

1 电子邮件消息：

电子邮件是一种用电子手段提供信息交换的通信方式，是互联网应用最广的服务。通过网络的电子邮件系统，用户可以以非常低廉的价格（不管发送到哪里，都只需负担网费）、非常快速的方式（几秒钟之内可以发送到世界上任何指定的目的地），与世界上任何一个角落的网络用户联系。

电子邮件可以是文字、图像、声音等多种形式。同时，用户可以得到大量免费的新闻、专题邮件，并实现轻松的信息搜索。电子邮件的存在极大地方便了人与人之间的沟通与交流，促进了社会的发展。

2 邮件头：

一封邮件的头部，正文和附件除外的信息

3 邮件体：

一封邮件除了头部以外的信息，一般包括正文和附件

4 Base64：

Base64 是网络上最常见的用于传输 8Bit 字节码的编码方式之一，Base64 就是一种基于 64 个可打印字符来表示二进制数据的方法。可查看 RFC2045～RFC2049，上面有 MIME 的详细规范。

Base64 编码是从二进制到字符的过程，可用于在 HTTP 环境下传递较长的标识信息。采用 Base64 编码具有不可读性，需要解码后才能阅读。

Base64 由于以上优点被广泛应用于计算机的各个领域，然而由于输出内容中包括两个以上“符号类”字符 (+, /, =)，不同的应用场景又分别研制了 Base64 的各种“变种”。为统一和规范化 Base64 的输出，Base62x 被视为无符号化的改进版本。

5 email.message.EmailMessage:

EmailMessage 类 实例化有以下参数，注：以下所有参数都是可选的并且可以在 send() 方法之前任意时间设置。

- subject: email 的主题
- body: 主体内容
- from_email: 发送者的邮箱地址
- to: 接受者的邮箱地址（是一个列表或者元组，发送给一个人或者多个人）
- bcc: 使用 'Bcc' 头的邮箱地址（一个列表或者元组）
- connection: 一个邮件备用实例。如果你想用同一个连接发送多个信息，那么需要使用这个参数。不然的话，发送多条信息给同一个接收人，每次调用 send() 方法都会重新创建一个新的连接。
- attachments: 一个附件的列表。附件可以使 email.MIMEBase.MIMBase 实例，也可以是元组（文件，目录，多媒体类型文件）。
- headers: 一个字典参数，放额外的头部信息。字典关键字是头部的名字，字典的键值是头部的内容。调用者在发送 email 信息时，要保证头部名和值都是正确的格式。与之对应的一个形参是 extra_headers。
- cc: 一个列表或者元组，存放使用 "Cc" 头的接受者的邮箱地址。
- replay_to: 一个列表或者元组， 存放使用 "Reply-To" 头的接受者的邮箱地址。

6 add_alternative():

If the message is a `multipart/alternative`, create a new message object, pass all of the arguments to its `set_content()` method, and `attach()` it to the `multipart`. If the message is a `non-multipart` or `multipart/related`, call `make_alternative()` and then proceed as above. If the message is any other type of `multipart`, raise a `TypeError`. If `content_manager` is not specified, use the `content_manager` specified by the current `policy`.

7 add_attachment():

If the message is a `multipart/mixed`, create a new message object, pass all of the arguments to its `set_content()` method, and `attach()` it to the `multipart`. If the message is a non-`multipart`, `multipart/related`, or `multipart/alternative`, call `make_mixed()` and then proceed as above. If `content_manager` is not specified, use the `content_manager` specified by the current `policy`. If the added part has no *Content-Disposition* header, add one with the value `attachment`. This method can be used both for explicit attachments (*Content-Disposition: attachment*) and `inline` attachments (*Content-Disposition: inline*), by passing appropriate options to the `content_manager`.

8 CC:

抄送, Carbon Copy, 又简称为 CC。在网络术语中, 抄送就是将邮件同时发送给收信人以外的人, 用户所写的邮件抄送一份给别人, 对方可以看见该用户的 E-mail。同收件人地址栏一样, 不可以超过 1024 个字符。

一般来说, 使用“抄送”服务时, 多人抄送的电子邮件地址使用“;”分隔。

9 BCC:

Blind Carbon Copy 的缩写。有时, 我们将一封信件寄给收件人时, 同时想寄给其他人, 可以使用抄送 (CC)。如果想将这些抄送对象隐藏起来, 就可以使用暗抄送, 即为 BCC。需要说明的是, 收件人可以看到“抄送[CC]”中的地址, 但看不到“密送[BCC]”中的地址。

10 SMTP:

SMTP 是一种提供可靠且有效的电子邮件传输的协议。SMTP 是建立在 FTP 文件传输服务上的一种邮件服务, 主要用于系统之间的邮件信息传递, 并提供有关来信的通知。SMTP 独立于特定的传输子系统, 且只需要可靠有序的数据流信道支持, SMTP 的重要特性之一是其能跨越网络传输邮件, 即“SMTP 邮件中继”。使用 SMTP, 可实现相同网络处理进程之

间的邮件传输，也可通过中继器或网关实现某处理进程与其他网络之间的邮件传输。

11 POP:

POP 是 TCP/IP 协议族中的一员，POP 允许本地计算机上的用户代理程序连接到邮件服务器，将用户的邮件取回到本地，这样用户就能在本地阅读邮件了。本协议主要用于支持使用客户端远程管理在服务器上的电子邮件。

12 IMAP:

IMAP (Internet Mail Access Protocol) 以前称作交互邮件访问协议 (Interactive Mail Access Protocol)，是一个应用层协议。IMAP 是斯坦福大学在 1986 年开发的一种邮件获取协议。它的主要作用是邮件客户端可以通过这种协议从邮件服务器上获取邮件的信息，下载邮件等。当前的权威定义是 RFC3501。IMAP 协议运行在 TCP/IP 协议之上，使用的端口是 143。它与 POP3 协议的主要区别是用户可以不用把所有的邮件全部下载，可以通过客户端直接对服务器上的邮件进行操作。

