**Строки**

**Создать строку**

* DEA! Строку можно написать в одинарных, двойных и тройных кавычках.
* \n \t и прочие - специальные значения (новая строка, табуляция, ...)
* DEA! r'строка' или R'строка' оставит \ без изменений.

Пустая строка

a = ''

b = ""

Не пустая строка

a = 'apple'

b = "banana"

c = '''текст на одной строке

и на другой строке'''

* a[0] - первый символ строки
* a[-1] - последний символ строки

**Срезы (Slice)**

* s[a:b:step] - часть строки с элементами от номера а (включительно) до номера b (не включительно), с шагом step

**Методы**

DEA! Нельзя изменить строку. Можно получить другую строку.

|  |  |  |  |
| --- | --- | --- | --- |
| [**Функция**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=0&table=1&up=0#sorted_table) | [**Что делает**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=1&table=1&up=0#sorted_table) | [**код**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=2&table=1&up=0#sorted_table) | [**результат**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=3&table=1&up=0#sorted_table) |
| a+b | строки вместе | "hel"+"lo" | "hello" |
| str\*n | повторить str n раз | "hi"\*3 | "hihihi" |
| **len**(str) | длина строки | len("hello") | 5 |
| str.**count**t(s) | сколько раз s встретилась в str | "alalabla".count("la")  "alalabla".count("ala") | 3  1 |
| str.**startswith**(s) | начинается str на s? | "hello".startswith("hel") "hello".startswith("el") | True False |
| str.**endswith**(s) | заканчивается str на s? | "hello".endswith("lo") "hello".endswith("el") | True False |
| s **in** str | строка s в str | "el" in "hello" "hel" in "hello" "Hel" in "hello" | True True False |
| str.**find**(s) | возвращает *первое* вхождение s в str или -1, если нет | "balalabla".find("la") "alalabla".find("zx") | 2 -1 |
| str.**index**(s) | возвращает *первое* вхождение s в str или ValueError[?](http://acm.mipt.ru/twiki/bin/edit/Cintro/ValueError?topicparent=Cintro.PythonString), если нет | "balalabla".index("la") | 2 |
| str.**rfind**(s) | возвращает *последнее* вхождение s в str или -1, если нет | "balalabla".rfind("la") "alalabla".rfind("zx") | 7 -1 |
| str.**rindex**(s) | возвращает *последнее* вхождение s в str или ValueError[?](http://acm.mipt.ru/twiki/bin/edit/Cintro/ValueError?topicparent=Cintro.PythonString), если нет | "balalabla".rindex("la") | 7 |
| str.**lstrip**(chars) | убирает символы из chars с начала строки | ' spacious '.lstrip() 'www.example.com'.lstrip('cmowz.') | 'spacious ' 'example.com' |
| str.**rstrip**(chars) | убирает символы из chars с конца строки | ' spacious '.rstrip() 'mississippi'.rstrip('ipz') | ' spacious' 'mississ' |
| str.**strip**(chars) | убирает символы из chars во всей строке | ' spacious '.strip() 'www.example.com'.strip('cmowz.') | 'spacious' 'example' |
| str.**replace**(old, new) str.replace(old, new, count) | Заменяет в str строку old на строку new (не более count раз) | "balalabla".replace("la", "xyz") "balalabla".replace("la", "xyz", 2) | "baxyzxyzbxyz" "baxyzxyzbla" |
| str.**partition**(sep) | возвращает 3 строки: до, sep, после |  |  |
| str.**rpartition**(sep) | возвращает 3 строки: до, sep, после |  |  |
| a = str.**split**(delimiter) a = str.split(delimiter, count) | Сделать список а, разбив строку str на элементы по разделителям delimeter; по умолчанию делится по пробелам |  |  |
| a = str.**rsplit**(delimiter) a = str.split(delimiter, count) | Сделать список а, разбив строку str на элементы по разделителям delimeter; по умолчанию делится по пробелам |  |  |
| s = str.join(a) | сделать из списка a строку, между элементами вставлять подстроку str |  |  |

Примеры:

before, s, after = "I have 2 small bombs and big bomb".partition("bomb")

print(before) # "I have 2 small "

print(s) # "bomb"

print(after) # "s and big bomb"

before, s, after = "I have 2 small bombs and big bomb".partition("dog")

print(before) # "I have 2 small bombs and big bomb"

print(s) # ""

print(after) # ""

>>> '1 2 3'.split()

['1', '2', '3']

>>> '1,2,3'.split(',')

['1', '2', '3']

>>> '1,2,3'.split(',', maxsplit=1)

['1', '2,3']

>>> '1,2,,3,'.split(',')

['1', '2', '', '3', '']

