
■ Лабораторная работа №4

Тема: Списки и строки. Циклы по коллекциям. Методы списков.

🎯 Цели работы

1. Научиться работать со **списками и строками**.
 2. Освоить перебор коллекций с помощью циклов for.
 3. Изучить **основные методы списков** (append, insert, extend, remove, pop, clear, index, count, sort, reverse, copy).
 4. Закрепить работу с числами в задачах обработки данных.
-

1. Создание и работа со списками чисел

```
nums = [10, 20, 30, 40, 50]
print(nums)      # [10, 20, 30, 40, 50]
print(nums[0])   # первый элемент → 10
print(nums[-1])  # последний элемент → 50
nums[2] = 99     # замена элемента
print(nums)      # [10, 20, 99, 40, 50]
```

2. Методы списков

Добавление

```
nums = [1, 2, 3]
nums.append(4)    # добавление в конец
nums.insert(1, 99) # вставка по индексу
nums.extend([5, 6]) # добавление другого списка
print(nums)       # [1, 99, 2, 3, 4, 5, 6]
```

Удаление

```
nums = [10, 20, 30, 40]
nums.remove(20)   # удаляет первый найденный элемент
x = nums.pop()    # удаляет последний (вернёт 40)
y = nums.pop(0)   # удаляет первый (вернёт 10)
print(nums)       # [30]
nums.clear()      # очищает список
print(nums)       # []
```

Поиск

```
nums = [5, 10, 15, 10, 20]
print(nums.index(10)) # индекс первого вхождения → 1
print(nums.count(10)) # количество элементов → 2
```

Сортировка и разворот

```
nums = [4, 2, 9, 1]
nums.sort()          # сортировка по возрастанию
print(nums)          # [1, 2, 4, 9]
nums.sort(reverse=True) # сортировка по убыванию
print(nums)          # [9, 4, 2, 1]
nums.reverse()       # просто переворот
print(nums)          # [1, 2, 4, 9]
```

Копирование

```
nums = [1, 2, 3]
copy_nums = nums.copy()
print(copy_nums)    # [1, 2, 3]
```

3. Циклы по спискам

Сумма элементов

```
nums = [2, 4, 6, 8, 10]
s = 0
for n in nums:
    s += n
print("Сумма =", s)
Поиск максимума
nums = [15, 3, 27, 8, 12]
m = nums[0]
for n in nums:
    if n > m:
        m = n
print("Максимум =", m)
Новый список (квадраты чисел)
nums = [1, 2, 3, 4, 5]
squares = []
for n in nums:
    squares.append(n**2)
print(squares) # [1, 4, 9, 16, 25]
Генерация списка чисел от 1 до 10
nums = []
for i in range(1, 11):
    nums.append(i)
print(nums)
```

4. Строки и их обработка

```
text = "Python"
print(text[0])    # P
print(text[-1])   # n
for ch in text:
    print(ch)      # перебор символов
Методы строк:
s = " hello, world! "
print(s.upper())   # HELLO, WORLD!
print(s.lower())   # hello, world!
print(s.strip())   # hello, world!
print(s.replace("world", "Python")) # hello, Python!
```

5. Полезные функции

```
nums = [3, 7, 2, 9, 5]
print(sum(nums))   # 26
print(max(nums))   # 9
print(min(nums))   # 2
print(sorted(nums)) # [2, 3, 5, 7, 9]
Подсчёт символов в строке:
text = "banana"
print(text.count("a")) # 3
```

6. Примеры задач

Сумма чётных чисел списка

```
nums = [1, 2, 3, 4, 5, 6, 7, 8]
s = 0
for n in nums:
    if n % 2 == 0:
```

```
s += n
print("Сумма чётных =", s)
Поиск второго максимума
nums = [10, 20, 50, 40, 30]
nums.sort()
print("Второй максимум =", nums[-2])
Перевернуть список вручную
nums = [1, 2, 3, 4, 5]
rev = []
for i in range(len(nums)-1, -1, -1):
    rev.append(nums[i])
print(rev)
Подсчёт гласных в строке
text = "Привет мир"
vowels = "аеёиоуыэюяАЕЁИОУЫЭЮЯ"
count = 0
for ch in text:
    if ch in vowels:
        count += 1
print("Гласных букв:", count)
```

7. Задания для самостоятельной работы

1. Создать список чисел от 1 до 20.
 2. Найти сумму всех нечётных чисел списка.
 3. Создать список квадратов чисел от 1 до 15.
 4. В списке [5, 1, 5, 2, 5, 3, 4] посчитать количество числа 5.
 5. Сгенерировать список чисел от 1 до 50, затем удалить все числа, которые делятся на 3.
 6. Создать список из 10 чисел, отсортировать его и вывести второй минимум и второй максимум.
 7. С помощью цикла сформировать список, где каждый элемент = сумма чисел от 1 до i (например [1, 3, 6, 10, 15, ...]).
 8. С помощью метода append() создать список чётных чисел от 2 до 20.
 9. Найти среднее арифметическое списка чисел.
 10. Объединить два списка чисел в один (extend или +) и вывести сумму всех элементов.
 11. Найти индекс максимального элемента в списке.
 12. Создать список случайных чисел и вывести его в обратном порядке.
 13. Дана строка, подсчитать количество пробелов.
 14. Проверить, является ли строка палиндромом (читается одинаково в обе стороны).
 15. Разбить строку "Python is very cool" на отдельные слова и вывести их построчно.
-