Программирование на Python

Урок №5

План урока:

1. Списки
2. Списки и циклы
3. Вывод списка
4. Циклы, списки и условия
5. Задачи

Результат:

```
print("Введите количество учеников")
chis_uchenikov = int(input())
spisok = []
i = 0
while i < chis_uchenikov:
    print("Введите баллы ученика")
    balli = int(input())
    spisok.append(balli)
    i = i + 1
i = 0
while i < chis_uchenikov:
    if (spisok[i] >= 9) and (10 >= spisok[i]):
        print("Ученик номер", i,"получил оценку", 5)
    if (spisok[i] >= 7) and (8 >= spisok[i]):
        print("Ученик номер", i,"получил оценку", 4)
    if (spisok[i] >= 5) and (6 >= spisok[i]):
        print("Ученик номер", i,"получил оценку", 3)
    if (spisok[i] >= 0) and (4 >= spisok[i]):
        print("Ученик номер", i,"получил оценку", 2)
    i = i + 1
```

Списки:

Вспомним нашу недавнюю задачу, где мы хотели определить самого сильного спортсмена и его номер. Но у нашей программы есть небольшой недостаток — она выводит только победителя, результаты остальных спортсменов же просто теряются. Понятно, что для этого нам нужно где-то запоминать их результаты. Пока что нам известен только один способ это сделать — хранить значение в переменной. Но в данном случае, этот способ просто неприменим. Мы не знаем заранее сколько спортсменов будет участвовать, но даже в случае, если мы предположим, что их количество не превышает 100 человек, то создавать и использовать 100 переменных будет просто невероятно тяжело. Нам необходим инструмент, который позволит хранить в нем данные о заранее неизвестном количестве спортсменов и при этом, позволяющий легко с ними работать. И такой инструмент называется списки.

Spisok						
Значение	8	1	0	10	5	12
i =	0	1	2	3	4	5
Вызов	spisok[0]	spisok[1]	spisok[2]	spisok[3]	spisok[4]	spisok[5]

Можно рассматривать список, как мешок с названием, в который складываются наши коробочки-переменные, только вместо имен у них просто цифры. Нулевая переменная, первая, вторая, и так далее. И чтобы отличать переменные в одном мешке от переменных в другом, мы указываем название мешка и в квадратных скобках номер переменной, который хотим достать. Мы будем называть переменную с номером, которая лежит в мешке-списке элементом. Добавление элемента со значением **x** в список с названием **spisok** производится командой **spisok.append(x)**.

Списки и циклы:

Циклы и списки крайне сильно связаны. И легко понять почему — списки предполагают хранение большого количества элементов,

точное количество которых, зачастую, заранее неизвестно. А циклы как раз позволяют осуществлять похожие действия большое количество раз, где конкретное количество выполнений мы можем определить сами. Давайте попробуем написать программу, которая будет просто принимать результаты участников соревнования и потом выводить их.

```
print("Введите число спортсменов")
chis_sports = int(input())
spisok = []
i = 0
while i < chis_sports:
    print("Введите вес, поднятный спортсменом")
    ves = int(input())
    spisok.append(ves)
    i = i + 1
```

Строчка spisok = [] создает новый пустой список. Можно считать, что мы достали новый мешок, сделали на него наклейку с названием и приготовили к наполнению, но в нем нет ни одного элемента. После каждого ввода нового веса, который был поднят участником, мы помещаем этот результат в наш мешок под его номером. В конце напишем **print(spisok)**, чтобы вывести список на экран. В конце мы видим просто список результатов спортсменов в том порядке, в котором они выступали.

