**Python 3: Строки. Функции и методы строк**

**Базовые операции**

**# Конкатенация (сложение)**

>>> s1 = 'spam'

>>> s2 = 'eggs'

>>> print(s1 + s2)

'spameggs'

**# Дублирование строки**

>>> print('spam' \* 3)

spamspamspam

**# Длина строки**

>>> len('spam')

4

**# Доступ по индексу**

>>> S = 'spam'

>>> S[0]

's'

>>> S[2]

'a'

>>> S[-2]

'a'

**# Срез**

>>> s = 'spameggs'

>>> s[3:5]

'me'

>>> s[2:-2]

'ameg'

>>> s[:6]

'spameg'

>>> s[1:]

'pameggs'

>>> s[:]

'spameggs'

**# Шаг, извлечения среза**

>>> s[::-1]

'sggemaps'

>>> s[3:5:-1]

''

>>> s[2::2]

'aeg'

**Другие функции и методы строк**

**# Литералы строк**

S = 'str'; S = "str"; S = '''str'''; S = """str"""

**# Экранированные последовательности**

S = "s\np\ta\nbbb"

**# Неформатированные строки (подавляют экранирование)**

S = r"C:\temp\new"

**# Строка байтов**

S = b"byte"

**# Конкатенация (сложение строк**)

S1 + S2

**# Повторение строки**

S1 \* 3

**# Обращение по индексу**

S[i]

**# Извлечение среза**

S[i:j:step]

**# Длина строки**

len(S)

**# Поиск подстроки в строке. Возвращает номер первого вхождения или -1**

S.find(str, [start],[end])

**# Поиск подстроки в строке. Возвращает номер последнего вхождения или -1**

S.rfind(str, [start],[end])

**# Поиск подстроки в строке. Возвращает номер первого вхождения или вызывает ValueError**

S.index(str, [start],[end])

**# Поиск подстроки в строке. Возвращает номер последнего вхождения или вызывает ValueError**

S.rindex(str, [start],[end])

**# Замена шаблона**

S.replace(шаблон, замена)

**# Разбиение строки по разделителю**

S.split(символ)

**# Состоит ли строка из цифр**

S.isdigit()

**# Состоит ли строка из букв**

S.isalpha()

**# Состоит ли строка из цифр или букв**

S.isalnum()

**# Состоит ли строка из символов в нижнем регистре**

S.islower()

**# Состоит ли строка из символов в верхнем регистре**

S.isupper()

**# Состоит ли строка из неотображаемых символов (пробел, символ перевода страницы ('\f'), "новая строка" ('\n'), "перевод каретки" ('\r'), "горизонтальная табуляция" ('\t') и "вертикальная табуляция" ('\v'))**

S.isspace()

**# Начинаются ли слова в строке с заглавной буквы**

S.istitle()

**# Преобразование строки к верхнему регистру**

S.upper()

**# Преобразование строки к нижнему регистру**

S.lower()

**# Начинается ли строка S с шаблона str**

S.startswith(str)

**# Заканчивается ли строка S шаблоном str**

S.endswith(str)

**# Сборка строки из списка с разделителем S**

S.join(список)

**# Символ в его код ASCII**

ord(символ)

**# Код ASCII в символ**

**chr(число)**

**# Переводит первый символ строки в верхний регистр, а все остальные в нижний**

S.capitalize()

**# Возвращает отцентрованную строку, по краям которой стоит символ fill (пробел по умолчанию)**

S.center(width, [fill])

**# Возвращает количество непересекающихся вхождений подстроки в диапазоне [начало, конец] (0 и длина строки по умолчанию)**

S.count(str, [start],[end])

**# Возвращает копию строки, в которой все символы табуляции заменяются одним или несколькими пробелами, в зависимости от текущего столбца. Если TabSize не указан, размер табуляции полагается равным 8 пробелам**

S.expandtabs([tabsize])

**# Удаление пробельных символов в начале строки**

S.lstrip([chars])

**# Удаление пробельных символов в конце строки**

S.rstrip([chars])

**# Удаление пробельных символов в начале и в конце строки**

S.strip([chars])

**# Возвращает кортеж, содержащий часть перед первым шаблоном, сам шаблон, и часть после шаблона. Если шаблон не найден, возвращается кортеж, содержащий саму строку, а затем две пустых строки**

S.partition(шаблон)

**# Возвращает кортеж, содержащий часть перед последним шаблоном, сам шаблон, и часть после шаблона. Если шаблон не найден, возвращается кортеж, содержащий две пустых строки, а затем саму строку**

S.rpartition(sep)

**# Переводит символы нижнего регистра в верхний, а верхнего – в нижний**

S.swapcase()

**# Первую букву каждого слова переводит в верхний регистр, а все остальные в нижний**

S.title()

**# Делает длину строки не меньшей width, по необходимости заполняя первые символы нулями**

S.zfill(width)

**# Делает длину строки не меньшей width, по необходимости заполняя последние символы символом fillchar**

S.ljust(width, fillchar=" ")

**# Делает длину строки не меньшей width, по необходимости заполняя первые символы символом fillchar**

S.rjust(width, fillchar=" ")