

# 제13강 Email

# 학습 목차

- CSV
- SMTP 를 활용한 메일 송신
- IMAP 을 활용한 메일 수신

# CSV(Comma Separated Value)

- 여러 개 항목값들의 집합을 텍스트 형식으로 저장
- 심표로 구분함. (기본 delimiter 는 심표)
- 스프레드시트 데이터의 기본 구조

연도	제조사	모델	설명	가격
1997	Ford	E350	ac, abs, moon	3000.00
1999	Chevy	Venture "Extended Edition"		4900.00
1999	Chevy	Venture "Extended Edition, Very Large"		5000.00
1996	Jeep	Grand Cherokee	MUST SELL! air, moon roof, loaded	4799.00

위의 데이터 표는 다음과 같이 CSV 형식으로 표현할 수 있다:

```
연도,제조사,모델,설명,가격
1997,Ford,E350,"ac, abs, moon",3000.00
1999,Chevy,"Venture ""Extended Edition""", "",4900.00
1999,Chevy,"Venture ""Extended Edition, Very Large""",,5000.00
1996,Jeep,Grand Cherokee,"MUST SELL!
air, moon roof, loaded",4799.00
```

첫줄에는 항목이름  
다음 줄부터는 각 항목의 값

# 기본 읽기 사용

```
id, height, weight
100, 176, 80
101, 180, 90
```

```
import csv
f = open('simple.csv')
reader = csv.reader(f)
for line in reader:
    print(line)
f.close()
```

```
['id', ' height', ' weight']
['100', ' 176', ' 80']
['101', ' 180', ' 90']
```

## skipinitialspace=True 옵션

- 기본 설정은 쉼표 사이의 모든 내용을 하나의 단위로 간주하기 때문에, 공백도 포함하게 됨. 쉼표 다음의 공백은 무시하는게 좋음.

```
import csv
f = open('simple.csv')
reader = csv.reader(f, skipinitialspace=True)
for line in reader:
    print(line)
f.close()
```

```
['id', 'height', 'weight']
['100', '176', '80']
['101', '180', '90']
```

# quotechar

- 쉼표 등의 특수 문자 등을 포함하고 싶을 때, 인용부호를 사용함.
- 기본값은 “
- 바꾸고 싶을 때는, quotechar= 로 따로 정의

```
year, title  
2020, "Orange, Apple and Tomato"
```

# 항목과 값들을 연관시키기 - DictReader

```
f = open('simple.csv')
reader = csv.DictReader(f, skipinitialspace=True)
for line in reader:
    print(line)
f.close()
```

```
{'id': '100', 'height': '176', 'weight': '80'}
{'id': '101', 'height': '180', 'weight': '90'}
```

# oscar\_age\_female.csv

```
"Index", "Year", "Age", "Name", "Movie"  
1, 1928, 22, "Janet Gaynor", "Seventh Heaven, Street Angel and Sunrise: A Song of Two Humans"  
2, 1929, 37, "Mary Pickford", "Coquette"  
3, 1930, 28, "Norma Shearer", "The Divorcee"  
4, 1931, 63, "Marie Dressler", "Min and Bill"
```



# 기본 쓰기

```
f = open('test.csv', 'w', newline='')  
writer = csv.writer(f)  
writer.writerow([101, 70, 90])  
writer.writerow(['hello, world', 20, 10])  
writer.writerows([[1,2,3], [3,4,5]])  
f.close()
```

101,70,90

"hello, world",20,10

1,2,3

3,4,5

# delimiter와 quotechar 재정의

```
f = open('test.csv', 'w', newline='')
writer = csv.writer(f, delimiter=':', quotechar='"')
writer.writerow([101, 70, 90])
writer.writerow(['hello: world', 20, 10])
writer.writerows([[1,2,3], [3,4,5]])
f.close()
```

