

Step 4

Step on Stepik: <https://stepik.org/lesson/3362/step/4>

Давайте рассмотрим несколько примеров работы со строками. Заведем пару переменных, в которые запишем строки и выведем их, передав наши переменные в качестве аргументов в функцию **print**.

```
In [25]: a = 'string'
         b = 'another string'
         print(a, b)
         string another string
```

In []:

Видим что наши строчки вывелись как одна строка. При этом они разделены пробелом. В случае, когда мы передаем строки в функцию `print` в качестве параметров, при вывод они будут разделены пробелами. Теперь выведем одну строку, предварительно сложив наши строки вместе:

```
In [26]: a = 'string'
         b = 'another string'
         print(a + b)
         stringanother string
```

In []:

Вначале мы конкатенируем две наши строки и выводим результат. В выводе видно, что в отличие от предыдущего случая, здесь нет пробела между двумя строчками, происходит это потому, что при конкатенации мы приписываем вторую строку сразу за последним символом первой строки. Никаких пробелов в этом случае не добавляется. Теперь выведем наши строки на разных строчках. Это можно сделать с помощью двух вызовов функции **print**.

```
In [27]: a = 'string'
        b = 'another string'
        print(a)
        print(b)

        string
        another string
```

In []:

Мы можем использовать один вызов функции print. В качестве параметра передадим результат сложения строк, добавив к нему символ перевода строки '\n':

```
In [28]: a = 'string'
        b = 'another string'
        print(a + '\n' + b)

        string
        another string
```

In []:

Получим такой же результат, как и в предыдущем случае.

Добавим комментарий к нашей операции:

```
In [29]: a = 'string'
        b = 'another string'
        print(a + '\n' + b)  # print two different lines

        string
        another string
```

In []:

Убедимся, что после запуска мы получим тот же результат. Комментарий не как ни повлиял на работу нашей программы. Теперь добавим многострочный комментарий.

```
In [30]: a = 'string'
        b = 'another string'
        print(a)
        ...
            multiline
            comment
        ...
        print(b)

        string
        another string
```

In []:

В очередной раз убеждаемся, что комментарии никоим образом не интерпретируются Python.