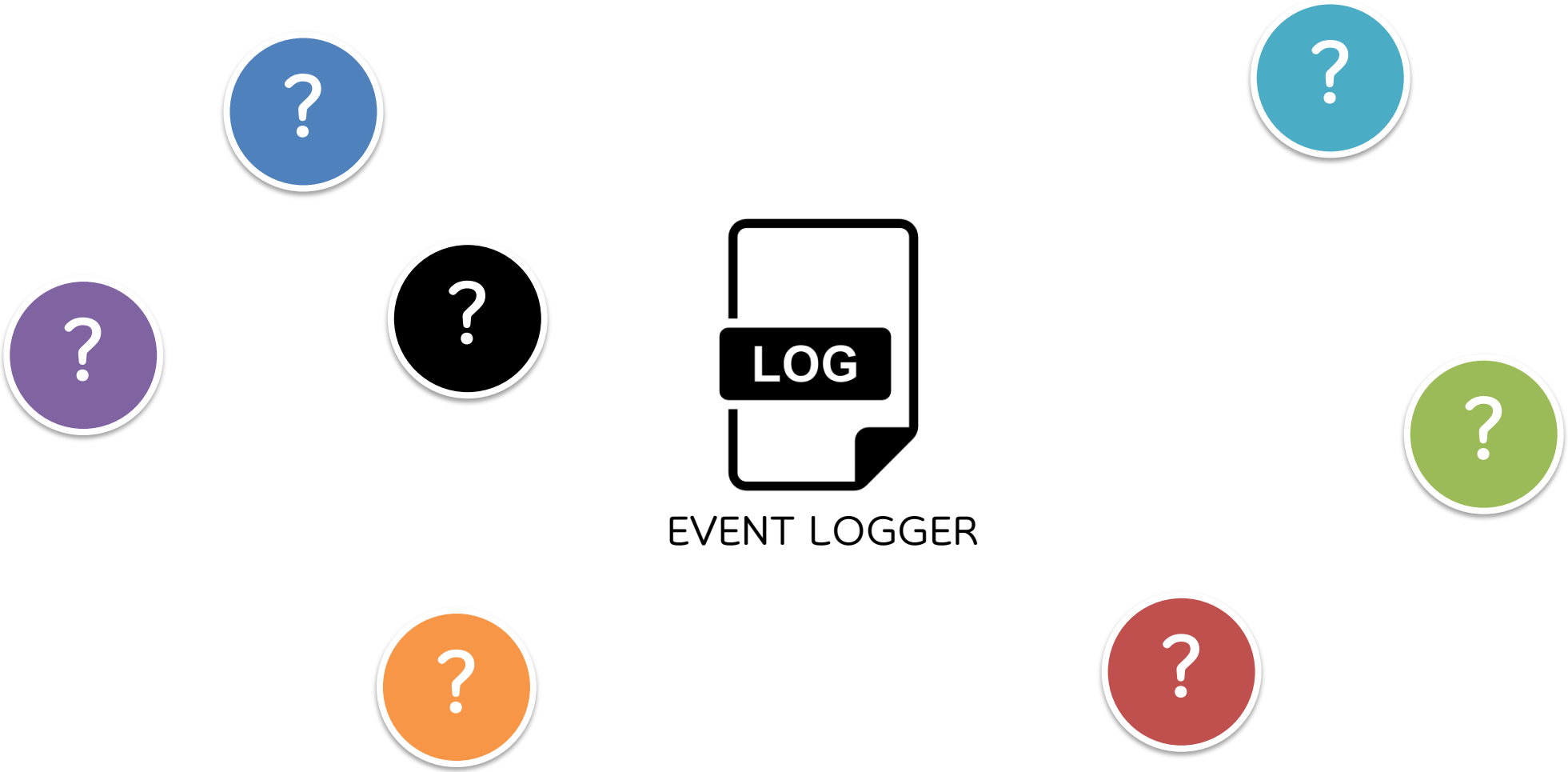
2020 SCB SUMMER PROJECT

실시간 파일 이벤트 로거 제작

CONTENTS

