

Списки в Python^a

1.1 Работа со списком целиком

Примеры работы со списком целиком

```
import math # нужно только для prod

a = [4, -9, 50, 19, 21, 4, -7] # инициализация списка
b = [] # инициализация пустого списка

# вывод списка на экран
print(a) # [4, -9, 50, 19, 21, 4, -7]
print(b) # []

# вывод списка на экран с "разыменованием"
print(*a) # 4 -9 50 19 21 4 -7

# вывод длины (количество элементов) списка
print(len(a)) # 7
print(len(b)) # 0

# вывод количества вхождений числа 3 в список
print(a.count(4)) # 2

# вывод минимального числа в списке
print(min(a)) # -9

# вывод максимального числа в списке
print(max(a)) # 50

# сумма элементов в списке
print(sum(a)) # 82

# среднее арифметическое всех элементов списка
print(sum(a)/len(a)) # 11.714285714285714

# целая часть среднего арифметического всех элементов списка
print(int(sum(a)/len(a))) # 11

# произведение элементов в списке (требует импорта из math)
print(math.prod(a)) # 20109600
```

^aАвтор: Е. В. Ширяева, <https://t.me/EVShiryaeva>, 2025.

Примеры работы со списком целиком (модификация списка)

```
a = [4, -9, 50, 19, 21, 4, -7] # инициализация списка

# переворот (инвертирование) списка
a.reverse() # или a = a[::-1]
print(a) # [-7, 4, 21, 19, 50, -9, 4]

# сортировка списка по неубыванию
a.sort() # изменяет список a
# или a = sorted(a) # действие sorted(a) не изменяет исходного списка
print(a) # [-9, -7, 4, 4, 19, 21, 50]

# сортировка списка по невозрастанию
a.sort(reverse=True)
# или a = sorted(a, reverse=True)
print(a) # [50, 21, 19, 4, 4, -7, -9]

# добавление одного элемента в КОНЕЦ списка
a += [100] # или a.append(100)
print(a) # [50, 21, 19, 4, 4, -7, -9, 100]

# добавление нескольких элементов в КОНЕЦ списка
a += [-3, -4] # или a.extend([-3, -4])
print(a) # [50, 21, 19, 4, 4, -7, -9, 100, -3, -4]

# добавление одного элемента в НАЧАЛО списка
a = [-100] + a
print(a) # [-100, 50, 21, 19, 4, 4, -7, -9, 100, -3, -4]

# добавление нескольких элементов в НАЧАЛО списка
a = [22, 55] + a
print(a) # [22, 55, -100, 50, 21, 19, 4, 4, -7, -9, 100, -3, -4]
```

1.2 Работа с фрагментами списка (срезы)

Примеры обращения к элементам списка

```
#      0   1   2   3   4   5   6   неотрицательная нумерация элементов
a = [2, 34, 6, -7, 12, -10, 13]
#   -7  -6  -5  -4  -3  -2  -1   отрицательная нумерация элементов

# первый (слева) элемент списка
print(a[0])    # 2

# последний элемент списка
print(a[-1])   # 13

# срез с третьего элемента и до конца
print(a[2:])   # [6, -7, 12, -10, 13]

# срез с первыми тремя элементами
print(a[:3])   # [2, 34, 6]

# срез всех элементов кроме первого и последнего
print(a[1:-1]) # [34, 6, -7, 12, -10]

# взять каждый второй элемент, начиная с нулевого
print(a[::-2]) # [2, 6, 12, 13]

# взять каждый второй второй, начиная с первого
print(a[1::2]) # [34, -7, -10]

# инвертирование списка (переворот)
print(a[::-1]) # [13, -10, 12, -7, 6, 34, 2]
```

1.3 Перебор элементов списка

Примеры печати элементов списка

```
a = [2, 34, 6, -7, 12, -10, 13]

# печать списка по элементам (в одну строку через пробел)
for x in a:
    print(x, end=' ') # 2 34 6 -7 12 -10 13

print() # пустая строка
# печать чётных элементов списка
for x in a:
    if x % 2 == 0:
        print(x, end=' ') # 2 34 6 12 -10

print()
# печать двузначных элементов списка
for x in a:
    if 10 <= abs(x) <= 99:
        print(x, end=' ') # 34 12 -10 13

print()
# печать двузначных элементов списка, меньших его последнего элемента
for x in a:
    if 10 <= abs(x) <= 99 and x < a[-1]:
        print(x, end=' ') # 12 -10
```

Печать пар элементов списка

```
a = [2, 34, 6, -7, 12, -10, 13]

# печать пар соседних элементов списка
for i in range(len(a)-1): # цикл по индексам
    print(a[i], a[i+1])
```

Результат

```
12 -10
2 34
34 6
6 -7
-7 12
12 -10
-10 13
```

Печать троек элементов списка

```
a = [2, 34, 6, -7, 12, -10, 13]

# печать троек соседних элементов списка
for i in range(len(a)-2): # цикл по индексам
    print(a[i], a[i+1], a[i+2])
```

