熱血講義

프리렉의 열혈강의 시리즈

Python 파이썬

열혈강의 Python 파이썬

❖제 16 강

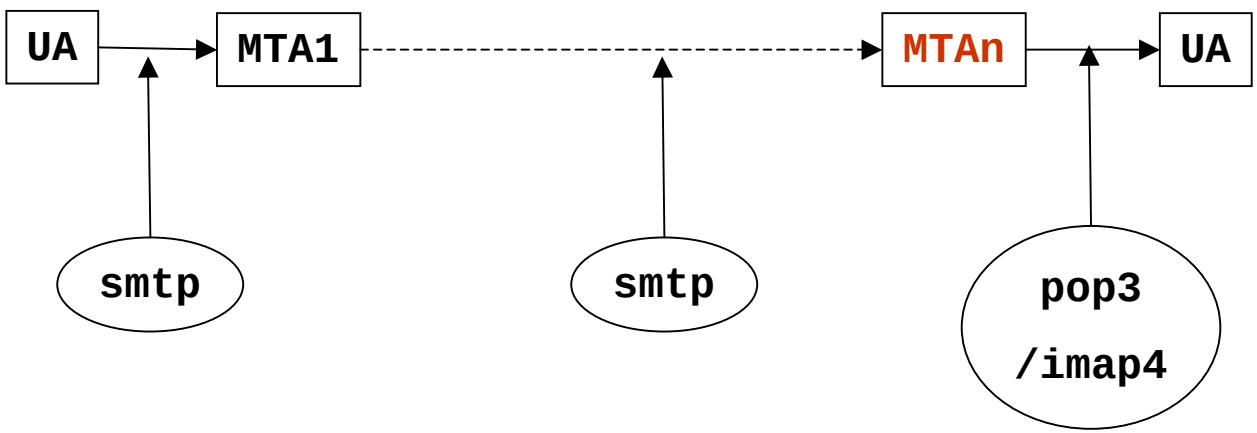
주제 : 이메일 보내기/읽기

이강성(gslee@mail.kw.ac.kr)

❖강의 목차

기본 프로토콜

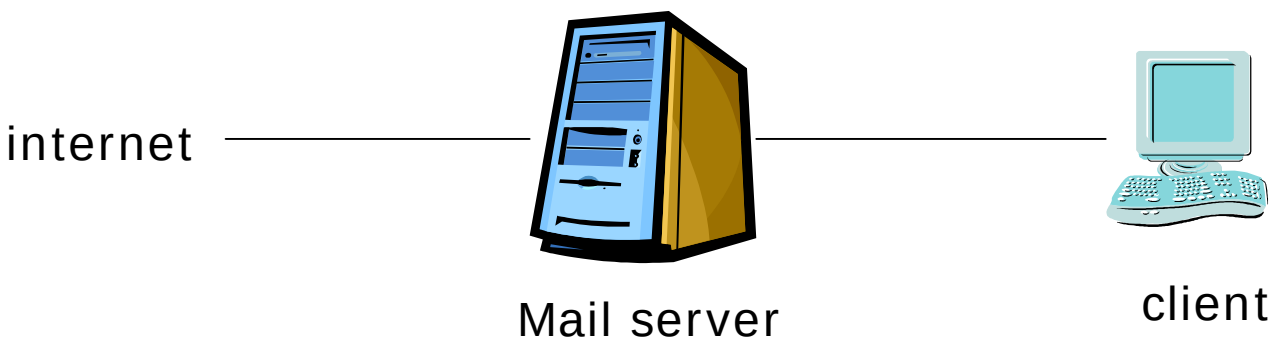
- 메일을 보내는 프로토콜 SMTP
  - Simple Mail Transfer Protocol
  - TCP/UDP 25번 포트 사용
  - 규약 : RFC-821



UA : User Agent (전자우편 송수신 프로그램)
MTA : Message Transfer Agent (메일 중계를 담당하는 컴퓨터)

기본 프로토콜

- 메일을 받는 프로토콜 **POP3**
  - **Post Office Protocol Version 3**
  - 수신인의 메일박스에서 사용자의 메일을 가져오는 프로토콜
  - **TCP 110** 포트 사용
  - 규약 : **RFC-821**



기본 프로토콜

- 메일을 받는 또 다른 프로토콜IMAP4
  - Internet Message Access Protocol version 4
  - POP3와 기본 기능은 동일
  - POP3의 단점을 보완
  - 원격지 서버에서 모든 메일을 관리
  - 자신의 컴퓨터의 메일처럼 다룰 수 있게 해준다
  - 메일 헤더만 따로 가져오는 것이 가능
  - 제목만을 통해 선택적으로 메일을 살펴볼 수도 있다
  - 특정 MIME 내용만 가져올 수 있는 기능

MIME ?

