

python

Занятие 4. Структуры данных. Начало. Списки.

Переменные

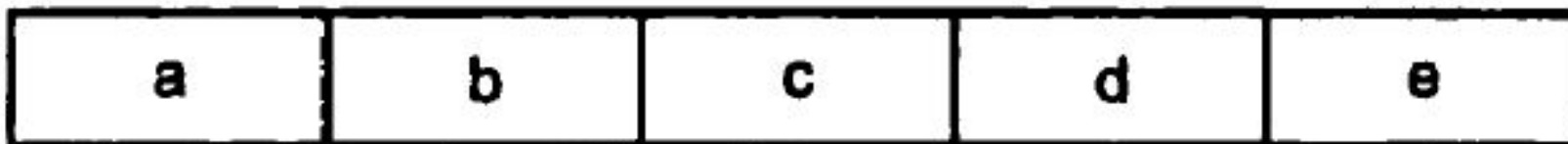
- До этого момента мы работали с переменными. Их удобно использовать для хранения одиночных значений.
- Если нужно обработать большое количество данных, то для каждого элемента придется создавать переменную
- Если объем данных заранее неизвестен, сколько создавать переменных?



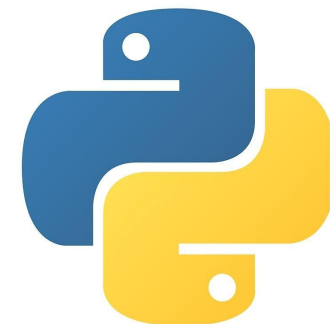
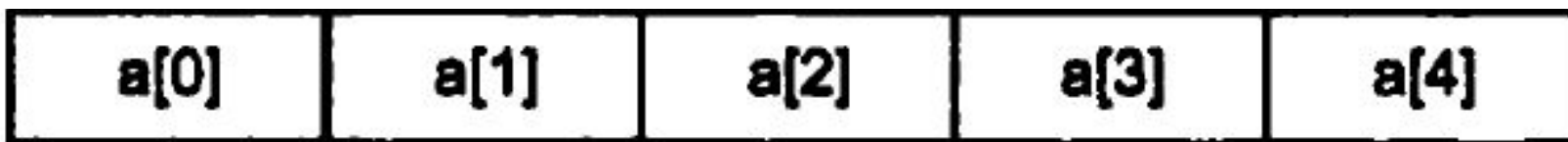
Список

Список – это набор разнотипных (в Python) элементов данных, расположенных в оперативной памяти друг за другом

Пять простых переменных



Список a из пяти элементов



Список

Поскольку элементы списка хранятся в памяти друг за другом, их можно пронумеровать

Номер элемента списка называется его индексом

В Python нумерация элементов списка начинается с нуля

Тогда, если в списке всего n элементов, последний его элемент будет иметь индекс $n-1$



Список



Создание списка

Создать список можно двумя способами:

1) `Название_списка = []`

Пример: `a = []`

2) `Название_списка = list()`

Пример: `b = list()`



Создание списка

Можно создать список с заранее заданными данными:

```
Название_списка=[данные, данные, ...]
```

Пример:

```
a=[5, 6, 7, 8]
```

```
b=["Понедельник", "Вторник", "Среда"]
```

```
c=[True, False, False]
```



Создание списка

В Python списки могут содержать разнотипные данные:

```
a = [5, True, "Привет"]
```

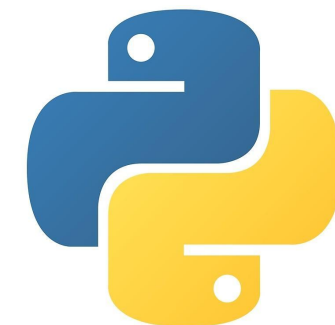


Добавление элементов в список

Добавление элемента в список осуществляется с помощью метода `append()`

```
a = []  
a.append("Hello")  
a.append(5)  
a.append("Intellect")  
print(a)
```

```
=====  
['Hello', 5, 'Intellect']
```



Обращение к элементам списка

- **Обращаться (для чтения или изменения) к элементам списка можно по их порядковым номерам.**
- Для этого нужно указать номер элемента в квадратных скобках после названия списка
- Нумерация начинается с 0!

