**Список (list)**

Списки в Python - упорядоченные изменяемые коллекции объектов произвольных типов (почти как массив, но типы могут отличаться).

Все методы списка (в ipython или тетради): **dir(list)** или **help(list)**

Help по 1 методу (например, append): list.append**?**

**Создать список**

Пустой список:

a = []

a = ['apple', 'banana', 'wildberry']

b = [12, 34, -5, 16]

c = [12, 'apple', [3.14, 9.81], 'orange' ]

d = list('hello')

**Срезы (Slice)**

* s[a:b:step] - подсписок с элементами от номера а (включительно) до номера b (не включительно), с шагом step

Можно изменять список при помощи срезов

>>> a = [1, 2, 3]

>>> a

[1, 2, 3] # все элементы числа

>>> a[2] = [4, 5]

>>> a

[1, 2, [4, 5]] # последний элемент - список, а не число

Изменять часть списка (удалить все, что слева, вставить в список то, что справа):

>>> a = [1, 2, 3]

>>> a

[1, 2, 3]

>>> a[1:2] = [4, 5]

>>> a

[1, 4, 5, 3]

DEA! Удаляемая и вставляемая части могут быть разной длины.

**Методы работы со списками**

|  |  |  |
| --- | --- | --- |
| [**Python**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=0&table=1&up=0#sorted_table) | [**Получилось**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=1&table=1&up=0#sorted_table) | [**Комментарий**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=2&table=1&up=0#sorted_table) |
| a=[1, 2, 3], a[0]=7 | [7, 2, 3] | элемент списка |
| len([1, 2, 3]) | 3 | Длина |
| [1, 2, 3] + [4, 5, 6] | [1, 2, 3, 4, 5, 6] | + Склеить |
| ['Hi!'] \* 4 | ['Hi!', 'Hi!', 'Hi!', 'Hi!'] | \* Повторить |
| [1, 2, 3] is [1, 2, 3] | True | Равны? |
| 3 in [1, 2, 3] | True | Проверить, что есть |
| 7 not in [1, 2, 3] | True | Проверить, что нет |
| for x in [1, 2, 3]:    print x | 1 2 3 | Напечатать все элементы |

**range(from, to, step)**

>>> range(1, 5)

[1, 2, 3, 4]

>>> range(10)

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

>>> range(1, 10, 2)

[1, 3, 5, 7, 9]

>>> range(20, 4, -5)

[20, 15, 10, 5]

>>> range(10, 20, -5)

[]

**Встроенные функции для работы со списками**

|  |  |
| --- | --- |
| [**Функция**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=0&table=2&up=0#sorted_table) | [**Что делает**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=1&table=2&up=0#sorted_table) |
| len(a) | Длина списка a |
| max(a) | Максимальный элемент списка а |
| min(a) | Минимальный элемент списка а |
| sum(a) | Сумма чисел (работает только с числами) |
| cmp(a, b) | Сравнить списки а и b |
| a = str.split(delimiter) | Сделать список а, разбив строку str на элементы по разделителям delimeter; по умолчанию делится по пробелам |
| s = str.join(a) | сделать из списка a строку, между элементами вставлять подстроку str |
| b = map(func, a) | применить функцию func к каждому элементу списка а b - не список, а map; чтобы его печатать, сделайте из него список print(list(b)) |

**Методы списков**

|  |  |
| --- | --- |
| [**Метод**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=0&table=3&up=0#sorted_table) | [**Что делает**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=1&table=3&up=0#sorted_table) |
| a.append(x) | Добавляет элемент х в список а |
| a.insert(i, x) | вставляет x на место номер i |
| a.extend(b) | a = a + b |
| a.remove(x) | удаляет x из списка а |
| a.count(x) | сколько раз элемент х входит в список а |
| a.index(x) | индекс первого вхождения х в список а или -1 |
| x = a.pop() | Удаляет последний элемент из списка а, возвращает этот элемент |
| a.reverse() | Лист в обратном порядке |
| a.sort() | Сортирует список |
| a.sort(функция) | Сортирует список элементов, пользуясь для сравнения двух элементов функцей |

