

Функции для работы со списками

Занятие № 23



Зачем изучать функции для работы со списками?

- знать встроенные функции, чтобы сэкономить себе время написания кода
- уметь создавать собственные функции под свою задачу,
 чтобы оптимизировать код
- делить задачу на части между сокомандниками

Своя функция

Как создать?

def имя(аргументы):

Как вернуть значение?

return что-то

Как вызвать?

переменная = имя(значения или переменные)

print(имя(значения или переменные))

Оптимизируем код сортировки выбором

```
for i in range(len(spisok) - 1):
   min indx = i
                                           Сделаем функцию
                                           вместо этого кода
   for j in range(i + 1, len(spisok)):
        if spisok[j] < spisok[min_indx]:</pre>
            \min \ indx = j # находим индекс минимального элемента
   # меняем местами минимальный и стартовый (і-ый)
   spisok[i], spisok[min indx] = spisok[min indx], spisok[i]
```

Задание

Написать функцию find_min_index(s), которая получает на вход список s и возвращает индекс минимального элемента

Оптимизируем код сортировки выбором

```
def find_min_index(s):
    for j in range(len(s)):
        if s[j] < s[min_indx]:
            min_indx = j
    return min_indx</pre>
```

Как теперь будет выглядеть код сортировки?



Встроенные функции



- sum(список) возвращает сумму всех элементов
- min(список) минимум из набора элементов
- тах(список) максимум из набора элементов

Какого типа объект возвращают?

Того типа, что были элементы в списке





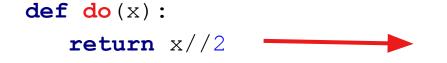
Функция тар



!!! Результат обязательно снова превращать в список, иначе получим объект типа map



Функция тар



это действие применяется к каждому элементу

```
s = [2, 5, 1, 6, 11]
s1 = list(map(do, s))
print(*s1)
```

Получим: 12035

Функция тар



это действие применяется к каждому элементу

важно!

Если вы обрабатываете список из чисел или строк, то ваша функция должна **принимать ровно 1 аргумент**

Получим: 12035





Метод в тар

```
names = ["olga", "anna", "kate"]
names = list(map(str.capitalize, names))
print(*names)
```

Получим: Olga Anna Kate