# :mod:`base64` --- Base16, Base32, Base64, Base85 Data Encodings

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.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)base64.rst`, line 4)

Unknown directive type "module".

```
.. module:: base64
   :synopsis: RFC 4648: Base16, Base32, Base64 Data Encodings;
              Base85 and Ascii85
```

**Source code:** :source:`Lib/base64.py`

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 8); *backlink*

Unknown interpreted text role "source".

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 10)

Unknown directive type "index".

```
.. index::
   pair: base64; encoding
   single: MIME; base64 encoding
```

---

This module provides functions for encoding binary data to printable ASCII characters and decoding such encodings back to binary data. It provides encoding and decoding functions for the encodings specified in RFC 4648, which defines the Base16, Base32, and Base64 algorithms, and for the de-facto standard Ascii85 and Base85 encodings.

The RFC 4648 encodings are suitable for encoding binary data so that it can be safely sent by email, used as parts of URLs, or included as part of an HTTP POST request. The encoding algorithm is not the same as the :program:`uuencode` program.

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 22); *backlink*

Unknown interpreted text role "program".

There are two interfaces provided by this module. The modern interface supports encoding :term:`bytes-like objects <bytes-like object>` to ASCII :class:`bytes`, and decoding :term:`bytes-like objects <bytes-like object>` or strings containing ASCII to :class:`bytes`. Both base-64 alphabets defined in RFC 4648 (normal, and URL- and filesystem-safe) are supported.

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 27); *backlink*

Unknown interpreted text role "term".

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 27); *backlink*

Unknown interpreted text role "class".

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 27); *backlink*

Unknown interpreted text role "term".

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 27); *backlink*

Unknown interpreted text role "class".

The legacy interface does not support decoding from strings, but it does provide functions for encoding and decoding to and from :term:`file objects <file object>`. It only supports the Base64 standard alphabet, and it adds newlines every 76 characters as per RFC 2045. Note that if you are looking for RFC 2045 support you probably want to be looking at the :mod:`email` package instead.

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 33); *backlink*

Unknown interpreted text role "term".

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 33); *backlink*

Unknown interpreted text role "mod".

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 41)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.3
   ASCII-only Unicode strings are now accepted by the decoding functions of
   the modern interface.
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst, line 45)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.4
   Any :term:`bytes-like objects <bytes-like object>` are now accepted by all
   encoding and decoding functions in this module.  Ascii85/Base85 support added.
```

The modern interface provides:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst, line 51)

Unknown directive type "function".

```
.. function:: b64encode(s, altchars=None)

   Encode the :term:`bytes-like object` *s* using Base64 and return the encoded
   :class:`bytes`.

   Optional *altchars* must be a :term:`bytes-like object` of at least
   length 2 (additional characters are ignored) which specifies an alternative
   alphabet for the ``+`` and ``/`` characters.  This allows an application to e.g.
   generate URL or filesystem safe Base64 strings.  The default is ``None``, for
   which the standard Base64 alphabet is used.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst, line 63)

Unknown directive type "function".

```
.. function:: b64decode(s, altchars=None, validate=False)

   Decode the Base64 encoded :term:`bytes-like object` or ASCII string
   *s* and return the decoded :class:`bytes`.

   Optional *altchars* must be a :term:`bytes-like object` or ASCII string of
   at least length 2 (additional characters are ignored) which specifies the
   alternative alphabet used instead of the ``+`` and ``/`` characters.

   A :exc:`binascii.Error` exception is raised
   if *s* is incorrectly padded.

   If *validate* is ``False`` (the default), characters that are neither
   in the normal base-64 alphabet nor the alternative alphabet are
   discarded prior to the padding check.  If *validate* is ``True``,
   these non-alphabet characters in the input result in a
   :exc:`binascii.Error`.

   For more information about the strict base64 check, see :func:`binascii.a2b_base64`
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst, line 84)

Unknown directive type "function".