**Разница между append и extend**

a = [1, 2, 3]

a.append([4, 5])

print(a) # [1, 2, 3, [4, 5]] в списке 4 элемента, последний элемент - список [4, 5]

b = [1, 2, 3]

b.extend([4, 5])

print(b) # [1, 2, 3, 4, 5] в списке 5 элементов

**Разница между append и +**

Конкатенация (+) создает новый объект, а метод append нет. Поэтому append работает быстрее.

a.append(x) в конец работает как a[len(a):] = [x]

a[:0] = [x] - добавить в начало списка. Работают так же быстро, как append.

**В списках хранятся ссылки**

В списке хранятся только ссылки на объекты.

Тут должен быть рисунок

m = [[1, 2, 3], [1, 2, 3], [1, 2, 3]]

print(m) # [[1, 2, 3], [1, 2, 3], [1, 2, 3]]

m[0][0] = 10

print(m) # [[10, 2, 3], [1, 2, 3], [1, 2, 3]]

a = [1, 2, 3]

m = [a, a, a] # m содержит 3 ссылки на один и тот же список a

print(m) # [[1, 2, 3], [1, 2, 3], [1, 2, 3]]

m[0][0] = 10

print(m) # [[10, 2, 3], [10, 2, 3], [10, 2, 3]]

UESTION? Как изменить код последнего примера, чтобы для создания списка использовался все тот же список а, но изменение элемента m[0][0]=10 не изменяло другие элементы матрицы?

**Вложенные списки (как хранить матрицу)**

Списки могут быть вложенными.

**Генераторы списков (list comprehensions)**

Про генераторы вообще поговорим в отдельном уроке. Сейчас получаем рецепты.

Общий вид как сделать список:

[выражение for переменная in последовательность]

или

[выражение for переменная in последовательность if условие]

Как сделать список, написав меньше кода?

Привычный вариант:

a = []

for x in range(5):

a.append(x\*\*2)

Если функция достаточно сложная, можно ее написать отдельно:

def sqr(x):

return x\*\*2

a = list(map(sqr, range(5)))

А если простая, то записать через lambda:

a = list(map(lambda x: x\*\*2, range(10)))

Или, как большинство программистов на питоне, использовать list comprehensions

a = [x\*\*2 for x in range(5)] # [0, 1, 4, 9, 16]

Примеры:

>>> [(x, y) for x in [1,2,3] for y in [3,1,4] if x != y]

[(1, 3), (1, 4), (2, 3), (2, 1), (2, 4), (3, 1), (3, 4)]

то же самое:

>>> combs = []

>>> for x in [1,2,3]:

... for y in [3,1,4]:

... if x != y:

... combs.append((x, y))

...

>>> combs

[(1, 3), (1, 4), (2, 3), (2, 1), (2, 4), (3, 1), (3, 4)]

Обратите внимание на порядок for и if.

>>> vec = [-4, -2, 0, 2, 4]

>>> [x\*2 for x in vec] # create a new list with the values doubled

[-8, -4, 0, 4, 8]

>>> [x for x in vec if x >= 0] # filter the list to exclude negative numbers

[0, 2, 4]

>>> [abs(x) for x in vec] # apply a function to all the elements

[4, 2, 0, 2, 4]

# call a method on each element

>>> freshfruit = [' banana', ' loganberry ', 'passion fruit ']

>>> [weapon.strip() for weapon in freshfruit]

['banana', 'loganberry', 'passion fruit']

>>> [(x, x\*\*2) for x in range(6)] # create a list of 2-tuples like (number, square)

[(0, 0), (1, 1), (2, 4), (3, 9), (4, 16), (5, 25)]

>>> # the tuple must be parenthesized, otherwise an error is raised

>>> [x, x\*\*2 for x in range(6)]

File "<stdin>", line 1, in <module>

[x, x\*\*2 for x in range(6)]

^

SyntaxError: invalid syntax

>>> vec = [[1,2,3], [4,5,6], [7,8,9]]

