:mod:`binascii` --- Convert between binary and ASCII

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst`, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst`, line 4)

Unknown directive type "module".

```
.. module:: binascii
   :synopsis: Tools for converting between binary and various ASCII-encoded binary
              representations.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst`, line 8)

Unknown directive type "index".

```
.. index::
   module: uu
   module: base64
```

The :mod:`binascii` module contains a number of methods to convert between binary and various ASCII-encoded binary representations. Normally, you will not use these functions directly but use wrapper modules like :mod:`uu` or :mod:`base64` instead. The :mod:`binascii` module contains low-level functions written in C for greater speed that are used by the higher-level modules.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst`, line 14); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst`, line 14); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst`, line 14); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst`, line 14); *backlink*

Unknown interpreted text role "mod".

Note

`a2b_*` functions accept Unicode strings containing only ASCII characters. Other functions only accept :term:`bytes-like objects <bytes-like object>` (such as :class:`bytes`, :class:`bytearray` and other objects that support the buffer protocol).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst`, line 23); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 23); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 23); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 28)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.3
   ASCII-only unicode strings are now accepted by the ``a2b_*`` functions.
```

The :mod:`binascii` module defines the following functions:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 32); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 35)

Unknown directive type "function".

```
.. function:: a2b_uu(string)

   Convert a single line of uuencoded data back to binary and return the binary
   data. Lines normally contain 45 (binary) bytes, except for the last line. Line
   data may be followed by whitespace.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 42)

Unknown directive type "function".

```
.. function:: b2a_uu(data, *, backtick=False)

   Convert binary data to a line of ASCII characters, the return value is the
   converted line, including a newline char. The length of *data* should be at most
   45. If *backtick* is true, zeros are represented by ``'`'`` instead of spaces.

   .. versionchanged:: 3.7
      Added the *backtick* parameter.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 52)

Unknown directive type "function".

```
.. function:: a2b_base64(string, strict_mode=False)

   Convert a block of base64 data back to binary and return the binary data. More
   than one line may be passed at a time.

   If *strict_mode* is true, only valid base64 data will be converted. Invalid base64
   data will raise :exc:`binascii.Error`.

   Valid base64:
      * Conforms to :rfc:`3548`.
```

```
      * Contains only characters from the base64 alphabet.
      * Contains no excess data after padding (including excess padding, newlines, etc.).
      * Does not start with a padding.

   .. versionchanged:: 3.11
      Added the *strict_mode* parameter.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 70)

Unknown directive type "function".

```
.. function:: b2a_base64(data, *, newline=True)

   Convert binary data to a line of ASCII characters in base64 coding. The return
   value is the converted line, including a newline char if *newline* is
   true.  The output of this function conforms to :rfc:`3548`.

   .. versionchanged:: 3.6
      Added the *newline* parameter.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 80)

Unknown directive type "function".

```
.. function:: a2b_qp(data, header=False)

   Convert a block of quoted-printable data back to binary and return the binary
   data. More than one line may be passed at a time. If the optional argument
   *header* is present and true, underscores will be decoded as spaces.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 87)

Unknown directive type "function".

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

   Convert binary data to a line(s) of ASCII characters in quoted-printable
   encoding.  The return value is the converted line(s). If the optional argument
   *quotetabs* is present and true, all tabs and spaces will be encoded.   If the
   optional argument *istext* is present and true, newlines are not encoded but
   trailing whitespace will be encoded. If the optional argument *header* is
   present and true, spaces will be encoded as underscores per :rfc:`1522`. If the
   optional argument *header* is present and false, newline characters will be
   encoded as well; otherwise linefeed conversion might corrupt the binary data
   stream.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 100)

Unknown directive type "function".

```
.. function:: crc_hqx(data, value)

   Compute a 16-bit CRC value of *data*, starting with *value* as the
   initial CRC, and return the result.  This uses the CRC-CCITT polynomial
   *x*:sup:`16` + *x*:sup:`12` + *x*:sup:`5` + 1, often represented as
   0x1021.  This CRC is used in the binhex4 format.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 108)

Unknown directive type "function".

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

   Compute CRC-32, the unsigned 32-bit checksum of *data*, starting with an
   initial CRC of *value*.  The default initial CRC is zero.  The algorithm
   is consistent with the ZIP file checksum.  Since the algorithm is designed for
   use as a checksum algorithm, it is not suitable for use as a general hash
   algorithm.  Use as follows::

      print(binascii.crc32(b"hello world"))
      # Or, in two pieces:
      crc = binascii.crc32(b"hello")
      crc = binascii.crc32(b" world", crc)
      print('crc32 = {:#010x}'.format(crc))

   .. versionchanged:: 3.0
      The result is always unsigned.
      To generate the same numeric value when using Python 2 or earlier,
      use ``crc32(data) & 0xffffffff``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 127)

Unknown directive type "function".

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

   Return the hexadecimal representation of the binary *data*.  Every byte of
   *data* is converted into the corresponding 2-digit hex representation.  The
   returned bytes object is therefore twice as long as the length of *data*.

   Similar functionality (but returning a text string) is also conveniently
   accessible using the :meth:`bytes.hex` method.

   If *sep* is specified, it must be a single character str or bytes object.
   It will be inserted in the output after every *bytes_per_sep* input bytes.
   Separator placement is counted from the right end of the output by default,
   if you wish to count from the left, supply a negative *bytes_per_sep* value.

      >>> 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'

   .. versionchanged:: 3.8
      The *sep* and *bytes_per_sep* parameters were added.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 155)

Unknown directive type "function".

```
.. function:: a2b_hex(hexstr)
              unhexlify(hexstr)

   Return the binary data represented by the hexadecimal string *hexstr*.  This
   function is the inverse of :func:`b2a_hex`. *hexstr* must contain an even number
   of hexadecimal digits (which can be upper or lower case), otherwise an
   :exc:`Error` exception is raised.

   Similar functionality (accepting only text string arguments, but more
   liberal towards whitespace) is also accessible using the
   :meth:`bytes.fromhex` class method.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 167)

Unknown directive type "exception".

```
.. exception:: Error
```

```
   Exception raised on errors. These are usually programming errors.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 172)

Unknown directive type "exception".

```
.. exception:: Incomplete

   Exception raised on incomplete data. These are usually not programming errors,
   but may be handled by reading a little more data and trying again.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]binascii.rst, line 178)

Unknown directive type "seealso".

```
.. seealso::

   Module :mod:`base64`
      Support for RFC compliant base64-style encoding in base 16, 32, 64,
      and 85.

   Module :mod:`uu`
      Support for UU encoding used on Unix.

   Module :mod:`quopri`
      Support for quoted-printable encoding used in MIME email messages.
```