- Multipart Internet Mail Extensions
- 초기의 메일은 영문자 중심으로만 전송
- 다국어 문자, 멀티미디어 데이터를 함께 전송하도록 기능이 확장된 프로토콜
- 여러 파트로 구성되고 각 파트는 별도의 데이터를 포함한다

파이썬 지원 모듈

- smtplib
- poplib
  - POP3
- imaplib
  - IMAP4
- email (2.2)

텍스트 메일 보내기

- **smtplib 모듈 이용**

```
import smtplib
s = smtplib.SMTP(HOST)
s.sendmail(from_addr, to_addr, msgString)
s.quit()

from_addr : 송신자 주소
to_addr : 수신자 주소 리스트
msgString : 보낼 메시지
```

텍스트 문자열 구성하기

- **email.MIMEText 모듈 이용**
  - **class MIMEText(_*text*[, _*subtype*[, _*charset*[, _*encoder*]]])**

_text : 변환할 문자열

_subtype : minor type(plain)

_charset : 문자셋(us-ascii)

_encoder : 7 or 8 (7)

- **예제**
  - **msg = MIMEText(contents, _charset='euc-kr')**
  - **기타 필요한 헤더 정보 추가(Subject, From, To)**
  - **msg.as_string() / str(msg)**

샘플 코드

```
# email01.py
import smtplib
from email.MIMEText import MIMEText

HOST = 'smtp.server'                # SMTP 서버
me = 'gslee@mail.kw.ac.kr'          # 내 주소
you = 'myfriend@somewhere.com'      # 받을 사람 주소
contents = '''
메일 시험중..
파이썬으로 보내는 메일임..
'''
msg = MIMEText(contents, _charset='euc-kr')
msg['Subject'] = 'I love 파이썬'    # 제목
msg['From'] = me
msg['To'] = you

s = smtplib.SMTP(HOST)
s.sendmail(me, [you], msg.as_string())
s.quit()
```

첨부 파일 보내기

- 전체 MIME 메시지 객체 생성

```
from email.MIMEBase import MIMEBase
outer = MIMEBase('multipart', 'mixed')
outer.attach(msg) # MIME 메시지 추가
```

- 개별 MIME 메시지 생성

```
from email.MIMEText import MIMEText
from email.MIMEImage import MIMEImage
from email.MIMEAudio import MIMEAudio
msg = MIMEText(텍스트_문자열, _charset='euc-kr')
msg = MIMEImage(이미지_문자열, _subtype=subtype)
msg = MIMEAudio(오디오_문자열, _subtype=subtype)
```

첨부 파일 보내기

- 기타의 메시지 생성

```
from email.MIMEBase import MIMEBase
from email import Encoders
msg = MIMEBase(maintype, subtype)# 객체를 만들고
msg.add_payload(데이터_문자열)# 내용을 덧 붙이고
Encoders.encode_base64(msg) # 인코딩 한다
```

인코딩

- **Encoders 모듈의 함수들**

```
from email import Encoders
```

  - **encode_quopri(*msg*)**
    - Quoted-printable로 인코딩하고 Content-Transfer-Encoding 헤더 quoted-printable로 설정
  - **encode_base64(*msg*)**
    - Base64로 인코딩하고 Content-Transfer-Encoding 헤더 base64로 설정
  - **encode_7or8bit(*msg*)**
    - Content-Transfer-Encoding 를 7bit 혹은 8bit로 설정
  - **encode_noop(*msg*)**
    - 아무 일도 하지 않는다.

인코딩

- **quoted-printable**
  - ASCII 문자는 그대로 전송
  - 그 이외의 문자는 세 개의 문자로 변환해서 전송
  - 변환 방법은
    - '=' + ASCII로 두 바이트로 16진수를 표현
    - 10011101 → =9D

인코딩

- **base64**
  - 데이터를 24비트(3바이트) 블록으로 분리
  - 이것을 다시 6비트 네 개로 분리
  - 각 6비트는 8비트 ASCII 문자로 변환
  - A-Z, a-z, 0-9, +, / 총 64개의 문자로 구성
  - 000100 → 'E'

첨부 파일 예제

- 디렉토리 파일 전체 첨부하기

```
# email02.py
import smtplib
import mimetypes
import glob

from email import Encoders
from email.Message import Message
from email.MIMEAudio import MIMEAudio
from email.MIMEBase import MIMEBase
from email.MIMEImage import MIMEImage
from email.MIMEText import MIMEText

HOST = 'smtp.server'                    # SMTP 서버
me = 'gslee@mail.kw.ac.kr'              # 내 주소
receiver = ['friend1@some.com', 'friend2@some.com']# 받을 사람 주소 리스트
```

