

# 27. Python Mailing

# 1. Python SMTP , IMAP 개념

## 1] SMTP Library

### 1) SMTP , POP3 , IMAP 개념

#### ① SMTP Protocol (Simple Mail Transfer Protocol , TCP Port 25)

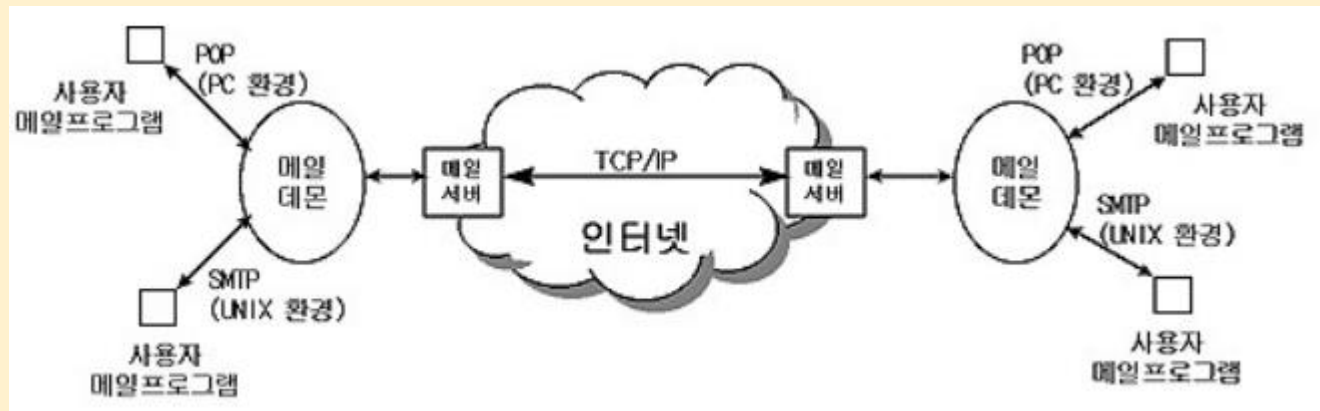
- eMail을 보낼때 사용하는 Mail 전송 Protocol , Client To Server , Server To Server 통신 모두 담당
- gMail : 하루 2000개 전송한도

#### ② POP3 Protocol (Post Office Protocol Version 3 , TCP Port 110)

- 서버-client 간 Protocol, Client에서 Server에 있는 mail 을 D/L받아 읽도록 해 주는 Protocol
- 구현용이, 서버로부터 Mail 가져온 후 삭제

#### ③ IMAP Potocol (Internet Mail Access Protocol , TCP Port 143)

- POP3 대체 Protocol
- 서버로부터 Mail 가져와도 서버에 Mail이 그대로 남아 있음
- 서버 Traffic이 많이 쓰임



- mail daemon이란 일종의 프로세스로서, 송신자의 메일과 정보를 메일 서버가 해석 가능하도록 재가공. 반대의 경우 또한 같은 방법으로 수신자의 정확한 메시지 수신과 릴레이를 지원

## 2. SMTP Process

### 2) SMTP 주요 Process

- ① Mail 보낼때 SMTP Protocol 사용 # gmail 에서는 Port 587번 사용  
smtp = smtplib.SMTP('smtp.gmail.com', 587) # Python 에 smtplib가 Core에 포함 되어 있음
- ② SSL일때 설정  
smtp = smtplib.SMTP\_SSL('smtp.gmail.com', 465) # SSL일때 Port 465 번 사용
- ③ Hello message 보내기  
smtp.ehlo()
- ④ TLS 암호화 시작(SMTP\_SSL 시 사용하지 않는다)  
smtp.starttls()
- ⑤ SMTP Server Login # google Mail(ID , Password 기술 필요)에 반드시 계정 설정 필요  
smtp.login(myEmail@gmail.com', 'PassWord')
- ⑥ eMail 보내기  
smtp.send\_message(msg)
- ⑦ SMTP Server 접속 끊기  
smtp.quit()

Google 계정관리

2단계 인증

### 3. Python IMAP

#### 1] IMAP 개념

- IMAP Server에 Login하고, email 검색 , 가져와서 Message를 읽는다
- imaplib와 email 두개의 모듈 사용

#### 2] 주요 Process

##### ① # IMAP Server에 접속

```
import imaplib  
imap = imaplib.IMAP4_SSL('imap.gmail.com')
```

##### ② IMAP Server Login

```
imap.login('myEmail@gmail.com', 'PassWord')
```

##### ③ INBOX 선택(받은 편지함)

```
imap.select('inbox')
```

##### ④ email 검색

```
# 전체를 검색(search) -> 고유 id(uid) 검색
```

```
result, data = imap.uid('search', None, "ALL")
```

```
# 확보된 uid Data중 가장 마지막 자료 가져오면 최근 자료 가져옴
```

```
latest_email_uid = data[0].split()[-1]
```

```
# fetch를 통해 확보된 uid Data중 실제 Mail Data를 가져옴
```

```
result, data = imap.uid('fetch', latest_email_uid, '(RFC822)')
```

```
# 사람이 읽기 힘든 가공되지 않은 Data
```

```
raw_email = data[0][1]
```

### 3. Python IMAP(계속)

#### 2] 주요 Process

##### ⑤ Raw Email Parsing : 파싱하기

```
import email # 파싱 필요 모듈
```

```
# byte가 string로 바뀜
```

```
email_message = email.message_from_string(raw_email.decode('utf-8'))
```

```
email_message['To'] # 받는 사람
```

```
email.utils.parseaddr(email_message['From']) # 보낸사람
```

##### ⑥ eMail 본문 내용 확인

```
if email_message.is_multipart():
```

```
    for p in email_message.get_payload():
```

```
        print(p.get_payload())
```

##### ⑦ 로그아웃

```
imap.close()
```

```
imap.logout()
```