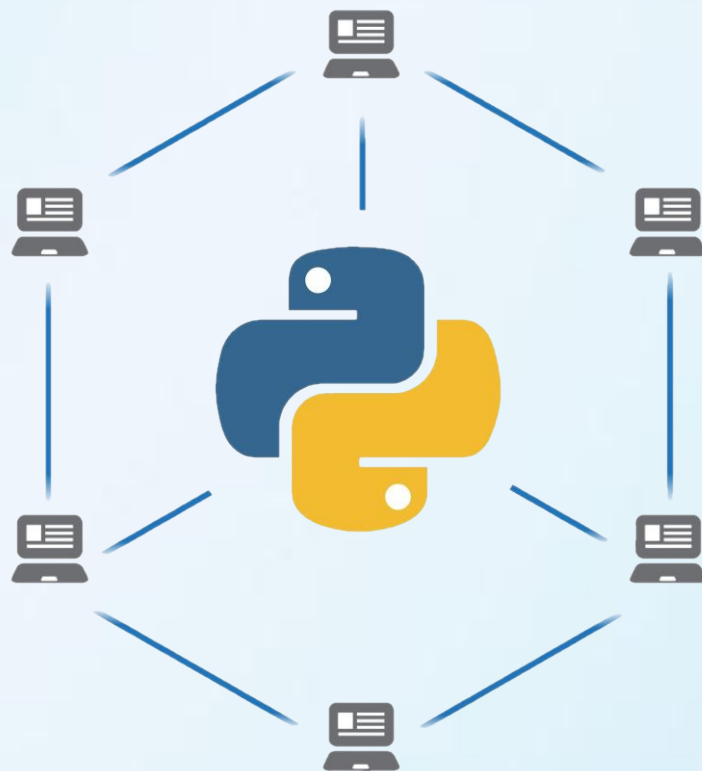


Списки



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

Раньше мы сталкивались с задачей обработки элементов последовательности, например, вычисляя наибольший элемент последовательности. Но при этом мы не сохраняли всю последовательность в памяти компьютера. Однако, во многих задачах нужно именно сохранять всю последовательность, например, если бы нам требовалось вывести все элементы последовательности в возрастающем порядке («отсортировать последовательность»).

Для хранения таких данных можно использовать структуру данных, называемую в Python список (в большинстве же языков программирования используется другой термин «массив»).

Список - последовательность элементов,
пронумерованных от 0.

```
a = ['Привет', 'мир', 2021, 'я', 'изучаю', 'Python']
```

В списке a - 6 элементов:

```
a[0] == 'Привет'
```

```
a[1] == 'мир'
```

```
a[2] == 2021
```

```
a[3] == 'я'
```

```
a[4] == 'изучаю'
```

```
a[5] == 'Python'
```

```
a = ['Привет', 'мир', 2021, 'я', 'изучаю', 'Python']
```

Длину списка, то есть количество элементов в нем, можно узнать при помощи функции len

```
len(a) == 6
```

Мы можем заменить значение элемента списка:

```
a = ['Привет', 'мир', 2021, 'я', 'изучаю', 'Python']
```

```
a[3] = 'мы'
```

```
a[4] = 'знаем'
```

```
print(a)
```

В выводе мы получим:

```
['Привет', 'мир', 2021, 'мы', 'знаем', 'Python']
```

Рассмотрим несколько способов создания и считывания списков. Прежде всего, можно создать пустой список (не содержащий элементов, длины 0), а в конец списка можно добавлять элементы при помощи метода `append`.

```
a = []
```

```
n = int(input())
```

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

```
    element = int(input())
```

```
    a.append(element)
```

```
print(a)
```



```
a = []  
for i in range(int(input())):  
    a.append(int(input()))  
print(a)
```

Для списков целиком определены следующие операции:

- конкатенация списков (сложение списков, т.е. приписывание к одному списку другого)
- повторение списков (умножение списка на число).

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

```
b = [4, 5]
```

```
c = a + b
```

```
print(c)
```

В выводе получим:

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

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

В выводе получим:
[1, 2, 1, 2, 1, 2]

Также в списках могут храниться другие списки
(вложенные списки):

```
a = [ ['PS1', 'PS2', 'PS3', 'PS4', 'PS5'], ['XBOX', 'XBOX 360', 'XBOX ONE', 'XBOX Series X'] ]
```

Также мы можем добавить каждый элемент строки в список:

```
a = list('Python')  
print(a)
```

В выводе получим:

```
['P', 'y', 't', 'h', 'o', 'n']
```

| Операция | Что делает |
|-----------------------|--|
| a.append(x) | Добавляет элемент в конец списка |
| a.insert(i, x) | Вставляет на элемент под номером i значение x |
| a.remove(x) | Удаляет первый элемент в списке, имеющий значение x. |
| a.index(x) | Индекс первого вхождения элемента x в список |
| a.count(x) | Выводит количество элементов со значением x |
| a.sort() | Сортирует список на основе функции |
| a.reverse() | Меняет список в обратном направлении |
| a.clear() | Очищает список |
| if x in a | Проверить, содержится ли элемент в списке |
| if x not in a | Проверяет, не содержится ли элемент в списке |
| min(a) | Выводит наименьший элемент списка |
| max(a) | Выводит наибольший элемент списка |
| a.pop(index) | Удаляет из списка элемент под номером index |

Спасибо за внимание!

