

# Практическая работа №5

1) Создайте функцию **reverse**(lst), которая принимает список и возвращает его перевёрнутый вариант.

2) Напишите функцию **sum\_range**(start, end), которая суммирует все целые числа от значения «start» до величины «end» включительно.

Если пользователь задаст первое число большее чем второе, просто поменяйте их местами.

3) Создайте функцию **my\_range**(start, end, step), которая возвращает список чисел в заданном диапазоне. Функция должна работать как и обычный **range**(если вы забыли как ей пользоваться, то можете прочесть в документации)

4) Создайте функцию **abbreviation**(text), которая принимает строку и возвращает аббревиатуру, где буквы разделены точкой.

Пример работы: `abbreviation('Python is ease')` => 'P.I.E'

5) Напишите функцию **to\_dict**(lst), которая принимает аргумент в виде списка и возвращает словарь, в котором каждый элемент списка является и ключом и значением.

6) Создайте функцию **to\_dict**(text), которая принимает строку и возвращает словарь, ключами которого будут символы строки, а значениями – количество вхождения этого символа.

7) Сделайте функцию **\_if**(value, fn1, fn2), которая принимает 3 аргумента: булево значение и 2 функции. Если булево значение правдиво, то следует вызвать 1 функцию, если ложно – вторую. Пример: `_if(True, truthy, falsey)` => `truthy()`. То есть вызвана 1 функция

8) Создать словарь **calculator**, у который имеет следующие методы: сложение(summ), умножение(mul), среднее арифметическое(average) и процент(percent). Последний метод находит количество процентов, которые составляет число 1 от числа 2(8 от 10 равно 80%).

9) Реализовать функцию **my\_map**(fn, lst). Она должна принимать два параметра: функцию и список и вернуть новый список.

Пример работы: `my_map(lambda x: x * 2, [1, 2, 3]) => [2, 4, 6]`. В данном примере мы переделали функцию, которая умножает переданный аргумент на два, таким образом все элементы списка увеличились в два раза.

10) Реализовать функцию **my\_filter**(fn, lst), которая принимает два параметра: функцию и список и возвращает новый список.

Пример работы: `my_filter(lambda x: x > 3, [1, 2, 3, 4, 5, 6]) => [4, 5, 6]`. Тут мы переделали функцию, которая проверяет, является ли число больше тройки. Если да, то число проходит в новый список.