

Двумерные списки



В. М. Гуровиц, gurovic@gmail.com

Элементы списка



- `[2, 'abcde', 3.14]` – список может состоять из элементов разного типа
- В том числе, и из других списков:
- `s = [1, 2, 3]`
- `t = [4, 5]`
- `a = [s, t]`
- `print(a)`
`[[1, 2, 3], [4, 5]]`

Список списков



□ $b = [[1, 2, 3, 4],$
 $[5, 6, 7, 8],$
 $[9, 0, 1, 2]]$

Таблицу можно представить как список списков.

Будем называть его “двумерным” списком.

Доступ к элементам



```
□ b = [[1, 2, 3, 4],  
       [5, 6, 7, 8],  
       [9, 0, 1, 2]]
```

```
□ print(b[1][2])
```

```
7
```

```
□ print(len(b))
```

```
3
```

```
□ print(len(b[1]))
```

```
4
```

```
□ print(b[1])
```

```
[5, 6, 7, 8]
```

Создание списков



```
□ s = [0, 0, 0]
□ a1 = s + s      # [0, 0, 0, 0, 0, 0]
□ a2 = s * 2     # [0, 0, 0, 0, 0, 0]
□ a3 = [s] + [s] # [[0, 0, 0], [0, 0, 0]]
□ a4 = [s] * 2   # [[0, 0, 0], [0, 0, 0]]
□ a4[0][0] = 1  # [[1, 0, 0], [1, 0, 0]]
```

□ **Все эти способы – НЕПРАВИЛЬНЫЕ!**

Правильное создание списков



- Вариант 1.
- `s = [[]] * n`
- `for i in range(n):`
- `s[i] = [0] * m`

- Вариант 2.
- `s = []`
- `for i in range(n):`
- `s.append([0] * m)`

Правильное создание списков



- Вариант 3.
- `s = [[0] * m for i in range(n)]`

Самый короткий вариант – пользуйтесь им!

Чтение таблицы чисел



```
s = [[]]*3
for i in range(3):
    s[i] = list(map(int, input().split()))
```

Печать таблицы чисел



```
for i in range(3):  
    print(" ".join(map(str,s[i])))
```

Работа с элементами



- Работа с одним элементом:

```
print(a[2][1])
```

```
a[0][0] = 5
```

- Работа со строкой:

```
s[1] = [1, 2]
```

```
s[0], s[1] = s[1], s[0]
```

Важно понимать



1. Python не знает слов СТРОКА И СТОЛБЕЦ. Вы сами выбираете, что означает первый индекс, а что – второй.
2. При считывании данных они вводятся по строкам (сначала читается первая введенная строка, затем вторая и т.д.)
3. Списки (строки) могут разной (в том числе и нулевой) длины.