Результат

```
2 34 6
34 6 -7
6 -7 12
-7 12 -10
12 -10 13
```

Примеры использования стандартных алгоритмов

```
a = [2, 34, 6, -7, 12, -10, 13]

# Поиск количества чётных элементов списка
cnt = 0
for x in a:
    if x % 2 == 0:
        cnt += 1
print(cnt) # 5

# Поиск суммы всех элементов списка
sm = 0
for x in a:
    sm += x
print(sm) # 50

# Поиск произведение положительных элементов списка
pr = 1
for x in a:
    if x > 0:
        pr *= x
print(pr) # 63648

# Поиск среднего арифметического двузначных элементов списка
sm = 0
cnt = 0
for x in a:
    if 10 <= abs(x) <= 99:
        sm += x
        cnt += 1
average = sm / cnt
print(average) # 12.25

# Формирование нового списка с индексами отрицательных элементов
t = []
for i in range(len(a)): # цикл по индексам
    if a[i] < 0:
        t += [i]
print(t) # [3, 5]
```

1.4 Создание списков на основе заданного списка

Синтаксис генератора списка

```
[eval for x in a]
```

eval — выражение для получения элемента списка; x — переменная для элемента списка; a — исходный список (задан некоторым способом).

Пример использования генератора списка

```
a = [34, -90, 56, 19, 21, 34, -78, 0]
```

```
# Создать новый список, в котором все элементы будут иметь противоположный знак  
a1 = [-x for x in a]
```

```
# вывод нового списка на экран  
print(a1) # [-34, 90, -56, -19, -21, -34, 78, 0]
```

Синтаксис генератора списка с условием отбора condition

```
[eval for x in a if condition]
```

eval — выражение для получения элемента списка; x — переменная для элемента списка; a — исходный список; condition — условие отбора.

Примеры использования генераторов списков с условием

```
a = [34, -90, 56, 19, 21, 34, -78, 0]
```

```
# Создать список с положительными элементами  
a2 = [x for x in a if x > 0]  
print(a2) # [34, 56, 19, 21, 34]
```

```
# Создать список с чётными элементами  
a3 = [x for x in a if x % 2 == 0]  
print(a3) # [34, -90, 56, 34, -78, 0]
```

```
# Создать список с положительными элементами, кратными трём  
a4 = [x for x in a if x > 0 and x % 3 == 0]  
print(a4) # [21]
```

```
# Создать список с положительными элементами, двоичная запись которых  
# содержит нечётное количество единиц  
a5 = [x for x in a if x > 0 and bin(x).count('1') % 2 != 0]  
print(a5) # [56, 19, 21]
```

Задания для самостоятельной работы

№ List.1 (➡). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Примечание. После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = [3, 24, -56, 10, 201, 234, -78, 0, 113, -45, 61]

# Создать список с удвоенными элементами списка а
a1 = [Здесь должен стоять Ваш код]
print(a1) # [6, 48, -112, 20, 402, 468, -156, 0, 226, -90, 122]

# Создать список из последних цифр каждого элемента списка
a2 = []
print(a2) # [3, 4, 6, 0, 1, 4, 8, 0, 3, 5, 1]

# Создать список умножением каждого элемента списка а на его 1 элемент
a3 = []
print(a3) # [9, 72, -168, 30, 603, 702, -234, 0, 339, -135, 183]
```

№ List.2 (➡). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Примечание. После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = [3, 24, 8, 1234, 9, 0, 127, 88]

# Создать список, в котором все элементы списка а переведены в 8-ную с.с.
a1 = [Здесь должен стоять Ваш код]
print(a1) # ['3', '30', '10', '2322', '11', '0', '177', '130']

# Создать список, в котором все элементы списка а переведены в 16-ную с.с.
a2 = []
print(a2) # ['3', '18', '8', '4d2', '9', '0', '7f', '58']

# Создать список с количеством цифр в 2-ной записи каждого элемента списка а
a3 = []
print(a3) # [2, 5, 4, 11, 4, 1, 7, 7]
```

№ List.3 (➡). Дан список строк, представляющих собой двоичные числа. Для данного списка запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Примечание. После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = ['1011', '101', '11101', '1101011', '1001', '111000110010']

# Создать список, в котором все элементы списка а заменены длиной двоичного числа
a1 = [Здесь должен стоять Ваш код]
print(a1) # [4, 3, 5, 7, 4, 12]

# Создать список, в котором все элементы списка а переведены из 2-ной с.с. в 10-ную
a2 = []
print(a2) # [11, 5, 29, 107, 9, 3634]

# Создать список, в котором все элементы списка а переведены из 2-ной с.с. в 8-ную
a3 = []
print(a3) # ['13', '5', '35', '153', '11', '7062']

# Создать список, в котором все элементы списка а переведены из 2-ной с.с. в 16-ную
a4 = []
print(a4) # ['b', '5', '1d', '6b', '9', 'e32']
```

№ List.4 (➡). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Примечание. После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = [3, 24, -56, 10, 201, 234, -78, 0, 113, -45, 61]

# Создать список с двузначными элементами списка а
a1 = [Здесь должен стоять Ваш код]
print(a1) # [24, -56, 10, -78, -45, 61]

# Создать список с неположительными элементами списка а, увеличенными на 100
a2 = []
print(a2) # [44, 22, 100, 55]

# Создать список с отрицательными элементами списка а, поделенными нацело на 3
a3 = []
print(a3) # [-19, -26, -15]
```

№ List.5 (➡). Для заданного списка строк *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Примечание. После операторов печати в коде стоят ожидаемые ответы для сравнения.

