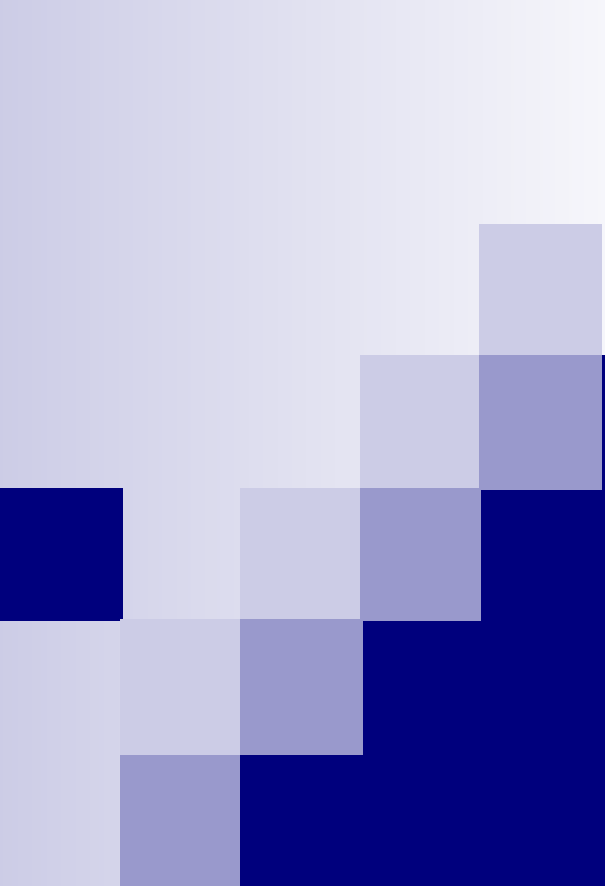




**Красивое лучше уродливого**



**Явное лучше неявного**  
Гвидо ван Россум



# Язык программирования Python (лекция 8)

**Валеева Н.Х. – кан. пед. наук,  
преподаватель ц/к ПЭВМ**



**Язык программирования Python  
был создан примерно в 1991 году  
голландцем  
Гвидо ван Россумом.**



**Официальный сайт <http://python.org>**

# Списки



**Списки в Python - упорядоченные  
изменяемые совокупности объектов  
произвольных типов, заключенные в  
квадратные скобки**

**Например:**

```
[23, 656, -20, 67, -45]           # список целых чисел  
[4.15, 5.93, 6.45, 9.3, 10.0, 11.6] # список из дробных чисел  
["Katy", "Sergei", "Oleg", "Dasha"] # список из строк  
["Москва", "Титова", 12, 148]      # смешанный список  
[[0, 0, 0], [0, 0, 1], [0, 1, 0]]  # список, состоящий из списков
```

# Действия над списками



## 1) Соединение (+)

```
>>> [45, -12, 'april'] + [21, 48.5, 33]  
[45, -12, 'april', 21, 48.5, 33]
```

## 2) Повторение (\*)

```
>>> [[0,0],[0,1],[1,1]] * 2  
[[0, 0], [0, 1], [1, 1], [0, 0], [0, 1], [1, 1]]
```



# Действия над списками

## 3) Срезы

```
>>> q = ['a', 'b', 'c', 'd', 'e', 'f']
>>> len(q) # 6 –длина списка
>>> q[0] # 'a'
>>> q[4] # 'e'
>>> q[0:3] # ['a', 'b', 'c']
>>> q[3:] # ['d', 'e', 'f']
```

*В отличие от строк*

```
>>> q[-1] = '%' # заменяем последний элемент
>>> q
['a', 'b', 'c', 'd', 'e', '%']
```



# Способы создания списков

## 1. С помощью функции `list`:

```
>>> list('список')
```

```
['с', 'п', 'и', 'с', 'о', 'к']
```

```
>>> list('колледж')
```

```
['к', 'о', 'л', 'л', 'е', 'д', 'ж']
```



# Способы создания списков

## 2. Список можно создать и при помощи перечисления элементов:

```
>>> s = [] # Пустой список
```

```
>>> m = ['s', 'p', ['isok'], 2]
```

```
>>> b=[23, 9, 'w', 'alpha', -100, 'ящик']
```

```
>>> s
```

```
[]
```

```
>>> m
```

```
['s', 'p', ['isok'], 2]
```

```
>>> b
```

```
[23, 9, 'w', 'alpha', -100, 'ящик']
```



# Способы создания списков



Длинные списки можно переносить в скрипте на другую строку:

```
days = ["Sunday", "Monday", "Tuesday",  
        "Wednesday", "Thursday", "Friday",  
        "Saturday"]
```



# Способы создания списков

3. Список можно создать при помощи генератора списков (списочное включение).

