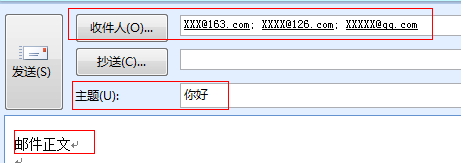
[**python自动发邮件总结及实例说明**](http://www.cnblogs.com/yufeihlf/p/5726619.html)

python发邮件需要掌握两个模块的用法，smtplib和email，这俩模块是python自带的，只需import即可使用。smtplib模块主要负责发送邮件，email模块主要负责构造邮件。

smtplib模块主要负责发送邮件：是一个发送邮件的动作，连接邮箱服务器，登录邮箱，发送邮件（有发件人，收信人，邮件内容）。

email模块主要负责构造邮件：指的是邮箱页面显示的一些构造，如发件人，收件人，主题，正文，附件等。



**1.smtplib模块**

smtplib使用较为简单。以下是最基本的语法。

导入及使用方法如下：

[复制代码](javascript:void(0);)

import smtplib

smtp = smtplib.SMTP()

smtp.connect('smtp.163.com,25')

smtp.login(username, password)

smtp.sendmail(sender, receiver, msg.as\_string())

smtp.quit()

[复制代码](javascript:void(0);)

说明：

smtplib.SMTP()：实例化SMTP()

connect(host,port):

host:指定连接的邮箱服务器。常用邮箱的smtp服务器地址如下：

新浪邮箱：smtp.sina.com,新浪VIP：smtp.vip.sina.com,搜狐邮箱：smtp.sohu.com，126邮箱：smtp.126.com,139邮箱：smtp.139.com,163网易邮箱：smtp.163.com。

port：指定连接服务器的端口号，默认为25.

login(user,password):

user:登录邮箱的用户名。

password：登录邮箱的密码，像笔者用的是网易邮箱，网易邮箱一般是网页版，需要用到客户端密码，需要在网页版的网易邮箱中设置授权码，该授权码即为客户端密码。

sendmail(from\_addr,to\_addrs,msg,...):

from\_addr:邮件发送者地址

to\_addrs:邮件接收者地址。字符串列表['接收地址1','接收地址2','接收地址3',...]或'接收地址'

msg：发送消息：邮件内容。一般是msg.as\_string():as\_string()是将msg(MIMEText对象或者MIMEMultipart对象)变为str。

quit():用于结束SMTP会话。

**2.email模块**

email模块下有mime包，mime英文全称为“Multipurpose Internet Mail Extensions”，即多用途互联网邮件扩展，是目前互联网电子邮件普遍遵循的邮件技术规范。

该mime包下常用的有三个模块：**text，image，multpart**。

导入方法如下：

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.image import MIMEImage

构造一个邮件对象就是一个Message对象，如果构造一个MIMEText对象，就表示一个文本邮件对象，如果构造一个MIMEImage对象，就表示一个作为附件的图片，要把多个对象组合起来，就用MIMEMultipart对象，而MIMEBase可以表示任何对象。它们的继承关系如下：

Message

+- MIMEBase

+- MIMEMultipart

+- MIMENonMultipart

+- MIMEMessage

+- MIMEText

+- MIMEImage

2.1 text说明

邮件发送程序为了防止有些邮件阅读软件不能显示处理HTML格式的数据，通常都会用两类型分别为"text/plain"和"text/html"

构造MIMEText对象时，第一个参数是邮件正文，第二个参数是MIME的subtype，最后一定要用utf-8编码保证多语言兼容性。

2.1.1添加普通文本

text = "Hi!\nHow are you?\nHere is the link you wanted:\nhttp://www.baidu.com"

text\_plain = MIMEText(text,'plain', 'utf-8')

查看MIMEText属性：**可以观察到MIMEText，MIMEImage和MIMEMultipart的属性都一样。**

print dir(text\_plain)

['\_\_contains\_\_', '\_\_delitem\_\_', '\_\_doc\_\_', '\_\_getitem\_\_', '\_\_init\_\_', '\_\_len\_\_', '\_\_module\_\_', '\_\_setitem\_\_', '\_\_str\_\_', '\_charset', '\_default\_type', '\_get\_params\_preserve', '\_headers', '\_payload', '\_unixfrom', 'add\_header', 'as\_string', 'attach', 'defects', 'del\_param', 'epilogue', 'get', 'get\_all', 'get\_boundary', 'get\_charset', 'get\_charsets', 'get\_content\_charset', 'get\_content\_maintype', 'get\_content\_subtype', 'get\_content\_type', 'get\_default\_type', 'get\_filename', 'get\_param', 'get\_params', 'get\_payload', 'get\_unixfrom', 'has\_key', 'is\_multipart', 'items', 'keys', 'preamble', 'replace\_header', 'set\_boundary', 'set\_charset', 'set\_default\_type', 'set\_param', 'set\_payload', 'set\_type', 'set\_unixfrom', 'values', 'walk']

2.1.2添加超文本

[复制代码](javascript:void(0);)

html = """

<html>

<body>

<p>

Here is the <a href="http://www.baidu.com">link</a> you wanted.