[Допишите генераторы списков](#)

```
a = ['194', '26', '11101', '1101013', '1001', '78', '37']

# Создать список из элементов списка a с длиной не более 4 символов
a1 = [Здесь должен стоять Ваш код]
print(a1) # ['194', '26', '1001', '78', '37']

# Создать список из элементов списка a, которые
#                               НЕ могут представлять собой восьмеричную запись
a2 = []
print(a2) # ['194', '78']

# Создать список из элементов списка a, которые представляют собой
#                               правильную запись двоичного числа
a3 = []
print(a3) # ['11101', '1001']
```

Задания для дополнительной отработки

№ List.6 (➡). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Примечание. После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = [-34, -90, 56, 19, 21, 134, -78, 0, 11, -345, 61, 84]

# Создать список из элементов списка a, умноженных на свою последнюю цифру
a1 = [Здесь должен стоять Ваш код]
print(a1) # [-136, 0, 336, 171, 21, 536, -624, 0, 11, -1725, 61, 336]

# Создать список из элементов списка a, умноженных на последнюю цифру 1 элемента
a2 = []
print(a2) # [-136, -360, 224, 76, 84, 536, -312, 0, 44, -1380, 244, 336]

# Создать список из элементов списка a, увеличенных на значение последнего элемента
a3 = []
print(a3) # [50, ...] Проверьте свой результат самостоятельно
```

№ List.7 (➡). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Примечание. После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = [30, 7, 32, 2050, 19, 63, 98, 13]

# Создать список, в котором все элементы списка a записаны в 2-ной с.с.
a1 = [Здесь должен стоять Ваш код]
print(a1) # ['11110', '111', '100000', '100000000010', '10011', ...]

# Создать список, в котором все элементы списка a записаны в 8-ной с.с.
a2 = []
print(a2) # Проверьте свой результат самостоятельно

# Создать список из элементов списка a заменой каждого элемента
# на количество цифр в его 16-ной записи
a3 = []
print(a3) # Проверьте свой результат самостоятельно
```

№ List.8 (➡). Дан список строк, представляющих собой двоичные числа. Для данного списка запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Допишите генераторы списков

```
a = ['10101', '10100', '100110', '1100110', '10011', '11001101101']

# Создать список, в котором все элементы списка а переведены из 2-ной с.с. в 10-ную
a1 = [Здесь должен стоять Ваш код]
print(a1) # Проверьте свой результат самостоятельно

# Создать список, в котором все элементы списка а переведены из 2-ной с.с. в 8-ную
a2 = []
print(a2) # Проверьте свой результат самостоятельно

# Создать список, в котором все элементы списка а переведены из 2-ной с.с. в 16-ную
a3 = []
print(a3) # Проверьте свой результат самостоятельно

# Создать список, в котором все элементы списка а заменены кол-вом пар 11 в строке
a4 = []
print(a4) # Проверьте свой результат самостоятельно
```

№ List.9 (➡). Для заданного списка строк *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Допишите генераторы списков

```
a = ['100', '194', '26', '111', '1013', '10101', '98', '37', '8']

# Записать в новый список элементы списка а, которые
#                               НЕ могут представлять собой 8-ую запись
a1 = [Здесь должен стоять Ваш код]
print(a1) # Проверьте свой результат самостоятельно

# Записать в новый список все элементы списка а с длиной не более двух символов
a2 = []
print(a2) # Проверьте свой результат самостоятельно

# Записать в новый список элементы списка а, которые представляют собой 2-ую запись
a3 = []
print(a3) # Проверьте свой результат самостоятельно
```

№ List.10 (➡). Для заданного списка *a* запишите генераторы списков для отбора значений, удовлетворяющих указанным в заданиях условиям.

Примечание. После операторов печати в коде стоят ожидаемые ответы для сравнения.

Допишите генераторы списков

```
a = [-34, -90, 56, 19, 21, 134, -78, 0, 11, -345, 61, 84]

# Создать список с неположительными элементами списка а
a1 = [Здесь должен стоять Ваш код]
print(a1) # Проверьте свой результат самостоятельно

# Создать список с элементами списка а, кратными трем
a2 = []
print(a2) # Проверьте свой результат самостоятельно

# Создать список с положительными чётными элементами списка а
a3 = []
print(a3) # Проверьте свой результат самостоятельно

# Создать список с положительными элементами, двоичная запись которых
# содержит нечётное количество значащих нулей
a4 = []
print(a4) # [56, 134, 11, 61]

# Создать список с положительными элементами списка а, запись которых в 8-ной с.с.
# содержит хотя бы одну двойку
a5 = []
print(a5) # [19, 21, 134, 84]
```