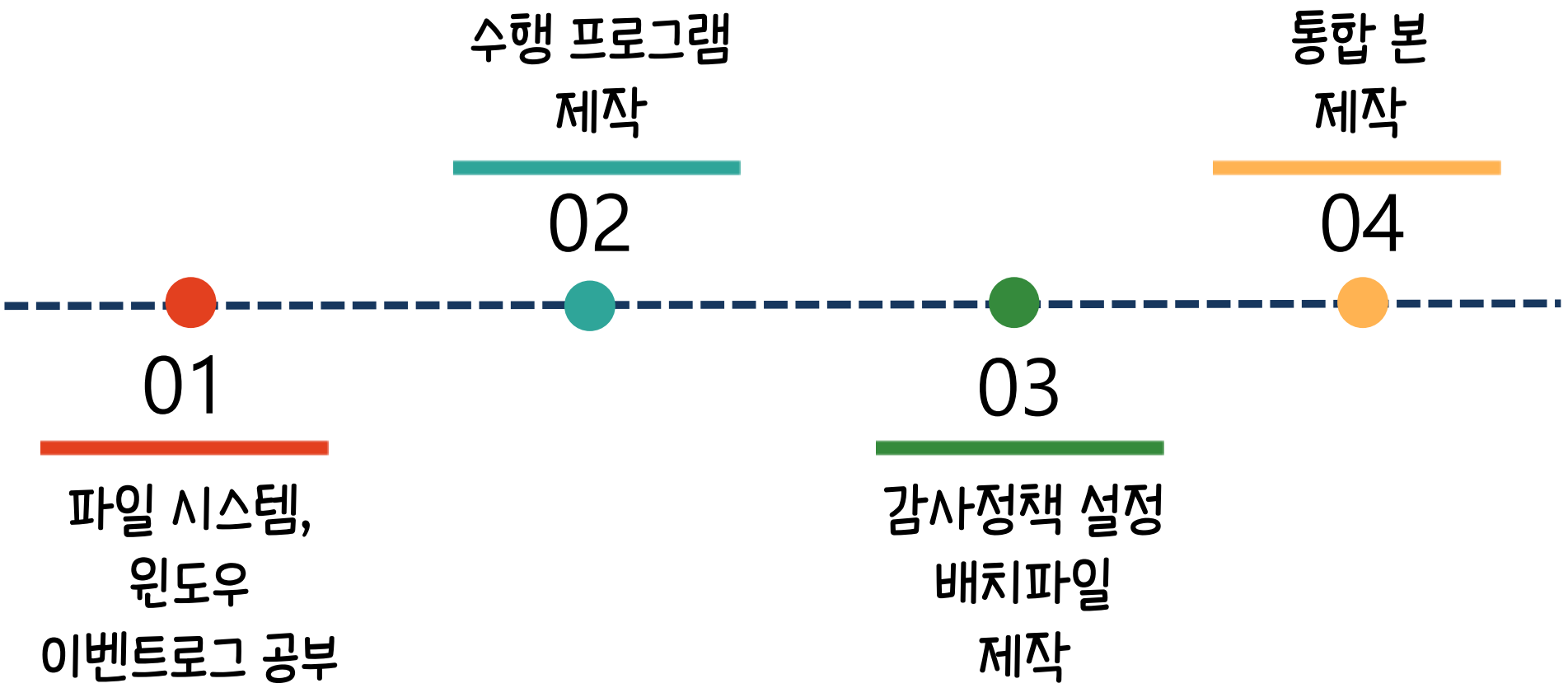
01 프로젝트 개요 – 주제 선정 배경



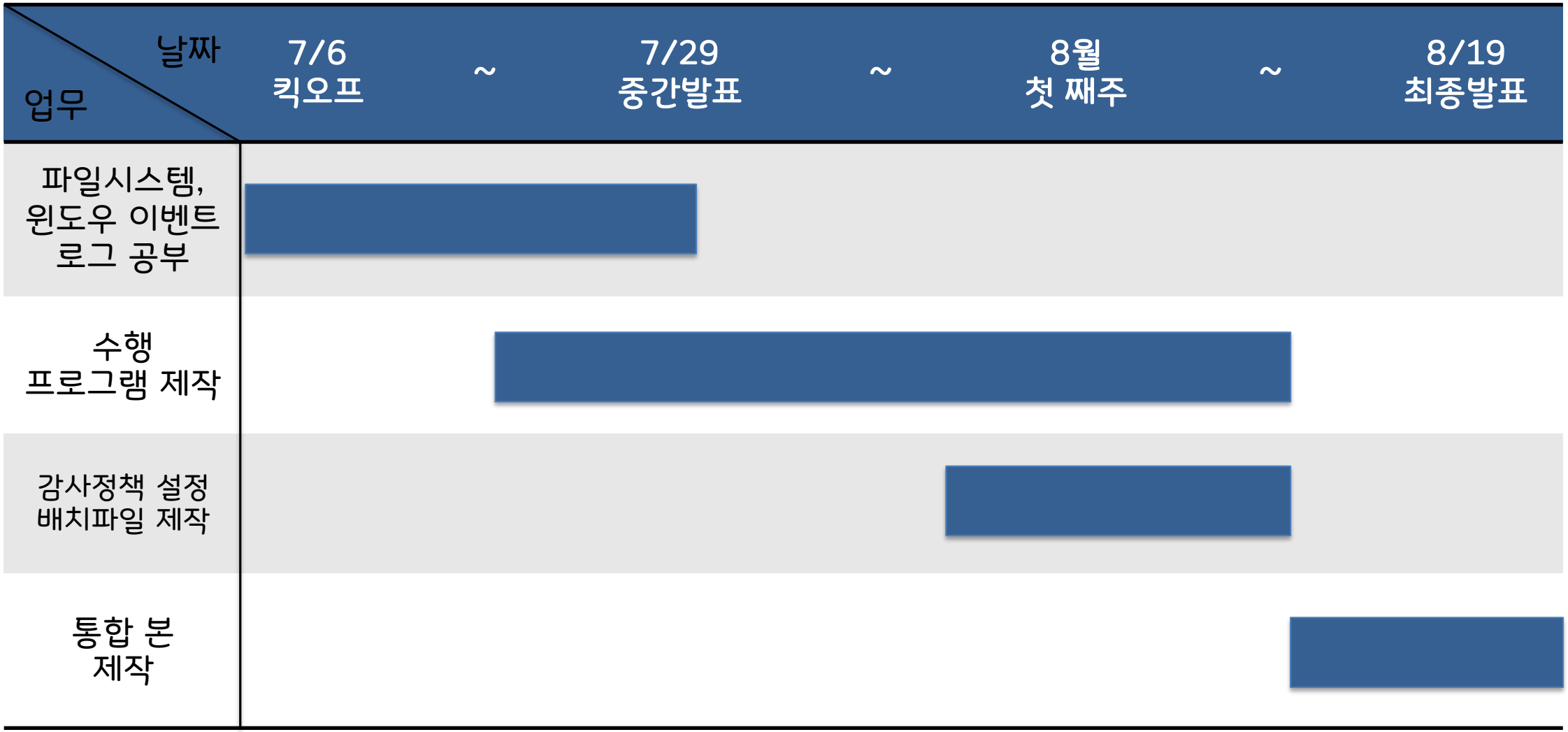
01 프로젝트 개요 - 목표

> “ 지정한 경로에 변경이 생길 시 이를 알리고
> 로그 기록을 볼 수 있게 하는 프로그램 ”

“관제 프로그램”


