

# python 模块介绍- base64 RFC 3548: Base16, Base32, Base64 数据编码

## 目录

项目简介.....	1
简介: .....	1
快速入门.....	2
Base64 编码解码.....	2
URL-Safe.....	4
其他编码.....	5

## 项目简介

Python 中文库 <https://bitbucket.org/xurongzhong/python-chinese-library> 主要基于个人的使用经验，收集一些重要的外部和内部模块的中文教程和实例。

发起人: [ouyangchongwu@gmail.com](mailto:ouyangchongwu@gmail.com), [xurongzhong@gmail.com](mailto:xurongzhong@gmail.com) 。 微博: <http://weibo.com/cizhenshi>

欢迎大家加入分享经验。另有交流 QQ 群 python 开发自动化测试 113938272, 软件测试进阶:6089740。

本文中的源代码地址: <https://bitbucket.org/xurongzhong/python-chinese-library/libraries/standard/base64>

文件下载:

- hg clone 克隆所有文件 hg clone <https://bitbucket.org/xurongzhong/python-chinese-library>。
- <https://bitbucket.org/xurongzhong/python-chinese-library/src> 浏览文件，右键点击文件，选另存为下载。
- <https://bitbucket.org/xurongzhong/python-chinese-library/downloads>。此目录不定时最新文档，但是收集了一些测试开发相关的书籍。

Bug 提交: <https://bitbucket.org/xurongzhong/python-chinese-library/issuest>。

版本管理

版本号	修订发布时间	修订人	备注
V1.0	2013-12-10	Ouyangchongwu#gmail.com	初始版本, 由《The Python Standard Library by Example 2011》和 <a href="http://docs.python.org/2/library/base64.html">http://docs.python.org/2/library/base64.html</a> 生成。
V1.1	2015-06-10	xurongzhong#gmail.com 徐荣中	python 代码染色; 结构调整

参考资料:

python2 官方网址: <http://docs.python.org/2/library/base64.html>

python3 官方网址: <https://docs.python.org/3/library/base64.html>

python 标准库 pymotw: <http://pymotw.com/2/base64/index.html#module-base64>

简介:

功能: RFC 3548: Base16, Base32, Base64 数据编码。转换二进制数据为适合明文协议传输的 ASCII 序列。转换 8bits 为每个字节包含 6, 5 或 4bits 的有效数据, 比如 SMTP, URL 的一部分或者 HTTP POST 的一部分。参考: [RFC 3548](#)。编码算法不同于 uuencode。

类型: 标准库

相关模块: uu, binhex, uu, quopri

Base64 是一种基于 64 个可打印字符来表示二进制数据的表示方法。由于 2 的 6 次方等于 64, 所以每 6 个位元为一个单元, 对应某个可打印字符。三个字节有 24 个位元, 对应于 4 个 Base64 单元, 即 3 个字节需要用 4 个可打印字符来表示。它可用来作为电子邮件的传输编码。在 Base64 中的可打印字符包括字母 A-Z、a-z、数字 0-9, 这样共有 62 个字符, 此外两个可打印符号在不同的系统中而不同。之后在 6 位的前面补两个 0, 形成 8 位一个字节的形势。一些如 uuencode 的其他编码方法, 和之后 binhex 的版本使用不同的 64 字符集来代表 6 个二进制数字, 但是它们不叫 Base64。

Base64 常用于在通常处理文本数据的场合, 表示、传输、存储一些二进制数据。包括 MIME 的 email, email via MIME, 在 XML 中存储复杂数据

Python Base64 模块提供了 RFC3548 中的数据编码和解码 (转换二进制数据为适合明文协议传输的 ASCII 序列, 如 RFC3548 中指定。该标准定义了 Base16, Base32 和 Base64 算法, 编码和解码的任意二进

制字符串转换为文本字符串，这样就可以通过电子邮件安全发送，作为网址的一部分，或包含在 HTTP POST 请求中。

**Base64** 模块提供两个接口。新式接口支持使用三个字母的编码和解码的字符串对象。传统接口提供了编码和解码文件对象和字符串，但只使用了标准的 **Base64** 字母。传统接口这里不做介绍。

`base64`、`base32`、`base16` 可以分别编码转化 8 位字节为 6 位、5 位、4 位。16,32,64 分别表示用多少个字符来编码。

更多 `base64` 的资料，参见

<http://zh.wikipedia.org/wiki/Base64>，<http://tools.ietf.org/html/rfc822>，<http://tools.ietf.org/html/rfc1421>，<http://tools.ietf.org/html/rfc2045>。

快速入门

请看 `python` 模块介绍中的实例：

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

- `base64.b64encode(s[, altchars])`: 使用 **Base64** 编码字符串。s 是要编码的字符串。altchars 是用来替换+和/的字符串，它们在 `url` 和文件系统中它们有特殊含义，通常需要替换。
- `base64.b64decode(s[, altchars])`: 解码 **Base64** 编码的字符串。s 为要解码的字符串。altchars 和 `b64encode` 相同。
  - `base64.standard_b64encode(s)`: 参考 `b64encode`。
  - `base64.standard_b64decode(s)`: 参考 `b64decode`。

