

9.1 Индексация

Шаг 1

Тема урока: строки

1. Индексация строк
2. Итерирование строк
3. Решение задач

Аннотация. Строковый тип данных. Вспомним основные операции над строками, научимся работать с отдельными символами, а также перебирать (итерировать) символы строк.

Повторение материала

Строки в Python используются, когда надо работать с текстовыми данными.

Создание строки. Для создания строк мы используем парные кавычки '' или "" :

```
s1 = 'Python'  
s2 = "Pascal"
```

Считывание строки. Для считывания текстовых данных в строковую переменную мы используем функцию input() :

```
s = input() # считали текст  
num = int(input()) # считали текст и преобразовали его в целое число
```

Пустая строка. Для создания пустой строки мы пишем s = '' или s = "" . Пустая строка – это аналог числа 0 .

Длина строки. Для определения длины строки (количество символов) мы используем встроенную функцию len() :

```
s = 'Hello'  
n = len(s) # значение переменной равно 5  
print(n)
```

Конкатенация и умножение на число. Операторы + и * можно использовать для строк. Оператор + сцепляет две и более строк. Это называется конкатенацией строк. Оператор * повторяет строку указанное количество раз.

Выражение	Результат
'AB' + 'cd'	'ABcd'
'A' + '7' + 'B'	'A7B'
'Hi'* 4	'HiHiHiHi'

Оператор принадлежности in . С помощью оператора in мы можем проверять, находится ли одна строка в составе другой. То есть, является ли одна строка подстрокой другой:

```
s = 'All you need is love'  
if 'love' in s:  
    print('❤')  
else:  
    print('💔')
```

Так как строка s содержит подстроку 'love' , то будет выведен смайлик ❤ .



В Python можно использовать смайлики emoji (<https://pypi.org/project/emoji/>) 😊

Индексация строк

Очень часто бывает необходимо обратиться к конкретному символу в строке. Для этого в Python используются квадратные скобки `[]`, в которых указывается индекс (номер) нужного символа в строке.

Пусть `s = 'Python'`. Таблица ниже показывает, как работает индексация:

Выражение	Результат	Пояснение
<code>s[0]</code>	P	первый символ строки
<code>s[1]</code>	y	второй символ строки
<code>s[2]</code>	t	третий символ строки
<code>s[3]</code>	h	четвертый символ строки
<code>s[4]</code>	o	пятый символ строки
<code>s[5]</code>	n	шестой символ строки



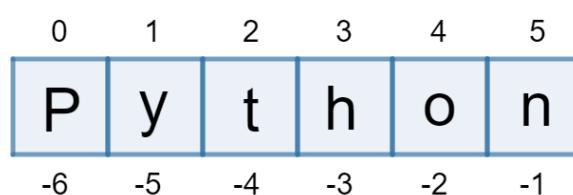
Обратите внимание, первый символ строки равен `s[0]`, а не `s[1]`. В Python индексация начинается с `0`, по аналогии с функцией `range(n)`, которая генерировала последовательность натуральных чисел от `0` до `n - 1`.

В отличие от многих языков программирования, в Python есть возможность работы с отрицательными индексами. Если первый символ строки имеет индекс `0`, то последнему элементу присваивается индекс `-1`.

Выражение	Результат	Пояснение
<code>s[-6]</code>	P	первый символ строки
<code>s[-5]</code>	y	второй символ строки
<code>s[-4]</code>	t	третий символ строки
<code>s[-3]</code>	h	четвертый символ строки
<code>s[-2]</code>	o	пятый символ строки
<code>s[-1]</code>	n	шестой символ строки

Таким образом, получаем

Положительные индексы	0	1	2	3	4	5
Строка	P	y	t	h	o	n
Отрицательные индексы	-6	-5	-4	-3	-2	-1



Частая ошибка у начинающих программистов – обращение по несуществующему индексу в строке.