수행 프로그램
제작
02
통합 본
제작
04
01
파일 시스템,
윈도우
이벤트로그 공부
03
감사정책 설정
배치파일
제작

02 프로젝트 설명 - 프로젝트 일정



02 프로젝트 설명 - 역할 분담

김평안

PM
프로젝트 총괄

김수현

파이썬 개발

최송이

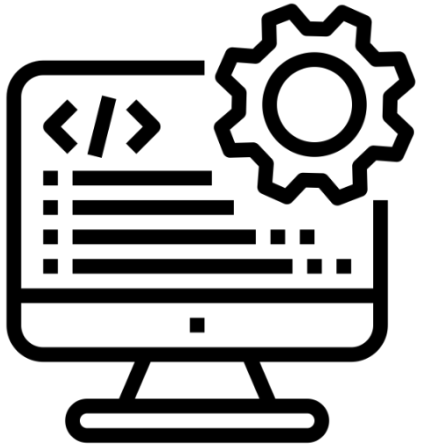
배치파일 개발

프로젝트 수행 결과 - 주요 기능

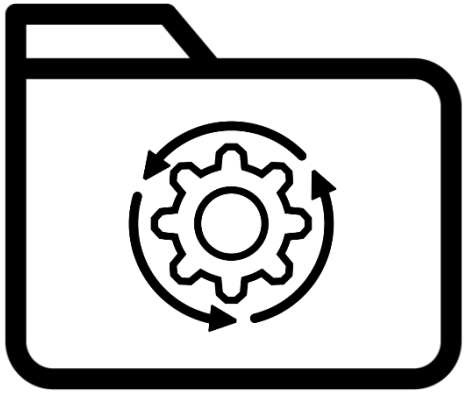
1. 지정된 경로 내에 변동사항 기록

2. 변동사항 알림

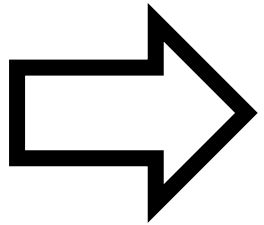
프로젝트 수행 결과 - 구성 및 상세 동작

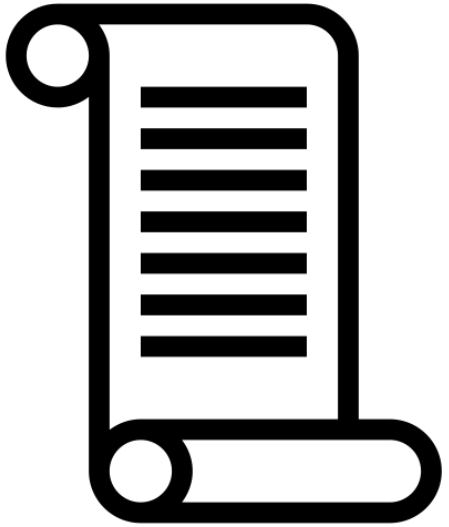


감사정책 설정



감사할 경로 지정





로그 작성

03 프로젝트 수행 결과 - 구성 및 상세 동작

```
@echo off
>nul 2>&1 "%SYSTEMROOT%₩system32₩cacls.exe" "%SYSTEMROOT%₩system32₩config₩system"
if '%errorlevel%' NEQ '0' (
    goto UACPrompt
) else ( goto gotAdmin )
:UACPrompt
    echo Set UAC = CreateObject^("Shell.Application"^) > "%temp%₩getadmin.vbs"
    set params = %*:"=""
    echo UAC.ShellExecute "cmd.exe", "/c %~s0 %params%", "", "runas", 1 >> "%temp%₩getadmin.vbs"
    "%temp%₩getadmin.vbs"
    rem del "%temp%₩getadmin.vbs"
    exit /B
:gotAdmin
pushd "%CD%"
    CD /D "%~dp0"
```

관리자 권한을 얻는 부분

```
echo [Event Audit] > hi.inf
echo AuditObjectAccess = 1 >> hi.inf
echo AuditAccountManage = 1 >> hi.inf
echo AuditAccountLogon = 1 >> hi.inf
echo AuditPrivilegeUse = 1 >> hi.in
echo AuditDSAccess = 1 >> hi.inff
echo AuditLogonEvents = 1 >> hi.inf
echo AuditSystemEvents = 1 >> hi.inf
echo AuditPolicyChange = 1 >> hi.inf
echo AuditProcessTracking = 1 >> hi.inf
echo [Version] >> hi.inf
echo signature= '$CHICAGO$' >> hi.inf
echo Revision=1 >> hi.inf
secedit /configure /db sysmain.sdb /cfg hi.inf
del hi.inf
del sysmain.sdb
```