Quoting 은 필요할 때만 하도록 하는 설정이 기본.  
만약 항상 하고 싶으면, QUOTE\_ALL option

```
f = open('test.csv', 'w', newline='')
writer = csv.writer(f, delimiter=':', quoting=csv.QUOTE_ALL)
writer.writerow([101, 70, 90])
writer.writerow(['hello: world', 20, 10])
writer.writerows([[1,2,3], [3,4,5]])
f.close()
```

# 항목과 값 쓰기

```
with open('test.csv', 'w', newline='') as f:
    writer = csv.DictWriter(f, fieldnames=['kor', 'math', 'eng'])
    writer.writeheader()
    writer.writerow({'kor':100, 'math':95, 'eng':90})
    writer.writerow({'kor':55, 'math':70, 'eng':20})
```

# CSV Read & Write

```
import csv

file_name = 'oscar_age_female.csv'

with open(file_name) as f:
    reader = csv.reader(f, skipinitialspace=True)

    data = list(reader)

    for row in data[1:]:
        print(row)

    with open('converted.csv', 'w', newline='') as wf:
        writer = csv.writer(wf, delimiter=',', quotechar='\\', quoting=csv.QUOTE_ALL)
        for row in data[1:]:
            writer.writerow(row)
```

# SMTP(Simple Mail Transfer Protocol)

- 이메일 전송 표준

# Email Test & Debug Server

- 테스트 SMTP 서버를 현재 PC에서 구동
- localhost:8025 에 SMTP 데몬 연결

```
pip install aiosmtpd
```

```
python -m aiosmtpd -n
```

# 기본 연결

```
import smtplib

mail_server = smtplib.SMTP('localhost:8025')

sender = 'daehyun.lee.python.text@gmail.com'
receiver = 'dustinlee@tukorea.ac.kr'
message = """\
Subject: Test Mail
This is automatically send mail by Python."""

mail_server.sendmail(sender, receiver, message)

mail_server.quit()
```

# EmailMessage 활용 연결

```
from email.message import EmailMessage

msg = EmailMessage()
msg['Subject'] = 'Test Mail' # 한글 제목 X
msg['From'] = 'daehyun.lee.python.test@gmail.com'
msg['To'] = 'dustinlee@tukorea.ac.kr'

msg.set_content('이것은 자동으로 발송된 테스트 메일입니다.')

with open('./키티.jpg', 'rb') as f:
    msg.add_attachment(f.read(), maintype='image', subtype='jpeg', filename='키티.jpg')

mail_server = smtplib.SMTP('localhost:8025')
mail_server.send_message(msg)
mail_server.quit()
```



# MIME(Multipurpose Internet Mail Extensions)

메인타입(주타입)	설명	서브타입(부타입)
text	텍스트를 표현.	text/plain, text/html, text/css, text/javascript, ...
image	이미지를 표현.	image/gif, image/png, image/jpeg, image/bmp, ...
video	동영상을 표현.	video/mp4, video/ogg, video/mpeg4-generic, ...
application	모든 종류의 이진 데이터를 표현.	application/octet-stream, application/vnd.mspowerpoint, application/xml, application/pdf, application/json, ...
multipart	여러가지 타입의 데이터들을 동시에 전송할 때 사용.	multipart/form-data, multipart/byteranges, ...

# 2단계 인증 설정

Google 계정

Google 계정 검색

?

I

홈

개인 정보

데이터 및 개인 정보 보호

보안

사용자 및 공유

결제 및 구독

정보

보안 활동 검토

Google에 로그인하는 방법

항상 Google 계정에 액세스할 수 있도록 최신 정보가 반영되었는지 확인하세요.

2단계 인증

✓ 사용 설정일: 2023. 4. 20.

>

\*\*\* 비밀번호

최종 변경일: 2021. 5. 1.

>

가능한 경우 비밀번호 건너뛰기

✓ 사용

>

2단계 인증 전화번호

010-9788-1745

>

## < 2단계 인증

### 계정이 2단계 인증으로 보호되고 있습니다

해커가 계정에 액세스하지 못하도록 보안을 강화하세요.

