

Здесь начинаются удивительные вещи! Помните, мы говорили, что операции зависят от типа данных? Над объектами определенного типа можно производить только определенные операции: числа – складывать, умножать и т.д. Так вот, для строк символ + будет объединять строки, а для чисел – складывать их. А, что если сложить число и строку?

```
>>> 'Mapc' + 5
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    'Mapc' + 5
TypeError: Can't convert 'int' object to str implicitly
>>>
```

Python запутался, т.к. не понял, что мы от него хотим: сложить числа или объединить строки. К примеру, мы хотим объединить строки. Для этого с помощью функции `str` преобразуем число 5 в строку '5' и выполним объединение:

```
>>> 'Mapc' + str(5)
'Mapc5'
>>>
```

Можно ли выполнить обратное преобразование типов? Можно!

```
>>> int("-5")
-5
>>>
```

Попросим Python повторить нашу строку заданное число раз:

```
>>> "СПАМ" * 10
'СПАМСПАМСПАМСПАМСПАМСПАМСПАМСПАМСПАМ'
>>>
```

Операция умножения для строк приобрела другой смысл.

Строки можно присваивать переменным и дальше работать с переменными:

```
>>> s = "Я изучаю программирование"
>>> s
'Я изучаю программирование'
>>> s*4
'Я изучаю программированиеЯ изучаю программированиеЯ изучаю
программированиеЯ изучаю программирование'
>>> s + " на языке Python"
'Я изучаю программирование на языке Python'
>>>
```

Если хотим поместить разные виды кавычек в строку, то сделать это можно несколькими способами:

```
>>> "Hello's"
"Hello's"
>>> 'Hello\'s'
"Hello's"
>>>
```