Вывод списка:

Пока что отображение результатов в нашей таблице не наглядно. Намного лучше было бы сразу видеть номер спортсмена, а потом его результат. Для этого нам надо пройтись по нашему списку еще раз, сразу после его заполнения, и выводить у каждого элемента списка spisok[i] его номер i.

```
print("Введите число спортсменов")
chis_sports = int(input())
spisok = []
i = 0
while i < chis_sports:
    print("Введите вес, поднятный спортсменом")
    ves = int(input())
    spisok.append(ves)
    i = i + 1
i = 0
while i < chis_sports:
    print("Результат спортсмена под номером", i, "равняется", spisok[i])
    i = i + 1
```

Мы обнулили переменную i перед началом второго цикла из-за того, что после конца первого цикла она имеет значение равное длине списка. Затем просто вывели пары **i** и **spisok[i]** с пояснениями.

Циклы, списки и условия.

Как мы уже говорили, списки и циклы практически не используются друг без друга. Но так же очень часто вместе с ними используются и условия. Потому что просто вывести значения зачастую бывает мало, очень часто нам надо вывести или определенные значения, или вывести определенные значения в нужном нам виде. Давайте рассмотрим задачу, которая довольно часто встает перед учителями. Класс написал контрольную работу, каждый выполнил определенное количество заданий. И теперь на основе количества выполненных заданий, учитель должен проставить оценки. Пусть всего заданий было 10, от 9 до 10 выполненных заданий идет 5, от 6 и до 8 идет 4, от 4 и до 5 идет 3, а все, что ниже — 2. Сначала введем количество учеников, которые написал контрольную и количество выполненных ими заданий.

```
print("Введите количество учеников")
chis_uchenikov = int(input())
spisok = []
i = 0
while i < chis_uchenikov:
    print("Введите баллы ученика")
    balli = int(input())
    spisok.append(balli)
    i = i + 1
```

Затем попробуем просто вывести все результаты

```
print("Введите количество учеников")
chis_uchenikov = int(input())
spisok = []
i = 0
while i < chis_uchenikov:
    print("Введите баллы ученика")
    balli = int(input())
    spisok.append(balli)
    i = i + 1
i = 0
while i < chis_uchenikov:
    print("Результат увеника номер", i, "равняется", spisok[i])
    i = i + 1
```

Пока что наша программа никак не отличается от нашей предыдущей. Но теперь мы добавим главное — условия на вывод.

```
print("Введите количество учеников")
chis_uchenikov = int(input())
spisok = []
i = 0
while i < chis_uchenikov:
    print("Введите баллы ученика")
    balli = int(input())
    spisok.append(balli)
    i = i + 1
i = 0
while i < chis_uchenikov:
    if (spisok[i] >= 9) and (10 >= spisok[i]):
        print("Ученик номер", i,"получил оценку", 5)
    if (spisok[i] >= 7) and (8 >= spisok[i]):
        print("Ученик номер", i,"получил оценку", 4)
    if (spisok[i] >= 5) and (6 >= spisok[i]):
        print("Ученик номер", i,"получил оценку", 3)
    if (spisok[i] >= 0) and (4 >= spisok[i]):
        print("Ученик номер", i,"получил оценку", 2)
    i = i + 1
```

Ключевое слово and переводится как “И”. Имеет точно такое е значение, то есть, для срабатывания if нужно сразу два условия в каждой из скобок. Знак >= означает больше или равно.

Задачи:

1. На вход подается несколько чисел, нужно вывести сколько из них положительные, а сколько отрицательные.
2. В последнем задании необходимо людям получившим 3 или 2 выдавать сообщение о том сколько баллов они набрали и сколько им не хватило до 4.
3. В задании номер два добавить вывод значений сколько человек на какие оценки написали.

Тайминг:

Тема	Время с начала занятия, мин
Списки	15
Списки и циклы	35
Вывод списка	50
Циклы, списки и условия	80
Задачи	85
Контрольные вопросы	90

Контрольные вопросы:

1. Что такое список?
2. С какого номера идет нумерация элементов?
3. Чем список отличается от переменной?
4. С помощью какой команды мы можем вызвать элемент списка в цикле?
5. Как используются вместе циклы и списки?