

Лекция 2

Работа со строками в Python.

Создание строк.

Преобразование типов данных с помощью функции `str()`.

Управляющие последовательности внутри строк.

Объединение и размножение строк. Функции `len()`, `split()`, `join()`, `replace()`.

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 (функция len)

```
>>> len('spam')
4
```

- Доступ по индексу

```
>>> S = 'spam'
```

```
>>> S[0]
```

```
's'
```

```
>>> S[2]
```

```
'a'
```

```
>>> S[-2]
```

```
'a'
```

Как видно из примера, в Python возможен и доступ по отрицательному индексу, при этом отсчет идет от конца строки.

Извлечение среза

Оператор извлечения среза: [X:Y]. X – начало среза, а Y – окончание;

символ с номером Y в срез не входит. По умолчанию первый индекс равен 0, а второй - длине строки.

>>>

```
>>> s = 'spameggs'
```

```
>>> s[3:5]
```

```
'me'
```

```
>>> s[2:-2]
```

```
'ameg'
```

```
>>> s[:6]
```

```
'spameg'
```

```
>>> s[1:]
```

```
'pameggs'
```

```
>>> s[:]
```

```
'spameggs'
```

Таблица "Функции и методы строк"

Функция или метод	Назначение
<code>S = 'str'; S = "str"; S = """str"""; S = """"str""""</code>	Литералы строк
<code>S = "s\np\ta\nbbb"</code>	Экранированные последовательности
<code>S = r"C:\temp\new"</code>	Неформатированные строки (подавляют экранирование)
<code>S = b"byte"</code>	Строка байтов
<code>S1 + S2</code>	Конкатенация (сложение строк)
<code>S1 * 3</code>	Повторение строки
<code>S[i]</code>	Обращение по индексу
<code>S[i:j:step]</code>	Извлечение среза

len(S)	Длина строки
S.find (str, [start],[end])	Поиск подстроки в строке. Возвращает номер первого вхождения или -1
S.rfind (str, [start],[end])	Поиск подстроки в строке. Возвращает номер последнего вхождения или -1
S.index (str, [start],[end])	Поиск подстроки в строке. Возвращает номер первого вхождения или вызывает ValueError
S.rindex (str, [start],[end])	Поиск подстроки в строке. Возвращает номер последнего вхождения или вызывает ValueError
S.replace (шаблон, замена[, maxcount])	Замена шаблона на замену. maxcount ограничивает количество замен
S.split (символ)	Разбиение строки по разделителю

S.isdigit()	Состоит ли строка из цифр
S.isalpha()	Состоит ли строка из букв
S.isalnum()	Состоит ли строка из цифр или букв
S.islower()	Состоит ли строка из символов в нижнем регистре
S.isupper()	Состоит ли строка из символов в верхнем регистре
S.isspace()	Состоит ли строка из неотображаемых символов (пробел, символ перевода строки ('\f'), "новая строка" ('\n'), "перевод каретки" ('\r'), "горизонтальная табуляция" ('\t') и "вертикальная табуляция" ('\v'))
S.istitle()	Начинаются ли слова в строке с заглавной буквы
S.upper()	Преобразование строки к верхнему регистру
S.lower()	Преобразование строки к нижнему регистру

S.startswith(str)	Начинается ли строка S с шаблона str
S.endswith(str)	Заканчивается ли строка S шаблоном str
S.join(список)	Сборка строки из списка с разделителем S
ord(символ)	Символ в его код ASCII
chr(число)	Код ASCII в символ
S.capitalize()	Переводит первый символ строки в верхний регистр, а все остальные в нижний
S.center(width, [fill])	Возвращает отцентрированную строку, по краям которой стоит символ fill (пробел по умолчанию)