감사정책을 적용하는 부분

```
python ./Log_Parsing.py

exit
```

파이썬 파일을 실행 후, 종료

03 프로젝트 수행 결과 - 구성 및 상세 동작

```python
def Forder_Check():
    Forder_Check = os.path.exists("SCB_Log")
    if Forder_Check == False:
        dir_path = "./"
        dir_name = "SCB_Log"
        os.mkdir(dir_path + "/" + dir_name + "/")
        DeleteLog_txt()
        ReadChangeLog_txt()
        DetailLog_txt()
```

로그를 저장할 폴더 생성

```python
def DeleteLog_txt():
    f = open("./SCB_Log\\DeleteLog(4660).txt", "w")
    f.write("\t\t<< Delete Log >>\n\n")
    f.write("*" * 50)
    f.close()

def ReadChangeLog_txt():
    f = open("./SCB_Log\\ReadChangeLog(4663).txt", "w")
    f.write("\t\t<< Read, Change Log >>\n\n")
    f.write("*" * 50)
    f.close()

def DetailLog_txt():
    f = open("./SCB_Log\\DetailLog.txt", "w")
    f.write("\t\t<< Detail Log >>\n\n")
    f.write("*" * 50)
    f.close()
```

로그를 기록할 메모장 생성

03 프로젝트 수행 결과 - 구성 및 상세 동작

```
def DeleteLog_Notepad():
    destFile = r"./SCB_Log\\DeleteLog(4660).txt"
    with open(destFile, 'a') as f:
        winsound.PlaySound('sound.wav', winsound.SND_FILENAME)
        print("\nLog Number      : {}".format(i), file=f)
        print("\nEvent ID        : {}".format(evt.EventID), file=f)
        print("\nTime Generated : {}".format(evt.TimeGenerated), file=f)
        f.write('\n파일 또는 폴더가 삭제되었습니다.\n')
        f.write('\nThe file or folder has been deleted.\n')
        f.write('*' * 50)

def ReadChangeLog_Notepad():
    destFile = r"./SCB_Log\\ReadChangeLog(4663).txt"
    with open(destFile, 'a') as f:
        print("\nLog Number      : {}".format(j), file=f)
        print("\nEvent ID        : {}".format(evt.EventID), file=f)
        print("\nTime Generated : {}".format(evt.TimeGenerated), file=f)
        f.write("\n파일 및 폴더가 열렸거나 수정되었을 수 있습니다.\n")
        f.write("\nThe files and folders may have been opened or modified.\n")
        f.write('*' * 50)

def DetailLog_Notepad():
    destFile = r"./SCB_Log\\DetailLog.txt"
    with open(destFile, 'a') as f:
        print("\nLog Number      : {}".format(k), file=f)
        print("\nEvent Category : {}".format(evt.EventCategory), file = f)
        print("\nEvent Type      : {}".format(evt.EventType), file = f)
        print("\nEvent ID        : {}".format(evt.EventID), file = f)
        print("\nTime Generated : {}".format(evt.TimeGenerated), file = f)
        print("\nSource Name     : {}".format(evt.SourceName), file = f)
        data = evt.StringInserts

        if data:
            f.write("\nEvent Data :")
            for msg in data:
                f.write("\n")
                f.write(msg)


        f.write('*' * 50)
```

