

# Программирование на Python

## Урок №11

### План урока:

1. Углубленное изучение функций
2. Еще немного о списках
3. Списки и функции
4. Задания

### Результат:

```
def obnulit(new_spisok):  
    i = 0  
    while i < len(new_spisok):  
        new_spisok[i] = 0  
        i = i + 1  
spisok = [1, 2, 3, 4]  
print(spisok)  
obnulit(spisok)  
print(spisok)
```

### Углубленное изучение функций:

На прошлом занятии мы изучили функции, которые принимают какие-то значения, а затем что-то передают обратно. Но функции могут как ничего не принимать, так ничего и не передавать. Такие функции встречаются реже, но их все равно необходимо знать. К примеру. функция, которая ничего не принимает и не передает.

```
def hello():  
    print("Hello world")  
hello()  
hello()  
hello()
```

В результате выполнения этой программы, на экран 3 раза выводится “Hello world”. Ее можно визуализировать как кнопку, при нажатии которой, автомат с вами здоровается.

Следующая функция ничего не принимает, но передает переменную.

```
def hello():
    name = input()
    name = "Hello " + name
    return name
privet = hello()
print(privet)
```

Данная функция просит ввести ваше имя, потом превращает добавляет к имени слово “Hello” и возвращает его.

Следующая функция ничего не возвращает, но принимает значения.

```
def hello(name):
    print("Hello ", name)
hello("Ilya")
```

Эта функция принимает имя и здоровается с пользователем по имени.

### Еще немного о списках:

При обращении с переменными, мы привыкли думать, что две разные переменные никак не влияют друг на друга. Что это совершенно разные объекты. Если мы изменим переменную x, то переменная y никак не изменится. Со списками все обстоит немного не так. К примеру. рассмотрим следующую программу.

```
spisok = [1, 2, 3, 4]
print(spisok)
new_spisok = spisok
new_spisok[0] = 10000
print(spisok)
print(new_spisok)
```

Значение списка spisok никак не менялось, менялось только значение списка new\_spisok. При этом изменился и список spisok. Получается, spisok и new\_spisok теперь по сути являются одним и тем же списком, только под разными именами. Это происходит

из-за того, что на самом деле название списка это не просто наклейка на мешке, а куда более сложная конструкция. Можно представлять это как наклейку на мешок, с передней стороны которой находится имя, а с задней стороны указатель, как найти этот мешок в памяти компьютера. И когда мы пишем, что один список равен другому, он копирует только путь до мешка, а не создает новый с такими же значениями. Скопировать список так, чтобы это было два новых объекта можно следующим образом.

```
spisok = [1, 2, 3, 4]
print(spisok)
new_spisok = []
i = 0
while i < len(spisok):
    new_spisok.append(spisok[i])
    i = i + 1
new_spisok[0] = 10000
print(spisok)
print(new_spisok)
```

Теперь spisok и new\_spisok это два разных объекта.

## Списки и функции:

В отличие от переменных, которые при передачи в функцию никак не изменяются, списки ведут себя по другому. Если мы передадим список в функцию, то он изменит свое значение. К примеру, данная функция обнуляет все значения в списке.

```
def obnulit(new_spisok):
    i = 0
    while i < len(new_spisok):
        new_spisok[i] = 0
        i = i + 1
spisok = [1, 2, 3, 4]
print(spisok)
obnulit(spisok)
print(spisok)
```

В функции наш список имеет другое название, но это не важно, список все равно обнуляется из-за того, что это один и тот же объект. В остальном же, передача в функцию списка и переменной не отличаются.

### **Задания:**

1. Реализовать функцию, которая будет увеличивать все числа в списке на 1
2. Реализовать функцию, которая будет изменять число в списке на противоположное по знаку
3. Реализовать функцию, которая будет изменять все число больше нуля на 1, равные нулю на 0, а меньше нуля на -1.
4. Реализовать функцию, которая будет складывать все числа в списке и возвращать эту сумму
5. Реализовать функцию, которая будет находить среднее значение чисел, лежащих в списке и выводить это значение.

### **Тайминг:**

Тема	Время с начала занятия, мин
Углубленное изучение функций	20
Еще немного о списках	50
Списки и функции	65
Задания	85
Контрольные вопросы	90

### **Контрольные вопросы:**

1. Почему списки не рекомендуется приравнивать друг к другу?
2. Что на самом деле означает название списка?
3. Что будет если приравнять два списка друг к другу, что станет с их именами?
4. Как обойти эту проблему?
5. Как передать список в функцию?
6. В чем отличие передачи переменной в функцию и списка?