```
.. function:: standard_b64encode(s)

   Encode :term:`bytes-like object` *s* using the standard Base64 alphabet
   and return the encoded :class:`bytes`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst, line 90)

Unknown directive type "function".

```
.. function:: standard_b64decode(s)

   Decode :term:`bytes-like object` or ASCII string *s* using the standard
   Base64 alphabet and return the decoded :class:`bytes`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst, line 96)

Unknown directive type "function".

```
.. function:: urlsafe_b64encode(s)

   Encode :term:`bytes-like object` *s* using the
   URL- and filesystem-safe alphabet, which
   substitutes ``-`` instead of ``+`` and ``_`` instead of ``/`` in the
   standard Base64 alphabet, and return the encoded :class:`bytes`.  The result
   can still contain ``=``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst, line 105)

Unknown directive type "function".

```
.. function:: urlsafe_b64decode(s)

   Decode :term:`bytes-like object` or ASCII string *s*
   using the URL- and filesystem-safe
   alphabet, which substitutes ``-`` instead of ``+`` and ``_`` instead of
   ``/`` in the standard Base64 alphabet, and return the decoded
   :class:`bytes`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst, line 114)

Unknown directive type "function".

```
.. function:: b32encode(s)

   Encode the :term:`bytes-like object` *s* using Base32 and return the
   encoded :class:`bytes`.
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 120)

Unknown directive type "function".

```
.. function:: b32decode(s, casefold=False, map01=None)

   Decode the Base32 encoded :term:`bytes-like object` or ASCII string *s* and
   return the decoded :class:`bytes`.

   Optional *casefold* is a flag specifying
   whether a lowercase alphabet is acceptable as input.  For security purposes,
   the default is ``False``.

   :rfc:`4648` allows for optional mapping of the digit 0 (zero) to the letter O
   (oh), and for optional mapping of the digit 1 (one) to either the letter I (eye)
   or letter L (el).  The optional argument *map01* when not ``None``, specifies
   which letter the digit 1 should be mapped to (when *map01* is not ``None``, the
   digit 0 is always mapped to the letter O).  For security purposes the default is
   ``None``, so that 0 and 1 are not allowed in the input.

   A :exc:`binascii.Error` is raised if *s* is
   incorrectly padded or if there are non-alphabet characters present in the
   input.
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 141)

Unknown directive type "function".

```
.. function:: b32hexencode(s)

   Similar to :func:`b32encode` but uses the Extended Hex Alphabet, as defined in
   :rfc:`4648`.

   .. versionadded:: 3.10
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 149)

Unknown directive type "function".

```
.. function:: b32hexdecode(s, casefold=False)

   Similar to :func:`b32decode` but uses the Extended Hex Alphabet, as defined in
   :rfc:`4648`.

   This version does not allow the digit 0 (zero) to the letter O (oh) and digit
   1 (one) to either the letter I (eye) or letter L (el) mappings, all these
   characters are included in the Extended Hex Alphabet and are not
   interchangeable.

   .. versionadded:: 3.10
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 162)

Unknown directive type "function".

```
.. function:: b16encode(s)

   Encode the :term:`bytes-like object` *s* using Base16 and return the
   encoded :class:`bytes`.
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 168)

Unknown directive type "function".

```
.. function:: b16decode(s, casefold=False)

   Decode the Base16 encoded :term:`bytes-like object` or ASCII string *s* and
   return the decoded :class:`bytes`.

   Optional *casefold* is a flag specifying whether a
   lowercase alphabet is acceptable as input.  For security purposes, the default
   is ``False``.

   A :exc:`binascii.Error` is raised if *s* is
   incorrectly padded or if there are non-alphabet characters present in the
   input.
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 182)

Unknown directive type "function".

```
.. function:: a85encode(b, *, foldspaces=False, wrapcol=0, pad=False, adobe=False)

   Encode the :term:`bytes-like object` *b* using Ascii85 and return the
   encoded :class:`bytes`.

   *foldspaces* is an optional flag that uses the special short sequence 'y'
   instead of 4 consecutive spaces (ASCII 0x20) as supported by 'btoa'. This
   feature is not supported by the "standard" Ascii85 encoding.
   *wrapcol* controls whether the output should have newline (``b'\n'``)
   characters added to it. If this is non-zero, each output line will be
   at most this many characters long.

