

Программирование на языке Python

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"
```

Операции со строками

Объединение (конкатенация) :

```
s1 = "Привет"
```

```
s2 = "Вася"
```

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

"Привет, Вася!"

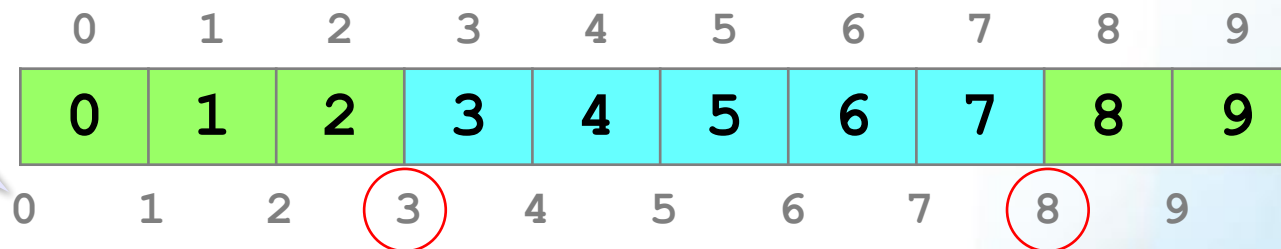
Срезы:

```
s = "0123456789"
```

```
s1 = s[3:8]
```

"34567"

срезы



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

Проверка на цифры:

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

... и много других.



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
```