>>> '1 2 3'.split()

['1', '2', '3']

>>> '1 2 3'.split(maxsplit=1)

['1', '2 3']

>>> ' 1 2 3 '.split()

['1', '2', '3']

s = ["abc", "de", "xyz"].join("-")

print(s) # "abc-de-xyz"

**Задачи на строки**

**1 - Срезы строки**

Дана строка 'Abrucadabra'. Последовательно на разных строках выведите:

|  |  |
| --- | --- |
| [**Что вывести**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=0&table=2&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=1&table=2&up=0#sorted_table) |
| третий символ этой строки; | u |
| предпоследний символ этой строки; | a |
| первые пять символов этой строки; | Abruc |
| всю строку, кроме последних двух символов; | Abrucadab |
| все символы с четными номерами (считая, что индексация начинается с 0); | Arcdba |
| все символы с нечетными номерами; | buaar |
| все символы в обратном порядке. | arbadacurbA |
| все символы строки через один в обратном порядке, начиная с последнего; | abdcrA |
| длину данной строки. | 11 |

**2 - Слов в строке**

Дана строка. Подсчитать, сколько в ней слов.

**3 - Вставить между буквами точки**

Получите новую строку, вставив между двумя символами исходной строки точки. Выведите полученную строку. Пример:

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=0&table=3&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=1&table=3&up=0#sorted_table) |
| python | p.y.t.h.o.n |

**4 - Замените буквы, но не все**

Замените в строке все появления буквы h на букву H, кроме первого и последнего вхождения. Пример:

|  |  |
| --- | --- |
| [**Input**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=0&table=4&up=0#sorted_table) | [**Output**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=1&table=4&up=0#sorted_table) |
| aahhhhhbb | aahHHHhbb |

**Капитан Флинт (int)**

Капитан Флинт зарыл клад на Острове сокровищ. Он оставил описание, как найти клад. Описание состоит из строк вида: "0 5", где первое число – одно из 0 (North, север), 2 (Sout, юг), 1 (East, восток), 3 (West, запад), а второе число – количество шагов, необходимое пройти в этом направлении.

Напишите программу, которая по описанию пути к кладу определяет точные координаты клада, считая, что начало координат находится в начале пути, ось OX направлена на восток, ось OY – на север.

Программа получает на вход последовательность строк указанного вида, завершающуюся строкой со словом "Treasure!". Программа должна вывести два целых числа: координаты клада.

Путь 5 шагов на север, 3 шага на восток, 1 шаг на юг мы запишем как:

0 5   
1 3   
2 1   
Treasure!

|  |  |
| --- | --- |
| [**Вход**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=0&table=5&up=0#sorted_table) | [**Выход**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=1&table=5&up=0#sorted_table) |
| 0 5  1 3  2 1  Treasure! | 3 4 |

# -\*- coding: utf-8 -\*-

x = 0

y = 0

while True :

# прочитали одну строку в line

line = raw\_input()

# если Treasure! надо напечатать координаты и остановиться

if line == 'Treasure!' :

print x, y

break

# line разбили на части и каждую часть сделали int

pair = line.split()

direction = int(pair[0])

d = int(pair[1])

if direction == 0 : # север

y = y + d

# тут нужно дописать решение для юг, восток, запад

**Капитан Флинт (string)**

Капитан Флинт зарыл клад на Острове сокровищ. Он оставил описание, как найти клад. Описание состоит из строк вида: "куда 5", где первое слово *куда* – одно из: north (север), south (юг), east (восток), west (запад), а второе число – количество шагов, необходимое пройти в этом направлении.

Напишите программу, которая по описанию пути к кладу определяет точные координаты клада, считая, что начало координат находится в начале пути, ось OX направлена на восток, ось OY – на север.

Программа получает на вход последовательность строк указанного вида, завершающуюся строкой со словом "Treasure!". Программа должна вывести два целых числа: координаты клада.

Путь 5 шагов на север, 3 шага на восток, 1 шаг на юг мы запишем как:

north 5   
east 3   
south 1   
Treasure!

|  |  |
| --- | --- |
| [**Вход**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=0&table=6&up=0#sorted_table) | [**Выход**](http://acm.mipt.ru/twiki/bin/view/Cintro/PythonString?sortcol=1&table=6&up=0#sorted_table) |
| north 5  east 3  south 1  Treasure! | 3 4 |

-- [TatyanaDerbysheva](http://acm.mipt.ru/twiki/bin/view/Main/TatyanaDerbysheva) - 21 Oct 2015

(с) Материалы раздела "Язык Си" публикуются под лиценцией [GNU Free Documentation License.](http://www.gnu.org/copyleft/fdl.html)