Номер	0	1	2	3
Значение	“Привет”	5	True	0,8

```
a = ["Привет", 5, True, 0.8]
print(a[2])
print(a[0])
print(a[3])
print(a[1])
```

```
=====
True
Привет
0.8
5
.
```



Обращение к элементам списка

- Список подразумевает хранение в нем большого кол-ва данных. Часто бывает неудобно обрабатывать их одиночными методами.
- Т.к. у списков есть индексы (порядковые номера), значит к ним можно удобно обращаться с помощью циклов

```
a=["Понедельник", "Вторник", "Среда"]  
for i in range(3):  
    print(a[i])
```

```
=====  
Понедельник  
Вторник  
Среда
```



Обращение к элементам списка

- Например, это можно использовать для изменения элементов

```
a=[1, 2, 3]
for i in range(3):
    print(a[i]+10)
```

```
=====
```

```
11
12
13
```



Обращение к элементам списка

- Для ввода большого количества элементов так же используем циклы:

```
n=int(input("Введите кол-во элементов: "))
a=[]
for i in range(n):
    a.append(int(input()))
print(a)
```



Обращение к элементам списка

Особенность цикла `for` в языке Python делает процесс обработки списков удобным.

```
a=[10,15,20]
for x in a:
    print(x)
```

Переменная **x** примет в себя по очереди все элементы списка **a** и выведет их на экран.

```
=====
10
15
20
```



Списки – ссылочный тип

Внимание! Списки – ссылочный тип.

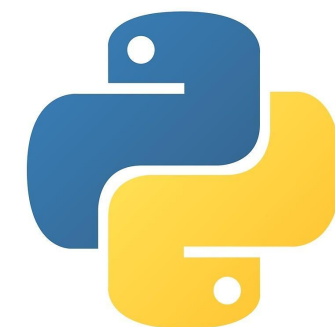
Это значит, что если вы выполните простое присвоение списков друг другу, то переменной `b` будет присвоена ссылка на тот же элемент данных в памяти, на который ссылается `a`, а не копия списка `a`. Т.е. если вы будете изменять список `a`, то и `b` тоже будет меняться.

Поэтому если вы хотите сделать копию списка, это можно сделать так:

```
b = list(a)
```



Средства обработки списков



Добавление элементов в список справа:

`имя_списка.append(данные)`

Пример:

```
a=[1, 2, 3, 4, 5]
```

```
a.append(6)
```

Список станет таким: `[1, 2, 3, 4, 5, 6]`



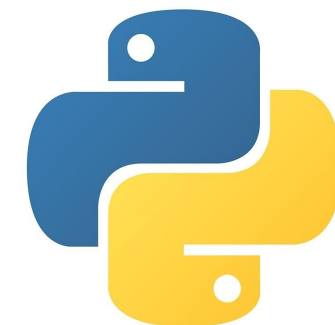
Добавление элементов в список по индексу:

```
имя_списка.insert(index, данные)
```

Пример:

```
a=[1, 2, 3, 4, 5]  
a.insert(2, 15)
```

Список станет таким: [1, 2, 15, 3, 4, 5]



Поиск и удаление элемента (первого):

`имя_списка.remove(данные)`

Пример:

```
a=[1, 2, 3, 4, 5]
```

```
a.remove(2)
```

Список станет таким: `[1, 3, 4, 5]`



Удаление всех элементов

```
имя_списка.clear()
```

Пример:

```
a=[1, 2, 3, 4, 5]
```

```
a.clear()
```

Список станет таким: []



Номер (индекс) найденного элемента

```
имя_списка.index(3)
```

Пример:

```
a=[1, 2, 3, 4, 5]  
print(index(3))
```

Вывод: 2



Сколько раз элемент есть в списке

`имя_списка.count(данные)`

Пример:

```
a=[3, 2, 3, 4, 5]  
print(a.count(3))
```

Вывод: 2



Сортировка

```
имя_списка.sort()
```

Пример:

```
a=[3,2,1,5,4]
```

```
a.sort()
```

Список станет таким: [1, 2, 3, 4, 5]