</p>

</body>

</html>

"""

text\_html = MIMEText(html,'html', 'utf-8')

[复制代码](javascript:void(0);)

2.1.3添加附件

sendfile=open(r'D:\pythontest\1111.txt','rb').read()

text\_att = MIMEText(sendfile, 'base64', 'utf-8')

text\_att["Content-Type"] = 'application/octet-stream'

text\_att["Content-Disposition"] = 'attachment; filename="显示的名字.txt"'

2.2 image说明

添加图片：

sendimagefile=open(r'D:\pythontest\testimage.png','rb').read()

image = MIMEImage(sendimagefile)

image.add\_header('Content-ID','<image1>')

查看MIMEImage属性：

print dir(image)

['\_\_contains\_\_', '\_\_delitem\_\_', '\_\_doc\_\_', '\_\_getitem\_\_', '\_\_init\_\_', '\_\_len\_\_', '\_\_module\_\_', '\_\_setitem\_\_', '\_\_str\_\_', '\_charset', '\_default\_type', '\_get\_params\_preserve', '\_headers', '\_payload', '\_unixfrom', 'add\_header', 'as\_string', 'attach', 'defects', 'del\_param', 'epilogue', 'get', 'get\_all', 'get\_boundary', 'get\_charset', 'get\_charsets', 'get\_content\_charset', 'get\_content\_maintype', 'get\_content\_subtype', 'get\_content\_type', 'get\_default\_type', 'get\_filename', 'get\_param', 'get\_params', 'get\_payload', 'get\_unixfrom', 'has\_key', 'is\_multipart', 'items', 'keys', 'preamble', 'replace\_header', 'set\_boundary', 'set\_charset', 'set\_default\_type', 'set\_param', 'set\_payload', 'set\_type', 'set\_unixfrom', 'values', 'walk']

2.3 multpart说明

常见的multipart类型有三种：multipart/alternative, multipart/related和multipart/mixed。

邮件类型为"multipart/alternative"的邮件包括纯文本正文（text/plain）和超文本正文（text/html）。

邮件类型为"multipart/related"的邮件正文中包括图片，声音等内嵌资源。

邮件类型为"multipart/mixed"的邮件包含附件。向上兼容，如果一个邮件有纯文本正文，超文本正文，内嵌资源，附件，则选择mixed类型。

msg = MIMEMultipart('mixed')

我们必须把Subject，From，To，Date添加到MIMEText对象或者MIMEMultipart对象中，邮件中才会显示主题，发件人，收件人，时间（若无时间，就默认一般为当前时间，该值一般不设置）。

msg = MIMEMultipart('mixed')

msg['Subject'] = 'Python email test'

msg['From'] = 'XXX@163.com <XXX@163.com>'

msg['To'] = 'XXX@126.com'

msg['Date']='2012-3-16'

查看MIMEMultipart属性：

msg = MIMEMultipart('mixed')

print dir(msg)

结果：

['\_\_contains\_\_', '\_\_delitem\_\_', '\_\_doc\_\_', '\_\_getitem\_\_', '\_\_init\_\_', '\_\_len\_\_', '\_\_module\_\_', '\_\_setitem\_\_', '\_\_str\_\_', '\_charset', '\_default\_type', '\_get\_params\_preserve', '\_headers', '\_payload', '\_unixfrom', 'add\_header', 'as\_string', 'attach', 'defects', 'del\_param', 'epilogue', 'get', 'get\_all', 'get\_boundary', 'get\_charset', 'get\_charsets', 'get\_content\_charset', 'get\_content\_maintype', 'get\_content\_subtype', 'get\_content\_type', 'get\_default\_type', 'get\_filename', 'get\_param', 'get\_params', 'get\_payload', 'get\_unixfrom', 'has\_key', 'is\_multipart', 'items', 'keys', 'preamble', 'replace\_header', 'set\_boundary', 'set\_charset', 'set\_default\_type', 'set\_param', 'set\_payload', 'set\_type', 'set\_unixfrom', 'values', 'walk']

