

Программирование (Python)

§ 19. Символьные строки

Что такое символьная строка?

Символьная строка – это последовательность СИМВОЛОВ.

- строка – единый объект
- длина строки может меняться во время работы программы

Символьные строки

Присваивание:

```
s = "Вася пошёл гулять"
```

Ввод с клавиатуры:

```
s = input()
```

Вывод на экран:

```
print(s)
```

Длина строки:

```
n = len(s)
```

length – длина

Сравнение строк

```
print("Введите пароль: ")
s = input()
if s == "sEzAm":
    print("Слушаюсь и повинуюсь!")
else:
    print("Пароль неправильный")
```



Какой правильный пароль?



Как одна строка может быть меньше другой?

стоит раньше в отсортированном списке

Сравнение строк

```
s1 = "паровоз"  
s2 = "пароход"  
if s1 < s2:  
    print(s1, "<", s2)  
elif s1 == s2:  
    print(s1, "=", s2)  
else:  
    print(s1, ">", s2)
```



Что выведет?

паровоз < пароход

первые отличающиеся
буквы

Сравниваем с начала: паровоз
пароход

«В»: код 1074

«Х»: код 1093



В < Х!

Обращение к символу по номеру

```
print ( s[5] )
```

```
print ( s[-2] )
```

| | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|----------------------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | $s[\text{len}(s)-2]$ |
| П | р | и | в | е | т | ! | |
| $s[0]$ | $s[1]$ | $s[2]$ | $s[3]$ | $s[4]$ | $s[5]$ | $s[6]$ | |



Символы нумеруются с нуля!

СОСТАВИТЬ «КОТ»

```
s = "информатика"  
kot = s[-2]+s[3]+s[-4]
```

Посимвольная обработка строк

`s[4] ≠ "a"`



Строка неизменна!

Задача. Ввести строку и заменить в ней все буквы «э» на буквы «е».

строим новую строку!

```
sNew = ""
for i in range(len(s)):
    if s[i] == "э":
        sNew += "е"
    else:
        sNew += s[i]
```

для каждого символа строки

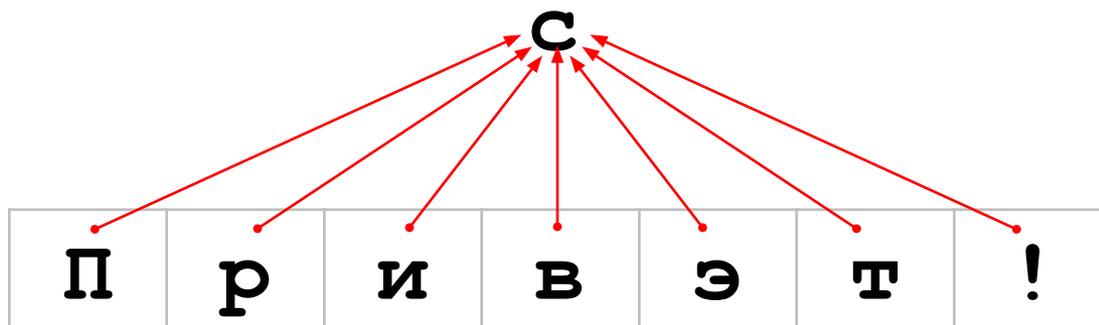
`len(s) - 1`

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| П | р | и | в | э | т | ! |

Цикл перебора символов

```
sNew = ""  
for c in s:  
    if c == "э":  
        sNew += "е"  
    else:  
        sNew += c
```

перебрать
все СИМВОЛЫ
строки



Задачи

«А»: Напишите программу, которая вводит строку, состоящую только из точек и букв X, и заменяет в ней все точки на нули и все буквы X на единицы.

Пример:

Введите строку: **..X.XX.**

Двоичный код: 0010110

«В»: Напишите программу, которая в символьной строке заменяет все нули на единицы и наоборот. Остальные символы не должны измениться.

