

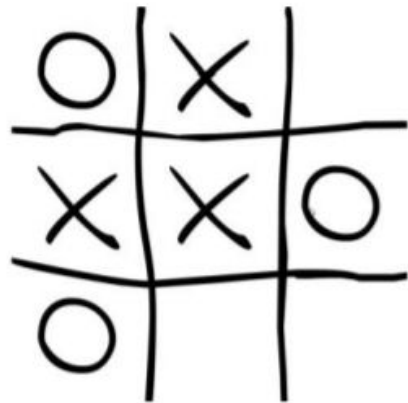
Вложенные списки

Двумерные вложенные списки (матрицы)

Язык **Python** не ограничивает нас в уровнях вложенности: элементами списка могут быть списки, элементами которых могут быть другие списки, элементами которых могут быть другие списки и так далее. Но для решения практических задач в первую очередь важно научиться работать с двумерными списками.

```
[ [-1, 0, 1],  
  [-1, 0, 1],  
  [0, 1, -1],  
  [1, 1, -1]]
```

Двумерные вложенные списки (матрицы)



	2		
	4	32	2
	8	64	128
2	4	2	2048

Создание двумерного списка



Список списков принципиально ничем не отличается от, например, списка чисел. Чтобы задать список списков в программе, мы также перечисляем элементы через запятую в квадратных скобках:

```
table = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

Если элементы списка вводятся через клавиатуру (каждая строка таблицы на отдельной строке, всего **n** строк, числа в строке разделяются пробелами), то для ввода списка можно использовать следующий код:

```
table = []  
for i in range(n):  
    row = [int(el) for el in input().split()]  
    table.append(row)
```

Или так:

```
table = [[int(el) for el in input().split()] for i in range(n)]
```


Перебор элементов. Вывод списка на экран.

Для доступа к элементу списка мы должны указать индекс этого элемента в квадратных скобках. В случае двумерных вложенных списков мы должны указать два индекса (каждый в отдельных квадратных скобках), в случае трёхмерного списка — три индекса и так далее. В двумерном случае сначала указывается номер строки, затем — номер столбца (сначала выбирается вложенный список, а затем — элемент из него).

```
table = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
print(table[0][0], table[0][1], table[1][0])
```

Перебор элементов. Вывод списка на экран. (ДЗ)

Для того, чтобы перебрать все элементы матрицы (чтобы, например, вывести их на экран), обычно используются вложенные циклы. Например, список из предыдущего примера можно вывести на экран таким образом:

```
table = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
  
for i in range(3):  
    for j in range(3):  
        print(table[i][j], end='\t')  
    print()
```

1. Как будет выведен список на экран?
2. Что будет, если поменять местами переменные `i` и `j` (печатать `table[j][i]`)?
3. Зачем нужен параметр `end` и почему используется символ табуляции, а не пробел?
4. Зачем нужен `print` без параметров? Можно ли без него обойтись?
5. Если нам нужно только вывести список на экран, можно ли обойтись одним циклом?