패스키가 아닌 다른 방법으로 로그인하는 경우, 계정에서 사용할 수 있는 가장 안전한 2단계를 완료하라는 메시지가 표시됩니다. 설정에서 언제든지 2단계와 로그인 옵션을 업데이트할 수 있습니다. [보안 설정으로 이동](#)



2단계 인증 사용 중지

### 2단계

Google 계정에 액세스할 수 있도록 이 정보를 최신 상태로 유지하고 로그인 옵션을 추가하세요.



패스키 및 보안 키



보안 키 추가



Google 메시지



OTP



OTP 앱 추가



전화번호



010-9788-1745



복구 코드



백업 코드 받기



### 앱 비밀번호

앱 비밀번호는 권장되지 않으며 대부분의 경우 필요하지 않습니다. 계정을 안전하게 보호하려면 'Google 계정으로 로그인'을 사용하여 앱을 Google 계정에 연결하세요.

앱 비밀번호

앱 비밀번호 1개



# gmail 송수신을 프로그램으로 자동화하려면?

- 2단계 인증 설정 -----> 앱 비밀번호 설정

## ← 2단계 인증

2023. 4. 20.에 2단계 인증을 사용하도록 설정했습니다.

사용 안함

사용 가능한 인증 2단계

비밀번호를 입력한 후 로그인하는 사용자의 본인 여부를 확인하는 2단계입니다. [자세히 알아보기](#)

참고: 지원 휴대전화에서 Google 계정에 로그인하면 Google 메시지가 2단계 인증을 위한 다른 수단으로 추가됩니다.

## ← 앱 비밀번호

앱 비밀번호를 사용하면 2단계 인증을 지원하지 않는 기기의 앱에서 Google 계정에 로그인할 수 있습니다. 비밀번호를 한 번만 입력하면 기억할 필요가 없습니다. [자세히 알아보기](#)

### 앱 비밀번호

이름	생성됨	최종 사용일	
python	오후 1:36	오후 8:15	

앱 비밀번호를 생성할 앱 및 기기를 선택하세요.

앱 선택

기기 선택

생성

# gmail 이메일 송신(보내기)

```
import smtplib
import email
from email.message import EmailMessage

id = 'daehyun.lee.python.test@gmail.com'
passwd = '' # this is app password

mail_server = smtplib.SMTP('smtp.gmail.com', 587)
mail_server.ehlo()
mail_server.starttls()
mail_server.login(id, passwd)
msg = EmailMessage()
msg['Subject'] = '자동 발송 메일'
msg['From'] = '이대현<daehyun.lee.python.test@gmail.com>'
msg['To'] = 'dustinlee@tukorea.ac.kr, daehyun.lee@gmail.com'
msg.set_content("이것은 자동 발송 메일입니다.")
mail_server.send_message(msg)
mail_server.quit()
```

# MIME type 추출

```
import mimetypes

def get_mime_type(fname):
    ctype, _ = mimetypes.guess_type(fname)
    if not ctype:
        ctype = 'application/octet-stream'
    maintype, subtype = ctype.split('/', 1)
    return maintype, subtype
```

# 파일 첨부 발송

```
fname = 'tukorea.jpg'
maintype, subtype = get_mime_type(fname)
with open(fname, 'rb') as f:
    msg.add_attachment(f.read(), maintype=maintype, subtype=subtype, filename=fname)
```

# gmail 읽기 - IMAP 연동 login & logout

```
import imapclient
import email
from email.policy import default

imap_server = imapclient.IMAPClient('imap.gmail.com', ssl=True)

id = 'daehyun.lee.python.test@gmail.com'
passwd = '' # this is app password

imap_server.login(id, passwd)
imap_server.logout()
```



# 모든 메일 정보 확인

```
# 모든 메시지 획득
messages = imap_server.gmail_search("")

for id, raw_message in imap_server.fetch(messages, 'RFC822').items():
    email_message = email.message_from_bytes(raw_message[b'RFC822'], policy=default)
    date = email_message['Date']
    sender = email_message['From']
    subject = email_message['Subject']
    print(f'{id=} {date=} {sender=} {subject=} ')
```