Например, если `s = 'Python'`, и мы попытаемся обратиться к `s[17]`, то мы получим ошибку:

```
IndexError: string index out of range
```

Ошибка возникает, поскольку строка содержит всего 6 символов.



Обратите внимание: если длина строки `s` равна `len(s)`, то при положительной нумерации слева направо, последний элемент имеет индекс равный `len(s) - 1`, а при отрицательной индексации справа налево, первый элемент имеет индекс равный `-len(s)`.

Итерирование строк

Очень часто нужно просканировать всю строку целиком, обрабатывая каждый ее символ. Для этого удобно использовать цикл `for`. Напишем программу, которая выводит каждый символ строки на отдельной строке:

```
s = 'abcdef'  
for i in range(len(s)):  
    print(s[i])
```

Результатом выполнения такой программы будут строки:

```
a  
b  
c  
d  
e  
f
```

Мы передаем в функцию `range()` длину строки `len(s)`. В нашем случае длина строки `s`, равна 6. Таким образом, вызов функции `range(len(s))` имеет вид `range(6)` и переменная цикла `i` последовательно перебирает все значения от 0 до 5. Это означает, что выражение `s[i]` последовательно вернет все символы строки `s`. Такой способ итерации строки удобен, когда нам нужен не только сам элемент `s[i]`, но и его индекс `i`.

Если нам не нужен индекс самого символа, то мы можем использовать более короткий способ итерации:

```
s = 'abcdef'  
for c in s:  
    print(c)
```

Этот цикл пройдет по строке `s`, придавая переменной цикла `c` значение каждого символа (!) в отличие от предыдущего цикла, в котором переменная цикла «бегала» по индексам строки.



Обратите внимание на обозначение переменных цикла. В первом цикле мы используем имя `i`, что соответствует стандартной идеологии наименования переменных цикла. Во втором цикле, мы назвали переменную буквой `c` – первая буква слова char (символ).

❤️ Happy Pythoning! 🧑

Шаг 2

Что покажет приведённый ниже код?

```
s = 'abcdefg'  
print(s[0] + s[2] + s[4] + s[6])
```

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/2>

Шаг 3

Что покажет приведённый ниже код?

```
s = 'abcdefg'  
print(s[0]*3 + s[-1]*3 + s[3]*2 + s[3]*2)
```

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/3>

Шаг 4

Что покажет приведённый ниже код?

```
s = '01234567891011121314151617'  
for i in range(0, len(s), 5):  
    print(s[i], end='')
```

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/4>

Шаг 5

Используя индексатор, дополните приведённый ниже код так, чтобы он вывел символ запятой.

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/5>

Шаг 6

Используя индексатор, дополните приведённый ниже код так, чтобы он вывел символ `w`.

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/6>

Шаг 7

В столбик 1

На вход программе подаётся одна строка. Напишите программу, которая выводит элементы строки с индексами `0, 2, 4, ...` в столбик.

Формат входных данных

На вход программе подаётся одна строка.

Формат выходных данных

Программа должна вывести элементы строки с индексами `0, 2, 4, ...`, каждое на отдельной строке.

► Тестовые данные 

Sample Input:

```
abcdefghijklmnpqr
```

Sample Output:

```
a  
c  
e  
g  
i  
k  
m  
o
```

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/7>

Шаг 8

В столбик 2

На вход программе подаётся одна строка. Напишите программу, которая выводит в столбик элементы строки в обратном порядке.

Формат входных данных

На вход программе подаётся одна строка.

Формат выходных данных

Программа должна вывести в столбик элементы строки в обратном порядке.

► Тестовые данные 

Sample Input:

```
abc
```

Sample Output:

```
c  
b  
a
```

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/8>

Шаг 9

ФИО

На вход программе подаются три строки: имя, фамилия и отчество (именно в таком порядке). Напишите программу, которая выводит инициалы человека.