## Base64 编码解码

```
#!/usr/bin/env python
# encoding: utf-8
#
# Copyright (c) 2008 Doug Hellmann All rights reserved.
#
''''
''''

__version__ = "$Id$"
#end_pymotw_header

import base64
import textwrap

# Load this source file and strip the header.
with open(__file__, 'rt') as input:
    raw = input.read()
    initial_data = raw.split('#end_pymotw_header')[1]

encoded_data = base64.b64encode(initial_data)

num_initial = len(initial_data)

# There will never be more than 2 padding bytes.
padding = 3 - (num_initial % 3)

print '%d bytes before encoding' % num_initial
print 'Expect %d padding bytes' % padding
print '%d bytes after encoding' % len(encoded_data)
print
print encoded_data
```

➤ 执行结果

```
$ python base64_b64encode.py
```

168 bytes before encoding

Expect 3 padding bytes

224 bytes after encoding

```
CgppbXBvcnQgYmFzZTY0CmltcG9ydCB0ZXh0d3JhcAoKIyBMb2FkIHRoaXMgc291cmNlIGZpbGUgYW5kIHN0cmlwIHRoZSBoZWFK
ZXIuCndpdGgg3BlbihfX2ZpbGVfYXwgJ3J0JykgYXMgaW5wdXQ6CiAgICByYXcgPSBpbmB1dC5yZWFKKCKKICAgIGluaXRpYWxfZGF0
YSA9IHJhdy5zcGxpCgn
```

Base64 编码的 4 个字节对应实际的 3 个字节，不足四个字节时，后面部分通常用等号填充。极端的情况下，一个字节需要用 4 个 Base64 编码来表示。

```
>>> import base64
>>> encoded = base64.b64encode('a')
>>> encoded
'YQ=='
```

Base64 解码参见快速入门部分介绍。

## URL-Safe

•base64.urlsafe\_b64encode(s):

•base64.urlsafe\_b64decode(s):

Base64 默认会使用+和/,但是这 2 个字符在 url 中也有特殊含义。使用 urlsafe 可以解决这个问题。+替换为-, /替换为\_。

```
import base64

encodes_with_pluses = chr(251) + chr(239)
encodes_with_slashes = chr(255) * 2

for original in [ encodes_with_pluses, encodes_with_slashes ]:
    print 'Original          :', repr(original)
    print 'Standard encoding:', base64.standard_b64encode(original)
    print 'URL-safe encoding:', base64.urlsafe_b64encode(original)
    print
```

➤ 执行结果

```
$ python base64_urlsafef.py
Original      : '\xfb\xef'
Standard encoding: ++8=
URL-safe encoding: --8=

Original      : '\xff\xff'
Standard encoding: //8=
URL-safe encoding: __8=
```

## 其他编码

Base32 包含 26 个大写字母和 2-7 的数字。

- `base64.b32encode(s)`: 使用 Base32 编码字符串。s 是要编码的字符串。
- `base64.b32decode(s[, casefold[, map01]])`: 解码 Base32 编码的字符串。s 为要解码的字符串。casefold 表示是否允许小写字母。map01 表示允许 0 表示 0, 1 表示 L。

```
import base64

original_string = 'This is the data, in the clear.'
print 'Original:', original_string

encoded_string = base64.b32encode(original_string)
print 'Encoded :', encoded_string

decoded_string = base64.b32decode(encoded_string)
print 'Decoded :', decoded_string
```

➤ 执行结果

```
$ python base64_base32.py
Original: This is the data, in the clear.
Encoded : KRUGS4ZANFZSA5DIMUQGIYLUMEWCA2LOEB2GQZJAMNWKYLSFY=====
```

Decoded : This **is** the data, **in** the clear.

Base16 包含 16 个 16 进制大写数字。类似的有 `base64.b16encode(s)` , `base64.b16decode(s[, casefold])`。

```
import base64

original_string = 'This is the data, in the clear.'
print 'Original:', original_string

encoded_string = base64.b16encode(original_string)
print 'Encoded :', encoded_string

decoded_string = base64.b16decode(encoded_string)
print 'Decoded :', decoded_string
```

➤ 执行结果

```
$ python base64_base16.py
Original: This is the data, in the clear.
Encoded : 546869732069732074686520646174612C20696E2074686520636C6561722E
Decoded : This is the data, in the clear.
```

Python3.4 中增加了 `Ascii85` 和 `base85` 支持。这里暂不做详细介绍。函数如下:

- `base64.a85encode(s, *, foldspaces=False, wrapcol=0, pad=False, adobe=False)`
- `base64.a85decode(s, *, foldspaces=False, adobe=False, ignorechars=b' tnrv')`
- `base64.b85encode(s, pad=False)`
- `base64.b85decode(b)`