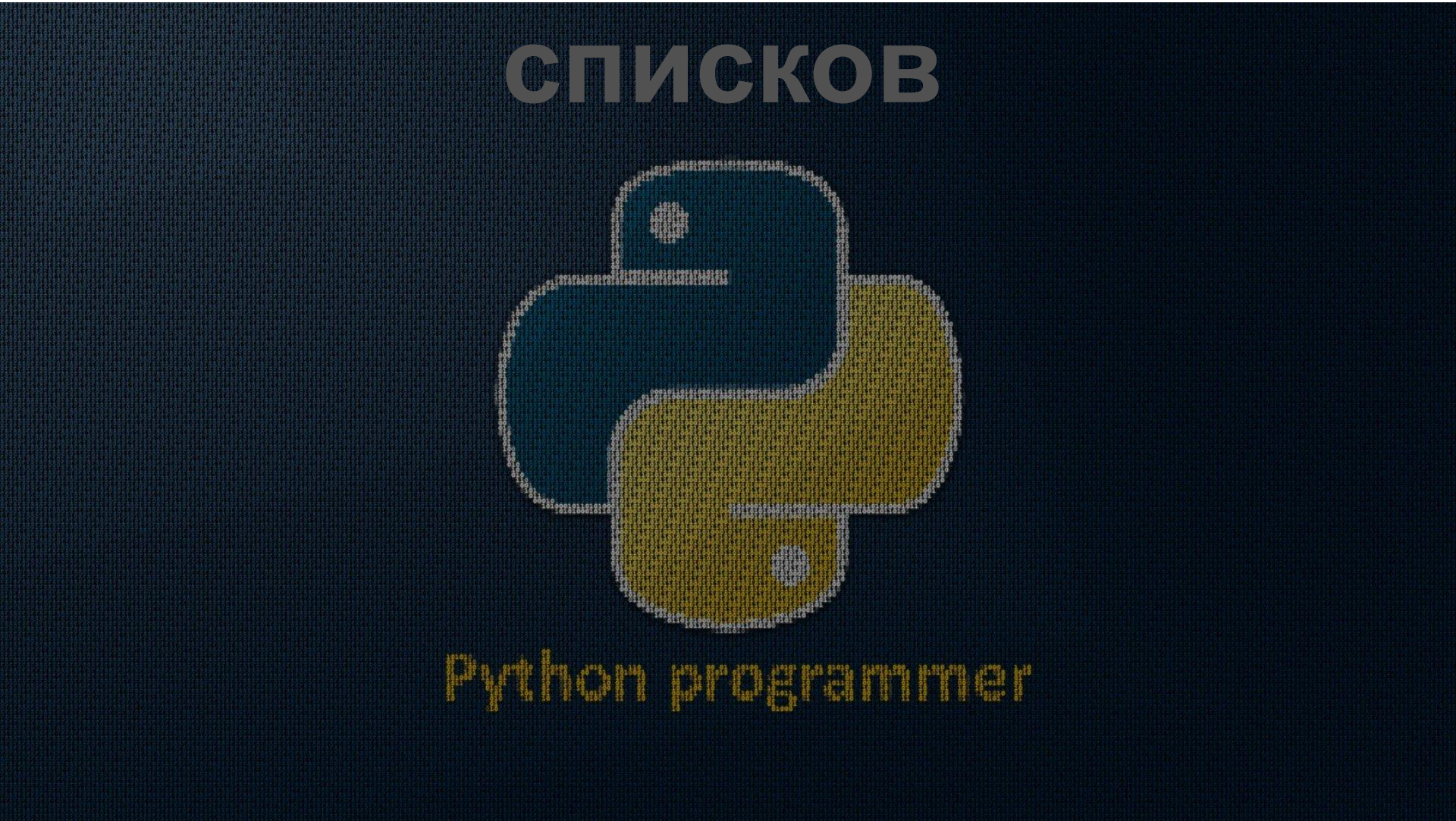
Списки.
Функции и методы списков



Списки в Python

- упорядоченные изменяемые коллекции объектов произвольных типов.

Базовые операции

Операция	Назначение	Пример
+	Конкатенация (сложение)	nomer=[12,34,56,32] print(nomer+[78,98]) Результат: [12,34,56,32,78,98]
*n	Дублирование списка	nomer=[12,34,56,32] print(nomer*2) Результат: [12, 34, 23, 45, 12, 34, 23, 45]
list[n]	Доступ по индексу	nomer=[12,34,23,45] print(nomer[2]) Результат: 23
list[x:y]	Извлечение среза (x-индекс начала среза y-индекс окончания среза)	nomer=[12,34,23,45] print(nomer[0:2]) Результат: 12,34

Базовые операции

Операция	Назначение	Пример
Max(list)	Максимальный элемент	nomer=[12,34,23,45] print(max(nomer))
Min(list)	Минимальный элемент	nomer=[12,34,23,45] print(min(nomer))
len(list)	Длина списка (функция)	nomer=[12,34,23,45] print(len(nomer))

Методы списков

Метод	Назначение	Пример
sort	Сортировка списка	nomer=[12,34,23,45] nomer.sort() print(nomer) Результат: [12,23,34,45]
reverse	Реверс строки	nomer=[12,34,23,45] nomer.reverse() print(nomer) Результат: [45,23,34,12]
count	Количество вхождений подстроки в списке, если нет совпадений - 0	list.count('то что ищем')
Index	Найти индекс первого вхождения конкретного элемента	nomer=[12,34,23,45] print(nomer.index(23)) Результат: 2

Задачи

- Дан список: rating=[15,87,54,23,14,78]

Определить:

1. длину списка;
2. номер минимального балла рейтинга;
3. вывести максимальный балл рейтинга.

Вывести список, упорядочив его по убыванию