Формат входных данных

На вход программе подаются три строки, каждая на отдельной строке.

Формат выходных данных

Программа должна вывести ФИО человека.

Примечание. Гарантируется, что имя, фамилия и отчество начинаются с заглавной буквы.

► Тестовые данные 

Sample Input:

```
Тимур  
Гуев  
Ахсарбекович
```

Sample Output:

```
ГТА
```

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/9>

Шаг 10

Цифра 1

На вход программе подаётся одна строка состоящая из цифр. Напишите программу, которая считает сумму цифр данной строки.

Формат входных данных

На вход программе подаётся одна строка, состоящая из цифр.

Формат выходных данных

Программа должна вывести сумму цифр данной строки.

► Тестовые данные 

Sample Input:

```
2514
```

Sample Output:

```
12
```

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/10>

Шаг 11

Цифра 2

На вход программе подаётся одна строка. Напишите программу, которая выводит сообщение «Цифра» (без кавычек), если строка содержит цифру. В противном случае вывести сообщение «Цифр нет» (без кавычек).

Формат входных данных

На вход программе подаётся одна строка.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

► Тестовые данные 

Sample Input 1:

Hi Python

Sample Output 1:

Цифр нет

Sample Input 2:

Hi 17 Python

Sample Output 2:

Цифра

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/11>

Шаг 12

Сколько раз?

На вход программе подаётся одна строка. Напишите программу, которая определяет, сколько раз в строке встречаются символы  и 

Символ + встречается <n> раз
Символ * встречается <m> раз

где  ,  и 

Формат входных данных

На вход программе подаётся одна строка.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

► Тестовые данные 

Sample Input:

bcd+a++++**31415

Sample Output:

Символ + встречается 5 раз
Символ * встречается 2 раз

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/12>

Шаг 13

Одинаковые соседи

На вход программе подаётся одна строка. Напишите программу, которая определяет сколько в ней одинаковых соседних символов.

Формат входных данных

На вход программе подаётся одна строка.

Формат выходных данных

Программа должна вывести количество одинаковых соседних символов.

► Тестовые данные 

Sample Input 1:

abcd

Sample Output 1:

0

Sample Input 2:

aabbcc

Sample Output 2:

3

Sample Input 3:

aaa

Sample Output 3:

2

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/13>

Шаг 14

Гласные и согласные

На вход программе подаётся одна строка с буквами русского языка. Напишите программу, которая определяет количество гласных и согласных букв и выводит текст в следующем формате:

Количество гласных букв равно <кол-во гласных букв>

Количество согласных букв равно <кол-во согласных букв>

Формат входных данных

На вход программе подаётся одна строка.

Формат выходных данных

Программа должна вывести текст в соответствии с условием задачи.

Примечание 1. В русском языке 9 гласных букв (ауыиэяю) и 21 согласная буква (бвгджзйклмнпрстфхцчш).

Примечание 2. Ваша программа должна игнорировать все небуквенные символы, а также букву ё .

► Тестовые данные

Sample Input 1:

Вдохновение – это умение приводить себя в рабочее состояние!

Sample Output 1:

Количество гласных букв равно 25

Количество согласных букв равно 24

Sample Input 2:

Глупая критика не так заметна, как глупая похвала.

Sample Output 2:

Количество гласных букв равно 18

Количество согласных букв равно 23

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/14>

Шаг 15

Decimal to Binary 10

На вход программе подаётся натуральное число, записанное в десятичной системе счисления. Напишите программу, которая переводит данное число в двоичную систему счисления

(https://ru.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F).

Формат входных данных

На вход программе подаётся одно натуральное число.

Формат выходных данных

Программа должна вывести число записанное в двоичной системе счисления.

► Тестовые данные

Sample Input 1:

5

Sample Output 1:

101

Sample Input 2:

128

Sample Output 2:

10000000

Чтобы решить это задание откройте <https://stepik.org/lesson/284101/step/15>