说明：

msg.add\_header(\_name,\_value,\*\*\_params):添加邮件头字段。

msg.as\_string():是将msg(MIMEText对象或者MIMEMultipart对象)变为str,**如果只有一个html超文本正文或者plain普通文本正文的话，一般msg的类型可以是MIMEText；如果是多个的话，就都添加到MIMEMultipart，msg类型就变为MIMEMultipart。**

msg.attach(MIMEText对象或MIMEImage对象):将MIMEText对象或MIMEImage对象添加到MIMEMultipart对象中。MIMEMultipart对象代表邮件本身，MIMEText对象或MIMEImage对象代表邮件正文。

以上的构造的文本，超文本，附件，图片都何以添加到MIMEMultipart('mixed')中：

msg.attach(text\_plain)

msg.attach(text\_html)

msg.attach(text\_att)

msg.attach(image)

**3.文字，html，图片，附件实现实例**

[复制代码](javascript:void(0);)

#coding: utf-8

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.image import MIMEImage

from email.header import Header

#设置smtplib所需的参数

#下面的发件人，收件人是用于邮件传输的。

smtpserver = 'smtp.163.com'

username = 'XXX@163.com'

password='XXX'

sender='XXX@163.com'

#receiver='XXX@126.com'

#收件人为多个收件人

receiver=['XXX@126.com','XXX@126.com']

subject = 'Python email test'

#通过Header对象编码的文本，包含utf-8编码信息和Base64编码信息。以下中文名测试ok

#subject = '中文标题'

#subject=Header(subject, 'utf-8').encode()

#构造邮件对象MIMEMultipart对象

#下面的主题，发件人，收件人，日期是显示在邮件页面上的。

msg = MIMEMultipart('mixed')

msg['Subject'] = subject

msg['From'] = 'XXX@163.com <XXX@163.com>'

#msg['To'] = 'XXX@126.com'

#收件人为多个收件人,通过join将列表转换为以;为间隔的字符串

msg['To'] = ";".join(receiver)

#msg['Date']='2012-3-16'

#构造文字内容

text = "Hi!\nHow are you?\nHere is the link you wanted:\nhttp://www.baidu.com"

text\_plain = MIMEText(text,'plain', 'utf-8')

msg.attach(text\_plain)

#构造图片链接

sendimagefile=open(r'D:\pythontest\testimage.png','rb').read()

image = MIMEImage(sendimagefile)

image.add\_header('Content-ID','<image1>')

image["Content-Disposition"] = 'attachment; filename="testimage.png"'

msg.attach(image)

#构造html

#发送正文中的图片:由于包含未被许可的信息，网易邮箱定义为垃圾邮件，报554 DT:SPM ：<p><img src="cid:image1"></p>

html = """

<html>

<head></head>

<body>

<p>Hi!<br>

How are you?<br>

Here is the <a href="http://www.baidu.com">link</a> you wanted.<br>

</p>

</body>

</html>

"""

text\_html = MIMEText(html,'html', 'utf-8')

text\_html["Content-Disposition"] = 'attachment; filename="texthtml.html"'

msg.attach(text\_html)

#构造附件

sendfile=open(r'D:\pythontest\1111.txt','rb').read()

text\_att = MIMEText(sendfile, 'base64', 'utf-8')

text\_att["Content-Type"] = 'application/octet-stream'

#以下附件可以重命名成aaa.txt

#text\_att["Content-Disposition"] = 'attachment; filename="aaa.txt"'

#另一种实现方式

text\_att.add\_header('Content-Disposition', 'attachment', filename='aaa.txt')

#以下中文测试不ok

#text\_att["Content-Disposition"] = u'attachment; filename="中文附件.txt"'.decode('utf-8')

msg.attach(text\_att)

#发送邮件

smtp = smtplib.SMTP()

smtp.connect('smtp.163.com')

#我们用set\_debuglevel(1)就可以打印出和SMTP服务器交互的所有信息。

#smtp.set\_debuglevel(1)

smtp.login(username, password)

smtp.sendmail(sender, receiver, msg.as\_string())

smtp.quit()

[复制代码](javascript:void(0);)

说明：

如果大家对add\_header的入参有更多的了解（Content-Disposition，Content-Type，Content-ID），希望告诉我，谢谢。

（尊重笔者的劳动哦，转载请说明出处哦。）