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
    for j in range(i + 1, len(spisok)):
        if spisok[j] < spisok[min_indx]:
            min_indx = j   # находим индекс минимального элемента

    # меняем местами минимальный и стартовый (i-ый)
    spisok[i], spisok[min_indx] = spisok[min_indx], spisok[i]
```

Сделаем функцию вместо этого кода

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
```

Как теперь будет выглядеть код сортировки?

```
for i in range(len(spisok) - 1):

    min_indx = i + find_min_index(spisok[i:])

    # меняем местами минимальный и стартовый (i-ый)
    spisok[i], spisok[min_indx] = spisok[min_indx], spisok[i]
```

находим индекс минимума в оставшейся части списка

Встроенные функции

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

Какого типа объект возвращают?
Того типа, что были элементы в списке

Функция map

```
x = list(map(функция, список))
```

любая функция

любой список

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

Функция map

```
def do(x):
    return x//2
```

это действие применяется к каждому элементу

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

Получим: 1 2 0 3 5

Функция map

```
def do(x):
    return x//2
```

это действие применяется к каждому элементу

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

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

Получим: 1 2 0 3 5

Метод в map

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

Получим: Olga Anna Kate