

# Программирование на Python

## Урок №13

### План урока:

1. Вложенные списки
2. Работа с вложенными списками
3. Задания

### Вложенные списки:

Списки могут содержать в себе не только какие-то числа или строки, но так же и другие списки. Это может показаться сложным, но на самом деле все очень просто и мы практически ежедневно сталкиваемся с такими конструкциями. Вот так можно визуализировать 5 вложенных списков длиной 5 внутри списка длиной в 5.

	0	1	2	3	4
0					
1					
2					
3					
4					

Это обычная таблица, если мы, к примеру, хотим обратиться к элементу номер 1 из строки 3, то это ни у кого не вызовет сложностей. К примеру, у нас есть список `spisok`. И в нем есть 5 элементов. Если мы обращаемся к элементу с индексом 0, то мы попадаем в строку под номер 0 в нашей таблице. Элемент номер 0 это тоже список и мы говорим к какому элементу этого списка мы хотим обратиться. К примеру, это будет элемент под номер 3. Значит, мы обращаемся к строке с нулевым номером и столбцу с третьим номером.

	0	1	2	3	4
0					
1					
2					
3					
4					

Искомый элемент подсвечен красным. Давайте заполним список, который лежит в элементе основного списка под номером 0, нулями. Список, который лежит под номером 1 в основном списке — единицами. У нас получится следующее.

	0	1	2	3	4
0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4

Попробуем написать код такого списка.

```

spisok = []
i = 0
while i < 5:
    new = [i, i, i, i, i]
    spisok.append(new)
    i = i + 1
print(spisok)

```

В конце, программа выводит строки нашей таблицы через запятую. И каждая строка в квадратных скобках. То есть, это тоже список. Необходимо отметить, что список перед добавлением в основной список, мы создаем заново. Это делается из-за того, что иначе в каждой строке основного списка будет лежать один и тот же список с одними и теми же значениями.

### Работа с вложенными списками:

Вложенные списки часто называют двумерными. Когда мы обращаемся к элементу обычного списка, то мы пишем `spisok[i]`. Но в случае вложенных списков, такое обращение вызовет не конкретный элемент, а список, который находится внутри нашего основного списка. По аналогии с обращением к строке и столбцу, мы обращаемся к элементу вложенного списка используя запись вида `spisok[i][j]`. Именно отсюда и происходит название “двумерный список” — нам надо сразу две переменные, переменная `i` показывает какую строку выбрать, а переменная `j` показывает столбец. Попробуем вывести все элементы двумерного списка.

```

spisok = []
i = 0
while i < 5:
    new = [i, i, i, i, i]
    spisok.append(new)
    i = i + 1
i = 0
j = 0
while j < len(spisok):
    i = 0
    while i < len(spisok[j]):
        print(spisok[j][i])
        i = i + 1
    j = j + 1

```

Сначала мы проходим по нулевой строчке, перебирая все столбцы, потом по первой строчке, перебирая все столбцы, и так далее. Во внешнем цикле мы перебираем все строчки, а количество строчек равно длине списка `spisok`. Во внутреннем списке мы перебираем столбцы, а количество столбцов может отличаться у каждой строчки(но мы будем работать только с прямоугольными таблицами). Поэтому необходимо узнавать длину каждого вложенного списка отдельно.

### **Задания:**

1. Пользователь вводит число, нужно вывести все элементы, которые находятся в строке с этим номером.
2. Пользователь вводит число, нужно вывести все элементы, которые находятся в столбце с этим номером.
3. Вывести все числа, которые стоят на диагонали нашей таблицы.
4. Вывести все числа, которые находятся выше диагонали в таблице.

### **Тайминг:**

Тема	Время с начала занятия, мин
Вложенные списки	25
Работа с вложенными списками	50
Задания	85
Контрольные вопросы	90

### **Контрольные вопросы:**

1. Что такое вложенный список?
2. Из чего он состоит?
3. Что означает строка в вложенном списке?
4. Что означает столбец в вложенном списке?
5. Почему вложенный список иногда называют двумерным?

6. Как обратиться к конкретному элементу во вложенном списке?