Введение в Python

Тип данных строка и методы для работы со строками

Амбросимова Виктория, 07.03.2022

Тип данных str

Строка для компьютера — это некоторая последовательность символов Юникода, заканчивающаяся специальным символом конца строки (например, символом ‘\n’ - перевода курсора на следующую строку).

Получение строки в python происходит с помощью команды input( ).

Строка в Python считается неизменяемым типом! Поэтому все «изменения» происходят путем перезаписи или пересоздания строки.

Тип данных str

- Строки в Python можно складывать и умножать на число

```
In [1]:
1 s ='hajgfkygf jhgqh ujhqd'
2 t = 'uewcb'
3 print(t+s)
4 print(t*3)

uewcbhajgfkygf jhgqh ujhqd
uewcbuewcbuewcb
```

- Если строку умножить на отрицательное число или ноль, то результатом будет пустая строка (проверьте)
- Операция сложения «+» называется конкатенацией строк, а оператор «*» создает копии строки

Тип данных str

- В Python есть оператор принадлежности in (или not in), который возвращает True или False в зависимости от вхождения подстроки в строку

```
In [2]:  1 s = 'qwerty'
         2 s in 'zxcvb asdfg qwerty'

Out[2]: True
```

- Оператор not in имеет обратную логику
- Далее будет представлен некоторый список функций и методов строк. С полным перечнем рекомендую ознакомиться в официальной документации языка Python

NORDIC IT SCHOOL
computer training courses

Встроеные функции строк

Функция	Описание
Type_int = len(‘str’)	Функция определения длины строки. Принимает string, возвращает int
Type_int = ord(‘char’)	Функция возвращает порядковый номер символа в таблице Unicode
Type_string = str( )	Функция приведения объекта к типу string
Type_string = chr(int)	Функция возвращает символ таблицы ASCII по его порядковому номеру

NORDIC IT SCHOOL
computer training courses

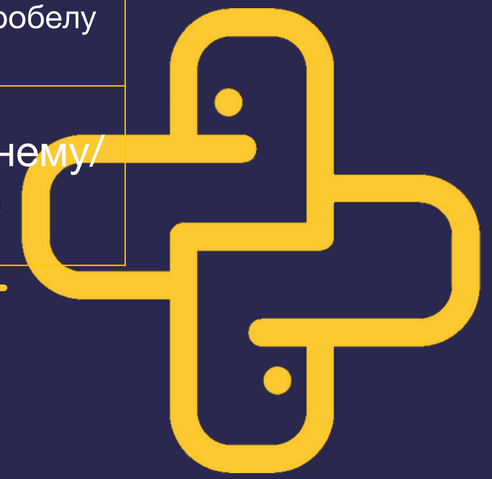
Базовые операции

- В строках возможен доступ по индексу $str[i]$. Причём индекс может быть отрицательным, но по модулю не больше длины самой строки. Отрицательный индекс позволяет проходить строку с конца
- Из строки можно взять срез $str[start : end : step]$ от символа с индексом $start$ до символа с индексом end с шагом $step$.

NORDIC IT SCHOOL
computer training courses

Методы строк

Метод	Описание
type_int = s.find (‘str’, [start], [end])	Поиск подстроки ‘str’ в строке s. Возвращается либо индекс первого вхождения, либо -1, если подстрока не найдена
type_str = s.replace(‘temp’, ‘subs’, [maxcount])	Осуществляет замену шаблона temp, на подстроку subs. Аргумент maxcount задает максимальное число замен. Его можно опустить
type_list = s.split(‘char’)	Разбиение строки по символу char. Возвращает список всех получившихся строк. По умолчанию разбивает по пробелу
type_str = s.upper( ) / s.lower( )	Методы преобразования строки к верхнему/ нижнему регистру соответственно



NORDIC IT SCHOOL
computer training courses

Методы строк, возвращающие bool

Метод	Описание
s.isdigit( ) / s.isalpha( ) / s.isalnum( )	Методы, проверяющие состоит ли строка только из цифр/букв/цифр и букв соответсвенно
s.isupper( ) / s.islower( )	Методы, проверяющие состоит ли строка из символов верхнего/нижнего регистра
s.startswith('str') / s.endswith('str')	Методы, проверяющие начинается/ заканчивается ли строка шаблоном str
s.isspase( )	Метод, проверяющиq состоит ли строка из неотображаемых символов (пробел, '\f', '\n', '\r', '\t', '\v')

Методы строк

Метод	Описание
type_str = s.join(list)	Сборка строки из списка с разделителем s
type_str = s.center(width, [fill])	Возвращает отцентрированную строку, по краям которой стоит символ fill (по умолчанию - пробел)
s.lstrip( ) / s.rstrip( ) / s.strip( )	Удаление пробельных символов в начале/в конце/в начале и конце строки
s.ljust(width, fillchar) / s.rjust(width, fillchar)	Делает длину строки не меньше width, по необходимости заполняя первые/последние символы символом fillchar

Возможный способ изменения строки

- Чтобы имитировать вставку символа в строку по индексу, следует перезаписать эту строку, используя такую конструкцию $t = t[: index] +' char' + t[index :]$
- Грубо говоря, мы берем два среза строки до нужного нам индекса и после, совершаем конкатенацию трех строк и перезаписываем результат в старую строку

```
In [5]:
1 t = 'asdfghjk'
2 index = 4
3 t = t[:index] + '0' + t[index:]
4 print(t)

asdf0ghjk
```