

ХОЧУ ПОМОЧЬ
ПРОЕКТУ

Модуль binascii в Python

Яндекс Взгляд · Опрос

Выберите 1 или несколько ответов

Какие сервисы проверки истории автомобилей вы знаете?

Avtocod

Автотека/Авито

Ни один из вариантов

ПроАвто/Auto.ru

Avinfobot

1 из 3 вопросов

Продолжить

[Стандартная библиотека Python3.](#) / Модуль binascii в Python

Преобразование между двоичным кодом и ASCII

Модуль `binascii` содержит ряд методов для преобразования между двоичными и различными двоичными представлениями в кодировке ASCII. Обычно эти функции не используются напрямую, а вместо них используются модули-оболочки, такие как `uu`, `base64` или `binhex`. Модуль `binascii` содержит низкоуровневые функции, написанные на C для большей скорости, для использования модулями более высокого уровня.

Обратите внимание, что функции `a2b_*` принимают строки Unicode, содержащие только символы ASCII. Другие функции принимают только объекты, подобные байтам (например, `байты`, `массив байтов` и другие объекты, поддерживающие протокол буфера).

Модуль binascii определяет следующие функции:

- Функция `binascii.a2b_uu()`,
- Функция `binascii.b2a_uu()`,
- Функция `binascii.a2b_base64()`,
- Функция `binascii.b2a_base64()`,
- Функция `binascii.a2b_qp()`,
- Функция `binascii.b2a_qp()`,
- Функция `binascii.a2b_hqx()`,
- Функция `binascii.rledecode_hqx()`,
- Функция `binascii.rlecode_hqx()`,
- Функция `binascii.b2a_hqx()`,
- Функция `binascii.crc_hqx()`,
- Функция `binascii.crc32()`,
- Функция `binascii.b2a_hex()`,
- Функция `binascii.hexlify()`,
- Функция `binascii.b2a_hex()`,
- Функция `binascii.hexlify()`,
- Функция `binascii.b2a_hex()`,
- Функция `binascii.b2a_hex()`,

Вверх

<https://docs-python.ru/standart-library/modul-binascii-python/>

1/4

- [Функция binascii.a2b_hex\(\)](#),
- [Функция binascii.unhexlify\(\)](#).

binascii.a2b_uu(string):

Функция `binascii.a2b_uu()` преобразует одну строку данных `string`, закодированных в формате uuencoded, обратно в двоичный формат и возвращает двоичные данные. Строки обычно содержат 45 (двоичных) байтов, за исключением последней строки. За строковыми данными может следовать пробел.

*binascii.b2a_uu(data, *, backtick=False):*

Функция `binascii.b2a_uu()` преобразует двоичные данные `data` в строку символов ASCII, возвращаемое значение - преобразованная строка, включая символ новой строки. Длина данных должна быть не более 45. Если аргумент `backtick=True`, то `'\n'` будет представлена нулями вместо пробелов.

Изменено в версии 3.7: Добавлен аргумент `backtick`.

binascii.a2b_base64(string):

Функция `binascii.a2b_base64()` преобразует блок данных, представленных как строка base64 обратно в двоичный формат, который и возвращает. За один раз можно пройти более одной строки.

*binascii.b2a_base64(data, *, newline=True):*

Функция `binascii.b2a_base64()` преобразует двоичные данные `data` в строку символов ASCII в кодировке base64. Возвращаемое значение представляет собой преобразованную строку, включая символ новой строки, если аргумент `newline=True`. Вывод этой функции соответствует RFC 3548.

Изменено в версии 3.6: Добавлен аргумент `newline`.

binascii.a2b_qp(data, header=False):

Функция `binascii.a2b_qp()` преобразует блок данных `data` для печати в кавычках - обратно в двоичные данные. За один раз можно пройти более одной строки. Если необязательный аргумент `header` присутствует и имеет значение `True`, то символы подчеркивания будут расшифрованы как пробелы.

binascii.b2a_qp(data, quotetabs=False, istext=True, header=False):

Функция `binascii.b2a_qp()` преобразует двоичные данные `data` в строку(и) символов ASCII в кодировке для печати в кавычках. Возвращаемое значение - преобразованная строка (строки).

- Если необязательный аргумент `quotetabs` присутствует и имеет значение `True`, все табуляции и пробелы будут закодированы.
- Если необязательный аргумент `istext` присутствует и имеет значение `True`, то новые строки не кодируются, но конечные пробелы будут кодироваться.
- Если необязательный аргумент `header` присутствует и имеет значение `True`, то пробелы будут кодироваться как символы подчеркивания в соответствии с RFC 1522. Если `header=False`, то символы новой строки также будут закодированы, в противном случае преобразование перевода строки `\n` может повредить поток двоичных данных.

binascii.a2b_hqx(string):

Функция `binascii.a2b_hqx()` преобразует данные ASCII в формате binhex4 в двоичный формат, не выполняя RLE-декомпрессию. Строка должна содержать полное количество двоичных байтов или (в случае последней части данных binhex4) содержать оставшиеся биты, равные нулю.