첨부 파일 예제(계속)

```
outer = MIMEBase('multipart', 'mixed')
outer['Subject'] = '첨부파일 메일 보내기'
outer['From'] = me
outer['To'] = ', '.join(receiver)
outer.preamble = 'This is a multi-part message in MIME format.\n\n'
outer.epilogue = '' # 이렇게 하면 멀티파트 경계 다음에 줄바꿈 코드가 삽입됨
msg = MIMEText('파이썬 파일 소스를 보냅니다..', _charset='euc-kr')
outer.attach(msg)
```

```
files = glob.glob('*.*')
for fileName in files:
    ctype, encoding = mimetypes.guess_type(fileName)
    if ctype is None or encoding is not None:
        ctype = 'application/octet-stream'
    maintype, subtype = ctype.split('/', 1)
    if maintype == 'text':
        fd = open(fileName)
        msg = MIMEText(fd.read(), _subtype=subtype)
    elif maintype == 'image':
        fd = open(fileName, 'rb')
        msg = MIMEImage(fd.read(), _subtype=subtype)
    elif maintype == 'audio':
        fd = open(fileName, 'rb')
        msg = MIMEAudio(fd.read(), _subtype=subtype)
    else:
        fd = open(fileName, 'rb')
        msg = MIMEBase(maintype, subtype)
        msg.add_payload(fd.read())
        # Encode the payload using Base64
        Encoders.encode_base64(msg)
    fd.close()
    msg.add_header('Content-Disposition', 'attachment', filename=fileName)
    outer.attach(msg)
```

첨부 파일 예제(계속)

```
# SMTP 서버를 통해서 메일 보내기
s = smtplib.SMTP(HOST)
s.sendmail(me, receiver, outer.as_string())
s.quit()
```

이메일 읽기

- 메일 가져오기
  - POP3를 이용하여 메일을 읽어오기 설명
  - 초기화 / 종료 방법

```
# 초기화 절차
import poplib
mbox = poplib.POP3(host)
mbox.user('userid') # 사용자 id를 넘겨준다
mbox.pass_('passwd')  # password를 넘겨준다
```

```
# 종료 절차
mbox.quit ()
```

이메일 읽기

- 메시지 개수, 크기 확인
  - `noMsg, tsize = mbox.stat()`
- 메시지 읽어 내기
  - `(server_msg, body, octets) = mbox.retr(n)`
  - 리턴 : (서버응답메시지, 메시지, 메시지크기)
  - 메시지 : [라인1, 라인2, ..., 라인m]
- 하나의 문자열 메시지로 구성
  - `message = '\n'.join(body)`

이메일 읽기

- ## 메일 읽기 예제

```
# readmail01.py

import poplib

host = 'mail.host.at.some.where'  # POP3를 지원하는 메일 서버를 지정한다.
mbox = poplib.POP3(host)
mbox.user('userid') # 사용자 id를 넘겨준다
mbox.pass_('passwd')  # password를 넘겨준다

noMsg, tsize = mbox.stat()
print noMsg, 'messages'
if noMsg > 0:
    (server_msg, body, octets) = mbox.retr(noMsg)
    message = '\n'.join(body)
    print message

mbox.quit ()
```

MIME 메시지 해석하기

- 문자열 메시지를 MIME 객체로 변환
  - `mimeMsg = email.message_from_string(strmsg)`
- 파일 객체를 MIME 객체로 변환
  - `mimeMsg = email.message_from_file(fp)`
- 헤더 정보 읽기
  - `mimeMsg['from']`
  - `mimeMsg['to']`
  - `mimeMsg['subject']`

이메일 목록 얻기

- 출력의 예

```
=?ks_c_5601-1987?B?IsDMsK28uil=?= <gslee111@naver.com>
=?ks_c_5601-1987?B?IsDMsK28uil=?= <gslee@mail.kw.ac.kr>
=?ks_c_5601-1987?B?wMy43sDPIMXXvbrGrg==?=
```

헤더의 부호화된 한글의 해석

- 일부 텍스트를 인코딩 형식
  - **=?char-set?encoding?encoded-text?=**
  - char-set : euc-kr, ksc_5601-1987,…
  - encoding : ‘B’ – base64, ‘Q’:quoted-printable
- 예

=?euc-kr?q?=B1=E8=20=C7=FC=BA=B9?= <gudwns_kim@yahoo.co.kr>

헤더의 부호화된 한글의 해석

- 디코딩 기법
  - email.Header.decode_header(string)
  - 각 디코딩되는 문자열을 (decoded_string, charset) 의 리스트로 리턴

```
>>> from email import Header
>>> s = '=?ks_c_5601-1987?B?IsDMsK28uiI=?= <gslee111@naver.com>')
>>> Header.decode_header(s)
[('"\xc0\xcc\xb0\xad\xbc\xba"', 'ks_c_5601-1987'),
    ('<gslee111@naver.com>', None)]

>>> def decodeHeader(headerMsg):
        L = Header.decode_header(headerMsg)
        return ''.join([t[0] for t in L])
>>> print decodeHeader(s)
"이강성"<gslee111@naver.com>
```

