

## Лабораторная работа 3. Задачи по переменным, циклам

### Умный калькулятор

Создайте программу, что будет получать от пользователя действия (+, -, \*, /). В зависимости от действия будет показываться сумма, разница, произведение или частное двух чисел, заданных пользователем.

Например:

"Первое число: 2"

"Операция: -"

"Второе число: 3"

2 - 3 = -1

### Проверка на ноль

К предыдущему заданию добавьте проверку при делении на ноль. Запишите проверку при помощи сокращенной формы условия.

### Тройная проверка

Создайте переменную со значением 14.

Создайте проверку, которая выведет значение «True» в случае если переменная больше за 10, при этом не равна 12 и при этом меньшая или равная числу 15 или же если переменная равна числу 18.

### Работа с циклами

Выведите столбец чисел от 34 до 67 с выводом только четных чисел. Используйте цикл while для этой задачи.

### Цикл do while

В языке Python нет цикла do..while, тем не менее его можно с легкостью создать собственноручно. Создайте такой цикл.

Для тех кто не знает, цикл do..while это цикл который выполняется хотя бы один раз, после чего проверяет условие и если оно не верно, то выходит из цикла, иначе продолжает следующую итерацию.

### Вывод чисел

Выведите числа от 1 до 100 с пропуском чисел 50 и 99.  
Создайте вывод при помощи цикла for, а также цикла while.

### **Небольшая игра**

Попросите пользователя ввести какое-либо слово, а также число. При помощи циклов выведите каждый символ строки, при этом символ должен повторяться количество раз равным числу, что ввел пользователь. Каждый последующий новый символ необходимо выводить с новой строки, например:

Привет # То что ввел пользователь  
3 # Число, которое ввел пользователь

# Результат

ППП

ррр

иии

ввв

еее

ттт

## Лабораторная работа 4. Списки

**Списки** можно также называть массивами, так как выполняют они схожую роль. В Python существует множество разновидностей таких "массивов". В уроке мы рассмотрели лишь первый тип, который называется списком и имеет следующие характеристики:

- его элементы можно переопределять;
- в него можно добавлять новые элементы;
- из него можно удалять элементы;
- в качестве ключей идут индексы (числа - 0, 1, 2...).

Это не все характеристики списка, тем не менее это ключевые моменты, которые отличают его от других "массивов" в языке Python.

Для списков существует большое количество функций, которые можно применять для добавления, удаления или же поиска элементов.

В списках можно хранить различную информацию. К примеру, вы можете хранить в нем сведения про всех студентов в группе.

Пример:

```
lis = ["Андрей", "Иван", "Василий", "Петро", "Максим", "Дима"]
```

Списки (list) в языке Python. Списки также часто называют массивами в Питоне. Благодаря спискам мы можем хранить большое количество данных всего в одной переменной.

```
l = []
lis = [1, 56, 'x', 34, 2.34, ['S', 't', 'r', 'o', 'k', 'a']]
print (lis)

a = [a + b for a in 'list' if a != 's' for b in 'soup' if b
    != 'u']
print (a)

l.append (23)
l.append (34)
b = [24, 67]
l.extend (b)
l.insert (1, 56)
l.append (34)
l.remove (34)
```

```
l.pop (0)
print (l.index (56))
print (l.count (34))
l.sort ()
l.reverse ()
l.clear ()

print (l)
```