Функция устарела с версии Python 3.9.

binascii.rledecode_hqx(data):

Функция `binascii.rledecode_hqx()` выполняет RLE-декомпрессию данных в соответствии со стандартом binhex4. Алгоритм использует `0x90` после байта в качестве индикатора повторения, за которым следует счетчик. Значение `0` указывает байтовое значение `0x90`. Процедура возвращает распакованные данные, если только входные данные не заканчиваются индикатором потерянного повтора, и в этом случае возникает неполное исключение.

Функция устарела с версии Python 3.9.

binascii.rlecode_hqx(data):

Функция `binascii.rlecode_hqx()` выполняет RLE-сжатие данных в стиле binhex4.

Функция устарела с версии Python 3.9.

`binascii.b2a_hqx(data):`

Функция `binascii.b2a_hqx()` выполняет преобразование двоичного кода `hexbin4` в ASCII и возвращает результирующую строку. Аргумент уже должен быть закодирован RLE и иметь длину, кратную 3 (за исключением, возможно, последнего фрагмента).

Функция устарела с версии Python 3.9.

`binascii.crc_hqx(data, value):`

Функция `binascii.crc_hqx()` вычисляет 16-битное значение CRC данных, начиная со значения в качестве начального CRC. При этом используется многочлен CRC-CCITT $x^{16} + x^{12} + x^5 + 1$, часто представленный как `0x1021`. Этот CRC используется в формате `binhex4`.

`binascii.crc32(data[, value]):`

Функция `binascii.crc32()` вычисляет CRC-32, 32-разрядную контрольную сумму данных без знака, начиная с начального значения CRC. Начальный CRC по умолчанию равен нулю. Алгоритм согласуется с контрольной суммой ZIP-файла. Так как алгоритм предназначен для использования в качестве алгоритма контрольной суммы, он не подходит для использования в качестве общего хэш-алгоритма.

Используйте следующим образом:

```
>>> print(binascii.crc32(b"hello world"))
# Или в двух частях
>>> crc = binascii.crc32(b"hello")
>>> crc = binascii.crc32(b" world", crc)
>>> print('crc32 = {:#010x}'.format(crc))
```

`binascii.hexlify(data[, sep[, bytes_per_sep=1]])`

`binascii.b2a_hex(data[, sep[, bytes_per_sep=1]]):`

Функция `binascii.hexlify()` возвращает шестнадцатеричное представление двоичных данных. Каждый байт данных преобразуется в соответствующее 2-разрядное шестнадцатеричное представление. Таким образом, возвращаемый объект байтов в два раза превышает длину данных.

У функции `binascii.b2a_hex()` аналогичная функциональность (но возвращающая текстовую строку) также удобно доступна с помощью метода `bytes.hex()`.

Если указан `sep`, то это должен быть односимвольный объект `str` или `bytes`. Он будет вставлен в вывод после каждого входного байта `bytes_per_sep`. Размещение разделителя, по умолчанию, считается с правого конца вывода, если необходимо считать слева, то укажите отрицательное значение `bytes_per_sep`.

```
>>> import binascii
>>> binascii.b2a_hex(b'\xb9\x01\xef')
# b'b901ef'
>>> binascii.hexlify(b'\xb9\x01\xef', '-')
# b'b9-01-ef'
>>> binascii.b2a_hex(b'\xb9\x01\xef', b'_', 2)
# b'b9_01ef'
>>> binascii.b2a_hex(b'\xb9\x01\xef', b' ', -2)
# b'b901 ef'
```

`binascii.unhexlify(hexstr)`

`binascii.a2b_hex(hexstr):`

Функция `binascii.unhexlify()` возвращает двоичные данные, представленные шестнадцатеричной строкой `hexstr`. Эта функция обратна `binascii.b2a_hex()`. Аргумент `hexstr` должен содержать четное количество шестнадцатеричных цифр (которые могут быть в верхнем или нижнем регистре), в противном случае возникает [исключение binascii.Error](#).

У функции `binascii.a2b_hex()` аналогичная функциональность (принимаящая только аргумент текстовой строки `hexstr`, но более либеральная по отношению к пробелам) также доступна с использованием метода класса `bytes.fromhex()`.

`binascii.Error:`

Исключение `binascii.Error` возникает при ошибках. Обычно это ошибки программирования.

[Вверх](#) `binascii.Incomplete:`

Исключение `binascii.Incomplete` поднимается на неполных данных. Обычно это не ошибки программирования, но их можно устранить, прочитав немного больше данных и повторив попытку.