MIME 메시지 해석하기

- 메시지 객체 탐색하기
  - mimeMsg.walk() 이용

```
for part in mimeMsg.walk():
    if part.get_main_type() == 'multipart':
        continue
```

- 메시지 종류 확인하기
  - get_main_type() / get_subtype()

MIME 메시지 해석하기

- 첨부 파일 이름 알아내기
  - `filename = part.get_filename()`
- 이름을 모를 때 파일 타입으로 확장자 추정하기
  - `ext = mimetypes.guess_extension(part.get_type())`
- 본문 내용 추출/디코딩하기
  - `part.get_payload(decode=1)`

예제:메일 하나 읽기

- 메일 읽고 첨부파일 저장하기

```
# readmail02.py
import poplib
import email
import mimetypes
from email import Header

def decodeHeader(headerMsg):
    L = Header.decode_header(headerMsg)
    return ''.join([t[0] for t in L])

host = ''       # 적당한 값을 써 넣을 것
userid = ''     # 적당한 값을 써 넣을 것
passwd = ''     # 적당한 값을 써 넣을 것

mbox = poplib.POP3(host)          # 서버에 연결
mbox.user(userid)                 # 로그인
mbox.pass_(passwd)                # 로그인
noMsg, tsize = mbox.stat()
print noMsg, 'messages'
```

```
if noMsg > 0:
    #(server_msg, body, octets) = mbox.retr(noMsg)
    (server_msg, body, octets) = mbox.retr(1)
    message = '\n'.join(body)
    msg = email.message_from_string(message)  # 문자열에서 Message 객체로
    print decodeHeader(msg['from'])   # 보낸 사람
    print decodeHeader(msg['to'])     # 받는 사람
    print decodeHeader(msg['subject'])# 제목
    print msg['date']         # 날짜
    counter = 1               # 마임 파트의 수
    for part in msg.walk(): # 각각의 파트에 대해
        if part.get_main_type() == 'multipart':
            continue
        filename = part.get_filename()      # 파일 이름을 얻고
        if not filename:    # 이름이 없으면 만든다.
            ext = mimetypes.guess_extension(part.get_type())
            if not ext:
                ext = '.bin'
            filename = 'part-%03d%s' % (counter, ext)
        counter += 1
        fp = open(filename, 'wb')
        fp.write(part.get_payload(decode=1)) # 디코드 해서 파일로 저장
        fp.close()
        print filename, 'saved..'
mbox.quit ()
```

이메일 목록 얻기

- 헤더 정보만 가져오기
  - **top(*which, howmuch*)**
  - **which – 메시지 번호**
  - **howmuch – 헤더 다음에 읽을 라인 수**

```
res = mbox.top(noMsg, 0)[1]
headerMsg = '\n'.join(res)
```

샘플 프로그램

```
# readmail03.py
... # 중복부분 생략
mbox = poplib.POP3(host)
mbox.user(userid) #
mbox.pass_(passwd)  #
noMsg, tsize = mbox.stat()

for k in range(1, noMsg+1):
    res = mbox.top(k, 0)[1]
    headerMsg = '\n'.join(res)
    f = StringIO.StringIO(headerMsg)
    msg = mimetools.Message(f)
    print k, '보낸이: %s, 받는이: %s, 제목: %s, 날짜: %s' %
(decodeHeader(msg['from']), decodeHeader(msg['to']),
decodeHeader(msg['subject']), msg['date'])
mbox.quit()
```

스팸 메일 삭제하기

- POP3 객체의 메일 삭제 메쏘드
  - dele(*메일번호*)
  - quit() 메쏘드 이후에 실제 삭제됨
- 제목에서 메일 삭제
  - 다음 슬라이드
- 본문에서 메일 삭제
  - 스스로 해보기

스팸 메일 삭제

```
mbox = poplib.POP3(host)
mbox.user(userid) #
mbox.pass_(passwd)  #
noMsg, tsize = mbox.stat()

spamKeyWords = ('광고', '홍보')
for k in range(noMsg, 0, -1):
    res = mbox.top(k, 0)[1]
    headerMsg = '\n'.join(res)
    f = StringIO.StringIO(headerMsg)
    msg = mimetools.Message(f)
    subject = decodeHeader(msg['subject'])
    if filter(lambda x: x>=0, map(subject.find, spamKeyWords)):
        print 'Deleting..', k, subject
        mbox.dele(k)

mbox.quit()
```