



# Введение в язык программирования Python

# Практическая работа



<https://play.myquiz.ru/>



# Списки



Список (list) - это тип данных, представляющий собой упорядоченный набор элементов.

Элементы одно списка должны иметь одинаковый тип. Например, список строк, список целых чисел.

Примеры СПИСКОВ:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
```

```
b = [17, 409, 88]
```



Каждый элемент списка имеет свой номер (индекс). Нумерация элементов начинается с нуля:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
```

0	1	2	3	4
Андрей	Вера	Даша	Коля	Юра
a[0]	a[1]	a[2]	a[3]	a[4]

С каждым элементом списка можно работать отдельно:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]  
print(a[2])
```

Чтобы перебрать все элементы списка используется цикл:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]  
for i in range(5):  
    print("Привет,", a[i])
```



1) Перечислением всех элементов:

```
b = [17, 409, 88]
```

Можно создать пустой список:

```
a = []
```

2) С помощью генератора списка:

```
a = []*5
```

Получится список, состоящий из пяти нулей: [0, 0, 0, 0, 0].

```
a = [ i for i in range(10) ]
```

Получится список [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

```
a = [ i*i for i in range(10) ]
```



3) Путём ввода элементов с клавиатуры (каждый элемент с новой строки):

```
a = [0]*5
for i in range(5):
    print("a[" + str(i) + "]= ", end = "")
    a[i] = int(input())
```

4) Путём ввода элементов с клавиатуры (все элементы в одной строке через пробел). Для этого используется метод `a.split()`, возвращающий список строк, которые получатся, если исходную строку разрезать на части по пробелам:

```
a = [0]*5
s = input() # человек вводит строку "1 2 3"
a = s.split()
print(a)
```

Вывод: ['1', '2', '3']

# Способы вывода списков

1) С помощью функции print():

```
b = [17, 409, 88]  
print(b)
```

Вывод: [17, 409, 88]

2) Вывод каждого элемента списка по-отдельности:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]  
for i in range(5):  
    print(a[i])
```



3) Вывод каждого элемента списка по-отдельности в одной строке:

```
a = ["Андрей", "Вера", "Даша", "Коля", "Юра"]
for i in range(5):
    print(a[i], end = " ")
```

4) Вывод элементов списка без обращения к индексам элементов:

```
fruits = ["Яблоко", "Банан", "Груша"]
for x in fruits:
    print(x, end = " ")
```

# Работа со списком

1) Элементы списка можно изменять:

```
a = [1, 2, 3, 4, 5, 6]
for i in range(6):
    if a[i] % 2 == 0:
        a[i] = 0
print(a)
```

Вывод: [1, 0, 3, 0, 5, 0]

2) Можно добавлять элементы в конец списка. Для этого используется метод `a.append(x)`:

```
++ a = [1, 2, 3]
++ a.append(4)
++ print(a)
```

Вывод: [1, 2, 3, 4]




3) Можно расширять список, добавляя в его конец элементы другого списка. Для этого используется метод **a.extend(b)**:

```
a = [1, 2, 3]
b = [4, 5]
a.extend(b)
print(a)
```

Вывод: [1, 2, 3, 4, 5]

4) Списки можно копировать:



```
a = [1, 2, 3]
b = a
print(b)
```

5) Функция нахождения длины списка **len(a)**:

```
a = ["Яблоко", "Банан", "Груша"]
x = len(a)
print(x)
```

6) Заполнение списка случайными элементами:

```
from random import randint
x = 7
a = [0]*x
for i in range(x):
    a[i] = randint(0,100)
```

7) Перестановка элементов списка в обратном порядке. Метод **a.reverse()**:

```
a = [0, 1, 2, 3, 4, 5]
a.reverse()
print(a)
```

Вывод: [5, 4, 3, 2, 1, 0]



8) Сортировка списка. Функция **sorted(a)**:

По возрастанию:

```
animals = ["кот", "еж", "собака", "барсук"]  
animals = sorted(animals)  
print(animals)
```

Вывод: ['барсук', 'еж', 'кот', 'собака']

По убыванию:

```
a = [5, 65, 14, 700, 8]  
a = sorted(a, reverse = True)  
print(a)
```

Вывод: [700, 65, 14, 8, 5]

# Матрицы в Python. Двумерные массивы

Матрица — математический объект, записываемый в виде прямоугольной таблицы элементов кольца или поля (например, целых, действительных или комплексных чисел), который представляет собой совокупность строк и столбцов, на пересечении которых находятся его элементы.

$A = [[1, 2, 3], [4, 5, 6]]$

# Матрицы в Python. Двумерные массивы

Для обработки и вывода списка как правило используется два вложенных цикла. Первый цикл по номеру строки, второй цикл по элементам внутри строки. Например, вывести двумерный числовой список на экран построчно, разделяя числа пробелами внутри одной строки, можно так:

```
for i in range(len(A)):  
    for j in range(len(A[i])):  
        print(A[i][j], end=' ')  
print()
```



# Матрицы в Python. Двумерные массивы

То же самое, но циклы не по индексу, а по значениям списка:

```
for row in A:  
    for elem in row:  
        print(elem, end=' ')  
    print()
```





# Матрицы в Python. Двумерные массивы



Используем два вложенных цикла для подсчета суммы всех чисел в списке:

```
S = 0
for i in range(len(A)):
    for j in range(len(A[i])):
        S += A[i][j]
```

Или то же самое с циклом не по индексу, а по значениям строк:

```
S = 0
for row in A:
    for elem in row:
        S += elem
```



# Создание вложенных списков

Первый способ: сначала создадим список из  $n$  элементов (для начала просто из  $n$  нулей). Затем сделаем каждый элемент списка ссылкой на другой одномерный список из  $m$  элементов:

```
A = [0] * n  
for i in range(n):  
    A[i] = [0] * m
```



# Создание вложенных списков

Другой (но похожий) способ: создать пустой список, потом  $n$  раз добавить в него новый элемент, являющийся списком-строкой:

```
A = []  
for i in range(n):  
    A.append([0] * m)
```

Но еще проще воспользоваться генератором: создать список из  $n$  элементов, каждый из которых будет списком, состоящих из  $m$  нулей:

```
+ A = [[0] * m for i in range(n)]
```



# Ввод двумерного массива

Используем два вложенных цикла для подсчета суммы всех чисел в списке:

```
A = []  
for i in range(n):  
    row = input().split()  
    for i in range(len(row)):  
        row[i] = int(row[i])  
    A.append(row)
```

Или:

```
+ array = [[input() for j in range(y)] for i in range(x)]
```



# Домашнее задание

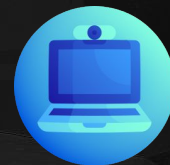


Дана квадратная матрица (минимум  $3 \times 3$ ), состоящая только из нулей. Заполнить ее случайными значениями. Вывести полученную матрицу в виде таблицы. Найти сумму элементов побочной диагонали (вне зависимости от размеров матрицы). Например:

В матрице вида

1	2	<u>3</u>
4	<u>5</u>	6
<u>7</u>	8	9

Элементы побочной диагонали выделены красным цветом, соответственно их сумма =  $3+5+7=15$





**ПОДВЕДЕМ ИТОГИ**

