

split, join, map, filter, lambda

В задачах этого листочка появятся новые *итерируемые* объекты.

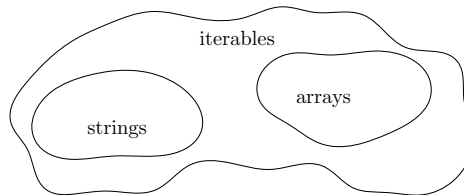
Итерируемый объект (iterable) — это такой объект X , к которому можно применить цикл

```
for elem in X
```

Для знакомых с классами в Python: это такой объект, для которого реализован метод `__iter__`.

Примерами итерируемых объектов являются строки и массивы. Однако, есть и другие итерируемые объекты. К ним всем можно применить цикл `for`, однако узнать длину или обратиться к итерируемому объекту по индексу нельзя.

Можно про них думать, как про конвейер, который по запросу отдаёт вам очередной элемент, но сам не знает сколько их там ещё (даже неизвестно, конечно ли их количество). И конечно, конвейер не «скажет», чему равен элемент с таким-то номером: те, что он отдал, уже забыты, а те, что ещё не отданы — пока неизвестны.



В тексте приняты следующие обозначения объектов и параметров функций:

$S (S1, S2, \dots)$ — строка

F — функция

IT — итерируемый объект

- $S1.split(S2)$ — разбивает строку $S1$ на части, используя строку $S2$, как разделитель. Возвращает массив строк.

Важно знать, как работает `split()`, в котором не указывается строка-разделитель.

В этом случае все лидирующие и финальные пробелы¹ игнорируются, все повторяющиеся пробелы интерпретируются как один и уже затем строка разбивается по пробелам.

Этот способ очень часто используется на практике, особенно когда есть риск «грязных» данных, где могут встретиться лишние пробелы. И важно понимать, что такой «пустой» `split()` принципиально отличается от `split('')` (где в качестве разделителя явно указан один пробел).

Явное указание пробела в качестве разделителя означает, что никакой предварительной «подготовки» строки не выполняется и **каждый** пробел это разделитель. И в массиве могут появляться, например, пустые строки.

Экспериментируйте!

- $S.join(IT)$ — возвращает строку, «склеенную» из элементов итерируемого объекта IT , используя строку S в качестве разделителя.

ВАЖНО! IT должен содержать только **строки**. Если это не строки, а например числа, надо их преобразовать в строки самостоятельно.

- $map(F, IT)$ — возвращает итерируемый объект, состоящий из элементов IT , к каждому из которых применена функция F .

```
print(map(int, "234"))
```

```
for elem in map(int, "234"):
    print(elem * elem)
```

```
print(list(map(int, "234")))
```

Первый `print()` напечатает что-то вроде такого:

```
<map object at 0x000002208E89E7C0>
```

Цикл напечатает числа 4 9 16.

А последний `print()` напечатает массив [4, 9, 16].

Обратите внимание на то, что в функцию `map()` первым параметром передаётся именно *функция*, как объект, а не результат её вызова на каких-то входных данных (после имени не стоят скобки).

¹На самом деле не только пробелы, но и все т.н. *пробельные* символы: пробелы, символы табуляции, переносы строк и прочие. Принадлежность символа к пробельным символам можно определить при помощи метода `str.isspace()`.

В качестве первого параметра функции `map()` можно использовать любые функции, в том числе самостоятельно написанные. Главное, чтобы эта функция принимала один параметр и чтобы она имела смысл при вызове с параметром — элементом ИТ.

Вот пример такого использования `map()`:

```
def extract_digits(s):
    ans = ""
    for char in s:
        if char.isdigit():
            ans += char
    return int(ans)

A = [
    "o2-r2",
    "password123",
    "hid9d3e2n1",
    "123!321?"
]

print(list(map(extract_digits, A))) # [22, 123, 9321, 123321]
```

- Анонимные функции `lambda` — ими обычно заменяют короткие функции, которые нужны «здесь и сейчас» и не предполагается их повторное использование в программе. Типичный пример — преобразование в `map()`, где в качестве первого параметра-функции можно указать `lambda`-функцию:

```
map(lambda x: x + 1, [2, 3, 5, 7])
```

Здесь задано отображение массива `[2, 3, 5, 7]`, а в качестве преобразования используется анонимная `lambda`-функция, увеличивающая аргумент на 1.

- `filter(F, ИТ)` — возвращает итерируемый объект, для всех элементов которого истинно значение функции `F`.
- К итерируемым объектам можно применять функции `sum()`, `min()`, `max()`.

Во всех задачах запрещается использование функции `eval()`.

Упражнения

- А. По числу получить массив его цифр.
- В. По массиву цифр получить число, этими цифрами записанное.

A. *Массив слов из строки*

Есть строка со словами. *Слово* — это последовательность непробельных символов, ограниченная слева и справа пробелом или началом или концом строки.