>>> [num for elem in vec for num in elem] # flatten a list using a listcomp with two 'for'

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

**Nested List Comprehensions (генерация вложенных списков)**

Дана матрица

>>> matrix = [

... [1, 2, 3, 4],

... [5, 6, 7, 8],

... [9, 10, 11, 12],

... ]

Надо ее транспонировать.

t = [[row[i] for row in m] for i in range(4)]

[[1, 5, 9], [2, 6, 10], [3, 7, 11], [4, 8, 12]]

Чуть подробнее:

transposed = []

for i in range(4):

transposed.append([row[i] for row in matrix])

еще подробнее:

transposed = []

for i in range(4):

# the following 3 lines implement the nested listcomp

transposed\_row = []

for row in matrix:

transposed\_row.append(row[i])

transposed.append(transposed\_row)

Дополнительный материал:

Разберитесь сами, как сделать то же самое с помощью функции [zip()](https://docs.python.org/3/library/functions.html#zip)

list(zip(\*matrix))

**Задачи (списки)**

**1 - Срезы строки**

Дан список a = [10, 7, -6, 11, 13, 5, 1, 8, 13]

|  |  |
| --- | --- |
| [**Что вывести**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=0&table=4&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=1&table=4&up=0#sorted_table) |
| первый элемент | 10 |
| элемент с номером 3 | 11 |
| последний элемент | 13 |
| предпоследний элемент | 8 |
| первые 6 элементов | 10 7 -6 11 13 5 |
| весь список, кроме последних 3 элементов | 10 7 -6 11 13 5 |
| все элементы с четными номерами (считая, что индексация начинается с 0); | 10 -6 13 1 13 |
| все элементы с нечетными номерами | 7 11 5 8 |
| все элементы в обратном порядке | 13 8 1 5 13 11 -6 7 10 |
| все элементы в обратном порядке, начиная с предпоследнего | 8 1 5 13 11 -6 7 10 |
| все элементы списка через один в обратном порядке, начиная с предпоследнего; | 8 5 11 7 |
| длину списка | 9 |

**3 - Вставить между буквами точки**

Получите новую строку, вставив между двумя символами исходной строки точки. Выведите полученную строку. Пример:

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=0&table=5&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=1&table=5&up=0#sorted_table) |
| python | p.y.t.h.o.n |

**5 - Найдите минимальное и максимальное число**

Дана последовательность целых чисел (на одной строке). Напечатайте минимальное и максимальное число.

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=0&table=6&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=1&table=6&up=0#sorted_table) |
| 7 19 -3 8 -11 0 56 | -11 56 |

**5a - Найдите минимальное и максимальное число**

Дана последовательность целых чисел (на многих строках). Напечатайте минимальное и максимальное число.

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=0&table=7&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=1&table=7&up=0#sorted_table) |
| 7 19 -3 8 -11  0 56 | -11 56 |

Рекомендуем из полученных списков чисел для строки сделать один общий список всех чисел.

**5b - Найдите минимальное и максимальное число**

Дана последовательность целых чисел (на многих строках). Напечатайте минимальное и максимальное число.

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=0&table=8&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=1&table=8&up=0#sorted_table) |
| 7 19 -3 8 -11  0 56 | -11 56 |

Надо экономить память.

Не делайте общий список всех чисел. Сделайте список минимальных в строках чисел и найдите в нем минимум.

Сделайте список максимальных= в строках чисел и найдите в нем максимум.

**6 - Отсортируйте числа**

Дана последовательность целых чисел (на одной строке). Напечатайте ее по возрастанию.

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=0&table=9&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonList?sortcol=1&table=9&up=0#sorted_table) |
| 7 19 -3 8 -11 0 56 | -11 -3 0 7 8 19 56 |

**7 - Оценки**

Студент за семестр получил оценки. Отбросьте 2 самых плохих оценки и посчитайте его средний балл.

-- [TatyanaDerbysheva](http://acm.mipt.ru/twiki/bin/view/Main/TatyanaDerbysheva) - 27 Oct 2015