Строки и методы строк

выполнила: Абдышева Бегимай / Python vol.6 / 06.07.20

Содержание:

- Строка (определение)
 - □ Как создавать строки
- 🖵 Конкатенация. Срезы
- Форматирование строк
- Методы строк
- **→** Q&A

Строка — неизменяемый тип данных; str от «string» — «строка». Строка представляет собой последовательность символов. Мы можем использовать одинарные или двойные кавычки для создания строки.

Строки всегда создаются одним из трех способов. Вы можете использовать одинарные, двойные и тройные скобки.

```
my string = "Добро пожаловать в Python!"
6
     another string = 'Я новый текст тут...'
8
     a long string = '''A это у нас
10
     новая строка
     в троичных скобках'''
12
13
14
15
16
```

```
my string = "I'm a Python programmer!"
     otherString = 'Слово "Python" обычно
     'подразумевает змею'
10
     tripleString = """В такой "строке" мы можем
12
13
     'использовать' все."""
14
15
16
18
```

Конкатенация — соединение или добавление двух объектов вместе.

```
3
     S1 = "Hello"
6
     S2 = "Maker's"
8
     print(S1 + ", " + S2 + "!")
10
     # "Hello, Maker's!"
11
12
13
```

Срезы – извлечение из данной строки одного символа или некоторого фрагмента по его индексу. Они работают по следующей схеме: [START: END: STEP].

```
s = 'I love programming!'
    5[2:6]
             # love
    s[2:-1]
             # love programming
10
    s[:6]
             # i love
11
12
    s[1:] # love programming!
13
14
    s[:] # i love programming!
15
16
```

Форматирование строк с помощью метода format()

```
print('Hello, %s' % name)
                                      #1 Форматирование строк "По старинке" (оператор %)
 6
    # Вывод: "Hello, Bob"
 8
 9
10
    print('Hello, {}'.format(name))
                                      #2 Форматирование строк "По новому" (str.format)
11
    # Вывод: 'Hello, Bob'
12
13
14
    name = 'Bob'
15
    print(f'Hello, {name}!')
#3 Интерполяция строк / f-Строки
16
17
18
    # Вывод: 'Hello, Bob!'
19
```

Методы строк

str.capitalize()

Возвращает копию строки в которой только первый символ находится в верхнем регистре, а все остальные в нижнем.

str.count

Возвращает количество непересекающихся вхождений подстроки в диапазоне [начало, конец]

str.isalnum()

Возвращает True если строка не пустая и состоит только из букв и цифр.

str.isalpha()

Возвращает True если строка не пустая и состоит только из букв.

str.islower()

Возвращает True если строка не пустая и состоит только из алфавитных строчных букв.

str.lower()

Возвращает новую строку в которой все символы исходной строки преобразованы к нижнему регистру

str.replace(old, new)

Возвращает копию исходной строки в которой все подстроки *old* заменены на подстроки *new*.

str.split

Выполняет разбиение исходной строки на подстроки по символу пробела и возвращает их в виде списка.

str.swapcase()

Возвращает копию исходной строки в которой все заглавные буквы преобразованы в строчные, строчные — в заглавные.

str.title()

Возвращает строку в которой каждое слово исходной строки начинается с буквы в верхнем регистре, а все остальные буквы в нижнем.

```
1
2
3    s_10 = 'hello world'
4    print(s_10.title())
5
6    # Hello World
```

Q & A