# Урок 11.01. Списки

Мы уже довольно давно работаем с такими типами данных как вещественные числа, целые числа и строки. И в программирование очень часто приходится работать с очень большим количеством тех же чисел или строк. И сегодня мы узнаем, что придумали программисты для облегчения работы с такими данными.

Для этого возьмем наш старый и немного измененный пример где мы только начинали изучать цикл for:

for number in 1, 5, 7, -2, 6:

 print(number, ' \*\* 2 =', number \*\* 2)

Эта программа идет по числам и возводит каждое из них в квадрат и выводит это на экран. Расширим нашу программа: например, нам необходимо добавить число, которое введет пользователь с клавиатуры. Для нас это не проблема:

user\_number = int(input('Введите число: '))

for number in 1, 5, 7, -2, 6, user\_number:

 print(number, ' \*\* 2 =', number \*\* 2)

Запускаем и всё работает. Но опыт уже подсказывает нам, что одна переменная это ещё не страшно, но вот 100 уже очень много. И все их нужно объявлять, писать через запятую в цикле… В общем, ужас. Но программисты были бы не программистами, если это нельзя было бы решить. И раз чисел много и все они должны храниться и обрабатываться в одном месте, то логично и называть все переменные одним именем. Давайте так и сделаем: создадим новую переменную b и в неё поместим все числа для цикла:

user\_number = int(input('Введите число: '))

b = 1, 5, 7, -2, 6, user\_number

for number in b:

 print(number, ' \*\* 2 =', number \*\* 2)

Запустим программу, и у нас всё получилось. Но если думать о будущем, то мы пока не решили нашу проблему, нужно сделать так, чтобы в переменную b числа добавлялись без каких-либо дополнительных переменных вроде нашей переменной user\_number. Для этого закомментируем переменную user\_number и узнаем новую команду:

# user\_number = int(input('Введите число: '))

b = 1, 5, 7, -2, 6

b.append(10)

for number in b:

 print(number, ' \*\* 2 =', number \*\* 2)

Давайте попробуем запустить наш код. У нас появилась ошибка AttributeError. Мы пока что не будем разбираться, что здесь написано, а исправим нашу программу вот так: числа для переменной b нужно заключить в квадратные скобки.

# user\_number = int(input('Введите число: '))

b = [1, 5, 7, -2, 6]

b.append(10)

for number in b:

 print(number, ' \*\* 2 =', number \*\* 2)

И снова запустим программу. Кажется, у нас всё получилось, но что же получилось, давайте разбираться.

Здесь мы познакомились с новым объектом – это список. Это такая переменная, которая хранит в себе не одно число, а сразу несколько. Инициализируется (создается) список так: сначала имя списка, потом знак равно, далее открывающая квадратная скобка, числа через запятую и в конце закрывающая квадратная скобка.

Вторая (b.append(10)) строчка это добавление нового элемента в список.

И давайте вернемся к нашей первоначальной задаче: добавление нескольких чисел. Для начала сделаем задачку с добавлением одного числа, которое вводится с клавиатуры

b = [1, 5, 7, -2, 6]

user\_number = int(input('Введите число: '))

b.append(user\_number)

for number in b:

 print(number, ' \*\* 2 =', number \*\* 2)

Получилось. Переходим к добавлению, например, 5 чисел. И тут нам уже нужен цикл.

b = [1, 5, 7, -2, 6]

for i in range(5):

 user\_number = int(input('Введите число: '))

 b.append(user\_number)

for number in b:

 print(number, ' \*\* 2 =', number \*\* 2)

Вот у нас и получилось всё. Однако имена у нас такие себе, надо поменять, а заодно узнаем, что если переменная не используется в цикле (как наша переменная i), то такие переменные принято обозначать \_ (символ нижнее подчеркивание). Сделаем нашу программу читабельнее

numbers\_list = [1, 5, 7, -2, 6]

for \_ in range(5):

 user\_number = int(input('Введите число: '))

 numbers\_list.append(user\_number)

for number in numbers\_list:

 print(number, ' \*\* 2 =', number \*\* 2)

Давайте для закрепления нашей темы решим небольшую задачу.

Задача «Библиотека»

У нас есть база данных с книгами, в ней хранятся ID (identification – уникальный номер для каждой книги) выданных книг, и если книгу возвращают, то идентификатор заменяется на -1. И в это утро мы уже выдали 10 книг, и из них уже вернули 3. Такой список выглядит так [50, 34, -1, -1, 2, 23, -1]. День продолжается и мы выдали еще 5 книг. Наша задача: в конце дня сформировать новый список книг, в котором останутся только те книги, которые не вернули, а те книги, которые вернули, нужно исключить. А также посчитает сколько книг вернули. Приступим

books\_ID = [50, 34, -1, -1, 2, 23, -1]

new\_books\_ID = []

returned = 0

for \_ in range(5):

 id = int(input('Введите ID книги: '))

 books\_ID.append(id)

for id in books\_ID:

 if id == -1:

 returned += 1

 else

 new\_books\_ID.append(id)

print('Новый список выданных книг:', new\_books\_ID)

print('Вернули за день:', returned)

Отлично. Задача решена.