



Фоксфорд

Кружки

Кружок по программированию на Python

Занятие №9



Кружок по программированию на Python

Списки в Python

- Списки в Python
- Срезы списков в Python



Введение

Что такое массив?

- Массив – это группа переменных одного типа, расположенных в памяти рядом (в соседних ячейках) и имеющих общее имя. Каждая ячейка в массиве имеет уникальный номер (индекс).
- Списки в Python обладают всеми вышеобозначенных свойствами, за одним исключением.



Введение

Что такое массив?

- Список -- изменяемая последовательность элементов произвольных типов.
- Примеры:
- `a = [1, 2, 3, 'fox']`
- `a = [[1, 2, 3, 4], [5, 6, 7], ['fox', 8, 9]]`



Введение

Объявление списков

- Есть множество способов объявить список:

```
a = []
```

```
b = list()
```

```
s = [0]*5
```

Первые два варианта объявляют пустой список, третий же – список из пяти нулей.

Индексация

- $a = [1, 2, 3, \text{'fox'}, 10]$

0	1	2	3	4
1	2	3	'fox'	10
-5	-4	-3	-2	-1

$a[0] = 1$

$a[4] = 10$

$a[-2] = \text{'fox'}$

Индексация

Срезы

- `a[1:3]` - ?

0	1	2	3	4
1	2	3	'fox'	10
-1	-2	-3	-4	-5

- `a[1:3] = [2, 3]`
- Выводятся символы с 1 по 3 неключительно!

`a[:] = [1, 2, 3, 'fox', 10]`

`a[-4:-1] = [2, 3, 'fox']`

`a[-1:-4] = []`

Индексация

Срезы

- `a[1::2]` - ?

0	1	2	3	4
1	2	3	'fox'	10
-5	-4	-3	-2	-1

- `a[1::2] = [2, 'fox']`
- Выводятся символы с через один, начиная с первого.

`a[::3] = [1, 'fox']`

`a[1:3:2] = [2]`

`a[1:3:-2] = []`



Задача №1

- Дан список чисел. Если в нем есть два соседних элемента одного знака, выведите эти числа. Если соседних элементов одного знака нет - не выводите ничего. Если таких пар соседей несколько - выведите первую пару.



Задача №1

```
a = list(map(int, input().split()))
for i in range(1, len(a)):
    if a[i - 1] * a[i] > 0:
        print(a[i - 1], a[i])
        break
```



Генераторы

- `A = [i**2 for i in range(10) if i%3==0]`
- `>>> [0, 9, 36, 81]`
- `A = [функция цикл итератор условие]`



Задача №2

- Циклически сдвиньте элементы списка вправо ($A[0]$ переходит на место $A[1]$, $A[1]$ на место $A[2]$, ..., последний элемент переходит на место $A[0]$).
- Используйте минимально возможное количество операций присваивания.



Задача №2

```
a = list(map(int, input().split()))
x = a[-1]
for i in range(len(a) - 1, 0, -1):
    a[i] = a[i - 1]
a[0] = x
print(*a)
```



Задача №2

```
a = list(map(int, input().split()))
x = a[-1]
for i in range(len(a) - 1, 0, -1):
    a[i] = a[i - 1]
a[0] = x
print(*a)
```

```
a = [1, 2, 3, 'fox', 10]
a[-1], a[:len(a)-1] = a[0], a[1:len(a)]
print(a)
```



Задача №3

Петя перешёл в другую школу. На уроке физкультуры ему понадобилось определить своё место в строю. Помогите ему это сделать.

Программа получает на вход невозрастающую последовательность натуральных чисел, означающих рост каждого человека в строю. После этого вводится число X – рост Пети. Все числа во входных данных натуральные и не превышают 200. Если в строю есть люди с одинаковым ростом, таким же, как у Пети, то он должен встать после них.



Задача №3

```
s=list(map(int,input().split()))
r=int(input())
count=0
for i in range (len(s)):
    if r<=s[i]:
        count+=1
print(count+1)
```




Задача №3

```
height = [int(x) for x in input().split()]
p_height = int(input())
left = -1
right = len(height)
while right - left > 1:
    middle = (left + right) // 2
    if height[middle] >= p_height:
        left = middle
    else:
        right = middle
print(left + 2)
```



Задача №4

Дан список целых чисел. Требуется “сжать” его, переместив все ненулевые элементы в левую часть списка, не меняя их порядок, а все нули - в правую часть. Порядок ненулевых элементов изменять нельзя, дополнительный список использовать нельзя, задачу нужно выполнить за один проход по списку. Распечатайте полученный список.



Задача №4

```
a = [int(i) for i in input().split()]
```

```
b = 0
```

```
for i in range(len(a)):
```

```
    if a[i] != 0:
```

```
        print(a[i], end=" ")
```

```
    else:
```

```
        b = b + 1
```

```
for i in range(b):
```

```
    print(0, end=" ")
```



Списки в Python

Функции

- `len(list)`: возвращает длину списка
- `min(list)`: возвращает наименьший элемент списка
- `max(list)`: возвращает наибольший элемент списка

1	2	3	25	10
----------	----------	----------	-----------	-----------

```
>>> len(a)
>>> 5
>>> min(a)
>>> 1
>>> max(a)
>>> 25
```



Списки в Python

Методы

Добавление элементов:

- `append(x)`: добавляет элемент `x` в конец списка;
- `insert(index, x)`: добавляет элемент `x` в список по индексу `index`;

```
a = [4, 5]
```

```
a.append(2)
```

```
>>>a
```

```
>>>[4, 5, 6]
```



Списки в Python

Методы

Удаление элементов:

- `remove(x)`: удаляет элемент `x`. Удаляется только первый найденный элемент `x`. Если элемент не найден, генерирует исключение `ValueError`;
- `clear()`: удаление всех элементов из списка;
- `pop([index])`: удаляет и возвращает элемент по индексу `index`. Если индекс не передан, то просто удаляет последний.



Списки в Python

Методы

- `index(x)`: возвращает индекс элемента `x`. Если элемент не найден, генерирует исключение `ValueError`;
- `count(x)`: возвращает количество вхождений элемента `x` в список;
- `sort([key])`: сортирует элементы. По умолчанию сортирует по возрастанию. Но с помощью параметра `key` мы можем передать функцию сортировки.



Задача №5

В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка.

Вводится список целых чисел. Все числа списка находятся на одной строке.



Задача №5

```
a = list(map(int, input().split()))
```

```
minind = a.index(min(a))
```

```
maxind = a.index(max(a))
```

```
a[minind], a[maxind] = a[maxind], a[minind]
```

```
print(*a)
```



Фоксфорд
Кружки



Спасибо за внимание!