

## Встроенные функции строк в python

Python предоставляет множество функций, которые встроены в интерпретатор. Вот несколько, которые работают со строками:

Функция	Описание
<code>chr()</code>	Преобразует целое число в символ
<code>ord()</code>	Преобразует символ в целое число
<code>len()</code>	Возвращает длину строки
<code>str()</code>	Изменяет тип объекта на <code>string</code>

Более подробно о них ниже.

**Функция `ord(c)`** возвращает числовое значение для заданного символа.

На базовом уровне компьютеры хранят всю информацию в виде цифр. Для представления символьных данных используется схема перевода, которая содержит каждый символ с его репрезентативным номером.

Самая простая схема в повседневном использовании называется [ASCII](#). Она охватывает латинские символы, с которыми мы чаще работаем. Для этих символов `ord(c)` возвращает значение ASCII для символа `c`:

```
>>> ord('a')
```

```
97
```

```
>>> ord('#')
```

```
35
```

ASCII прекрасен, но есть много других языков в мире, которые часто встречаются. Полный набор символов, которые потенциально могут быть представлены в коде, намного больше обычных латинских букв, цифр и символом.

**Unicode** — это современный стандарт, который пытается предоставить числовой код для всех возможных символов, на всех возможных языках, на каждой возможной платформе. Python 3 поддерживает Unicode, в том числе позволяет использовать символы Unicode в строках.

Функция `ord()` также возвращает числовые значения для символов Юникода:

```
>>> ord('€')
```

```
8364
```

```
>>> ord('Σ')
```

```
8721
```

Функция `chr(n)` возвращает символьное значение для данного целого числа.

`chr()` действует обратно `ord()`. Если задано числовое значение `n`, `chr(n)` возвращает строку, представляющую символ `n`:

```
>>> chr(97)
```

```
'a'
```

```
>>> chr(35)
```

```
'#'
```

`chr()` также обрабатывает символы Юникода:

```
>>> chr(8364)
```

```
'€'
```

```
>>> chr(8721)
```

```
'Σ'
```

Функция `len(s)` возвращает длину строки.

`len(s)` возвращает количество символов в строке `s`:

```
>>> s = 'Простая строка.'
```

```
>>> len(s)
```

Функция `str(obj)` возвращает строковое представление объекта.

Практически любой объект в Python может быть представлен как строка. `str(obj)` возвращает строковое представление объекта `obj`:

```
>>> str(49.2)
```

```
'49.2'
```

```
>>> str(3+4j)
```

```
'(3+4j)'
```

```
>>> str(3 + 29)
```

```
'32'
```

```
>>> str('py')
```

```
'py'
```