

Программирование на языке Python

§ 66. Символьные строки

Символьные строки

Начальное значение:

```
s = "Привет!"
```



Строка – это последовательность символов!

Вывод на экран:

```
print ( s )
```

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

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

0	1	2	3	4	5	6
П	р	и	в	е	т	!
s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]

s[len(s)-2]

Длина строки:

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

Символьные строки

Ввод с клавиатуры:

```
s = input ( "Введите имя: " )
```

Изменение строки:

```
s[4] = "a"
```



Строка – это неизменяемый объект!

... но можно составить новую строку:

```
s1 = s + "a"
```

Символьные строки

Задача: заменить в строке все буквы "а" на буквы "б".

```
s = input ( "Введите строку: " )  
s1 = ""      # строка-результат  
for c in s:  
    if c == "а":  
        c = "б"  
        s1 = s1 + c  
print ( s1 )
```

перебрать все
символы в строке

добавить символ к
строке-результату

Операции со строками

Объединение (конкатенация) :

```
s1 = "Привет"
```

```
s2 = "Вася"
```

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

"Привет, Вася!"

Срезы:

```
s = "0123456789"
```

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

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

разрезы

0 1 2 3 4 5 6 7 8 9

Операции со строками

Срезы:

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

от начала строки

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

до конца строки

```
s1 = s[::-1] # "9876543210"
```

реверс строки

Операции со строками

Срезы с отрицательными индексами:

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

N-2

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

N-6

N-2

Операции со строками

Удаление:

```
s = "0123456789"  
s1 = s[:3] + s[9:] # "0129"  
      "012"      "9"
```

Вставка:

```
s = "0123456789"  
s1 = s[:3] + "ABC" + s[3:]  
      "012ABC3456789"
```

Стандартные функции

Верхний/нижний регистр:

```
s = "aAbVcC"  
s1 = s.upper()    # "AABVCC"  
s2 = s.lower()    # "aabbcc"
```

Проверка на цифры:

```
s = "abc"  
print ( s.isdigit() )    # False  
s1 = "123"  
print ( s1.isdigit() )    # True
```

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
```

Пример обработки строк

Задача: Ввести имя, отчество и фамилию. Преобразовать их к формату «фамилия-инициалы».

Пример:

Введите имя, отчество и фамилию:

Василий Алибабаевич Хрюндиков

Результат:

Хрюндиков В.А.

Алибабаевич Хрюндиков

Алгоритм:

- найти первый пробел и выделить имя
- удалить имя с пробелом из основной строки
- найти первый пробел и выделить отчество
- удалить отчество с пробелом из основной строки
- «сцепить» фамилию, первые буквы имени и фамилии, точки, пробелы...

Хрюндиков

Хрюндиков В.А.

Пример обработки строк

```
print ( "Введите имя, отчество и фамилию:" )
s = input ()
n = s.find ( " " )
name = s[:n]      # вырезать имя
s = s[n+1:]
n = s.find ( " " )
name2 = s[:n]     # вырезать отчество
s = s[n+1:]      # осталась фамилия
s = s + " " + name[0] + "." + name2[0] + "."
print ( s )
```

Пример обработки строк

Решение в стиле Python:

```
print ( "Введите имя, отчество и фамилию:" )
s = input ()
fio = s.split ()
s = fio[2] + " " + fio[0][0] + "." + fio[1][0] + "."
print ( s )
```

Василий	Алибабаевич	Хрюндиков
fio[0]	fio[1]	fio[2]

Замена всех экземпляров подстроки

Встроенная функция:

```
s = "12.12.12"  
s = s.replace( "12", "A12B" )  
print ( s )
```

Сравнение строк

Пар ? пар ? парк

Сравнение по кодам символов:

	0	1	...	8	9
CP-1251	48	49	...	56	57
UNICODE	48	49	...	56	57
	A	B	...	Y	Z
CP-1251	65	66	...	89	90
UNICODE	65	66	...	89	90
	a	b	...	y	z
CP-1251	97	98	...	121	122
UNICODE	97	98	...	121	122

Сравнение строк

	А	Б	...	Ё	...	Ю	Я
CP-1251	192	193	...	168	...	222	223
UNICODE	1040	1041	...	1025	...	1070	1071

	а	б	...	ё	...	ю	я
CP-1251	224	225	...	184	...	254	255
UNICODE	1072	1073	...	1105	...	1102	1103

5STEAM < STEAM < Steam < steam

steam < ПАР < Пар < пАр < пар < парк

Основные методы строк

Операция	Описание	Пример
<code>s2 in s</code>	Проверка, что подстрока <code>s2</code> содержится в <code>s</code>	<code>'m' in 'team'</code>
<code>s2 not in s</code>	Проверка, что подстрока <code>s2</code> не содержится в <code>s</code> то же, что <code>not (s2 in s)</code>	<code>'l' not in 'team'</code>
<code>s + s2</code>	Конкатенация (склейка) строк, то есть строка, в которой сначала идут все символы из <code>s</code> , а затем все символы из <code>s2</code>	<code>'tea' + 'm' == 'team'</code>
<code>s * k</code>	Строка <code>s</code> , повторенная <code>k</code> раз	<code>'ha' * 3 == 'hahaha'</code>
<code>s[n]</code>	<code>n</code> -й элемент строки, отрицательные <code>n</code> — для отсчета с конца	<code>'team'[2] == 'a'</code> <code>'team'[-1] == 'm'</code>
<code>s[start:stop:step]</code>	Срез строки	<code>'mama'[:2] == 'ma'</code>
<code>s2 in s</code>	Проверка, что подстрока <code>s2</code> содержится в <code>s</code>	<code>'m' in 'team'</code>

Основные методы строк

Операция	Описание	Пример
<code>s.find(s2)</code> <code>s.rfind(s2)</code>	Индекс начала первого или последнего вхождения подстроки <code>s2</code> в <code>s</code> (вернет <code>-1</code> , если <code>s2 not in s</code>)	<code>s = 'abracadabra'</code> <code>s.find('ab') == 0</code> <code>s.rfind('ab') == 7</code> <code>s.find('x') == -1</code>
<code>s.count(s2)</code>	Количество неперекрывающихся вхождений <code>s2</code> в <code>s</code>	<code>'abracadabra'.count('a') == 5</code>
<code>s.startswith(s2)</code> <code>s.endswith(s2)</code>	Проверка, что <code>s</code> начинается с <code>s2</code> или оканчивается на <code>s2</code>	<code>'abracadabra'.startswith('abra')</code>
<code>s += s2</code> <code>s *= k</code>	Заменить содержимое строки на <code>s + s2</code> и <code>s * k</code> соответственно	
<code>s.isdigit()</code> <code>s.isalpha()</code> <code>s.isalnum()</code>	Проверка, что в строке <code>s</code> все символы — цифры, буквы (включая кириллические), цифры или буквы соответственно	<code>'100'.isdigit()</code> <code>'abc'.isalpha()</code> <code>'E315'.isalnum()</code>

Основные методы строк

Операция	Описание	Пример
<code>s.islower()</code> <code>s.isupper()</code>	Проверка, что в строке <code>s</code> не встречаются большие буквы, маленькие буквы. Обратите внимание, что для обеих этих функций знаки препинания и цифры дают <code>True</code>	<code>'hello!'.islower()</code> <code>'123PYTHON'.isupper()</code>
<code>s.lower()</code> <code>s.upper()</code>	Строка <code>s</code> , в которой все буквы (включая кириллические) приведены к верхнему или нижнему регистру, т. е. заменены на строчные (маленькие) или заглавные (большие)	<code>'Привет!'.lower() == 'привет!'</code> <code>'Привет!'.upper() == 'ПРИВЕТ!'</code>
<code>s.capitalize()</code>	Строка <code>s</code> , в которой первая буква — заглавная	<code>'привет'.capitalize() == 'Привет'</code>

Основные методы строк

Операция	Описание	Пример
<code>s.lstrip()</code> <code>s.rstrip()</code> <code>s.strip()</code>	Строка <code>s</code> , у которой удалены символы пустого пространства (пробелы, табуляции) в начале, в конце или с обеих сторон	<code>'Привет!'.strip() == 'Привет!'</code>
<code>s.ljust(k, c)</code> <code>s.rjust(k, c)</code>	Добавляет справа или слева нужное количество символов <code>c</code> , чтобы длина <code>s</code> достигла <code>k</code>	<code>'Привет'.ljust(8, '!') == 'Привет!!'</code>
<code>s.join(a)</code>	Склеивает строки из списка <code>a</code> через символ <code>s</code>	<code>'+'.join(['Вася', 'Маша']) == 'Вася+Маша'</code>
<code>s.split(s2)</code>	Список всех слов строки <code>s</code> (подстроки, разделенных строками <code>s2</code>)	<code>'Раз два три!'.split('а') == ['Р', 'з дв', ' три!']</code>

Основные методы строк

Операция	Описание	Пример
<code>s.replace(s2, s3)</code>	Строка <code>s</code> , в которой все неперекрывающиеся вхождения <code>s2</code> заменены на <code>s3</code> Есть необязательный третий параметр, с помощью которого можно указать, сколько раз производить замену	<code>'Раз два три!'.replace('а', 'я')</code> <code>== 'Ряз два три!'</code> <code>'Раз два три!'.replace('а', 'я', 1)</code> <code>== 'Ряз два три!'</code>
<code>list(s)</code>	Список символов из строки строки <code>s</code>	<code>list('Привет') == ['П', 'р', 'и', 'в', 'е', 'т']</code>
<code>bool(s)</code>	Проверка, что строка не пустая (возвращает <code>True</code> , если не пустая, и <code>False</code> в противном случае)	
<code>int(s)</code> <code>float(s)</code>	Если в строке <code>s</code> записано целое (дробное) число, получить это число, иначе — ошибка	<code>int('25') == 25</code>
<code>str(x)</code>	Представить любой объект <code>x</code> в виде строки	<code>str(25) == '25'</code>