



Курс по основам
программирования на Python

Списки

Большинство программ работает не с отдельными переменными, а с набором переменных.

```
Primes = [2, 3, 5, 7, 11, 13]  
Rainbow = ['Red', 'Orange', 'Yellow', 'Green', 'Blue', 'Indigo', 'Violet']
```

Для хранения таких данных можно использовать структуру данных, называемую в Питоне **список** (в большинстве же языков программирования используется другой термин “массив”). Список представляет собой **последовательность элементов, пронумерованных от 0, как символы в строке.**

Списки

- 0 Длину списка, то есть количество элементов в нём, можно узнать при помощи функции len.

```
len(Primes) == 6
```

Списки

0 Можно создать пустой список (не содержащий элементов длины 0).

a=[] – пустой список

В конец списка можно добавлять элементы при помощи метода **append**.

```
a = [] # заводим пустой список
n = int(input()) # считываем количество элемент в списке
for i in range(n):
    new_element = int(input()) # считываем очередной элемент
    a.append(new_element) # добавляем его в список
    # последние две строки можно было заменить одной:
    # a.append(int(input()))
print(a)
```


Списки

- 0 Для списков целиком определены операции сложения списков и умножение списка на число.

```
print([7, 8] + [9])  
print([0, 1] * 3)
```

```
[7, 8, 9]  
[0, 1, 0, 1, 0, 1]
```

Split

- 0 Элементы списка могут вводиться по одному в строке, в этом случае строку целиком можно считать функцией **input()**. После этого можно использовать метод строки **split()**, возвращающий список строк, которые получаются, если исходную строку разрезать на части по пробелам.

```
# на вход подаётся строка
# 1 2 3
s = input() # s == '1 2 3'
a = s.split() # a == ['1', '2', '3']
```


Split

- 0 У этого метода есть необязательный параметр, который определяет, какая строка будет использоваться в качестве разделителя между элементами списка.

```
a = '192.168.0.1'.split('.')
```

Join

- 0 Join – метод, позволяющий вывести список. У этого метода один параметр: список. Возвращается строка. Полученная соединением элементов переданного списка в одну строку.

```
a = ['red', 'green', 'blue']  
print(' '.join(a))  
# вернёт red green blue  
print(''.join(a))  
# вернёт redgreenblue
```


Задача 1

Выведите все элементы списка с четными индексами (то есть `A[0]` , `A[2]` , `A[4]` , ...).

Решение

```
s = input() # s == '1 2 3'  
a = s.split() # a == ['1', '2', '3']  
for i in range(len(a)):  
    if i % 2 == 0:  
        print(a[i])
```


Задача 2

Дан список чисел. Выведите все элементы списка, которые больше предыдущего элемента.

Решение

```
a = input().split()
for i in range(1, len(a)):
    if int(a[i-1]) < int(a[i]):
        print(a[i])
```


Задача 3

Дан список чисел. Выведите значение наибольшего элемента в списке, а затем индекс этого элемента в списке. Если наибольших элементов несколько, выведите индекс первого из них.

Решение

```
A = input().split()
max, index = A[0], 0
for i in range(1, len(A)):
    if A[i] > max:
        max, index = A[i], i
print(max, index)
```


Задача 4

Переставьте соседние элементы списка ($A[0]$ с $A[1]$, $A[2]$ с $A[3]$ и т. д.).
Если элементов нечетное число, то последний элемент остается на своем месте.

Решение

```
A = input().split()
for i in range(1, len(A), 2):
    A[i], A[i - 1] = A[i - 1], A[i]
for i in range(len(A)):
    print(A[i], end = ' ')
```