Пример:

Введите строку: **10a01Vx1010c**

Инверсия: 01a10Vx0101c

Задачи

«С»: Введите битовую строку и дополните её последним битом, который должен быть равен 0, если в исходной строке чётное число единиц, и равен 1, если нечётное (в получившейся строке должно всегда быть чётное число единиц).

Пример:

Введите битовую строку: **01101010110**

Результат: **011010101100**

Операции со строками

Объединение (конкатенация) :

```
s1 = "Привет"  
s2 = "Вася"  
s  = s1 + ", " + s2 + "!"
```

"Привет, Вася!"

Умножение:

```
s = "Ау"  
s5 = s*5
```

s5 = s + s + s + s + s

АУАУАУАУАУ

? Что получим?

Срезы строк (выделение части строки)

```
s = "0123456789"  
s1 = s[3:8]      # "34567"
```

с какого
символа

до какого
(не включая 8)

```
s = "0123456789"  
s1 = s[:8]      # "01234567"
```

от начала строки

```
s = "0123456789"  
s1 = s[3:]      # "3456789"
```

до конца строки

Срезы строк

Срезы с отрицательными индексами:

```
s = "0123456789"  
s1 = s[:-2] # "01234567"
```

`len(s) - 2`

```
s = "0123456789"  
s1 = s[-6:-2] # "4567"
```

`len(s) - 6`

`len(s) - 2`

Операции со строками

Удаление:

```
s = "0123456789"
```

```
s1 = s[:3] + s[9:]
```

"012"

"9"

"0129"

Вставка:

```
s = "0123456789"
```

```
s1 = s[:3] + "ABC" + s[3:]
```

"012"

"3456789"

"012ABC3456789"

Поиск в строках

```
s = "Здесь был Вася."  
n = s.find ( "с" )      # n = 3  
if n >= 0:  
    print ( "Номер символа", n )  
else:  
    print ( "Символ не найден." )
```



Находит первое слева вхождение подстроки!

Поиск с конца строки:

```
s = "Здесь был Вася."  
n = s.rfind ( "с" )     # n = 12
```

Задачи

«А»: Ввести с клавиатуры в одну строку фамилию и имя, разделив их пробелом. Вывести первую букву имени с точкой и потом фамилию.

Пример:

Введите фамилию, имя и отчество:

Иванов Петр

П. Иванов

«В»: Ввести с клавиатуры в одну строку фамилию, имя и отчество, разделив их пробелом. Вывести фамилию и инициалы.

Пример:

Введите фамилию, имя и отчество:

Иванов Петр Семёнович

П.С. Иванов

Задачи

«С»: Ввести адрес файла и «разобрать» его на части, разделенные знаком " / ". Каждую часть вывести в отдельной строке.

Пример:

Введите адрес файла:

C: /фото/2015/Байкал/shaman.jpg

C:

фото

2015

Байкал

shaman.jpg

Преобразования «строка» → «число»

Из строки в число:

```
s = "123"  
N = int ( s )          # N = 123  
s = "123.456"  
X = float ( s )       # X = 123.456
```

Из числа в строку:

```
N = 123  
s = str ( N )         # s = "123"  
s = "{:5d}".format(N) # s = " 123"  
  
X = 123.456  
s = str ( X )        # s = "123.456"  
s = "{:7.2f}".format(X) # s = " 123.46"  
s = "{:10.2e}".format(X) # s = " 1.23e+02"
```

Задачи

«А»: Напишите программу, которая вычисляет сумму двух чисел, введенную в форме символьной строки. Все числа целые.

Пример:

Введите выражение :

12+3

Ответ: 15

«В»: Напишите программу, которая вычисляет сумму трёх чисел, введенную в форме символьной строки. Все числа целые.

Пример:

Введите выражение :

12+3+45

Ответ: 60

Задачи

«С»: Напишите программу, которая вычисляет сумму произвольного количества чисел, введенную в форме символьной строки. Все числа целые.

Пример:

Введите выражение :

12+3+45+10

Ответ: 70

«D»: Напишите программу, которая вычисляет выражение, содержащее целые числа и знаки сложения и вычитания.

Пример:

Введите выражение :

12+134-45-17

Ответ: 84