

Python

Строковые и символьные величины. Строковые функции.

Тип `str` - это строка символов. Нумерация символов начинается с 0.

`s = 'python'` можно обращаться как к массиву `s[3]='h'` `s[-2]='o'`

0	1	2	3	4	5
p	y	t	h	o	n
-6	-5	-4	-3	-2	-1

Код	Результат
<pre>a = 'pyt' b = 'hon' s = a+b</pre>	<pre>s = 'python'</pre>
<pre>s = 'py.' s=s * 4 аналогично s=4*s</pre>	<pre>s='py.py.py.py.'</pre>
<pre>s = 'py.' s=s * -5</pre>	<pre>s="" – пустая строка</pre>

Оператор принадлежности подстроки `in`

<pre>>>> s = 'Python' >>> s in 'I love Python.' True >>> s in 'I love Java.' False</pre>	<pre>>>> 'z' not in 'abc' True >>> 'z' not in 'xyz' False</pre>
---	---

Срезы строк

Если пропустить первый индекс, срез начинается с начала строки. Таким образом

`s[:m] = s[0:m]:`

<pre>>>> s = 'python' >>> s[2:5] 'tho'</pre>	<pre>>>> s = 'python' >>> s[:4] 'pyth' >>> s[0:4] 'pyth'</pre>	<pre>>>> s = 'python' >>> s[2:] 'thon' >>> s[2:len(s)] 'thon'</pre>	<pre>>>> s = 'python' >>> s[-5:-2] 'yth' >>> s[1:4] 'yth'</pre>
--	---	--	--

Срезы строк с шагом

<pre>>>> s = 'python' >>> s[0:6:2] 'pto' >>> s[1:6:2] 'yhn'</pre>	<pre>>>> s = '12345' * 5 '123451234512345123451234512345' >>> s[::5] '11111' >>> s[4::5] '55555'</pre>	<pre>>>> s = 'python' >>> s[5:0:-2] 'nhy'</pre>	<pre>s = 'python' s[::-1] 'nohtyp' По сути, разворот строки</pre>
--	---	---	---

Задачи на строки

1. Дано предложение. Определить число букв "W" в нем.
2. Дано предложение. Определить число вхождений в него некоторого символа.
3. Дано предложение. Определить сколько в нем одинаковых соседних букв.
4. Дано предложение. Определить число вхождений в него буквосочетания "IQ".
5. Дано предложение. В нем слова разделены одним пробелом. Определить количество слов в предложении.
6. Дано предложение. В нем слова разделены одним или несколькими пробелами. Определить количество слов в предложении.
7. Дано предложение. Все буквы "S" в нем заменить на букву "F".

8. Дано предложение. Все его символы, стоящие на четных местах заменить на "R".
9. Дано предложение. Заменить в нем все вхождения буквосочетания "ах" на "ух".
10. Дана последовательность слов. Проверить, правильно ли в ней записаны сочетания "жи" и "ши". Исправить ошибки.
11. Найти в строке указанную подстроку и заменить ее на новую. Строку, ее подстроку для замены и новую подстроку вводит пользователь. При невозможности замены вывести соответствующее предложение.
12. Вводится строка. Удалить из нее все пробелы. После этого определить, является ли она палиндромом (перевертышем), т.е. одинаково пишется как с начала, так и с конца.
13. Вводится строка, содержащая буквы, целые неотрицательные числа и иные символы. Требуется все числа, которые встречаются в строке, поместить в отдельный целочисленный массив. Например, если дана строка "data 48 call 9 read13 blank0a", то в массиве должны оказаться числа 48, 9, 13 и 0.
14. Вводится строка. Требуется удалить из нее повторяющиеся символы и все пробелы. Например, если было введено "abc cde def", то должно быть выведено "abcdef".
15. Вводится ненормированная строка, у которой могут быть пробелы в начале, в конце и между словами более одного пробела. Привести ее к нормированному виду, т.е. удалить все пробелы в начале и конце, а между словами оставить только один пробел.
16. Вводится строка слов, разделенных пробелами. Найти самое длинное слово и вывести его на экран. Случай, когда самых длинных слов может быть несколько, не обрабатывать.
17. Дано слово. Определить, сколько различных букв в нем.
18. Даны два слова. Определить, можно ли из букв первого из них получить второе. Рассмотреть два варианта:
 - 1) повторяющиеся буквы второго слова могут в первом слове не повторяться;
 - 2) каждая буква второго слова должна входить в первое слово столько же раз, сколько и во второе.
19. Даны три слова. Напечатать только те буквы слов, которые есть только в одном из слов. Рассмотреть два варианта:
 - 1) повторяющиеся буквы каждого слова рассматриваются;
 - 2) повторяющиеся буквы каждого слова не рассматриваются.
20. Даны три слова. Напечатать их общие буквы. Повторяющиеся буквы каждого слова не рассматривать.
21. Даны три слова. Напечатать неповторяющиеся в них буквы (одним словом, через пробел).