알림 음 재생

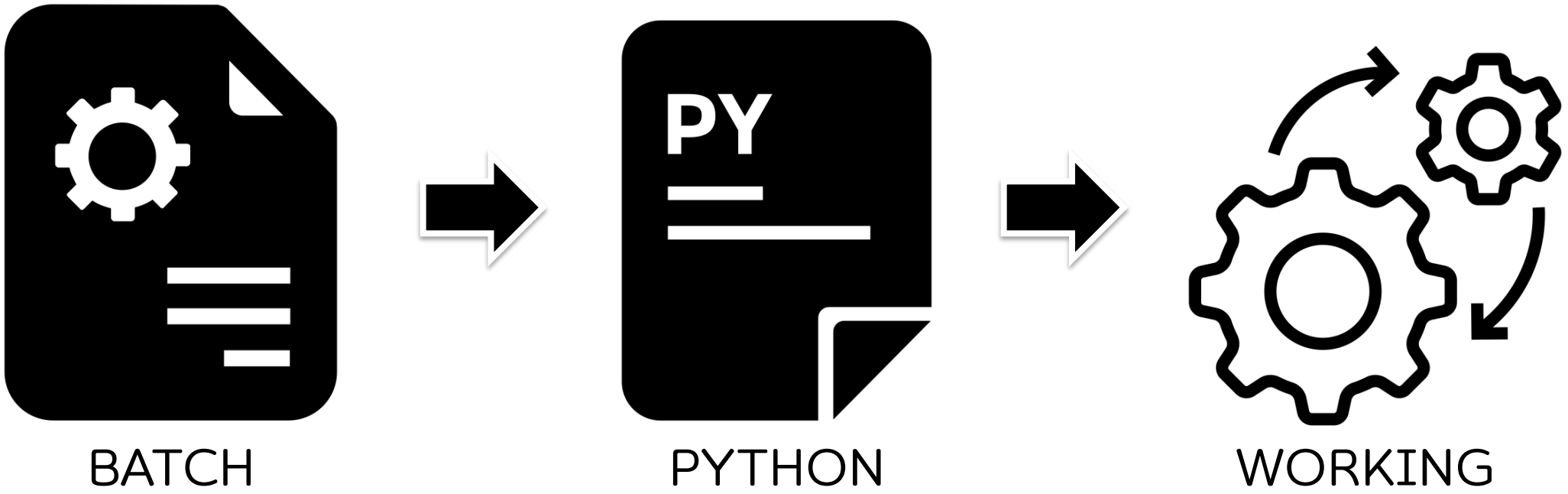
기록된 로그를 메모장에 저장

03 프로젝트 수행 결과 - 구성 및 상세 동작

```
if __name__ == "__main__":
    Forder_Check()
    win32gui.ShowWindow(win32console.GetConsoleWindow(), 0)

    server = 'localhost'
    logtype = 'Security'

    hand = wevt.OpenEventLog(server,logtype)
    flags = wevt.EVENTLOG_FORWARDS_READ|wevt.EVENTLOG_SEQUENTIAL_READ
    total = wevt.GetNumberOfEventLogRecords(hand)

    i = 0
    j = 0
    k = 0

    while True:
        events = wevt.ReadEventLog(hand, flags,0)

        if events:
            for evt in events:
                today = datetime.datetime.now().date()
                day_ago = today - datetime.timedelta(days=1)
                if evt.EventID == 4660:
                    i = i + 1
                    DeleteLog_Notepad()
                    DetailLog_Notepad()
                if evt.EventID == 4663:
                    j = j + 1
                    k = k + 1
                    ReadChangeLog_Notepad()
                    DetailLog_Notepad()
```

Main 함수

프로젝트 수행 결과 - 구성 및 상세 동작

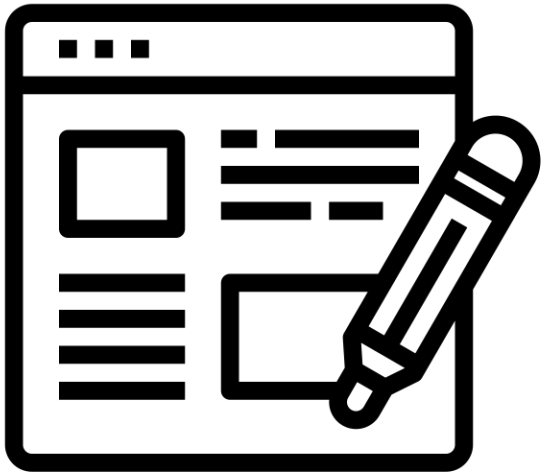


프로젝트 수행 결과 - 결과

1. 관련 분야 종사자에게 도움이 되는 기능을 제공

2. 보안 분야를 잘 모르는 사람에게 보안분야 보조

04 기대효과 및 계획 - 추후 계획



블로그 포스팅

감지 범위 추가

GUI추가로 편의성 개선

THANK YOU

Q&A