МЕТОДЫ СПИСКОВ

Списки в Python - упорядоченные изменяемые коллекции объектов произвольных типов (почти как массив, но типы могут отличаться).

list.append(x) - добавляет элемент в конец списка.
x - параметр типа (string, number, object etc.)
list.insert(pos, x) - вставляет определенное значение на обозначенную позицию в списке.
pos - число, определяющее позицию x в списке
x - элемент типа (string, number, object etc.)
list.pop(i) - удаляет i-ый элемент и возвращает его. Если индекс не указан, удаляется последний элемент индексом [-1]. Идентичен методу .remove(), с той лишь разницей, что последний не возвращает удаленный элемент.
list.count(x) - возвращает частоту с которой элемент со значением x встречается в списке.
list.sort(key=..., reverse=...) - не требует обязательного параметра, однако может принимать дополнительные параметры:
key - (необязательный параметр) служит ключом для сортировки списка
reverse - (необязательный параметр) если True - сортирует список по нисходящей

МЕТОДЫ СЛОВАРЕЙ

Словари в Python - неупорядоченные коллекции произвольных объектов с доступом по ключу. Их иногда ещё называют ассоциативными массивами или хеш-таблицами.

dict.fromkeys(seq[, value]) - создает словарь с ключами из seq и значением value (по умолчанию None)
sequence - последовательность элементов которые будет внесены в словарь как ключи
value - (необязательный параметр) будет записан как значение для последовательности ключей
dict.get(key[, value]) - возвращает значение по ключу, при его отсутствии возвращает None
key - ключ по которому будет проходить поиск значения;
value - (не обязательный параметр) возвращает:
а) значение по ключу если ключ есть в словаре
б) None если ключ не найден и value не указано
в) сам value, если ключ не найден, но указан value
dict.items() - возвращает пары ключ:значение. Метод идентичен методу .viewitems() в Python. Метод .items() не принимает никаких параметров
dict.popitem() - удаляет и возвращает последнюю добавленную в словарь пару ключ:значение
dict.update([other]) - делает обновление словаря, учитывая параметр other, в который подается другой словарь или пары ключ:значение

МЕТОДЫ МНОЖЕСТВ

Множество в python - "контейнер", содержащий не повторяющиеся элементы в случайном порядке. Элементы множеств сами по себе не изменяемые, но само множество может изменятся путем операций добавления, пересечения и т.д. Множества - не упорядоченные последовательности, следовательно индексация в них не имеет смысла. Мы не можем также делать срезы с множеств.

set.add(elmnt) - метод добавляет элемент в множество при условии, что элемент уникальный и не повторяет другой такой же во множестве
set.remove(elmnt) - метод удаляет заданный элемент из множества
A.difference(B) - метод возвращает разницу двух множеств также в виде множества
A.intersection(*other_sets) - метод возвращает общие элементы двух множеств
*Метод позволяет принимать как параметр несколько множеств
A.union(*other_sets) - метод объединяет и возвращает множество из уникальных элементов из объединенных множеств

МЕТОДЫ СТРОК

Строка - это последовательность символов.
В Python строка - это последовательность символов Unicode.

str.split(separator, maxsplit) - разбивает строку на символы через указанные разделители и возвращает список строк, принимает 2 параметра:
separator (необязательный) - разделитель, по которому происходит разбиение. Если не указан, строка разделяется на символы через пробел.
maxsplit (необязательный) - максимальное количество разбиений. Если параметр не указан, количество разбиений не ограничено

str.replace(old, new [, count]) - заменяет указанный символ/текст в строке на новый символ/текст, может принимать 3 параметра:
old - старая подстрока, которую вы хотите заменить
new - новая подстрока, которая заменит старую подстроку
count (необязательный) - количество раз, которое вы хотите заменить старую подстроку на новую. Если значение count не указано, заменяет все вхождения старой подстроки на новую подстроку.

string.lower() - преобразует все прописные символы в строке в строчные и возвращает их. Не принимает никаких параметров

string.count(substring, start=..., end=...) - возвращает количество вхождений подстроки в строку.
substring - подстрока, количество которой считает метод
start (Необязательный) - начальный индекс в строке, с которого начинается поиск
end (Необязательно) - конечный индекс в строке, по которому заканчивается поиск

string.join(iterable) - возвращает строку, соединяя все элементы итерабельной переменной (список, строка, кортеж), разделенные строковым разделителем.