

## 12. Строки. Операции над строками.

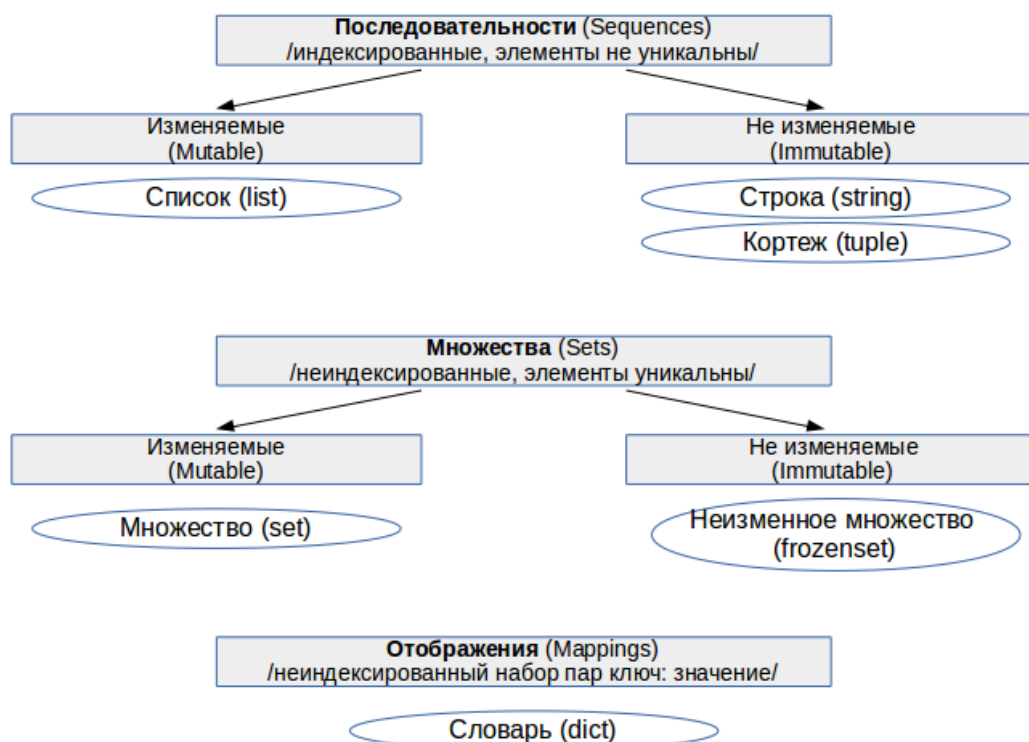
Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

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
\v	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
```

Базовые операции со строками:  
— Конкатенация (сложение)

```
>>> S1 = 'spam'
>>> S2 = 'eggs'
>>> print(S1 + S2)
'spameggs'
```

— Дублирование (умножение)

```
>>> print('spam' * 3)
spamspamspam
```

— Получение длины строки len

```
>>> len('spam')
4
```

Строки полностью поддерживают работу со срезами:

```
Python 3.7.6 (default, Dec 30 2019, 19:38:28)
[Clang 11.0.0 (clang-1100.0.33.16)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> str = "kotiki_meow"
>>> str[:1]
'k'
>>> str[1:]
'otiki_meow'
>>> █
```