## Генераторами списков в Python

называются однострочные конструкции, которые позволяют создавать новые списки.

Простейший генератор имеет вид:

```
>>> new_list = [x for x in range(10)]
```

```
>>> new_list
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```



# Функция диапазон

Функция **range()** (диапазон) предусмотрена для получения диапазона целых чисел.

Синтаксис: **range(a, b, h)**, где

a – начальное значение;

b – конечное значение;

h – шаг.

Примеры:

`range(10, 50, 5)` – 10, 15, 20, 25, 30, 35, 40, 45

`range(0, 10, 1)` = `range(10)` –  
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

# Функция диапазон



Для создания диапазона чисел необходимо использовать цикл for:

```
>>> for i in range(10): print (i, end=' ')
```

```
0 1 2 3 4 5 6 7 8 9
```

```
>>> for i in range(2, 20, 2): print (i, end=' ')
```

```
2 4 6 8 10 12 14 16 18
```

```
>>> for i in range(20, 2, -2): print (i, end=' ')
```

```
20 18 16 14 12 10 8 6 4
```

# Способы создания списков



```
>>> a = [ k for k in range(1, 15) ]
```

что делаем с  
элементом

что берем

откуда берем

Примеры:

```
1) >>> a = [ i**2 for i in range(1,15) ]
```

```
>>> a
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196]
```

```
2) >>> m = [c * 3 for c in 'money']
```

```
>>> m
```

```
['mmm', 'ooo', 'nnn', 'eee', 'yyy']
```

```
3) >>> t = [c * 3 for c in 'list' if c != 'i']
```

```
>>> t
```

```
['lll', 'sss', 'ttt']
```

# Способы создания списков



Вместо диапазонов генератор списков позволяет указывать существующий список:

```
>>> a = [2, -2, 4, -4, 7, 5]
```

```
>>> b = [i**2 for i in a]
```

```
>>> b
```

```
[4, 4, 16, 16, 49, 25]
```

# Способы создания списков



4. Список можно создать с помощью генератора случайных чисел.

```
>>> from random import randint
>>> A = [ randint(1, 9) for i in range(5) ]
>>> A
[2, 1, 1, 7, 8]
```



# Способы создания списков

## 5. Список можно ввести с клавиатуры (ручной ввод)

```
a = [] # объявляем пустой список
n = int (input('Сколько элементов в списке? ')) # считываем количество
print('Введите ',n,' элементов через ENTER')
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)
```



# Способы создания списков



Результат работы скрипта:

```
Сколько элементов в списке? 5
Введите 5 целых чисел через ENTER
10
-6
120
1
-56
[10, -6, 120, 1, -56]
>>> |
```

# Способы создания списков



При формировании списка из строк достаточно удалить `int` в строке ввода:

```
File Edit Format Run Options Window Help
a = [] # объявляем пустой список
n = int (input('Сколько элементов в списке? ')) # считываем количество
print('Введите ',n, ' элементов через ENTER')
for i in range(n):
    new_element = 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)
```

# Способы создания списков



## Пример «ручного ввода»:

```
Сколько элементов в списке? 6
Введите 6 элементов через ENTER
100
аптека
pi
end
-56
@
['100', 'аптека', 'pi', 'end', '-56', '@']
>>> |
```

# Способы создания списков



Можно записать решение этой задачи через генератор списка в одну строку:

```
>>> A = [ int (input() ) for i in range(int (input())) ]
3
9
103
-7
>>> A
[9, 103, -7]
>>>
```

# Методы списков



- | Метод                                   | Что делает   |
|---|--|
| ■ <b>list.append(x)</b>                 | Добавляет элемент в конец списка   |
| ■ <b>list.extend(L)</b>                 | Расширяет список list, добавляя в конец все элементы списка L                                  |
| ■ <b>list.insert(i, x)</b>              | Вставляет на i-ый элемент значение x   |
| ■ <b>list.remove(x)</b>                 | Удаляет первый элемент в списке со значением x. ValueError, если такого элемента не существует |
| ■ <b>list.pop([i])</b>                  | Удаляет i-ый элемент и возвращает его. Если индекс не указан, удаляется последний элемент      |
| ■ <b>list.index(x, [start [, end]])</b> | Возвращает положение первого элемента со значением x (при этом поиск ведется от start до end)  |
| ■ <b>list.count(x)</b>                  | Возвращает количество элементов со значением x   |
| ■ <b>list.sort([key=функция])</b>       | Сортирует список на основе функции   |
| ■ <b>list.reverse()</b>                 | Разворачивает список   |
| ■ <b>list.copy()</b>                    | Поверхностная копия списка   |
| ■ <b>list.clear()</b>                   | Очищает список   |

# Примеры



```
>>> a = [66.25, 333, 333, 1, 1234.5]
>>> print(a.count(333), a.count(66.25), a.count('x'))
2 1 0
>>> a.insert(2, -1)
>>> a.append(333)
>>> a
[66.25, 333, -1, 333, 1, 1234.5, 333]
>>> a.index(333)
1
>>> a.remove(333)
>>> a
[66.25, -1, 333, 1, 1234.5, 333]
>>> a.reverse()
>>> a
[333, 1234.5, 1, 333, -1, 66.25]
>>> a.sort()
>>> a
[-1, 1, 66.25, 333, 333, 1234.5]
```

# Распаковка списка



```
person_data = ["Олег", "Петров", 23, "матрос"]
```

Тогда чтобы извлечь из этого списка и сохранить по-отдельности все элементы следует написать такой код:

```
name, surname, age, profession=person_data
```

После этого мы сможем использовать отдельно каждую созданную переменную.



# Задача



```
#Дан список числовых значений, насчитывающий n элементов.  
#Поменяйте местами первую и вторую половины списка.  
from random import randint  
n = int(input('Введите количество элементов в списке '))  
A = [ randint(1, 50) for i in range(n) ]  
print('Исходный список')  
print(A)  
m = n // 2  
if n % 2 == 0:  
    A = A[m:] + A[:m]  
else:  
    A= A[m+1:] + [A[m]] + A[:m]  
print('Преобразованный список')  
print(A)
```





# Результаты теста скрипта

Введите количество элементов в списке 10

Исходный список

[20, 23, 12, 37, 7, 39, 16, 31, 17, 47]

Преобразованный список

[39, 16, 31, 17, 47, 20, 23, 12, 37, 7]

>>>

```
RESTART: C:/Users/USER/Documents/Наиля/ОП/П  
на Питоне/list.py
```

Введите количество элементов в списке 11

Исходный список

[6, 47, 43, 22, 20, 44, 12, 45, 6, 12, 25]

Преобразованный список

[12, 45, 6, 12, 25, 44, 6, 47, 43, 22, 20]

# Выполнить задание:

1. Создайте два списка и свяжите их с переменными. При этом первый список должен содержать не менее 10 целых чисел, а второй – не менее 6 различных слов.
2. Извлеките из первого списка второй элемент.
3. Измените во втором списке последний объект. Выведите список на экран.
4. Соедините оба списка в один, присвоив результат новой переменной. Выведите получившийся список на экран.
5. "Снимите" срез из соединенного списка так, чтобы туда попали некоторые части обоих первых списков, а количество элементов было не менее 10. Срез свяжите с очередной новой переменной. Выведите значение этой переменной.
6. Добавьте в список-срез два новых элемента-символа и снова выведите его.
7. Создайте новый список, который будет состоять из четных чисел первого списка. Выведите этот список и его длину.
8. Найдите индекс самого длинного слова во втором списке.



**Сейчас лучше, чем никогда!**  
**В списках наша сила!**  
**Успехов!**