

# Sendmail with many attach and recipients

**Source:** <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2005-02/0699.html>

---

*fernandestb\_at\_gmail.com*

**Date:** 02/03/05

Date: 3 Feb 2005 08:50:45 -0800

Here, my code of sendmail  
If Somebody had a doubt please contact me?  
fernandestb@gmail.com

```
# -*- coding: cp1252 -*-
## Faz o import das bibliotecas necessarias.
import mimetypes
import os.path
import smtplib
import sys

from email.Encoders import encode_base64
from email.MIMEAudio import MIMEAudio
from email.MIMEBase import MIMEBase
from email.MIMEImage import MIMEImage
from email.MIMEMultipart import MIMEMultipart
from email.MIMEText import MIMEText

## Funcao para criar lista baseada nos argumentos
## Isto vai ser utilizado para os destinatarios
## como para os arquivos anexados.
def cria_lista(arg):
    lista=[]
    for elemento in arg.split():
        lista.append(elemento)

    return lista

##Faz o anexo dos arquivos a mensagem
##os parametros é o caminho completo do arquivo e o nome do arquivo
##para fazer o anexo.
def getAttachment(path, filename):
    ctype, encoding = mimetypes.guess_type(path)
    if ctype is None or encoding is not None:
        ctype = 'application/octet-stream'
    maintype, subtype = ctype.split('/', 1)
    fp = open(path, 'rb')
    if maintype == 'text':
```

## comp.lang.python: Sendmail with many attach and recipients

```
        attach = MIMEText(fp.read(),_subtype=subtype,
        _charset="iso-8859-15")
        elif maintype == 'message':
            attach = email.message_from_file(fp)
        elif maintype == 'image':
            attach = MIMEImage(fp.read(),_subtype=subtype)
        elif maintype == 'audio':
            attach = MIMEAudio(fp.read(),_subtype=subtype)
        else:
            print maintype, subtype
            attach = MIMEBase(maintype, subtype)
            attach.set_payload(fp.read())
            encode_base64(attach)
        fp.close
        attach.add_header('Content-Disposition',
        'attachment',filename=filename)
        return attach

## Verifica se o primeiro parametro passado para o programa e --h
## chama o help.
if sys.argv[1] == '--h' :
    print 'sendmail <servidor de email> <From> <To> [Assunto]
[Mensagem] [Anexos] [Arquivo de log]\n'
    print '<servidor>: Servidor SMTP de email.\n'
    print '<From>: Quem esta enviando o email. (Ex.: "fx@fx.com.br")
\n'
    print '<To>: lista de destinatarios do email. (Ex.:
"Jose.Salles@sfw.com.br Francisco.Moreira@sfw.com.br") \n'
    print '[Assunto]: Assunto do email \n'
    print '[Mensagem]: Mensagem do corpo do email \n'
    print '[Anexos]: Lista de arquivos anexos
(Ex:"C:\\relatorios\\rel1.pdf C:\\relatorios\\rel2.pdf") \n'
    print '[Arquivo de log] Diretório onde será gerado o log da
aplicação ex: "C:\\temp\\sendmail.log"'
    print 'Ex: sendmail "smtp.fx.com.br" "fx@fx.com.br"
"Jose.Salles@sfw.com.br Francisco.Moreira@sfw.com.br" "teste de email"
"isto e apenas um teste" "C:\\temp\\rel2.pdf C:\\temp\\rel1.pdf"
"C:\\temp\\sendmail.log" \n'
    print 'Parâmetros entre chaves[] são opcionais, os outros
parâmetros são obrigatórios'

else:
    msg = MIMEMultipart()
    ## o 1º parametro e o servidor SMTP.
    servidor_smtp=sys.argv[1]
    ## o 2º parametro e quem esta enviando o anexo.
    msg['From']=sys.argv[2]
    ## o 3º parametro e a lista de destinatários.
    toaddrs=cria_lista(sys.argv[3])
    ## o 4º parametro e o assunto da mensagem.
    msg['Subject'] = sys.argv[4]
```

## comp.lang.python: Sendmail with many attach and recipients

```
## o 5º parametro e a mensagem do corpo do email.
mensagem=sys.argv[5]
## o 6º parametro cria a lista de arquivos.
arquivos=cria_lista(sys.argv[6])
## o 7º parametro especifica onde será gerado o arquivo de log
dir_log=sys.argv[7]
##Verifica se é vazio o arquivo
if dir_log == "":
    dir_log="C:\\temp\\sendmail.log"

## executa um loop na lista de arquivos
## e chama a funcao que anexa os arquivos.
for arq in arquivos:
    path = arq
    filename = os.path.basename(arq)
    attach = getAttachment(path, filename)
    msg.attach(attach)

## texto do corpo do email que sera anexado.
text=MIMEText(mensagem,_charset="iso-8859-15")
msg.attach(text)

## Busca o servidor de SMTP e faz a conexao.
try:
    server = smtplib.SMTP(servidor_smtp)
    ## Faz o envio do email.
    server.sendmail(msg['From'],toaddrs,msg.as_string())
    ## Fecha a conexão com o servidor.
    server.quit()

except Exception, e:
    arq=open(dir_log,'a')
    print >> arq, "Falha no envio da mensagem de email, não foi
possível a conexão ao servidor %s" % servidor_smtp
    arq.close
```