# Gmail 스타일 검색

- 검색 결과는 메일 ID 의 리스트

```
messages = imap_server.gmail_search('Hello')
```

<https://support.google.com/mail/answer/7190?hl=en/>

## Gmail에서 사용할 수 있는 검색 연산자

검색 연산자라고 부르는 단어나 기호를 사용해 Gmail 검색결과를 필터링할 수 있습니다. 또한 검색 연산자를 조합해 검색결과를 더 세부적으로 필터링할 수도 있습니다.

### 검색 연산자 사용 방법

- 1. Gmail 로 이동합니다.
- 2. 검색창에 검색 연산자를 입력합니다.

### 사용할 수 있는 검색 연산자

**도움말:** 검색 연산자를 사용해 검색하면 검색결과를 사용하여 다음과 같은 메일 **필터**를 **설정**할 수 있습니다.

검색 기준	검색 연산자 및 예
보낸 사람 검색	from: 예: from:지현
받는 사람 검색	to: 예: to:인성
메일 사본을 받는 사람 검색	cc: bcc: 예: cc:인성
제목에 포함된 단어	subject: 예: subject:저녁식사
여러 검색어와 일치하는 메일	OR 또는 {} 예: from:지현 OR from:인성 예: {from:지현 from:인성}
특정 단어가 포함된 결과 제외	- 예: 저녁식사 -영화
근처에 있는 단어로 메일 찾기. 단어와 단어 사이에 들어갈 수 있는 단어 수를 숫자로 지정하세요. 따옴표를 추가하여 처음에 입력한 단어가 먼저 나오는 메일을 찾습니다.	AROUND 예: 휴일 AROUND 10 휴가 예: "비밀 AROUND 25 생일"
특정 라벨이 있는 메일	label:

# 검색량 한계 설정

```
import imaplib  
imaplib._MAXLINE = 10000000
```

## sender 포맷

```
>>> msg['From']  
'이대현 <daehyun.lee.python.test@gmail.com>'  
>>> msg['From'].addresses[0]  
Address(display_name='이대현', username='daehyun.lee.python.test', domain='gmail.com')
```

# 등록 요청 메일로부터 주소록 만들기

```
msg_id_list = imap_server.gmail_search('등록')
address_list = []

for id, raw_message in imap_server.fetch(msg_id_list, 'RFC822').items():
    email_message = email.message_from_bytes(raw_message[b'RFC822'], policy=default)
    date = email_message['Date']
    sender = email_message['From']
    sender_address = f'{sender.addresses[0].username}@{sender.addresses[0].domain}'
    name = email_message['Subject'].split()[0]
    subject = email_message['Subject']
    address_list.append([name, sender_address])

with open('address.csv', 'w', newline='') as wf:
    writer = csv.writer(wf)
    for address in address_list:
        writer.writerow(address)
```

# Message 분석, 출력 및 파일 저장

```
def print_message(msg):
    print("=" * 200)
    print(f"""\
수신시각: {msg["Date"]}
제목:{msg["Subject"]}
보내는사람:{msg["From"]}
받는사람:{msg["To"]}
""")

    for part in msg.walk():
        if part.get_content_maintype() == 'multipart':
            continue
        if part.get_content_maintype() == 'text':
            if part.get_content_subtype() == 'plain':
                body = str(part.get_payload(decode=True), part.get_content_charset())
                print(body)
            else: # 첨부 파일 존재
                fname = part.get_filename()
                if fname:
                    print(f'{fname}을 저장합니다.....')
                    with open(fname, 'wb') as fp:
                        fp.write(part.get_payload(decode=True))

for id, raw_message in imap_server.fetch(messages, 'RFC822').items():
    email_message = email.message_from_bytes(raw_message[b'RFC822'], policy=default)
    print_message(email_message)
```