# Урок 14.02. Методы строк split и join.

Как мы уже заметили ранее, методы строк помогают нам лучше и проще работать со строками. Сегодня мы поговорим о еще двух методах, которые используются практически везде при работе со строковыми переменными.

Допустим мы работаем с текстовым файлом и нам надо создать список слов убрав все разделители в тексте. Раньше мы бы решили эту задачу вот так:

words\_list = []

while True:

word = input('Введите слово: ')

if word != 'end':

words\_list.append(word)

else:

break

print(words\_list)

Т.е. вводили бы до бесконечности слова по отдельности, пока не написали бы слово «end». Однако пришло время нам стать немного серьезнее и изменить условия задачи на более актуальные. А именно: пользователь вводит все нужные слова в одной строчке через пробел. Для этого сотрем весь наш код и напишем программу с нуля. Для начала нам нужно запросить само содержимое – строку. Далее создадим переменную список. Вот как это будет выглядеть:

text = input('Введите строку: ')

words\_list = []

А дальше нам поможет один из методов строк, которые разделяет текст на элементы списка – split (переводится как «разделить»). И пишется он следующим образом:

text = input('Введите строку: ')

words\_list = text.split()

print(words\_list)

Запустим и убедимся, что программа работает. Также стоит обратить внимание на то, что он убирает все пробелы. Отлично.

А теперь нам нужно вставить вместо разделителей пробелы три тире (---). Сейчас мы умеем решать это следующим образом:

text = input('Введите строку: ')

words\_list = text.split()

print(words\_list)

for word in words\_list:

print(f'---{word}', end='')

Однако при запуске мы получаем в начале лишнее тройное тире и вообще выглядит не очень красиво. И теперь пришла пора узнать еще один метод – метод соединения списка строковых элементов в одну строку с указанным разделителем. Для этого сначала указывается разделитель, потом у этого разделителя вызывается метод «join» (переводится как «присоединить»), после чего вы функцию этого разделителя передается сам список строк. Программа будет выглядеть следующим образом:

text = input('Введите строку: ')

words\_list = text.split()

print(words\_list)

print('---'.join(words\_list))

Вот теперь и программа выглядит красиво и работает этого тоже быстрее (да, да, этот метод тоже быстрее), лучше и без ошибок.