Надо написать программу, которая выведет массив из этих слов без пробелов.

Напишите решение на Python в одну строчку.

Input			Output
abcd	dcba	adcb	["abcd", "dcba", "adcb"]

B. *Другой разделитель*

Есть строка со словами. *Слово* — это последовательность непробельных символов, ограниченная слева и справа пробелом или началом или концом строки.

Вывести строку, где после каждого слова, кроме последнего, стоит запятая и пробел. Других пробелов в выводимой строке быть не должно.

Напишите решение на Python в одну строчку.

Input			Output
abcd	dcba	adcb	abcd, dcba, adcb

C. *Слова наоборот*

На вход программе подаётся строка, содержащая слова, разделённые пробелами (можно считать, что строка содержит только строчные буквы и пробелы и есть как минимум одно слово).

Программа должна напечатать строку, содержащую те же слова в обратном порядке, которые разделены *одним* пробелом (сами слова не меняются, меняется их порядок).

Напечатанная строка не должна начинаться с пробела или заканчиваться им.

Напишите решение на Python в одну строчку.

Input			Output
abcd	efgh	prst	prst efgh abcd

D. *Массив чисел*

На вход программе подаётся строка, содержащая целые числа, разделённые пробелами.

Программа должна напечатать массив из этих чисел.

Напишите решение на Python в одну строчку.

Input	Output
1 2 3 -2 19	[1, 2, 3, -2, 19]

E. *На один больше*

На вход программе подаётся строка, содержащая целые числа, разделённые пробелами.

Программа должна напечатать массив из чисел на 1 больше.

Напишите решение на Python в одну строчку.

Input	Output
1 2 3 -2 19	[2, 3, 4, -1, 20]

F. *Сумма чисел*

На вход программе подаётся строка, содержащая целые числа, разделённые пробелами.

Выведите их сумму.

Напишите решение на Python в одну строчку.

Input	Output
4 3 -17 2	-8

G. *Сумма цифр числа*

Дано неотрицательное целое число. Выведите сумму его цифр.

Запишите решение одной строчкой.

Input	Output
12345	15

Н. IP-адрес

IP-адрес (версия IPv4) это четырёхбайтовый код, который принято записывать в виде четырёх десятичных чисел, разделённых точками. Каждое из чисел может принимать значения от 0 до 255.

Вот примеры правильных IP-адресов:

127.0.0.0

192.168.0.1

255.0.255.255

Обратите внимание, что числа не могут начинаться с нуля. То есть такая строка не является IP-адресом: 127.0.0.01

Напишите функцию, которая будет возвращать `True`, если переданная строка является правильным IP-адресом, и `False` в противном случае.

На вход программе подаётся произвольная строка. Программа должна вывести строку `YES`, если это правильный IP-адрес и `NO` в противном случае.

Input	Output
127.0.0.1	YES

I. Сумма чисел - 2

Есть строка из положительных целых чисел, между которыми расставлены знаки «плюс».

Напишите решение на Python в одну строчку.

Input	Output
4+5+6+3+2	20

J. Только вычитания

Есть строка из положительных целых чисел, между которыми расставлены знаки «минус».

Выведите значение этого выражения. Запишите решение двумя строчками.

Input	Output
24-5-6-3-2	8

К. Две операции

Теперь в строке есть и знаки «плюс» и знаки «минус». Для простоты будем считать, что первое число положительное (строка начинается с цифры).

Выведите значение этого выражения. Используйте обе предыдущие задачи (в Python нельзя разбивать строку при помощи метода `split()` по нескольким разделителям, да это и не поможет).

Эту задачу тоже можно решить одной строчкой кода, но это необязательное требование.

Input	Output
8+1-8+3-6+2	0
8-1+7+4+4+10-6-9+2	19

L. Сумма квадратов

По строке вида $x\ y\ \dots\ z$ (целые числа, разделённые пробелами) вывести выражение вида

$$x^2 + y^2 + \dots + z^2.$$

Запишите решение одной строчкой. Обратите внимание: знаки «+» отделены от слагаемых одним пробелом.

Input	Output
1 2 3 4	1 + 4 + 9 + 16

M. Сумма некоторых чисел

Есть строка, где записаны целые числа, разделённые пробелами. Вывести сумму тех из них, которые записаны 3 цифрами. Если таких чисел нет, выведите число 0.

Input	Output
123 45 100 -87 3 999	1222

N. Нулей не меньше, чем единиц

Есть строка, где записаны неотрицательные целые числа, разделённые пробелами.

Вывести только те из них, в двоичной записи которых нулей не меньше, чем единиц. Числа при выводе разделите пробелами.

Указание: используйте функцию `bin()`.

Input	Output
1 2 3 4 5 6 7 8 9 0	2 4 8 9 0