

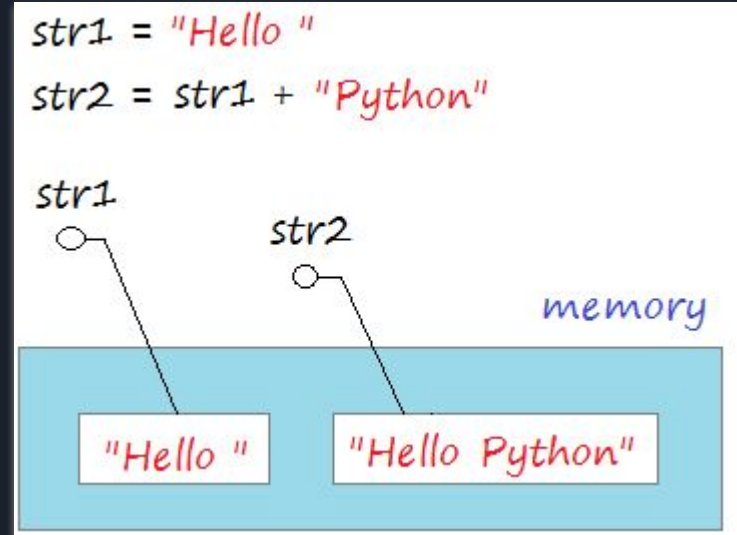


Python 3. Строки

Сулыз Андрей
iistclub.ru

Строки в программировании


В программировании, строковый тип (англ. string «нить, вереница») — тип данных, значениями которого является произвольная последовательность (строка) символов алфавита. Каждая переменная такого типа (строковая переменная) может быть представлена фиксированным количеством байтов либо иметь произвольную длину.





Базовые операции

- Конкатенация строк (сложение)
 1. `str1 = "Hello "`
 2. `str2 = "World"`
 3. `str1 + str2 => "Hello World"`
- Дублирование строк
 1. `"Hello" * 3 => "HelloHelloHello"`
- Доступ по индексу
 1. `str1 = "Hello"`
 2. `str1[0] => "H"`
- Срез
 1. `str1 = "Hello"`
 2. `str1[1:4] => "ell"`
- Длина строки
 1. `len("Hello") => 5`



Экранированные последовательности. Подавление экранирования

- Экранирование символов — замена в тексте управляющих символов на соответствующие текстовые подстановки.
- Неформатированные строки (подавляют экранирование):
 - `S = r"C:\temp\new"`

<code>\</code> в самом конце строки	Игнорируется, строка продолжается на новой строке
<code>\\</code>	Сам символ обратного слеша (остается один символ <code>\</code>)
<code>\'</code>	Апостроф (остается один <code>'</code>)
<code>\"</code>	Кавычка (остается один символ <code>"</code>)
<code>\n</code>	Новая строка (перевод строки)
<code>\r</code>	Возврат каретки
<code>\t</code>	Горизонтальная табуляция
<code>\u...</code>	16-битовый символ Юникода в 16-ричном представлении
<code>\U...</code>	32-битовый символ Юникода в 32-ричном представлении
<code>\x...</code>	16-ричное значение
<code>\o...</code>	8-ричное значение
<code>\0</code>	Символ Null (не признак конца строки)



Основные функции

- Поиск подстроки в строке.
Возвращает номер первого вхождения или -1
 1. `S.find(str, [start],[end])`
- Поиск подстроки в строке.
Возвращает номер последнего вхождения или -1
 1. `S.rfind(str, [start],[end])`
- Поиск подстроки в строке.
Возвращает номер первого вхождения или вызывает ValueError
 1. `S.index(str, [start],[end])`
- Поиск подстроки в строке.
Возвращает номер последнего вхождения или вызывает ValueError
 1. `S.rindex(str, [start],[end])`

Регулярные выражения

Это формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов. Для поиска используется строка-образец (маска), состоящая из символов и метасимволов и задающая правило поиска. Для манипуляций с текстом дополнительно задаётся строка замены, которая также может содержать в себе специальные символы.





Основные функции

- Замена шаблона
 1. `S.replace(шаблон, замена)`
- Разбиение строки по разделителю
 1. `S.split(символ)`
- Состоит ли строка из цифр
 1. `S.isdigit()`
- Состоит ли строка из цифр или букв
 1. `S.isalnum()`
- Состоит ли строка из букв
 1. `S.isalpha()`



Основные функции

- Состоит ли строка из символов в нижнем регистре
 1. `S.islower()`
- Состоит ли строка из символов в верхнем регистре
 1. `S.isupper()`
- Начинаются ли слова в строке с заглавной буквы
 1. `S.istitle()`
- Преобразование строки к верхнему регистру
 1. `S.upper()`
- Преобразование строки к нижнему регистру
 1. `S.lower()`



Основные функции

- Начинается ли строка `S` с шаблона `str`
 1. `S.startswith(str)`
- Заканчивается ли строка `S` шаблоном `str`
 1. `S.endswith(str)`
- Сборка строки из списка с разделителем `S`
 1. `S.join(список)`
- Удаление пробельных символов в начале строки
 1. `S.lstrip([chars])`
- Удаление пробельных символов в конце строки
 1. `S.rstrip([chars])`
- Удаление пробельных символов в начале и в конце строки
 1. `S.strip([chars])`

Иммутабельность

- В программировании неизменяемым (англ. *immutable*) называется объект, состояние которого не может быть изменено после создания.
- Строки в Питоне неизменяемы, поэтому каждая `+=` операция создаёт новую всё бóльшую строку

