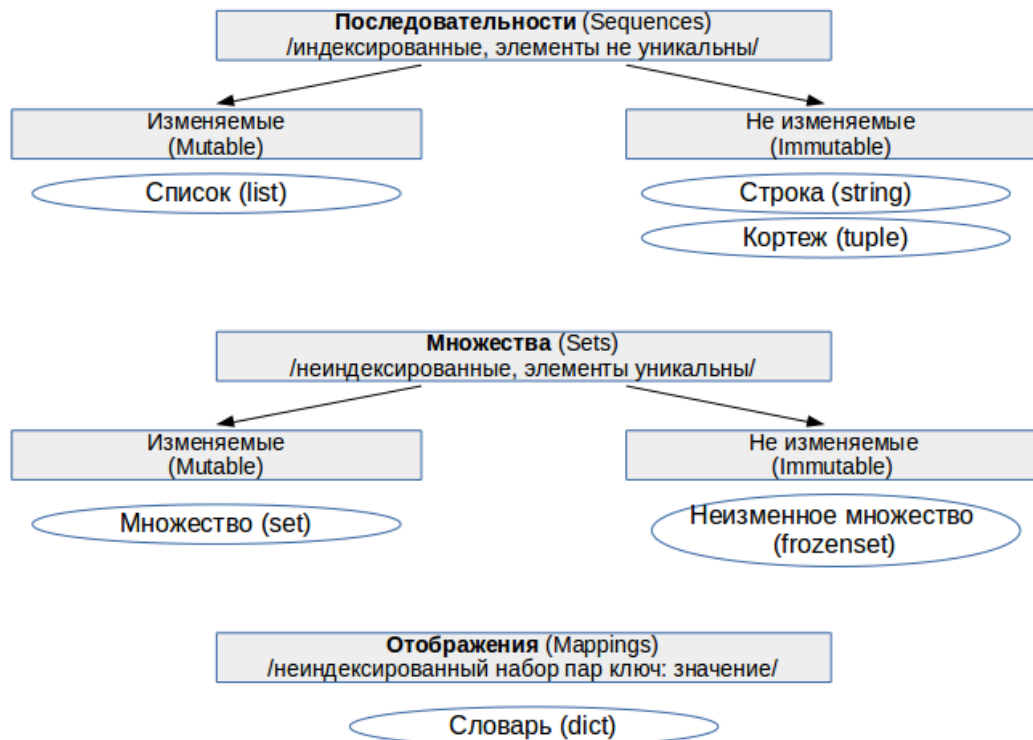
13. Строки. Основные методы строк.

Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

Строка может объявляться с помощью:

- “ - двух кавычек
- ‘ - одной кавычки

Строка - это тоже коллекция, которая относится к неизменяемым последовательностям



Следовательно, к элементам строки также можно обращаться по индексу, как и в случае с list и tuple:

```
Users > georgiydemo > Desktop > 🐍 check.py > ...
1   s = "тест"
2   print(s[0])
3   #s[0] = "у" <- ошибка
```

Экранирование символов происходит с помощью \
Если перед открывающей кавычкой стоит символ 'r' (в любом регистре),
то механизм экранирования отключается.
Существует ряд спецсимволов работы со строками:

\n	Перевод строки
\a	Звонок
\b	Забой
\f	Перевод страницы
\r	Возврат каретки
\t	Горизонтальная табуляция
\w	Вертикальная табуляция

Наиболее часто используемые - \n и \r\n при работе с многострочными файлами. Это зависит от кодировки файла и ОС.

Также можно писать многострочные комментарии при помощи “””:

```
Users > georgiydemo > Desktop > 🐍 check.py > ...
1   """
2   Пример многострочного
3   комментария в исходном коде Python
4   """
5   s = 0
```

Методы строк:

- **isalpha():** возвращает True, если строка состоит только из алфавитных символов
- **islower():** возвращает True, если строка состоит только из символов в нижнем регистре
- **isupper():** возвращает True, если все символы строки в верхнем регистре
- **isdigit():** возвращает True, если все символы строки - цифры
- **isnumeric():** возвращает True, если строка представляет собой число
- **startswith(str):** возвращает True, если строка начинается с подстроки str
- **endswith(str):** возвращает True, если строка заканчивается на подстроку str
- **lower():** переводит строку в нижний регистр
- **upper():** переводит строку в верхний регистр
- **title():** начальные символы всех слов в строке переводятся в верхний регистр
- **capitalize():** переводит в верхний регистр первую букву только самого первого слова строки
- **lstrip():** удаляет начальные пробелы из строки
- **rstrip():** удаляет конечные пробелы из строки
- **strip():** удаляет начальные и конечные пробелы из строки
- **ljust(width):** если длина строки меньше параметра width, то справа от строки добавляются пробелы, чтобы дополнить значение width, а сама

строка выравнивается по левому краю

- **rjust(width)**: если длина строки меньше параметра width, то слева от строки добавляются пробелы, чтобы дополнить значение width, а сама строка выравнивается по правому краю
- **center(width)**: если длина строки меньше параметра width, то слева и справа от строки равномерно добавляются пробелы, чтобы дополнить значение width, а сама строка выравнивается по центру
- **find(str[, start [, end]])**: возвращает индекс подстроки в строке. Если подстрока не найдена, возвращается число -1
- **replace(old, new[, num])**: заменяет в строке одну подстроку на другую
- **split([delimiter[, num]])**: разбивает строку на подстроки в зависимости от разделителя
- **join(strs)**: объединяет строки в одну строку, вставляя между ними определенный разделитель

Можно упомянуть про срезы

Можно упомянуть про оператор принадлежности in