

11. Агрегирующие функции `sum()`, `min()`, `max()`. Сортировка.


В Python существует ряд агрегирующих функций для быстрого получения минимального, максимального элемента, а также суммы элементов. Данные функции применяются только к числовым типам данных, а именно к `int`, `float` и `complex`. Элементы коллекции необязательно должны быть только лишь одного типа, например:

```
sum([3.14j, 5.2, 5])
```

Отработает без ошибок

По своей сути агрегирующие функции являются просто удобной надстройкой для красивого синтаксиса, но можно их и не использовать.

К примеру, в коде ниже функция `my_max` является аналогом встроенной функции `max()` в Python, тоже самое можно сделать и с другими

```
Users > georgiydemo > Desktop >  check.py > ...  
1  def my_max(l):  
2      max_e = 0  
3      for e in l:  
4          if e > max_e:  
5              max_e = e  
6      return max_e  
7  
8  print(my_max([5,7,3,9,3]))
```

Агрегирующие функции могут применяться к большинству коллекций (`tuple`, `list`, `set`) где возможно хранение численных типов.

К `dict` может применяться только по методам `.keys()` или `.values()`, по теории можно применить и к `.items()`, но все эти методы все равно сводятся к простому `list`.

Сортировка элементов чаще всего применяется к `list`. Она может осуществляться несколькими способами:

- функция `sorted()`
- метод `sort()` у `list`

Основное их различие состоит только в том, что `sorted` возвращает новый отсортированный список, а метод `list.sort()` изменяет текущий список:

```
[Clang 11.0.0 (clang-1100.0.33.16)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> l = [5,4,7,1]
>>> l.sort()
>>> l
[1, 4, 5, 7]
>>>
>>> l = [5,4,7,1]
>>> sorted(l)
[1, 4, 5, 7]
>>> █
```

С методами сортировки очень часто применяется несколько аргументов:

- `key` - ключ по которому происходит сортировка элементов. Чаще всего пишут лямбда-выражение вида `key=lambda x: x`
- `reverse` - принимает `True/False`, за счет него можно регулировать сортировку по возрастанию или убыванию элементов

Тут можно рассказать про лямбда-выражения, но по факту ответ на билет есть