   *pad* controls whether the input is padded to a multiple of 4
   before encoding. Note that the ``btoa`` implementation always pads.

   *adobe* controls whether the encoded byte sequence is framed with ``<~``
   and ``~>``, which is used by the Adobe implementation.

   .. versionadded:: 3.4
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 204)

Unknown directive type "function".

```
.. function:: a85decode(b, *, foldspaces=False, adobe=False, ignorechars=b' \\t\\n\\r\\v')

   Decode the Ascii85 encoded :term:`bytes-like object` or ASCII string *b* and
   return the decoded :class:`bytes`.

   *foldspaces* is a flag that specifies whether the 'y' short sequence
   should be accepted as shorthand for 4 consecutive spaces (ASCII 0x20).
   This feature is not supported by the "standard" Ascii85 encoding.

   *adobe* controls whether the input sequence is in Adobe Ascii85 format
   (i.e. is framed with <~ and ~>).

   *ignorechars* should be a :term:`bytes-like object` or ASCII string
   containing characters to ignore
   from the input. This should only contain whitespace characters, and by
   default contains all whitespace characters in ASCII.

   .. versionadded:: 3.4
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 224)

Unknown directive type "function".

```
.. function:: b85encode(b, pad=False)

   Encode the :term:`bytes-like object` *b* using base85 (as used in e.g.
   git-style binary diffs) and return the encoded :class:`bytes`.

   If *pad* is true, the input is padded with ``b'\0'`` so its length is a
   multiple of 4 bytes before encoding.

   .. versionadded:: 3.4
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 235)

Unknown directive type "function".

```
.. function:: b85decode(b)

   Decode the base85-encoded :term:`bytes-like object` or ASCII string *b* and
   return the decoded :class:`bytes`.  Padding is implicitly removed, if
   necessary.

   .. versionadded:: 3.4
```

The legacy interface:

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 246)

Unknown directive type "function".

```
.. function:: decode(input, output)

   Decode the contents of the binary *input* file and write the resulting binary
   data to the *output* file. *input* and *output* must be :term:`file objects
   <file object>`. *input* will be read until ``input.readline()`` returns an
   empty bytes object.
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 254)

Unknown directive type "function".

```
.. function:: decodebytes(s)

   Decode the :term:`bytes-like object` *s*, which must contain one or more
   lines of base64 encoded data, and return the decoded :class:`bytes`.

   .. versionadded:: 3.1
```

**System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 262)

Unknown directive type "function".

```
.. function:: encode(input, output)

   Encode the contents of the binary *input* file and write the resulting base64
   encoded data to the *output* file. *input* and *output* must be :term:`file
   objects <file object>`. *input* will be read until ``input.read()`` returns
   an empty bytes object. :func:`encode` inserts a newline character (``b'\n'``)
   after every 76 bytes of the output, as well as ensuring that the output
   always ends with a newline, as per :rfc:`2045` (MIME).
```

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 272)**

Unknown directive type "function".

```
.. function:: encodebytes(s)

   Encode the :term:`bytes-like object` *s*, which can contain arbitrary binary
   data, and return :class:`bytes` containing the base64-encoded data, with newlines
   (``b'\n'``) inserted after every 76 bytes of output, and ensuring that
   there is a trailing newline, as per :rfc:`2045` (MIME).

   .. versionadded:: 3.1
```

An example usage of the module:

```
>>> import base64
>>> encoded = base64.b64encode(b'data to be encoded')
>>> encoded
b'ZGF0YSB0byBiZSBlbmNvZGVk'
>>> data = base64.b64decode(encoded)
>>> data
b'data to be encoded'
```

## Security Considerations

A new security considerations section was added to RFC 4648 (section 12); it's recommended to review the security section for any code deployed to production.

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)base64.rst`, line 300)**

Unknown directive type "seealso".

```
.. seealso::

   Module :mod:`binascii`
      Support module containing ASCII-to-binary and binary-to-ASCII conversions.

   :rfc:`1521` - MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format
      Section 5.2, "Base64 Content-Transfer-Encoding," provides the definition of the
      base64 encoding.
```