Массивы из строк.

1. В массиве из строк определить количество строк начинающихся на букву «А» и ввести их.
2. В массиве найти строки, длина которых равна заданному числу, либо сообщить о их отсутствии.
3. В массиве найти строки, начинающиеся гласной русской буквой.
4. Отсортировать массив из строк по убыванию их длин.
5. Отсортировать массив из строк по алфавиту.
6. Дано предложение из 10 слов. Заполнить ими массив из 10 элементов.
7. Дано предложение. Напечатать его в обратном порядке слов, например, предложение "мама мыла раму" должно быть напечатано в виде "раму мыла мама".
8. Даны две строки. определить можно ли вторую строчку получить вычеркиванием символов первой строки. если можно то указать номера оставшихся символов первой строки.
9. Даны несколько строк текста, в которых нет начальных и конечных пробелов. Необходимо изменить эти строки так, чтобы их длины были одинаковыми. Это следует сделать путем вставки между словами дополнительных пробелов. Количество пробелов между отдельными словами внутри отдельной строки должно отличаться не более чем на 1.

Функция или метод	Назначение
len()	возвращает длину строки k=len('мир') k=3
ord()	возвращает числовое значение для заданного символа ord('a') → 97
chr()	возвращает символьное значение для данного целого числа chr(97) → a, действует обратно ord()
str()	возвращает строковое представление объекта str(3 + 29) → строка '32'
int()	int('145') → целое число 145
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.count(str, [start],[end])	Возвращает количество непересекающихся вхождений подстроки в диапазоне [начало, конец] (0 и длина строки по умолчанию)
S.replace(шаблон, замена,[maxcount])	Замена шаблона на замену. maxcount ограничивает количество замен
S.startswith(str)	Начинается ли строка S с шаблона str
S.endswith(str)	Заканчивается ли строка S шаблоном str
S.lstrip([chars])	Удаление пробельных символов в начале строки
S.rstrip([chars])	Удаление пробельных символов в конце строки
S.strip([chars])	Удаление пробельных символов в начале и в конце строки
S.istitle()	Начинаются ли слова в строке с заглавной буквы
S.isdigit()	Состоит ли строка из цифр
S.isalpha()	Состоит ли строка из букв
S.isalnum()	Состоит ли строка из цифр или букв
S.isspace()	Состоит ли строка из неотображаемых символов (пробел, символ перевода страницы ('\f'), "новая строка" ('\n'), "перевод каретки" ('\r'), "горизонтальная или" ('\t') и "вертикальная табуляция" ('\v'))
S.islower()	Состоит ли строка из символов в нижнем регистре

Функция или метод	Назначение
S.isupper()	Состоит ли строка из символов в верхнем регистре
S.title()	Первую букву каждого слова переводит в верхний регистр, а все остальные в нижний
S.upper()	Преобразование строки к верхнему регистру
S.lower()	Преобразование строки к нижнему регистру
S.swapcase()	Переводит символы нижнего регистра в верхний, а верхнего – в нижний
S.join(список)	Сборка строки из списка с разделителем S
S.capitalize()	Переводит первый символ строки в верхний регистр, а все остальные в нижний
S.center(width, [fill])	Возвращает отцентрованную строку, по краям которой стоит символ fill (пробел по умолчанию)
S.join(список)	Сборка строки из списка с разделителем S
S.expandtabs([tabsize])	Возвращает копию строки, в которой все символы табуляции заменяются одним или несколькими пробелами, в зависимости от текущего столбца. Если TabSize не указан, размер табуляции полагается равным 8 пробелам
S.partition(шаблон)	Возвращает кортеж, содержащий часть перед первым шаблоном, сам шаблон, и часть после шаблона. Если шаблон не найден, возвращается кортеж, содержащий саму строку, а затем две пустых строки
S.rpartition(sep)	Возвращает кортеж, содержащий часть перед последним шаблоном, сам шаблон, и часть после шаблона. Если шаблон не найден, возвращается кортеж, содержащий две пустых строки, а затем саму строку
S.zfill(width)	Делает длину строки не меньшей width, по необходимости заполняя первые символы нулями
S.ljust(width, fillchar=" ")	Делает длину строки не меньшей width, по необходимости заполняя последние символы символом fillchar
S.rjust(width, fillchar=" ")	Делает длину строки не меньшей width, по необходимости заполняя первые символы символом fillchar
S.format(*args, **kwargs)	Форматирование строки