13 Email 客户端:

邮件客户端通常指使用 IMAP/APOP/POP3/SMTP/ESMTP/协议收发电子邮件的软件。用户不需要登入邮箱就可以收发邮件。

14 webmail:

Webmail (基于万维网的电子邮件服务) 是因特网上一种主要使用网页浏览器来阅读或发送电子邮件的服务，与使用 Microsoft Outlook、Mozilla Thunderbird 等电子邮件客户端软件的电子邮件服务相对。因特网上的许多公司，诸如 Google、雅虎、中国的新浪等，都提供有 webmail 服务。此外，个人的因特网服务提供商也可能提供 webmail，这样他们的用户可以不必要通过他们一般使用的上网终端而直接使用他们的邮件服务。

15 RFC5322:

Internet Message Format

This document specifies the Internet Message Format (IMF), a syntax for text messages that are sent between computer users, within the framework of "electronic mail" messages. This specification is a revision of Request For Comments (RFC) 2822, which itself superseded Request For Comments (RFC) 822, "Standard for the Format of ARPA Internet Text Messages", updating it to reflect current practice and incorporating incremental changes that were specified in other RFCs.

16 CRLF:

CRLF 是 Carriage-Return Line-Feed 的缩写，意思是回车换行，就是回车(CR, ASCII 13, \r) 换行(LF, ASCII 10, \n)。换行在有的 ASCII 码表也用 newline（简 nl）来进行表示,这里的 lf 是 line feed 的概念，意思是一样的。

17 MIME:

MIME(Multipurpose Internet Mail Extensions)多用途互联网邮件扩展类型。是设定某种扩展名的文件用一种应用程序来打开的方式类型，当该扩展名文件被访问的时候，浏览器会自动使用指定应用程序来打开。多用于指定一些客户端自定义的文件名，以及一些媒体文件打开方式。

它是一个互联网标准，扩展了电子邮件标准，使其能够支持：

非 ASCII 字符文本；非文本格式附件（二进制、声音、图像等）；由多部分（multiple parts）组成的消息体；包含非 ASCII 字符的头信息（Header information）。

18 电子邮件消息:

通过电子邮件发送的消息

19 MIME 边界:

邮件通过 multipart 类型将内容分隔成不同的段, 各段之间的边界标识由对应 multipart 类型的 boundary 属性定义。要从邮件中提取出需要的内容, 首先需要提取出邮件中的分段信息。

20 邮件的传递:

20.1 提交与传输:

当用户编写完电子邮件消息并点击“发送”按钮后, 将电子邮件传输回邮件服务器这一任务的官方叫法是电子邮件提交, 电子邮件提交是 SMTP 协议的两个用途之一。

20.2 SMTP 认证:

SMTP 可以用于电子邮件客户端和提供电子邮件地址的机构服务器之间的电子邮件提交。这一过程中的连接通常都需要进行认证, 这样就可以防止垃圾邮件发送者在不需要提供密码的情况下就轻松盗用其他用户的账号发送大量垃圾消息。

20.3 多跳 (hop) :

以前, 电子邮件通常只需要经过一次 SMTP “跳” (hop) 就可以从源机器传输到存储接收者收件箱的目标机器了, 但是现在电子邮件消息在从源地址发送到目标地址的过程中通常会经过多个服务器, 这意味着 SMTP 信封接收者在邮件的传输过程中是不断变化的。

20.4 smtplib:

The `smtplib` module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon. For details of SMTP and ESMTP operation, consult [RFC 821](#) (Simple Mail Transfer Protocol) and [RFC 1869](#) (SMTP Service Extensions).

20.5 ehlo():

Identify yourself to an ESMTP server using `EHLO`. The `hostname` argument defaults to the fully qualified domain name of the local host. Examine the response for ESMTP option and store them for use by `has_extn()`. Also sets several informational attributes: the message returned by the server is stored as the `ehlo_resp` attribute, `does_esmtp` is set to true or false depending on whether the server supports ESMTP, and `esmtp_features` will be a dictionary containing the names of the SMTP service extensions this server supports, and their parameters (if any).

Unless you wish to use `has_extn()` before sending mail, it should not be necessary to call this method explicitly. It will be implicitly called by `sendmail()` when necessary.

21 poplib:

This module defines a class, `POP3`, which encapsulates a connection to a POP3 server and implements the protocol as defined in [RFC 1725](#). The `POP3` class supports both the minimal and optional command sets. Additionally, this module provides a class `POP3_SSL`, which provides support for connecting to POP3 servers that use SSL as an underlying protocol layer.

22 imaplib:

This module defines three classes, `IMAP4`, `IMAP4_SSL` and `IMAP4_stream`, which encapsulate a connection to an IMAP4 server and implement a large subset of the IMAP4rev1 client protocol as defined in [RFC 2060](#). It is backward compatible with IMAP4 ([RFC 1730](#)) servers, but note that the `STATUS` command is not supported in IMAP4.

23 IMAPClient:

IMAPClient is an easy-to-use, Pythonic and complete IMAP client library.

Although IMAPClient actually uses the `imaplib` module from the Python standard library under the hood, it provides a different API. Instead of requiring that the caller performs extra parsing work, return values are full parsed, readily usable and use sensible Python types. Exceptions are raised when problems occur (no error checking of return values is required).

IMAPClient is straight forward to use, but it can be useful to have at least a general understanding of the IMAP protocol. RFC 3501 explains IMAP in detail. Other RFCs also apply to various extensions to the base protocol. These are referred to in the documentation below where relevant.