Обратный порядок

```
имя_списка.reverse()
```

Пример:

```
a=[1, 2, 3, 4, 5]
```

```
a.reverse()
```

Список станет таким: [5, 4, 3, 2, 1]



Длина списка

```
x=len (имя_списка)
```

Пример:

```
a=[1, 2, 3, 4, 5]
```

```
x=len(a)
```

```
print(x)
```

Вывод: 5



Максимальный элемент

```
x=max (имя_списка)
```

Пример:

```
a=[10, 20, 30, 40, 50]
```

```
x=max(a)
```

```
print(x)
```

Вывод: 50



Минимальный элемент

```
x=min (имя_списка)
```

Пример:

```
a=[10, 20, 30, 40, 50]
```

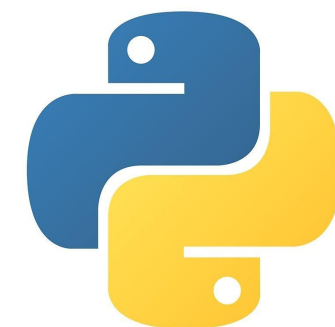
```
x=min(a)
```

```
print(x)
```

Вывод: 10



Задачки!



Объявить список из 7 элементов – названий дней недели. На ввод числа от 1 до 7 программа должна вывести соответствующий день недели.



Найти сумму максимального и минимального
элементов целочисленного списка.
Не использовать методы `max` и `min`

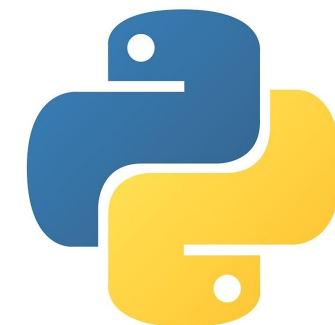


Вывести элементы списка, которые меньше среднего арифметического

1. Ввод списка

2. Найти среднее арифметическое списка

3. Вывести эл., $<$ ср.арифметического



Вывести в порядке возрастания цифры, из которых состоит число. Использовать списки



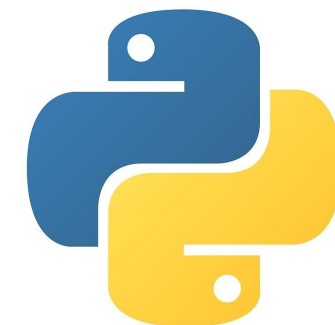
Посчитать кол-во отрицательных и
положительных элементов списка



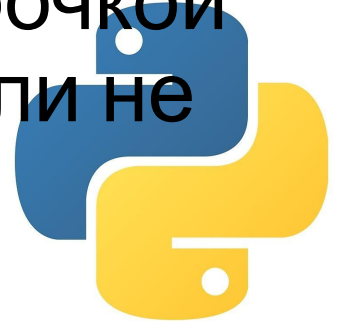
Вычислить номер дня с начала года



Определить самое большое количество
подряд идущих элементов списка



Вася не любит английский язык, но каждый раз старается получить хотя бы четверку за четверть, чтобы оставаться ударником. В текущей четверти Вася заметил следующую закономерность: по нечетным дням месяца он получал тройки, а по четным – четверки. Так же он помнит, в какие дни он получал эти оценки. Поэтому он выписал на бумажке все эти дни для того, чтобы оценить, сколько у него троек и сколько четверок. Помогите Васе это сделать, расположив четные и нечетные числа в разных строчках. Вася может рассчитывать на оценку 4, если четверок не меньше, чем троек. Третьей строчкой напишите «Да», если Вася получит 4 и «Нет», если не получит



Для данной буквы английского алфавита.
Нужно вывести справа стоящую букву на стандартной клавиатуре. При этом клавиатура замкнута, т.е. справа от буквы «p» стоит буква «a», от буквы «l» стоит буква «z», а от буквы «m» — буква «q».



Сортировка пузырьком



Школьный врач решил начать вести базу данных роста и веса школьников. Помогите ему, написать программу, которая запрашивает количество учеников, их ФИО, рост и вес, а после ввода выводит данные, отсортированные:

- 1) по ФИО
- 2) возрастанию роста
- 3) возрастанию веса



На сегодня достаточно



Жду вас на следующих занятиях!
На забывайте тетрадь и ручку!

