# 📘 Теория: сокращённый синтаксис и срезы в Python

## 1. **List comprehension** (генераторы списков)

Иногда нужно создать новый список из старого. Вместо цикла можно использовать короткую запись:

### 🔹 Обычный цикл:

numbers = [1, 2, 3, 4, 5]

squares = []

for n in numbers:

squares.append(n \*\* 2)

print(squares) # [1, 4, 9, 16, 25]

### 🔹 Сокращённый вариант (list comprehension):

numbers = [1, 2, 3, 4, 5]

squares = [n \*\* 2 for n in numbers]

print(squares) # [1, 4, 9, 16, 25]

👉 Можно добавлять условия:

evens = [n for n in numbers if n % 2 == 0]

print(evens) # [2, 4]

## 2. **Срезы (slicing)**

Срезы позволяют удобно брать куски списков и строк. Запись:

список[начало:конец:шаг]

* начало — индекс, с которого начинаем (включительно)
* конец — индекс, на котором останавливаемся (не включая)
* шаг — через сколько брать элементы

### 🔹 Примеры:

letters = ['a', 'b', 'c', 'd', 'e']

print(letters[1:4]) # ['b', 'c', 'd'] (с индекса 1 до 4, не включая 4)

print(letters[:3]) # ['a', 'b', 'c'] (с начала до 3)

print(letters[2:]) # ['c', 'd', 'e'] (с 2 до конца)

print(letters[::2]) # ['a', 'c', 'e'] (каждый второй элемент)

print(letters[::-1]) # ['e', 'd', 'c', 'b', 'a'] (разворот списка)

## 3. **Срезы со строками**

То же самое работает и для строк (ведь строка — это тоже последовательность):

word = "python"

print(word[0:2]) # 'py'

print(word[:3]) # 'pyt'

print(word[::2]) # 'pto' (каждая 2-я буква)

print(word[::-1]) # 'nohtyp' (разворот строки)

## 4. **Комбинация: comprehension + срезы**

Можно комбинировать:

numbers = [1, 2, 3, 4, 5, 6, 7, 8]

even\_squares = [n\*\*2 for n in numbers if n % 2 == 0]

print(even\_squares) # [4, 16, 36, 64]

print(even\_squares[::-1]) # [64, 36, 16, 4] (разворот списка)

**Задача 1. Минимум и максимум в списке**

**ТЗ:**  
Напиши функцию find\_min\_max(numbers), которая принимает список чисел и возвращает кортеж (минимум, максимум).  
Программа должна запросить у пользователя числа через пробел, превратить их в список и вывести результат.

### Задача 2. Среднее арифметическое

**ТЗ:**  
Напиши функцию average(numbers), которая принимает список чисел и возвращает их среднее арифметическое.  
Программа должна обработать случай, если список пуст (вернуть 0).

### Задача 3. Фильтрация чётных чисел

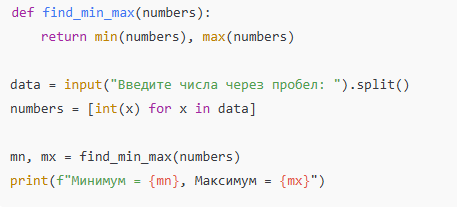
**ТЗ:**  
Напиши функцию filter\_even(numbers), которая возвращает список только чётных чисел.  
Программа должна запросить список у пользователя и вывести результат.

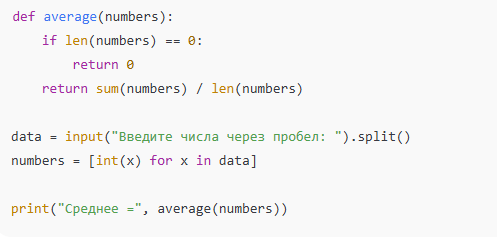
### Задача 4. Сумма положительных и отрицательных чисел

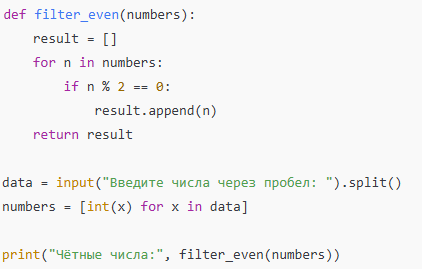
**ТЗ:**  
Напиши функцию sum\_pos\_neg(numbers), которая возвращает кортеж (сумма\_положительных, сумма\_отрицательных).  
Ноль игнорируем.

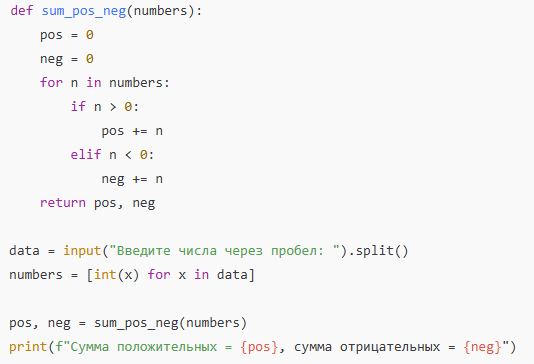
### Задача 5. Палиндром

**ТЗ:**  
Напиши функцию is\_palindrome(word), которая проверяет, является ли слово палиндромом (читается одинаково слева и справа).  
Программа должна попросить у пользователя слово и вывести результат.

1 

2

3

4

5