<http://ourcstory.tistory.com/97>

[**프로그래밍/Python**](http://ourcstory.tistory.com/category/%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D/Python)

### [[Python] Python Logging 파이썬 로깅 모듈](http://ourcstory.tistory.com/97)

쌍쌍바나나 2016.05.20 10:51

## 들어가며

  파이썬 표준 라이브러리인 logging은 아주 강력한놈입니다. 이제 다시는... print문과 file.write를 이용해서 로그를 남기지 마세요. 우리 스마트하게 로그를 남깁시다. logging은 다양한 기능을 제공하고 있습니다. 스트림과 파일에 동시에 로그를 남기거나, 로그 레벨에 따라 출력되는 로그를 제한 할 수 있습니다. 또한 로그는 축적되는 양이 많기 때문에 예외를 처리해야 하는 내용이 많습니다. 그 중에 파일 크기가 일정 크기를 넘어가면 새로운 로그 파일을 자동으로 생성하는 기능 또한 제공합니다. 아래 래퍼런스를 통해 더 자세한 내용을 확인 하실 수 있습니다.  Singleton Pattern으로 CustomLogger를 작성하여 사용한 사례를 포스팅한 내용이 있습니다. 어떤식으로 로깅 인스턴스를 사용하는지 확인이 가능합니다. **[[ CustomLogger 생성 및 활용 바로가기]](http://ourcstory.tistory.com/105" \t "_blank)**

### 로그 레벨 제한 및 로그 출력

로깅 정보는 로그의 레벨에 따라서 출력을 제한 할 수 있습니다.  
DEBUG > INFO > WARNING > ERROR > Critical

import logging

# 파일로 남기기 위해 filename='./log/test.log' parameter로 추가한다.

logging.basicConfig(filename='./log/test.log', level=logging.DEBUG)

logging.debug("debug")

logging.info("info")

logging.warning("warning")

logging.error("error")

logging.critical("critical")

'''

# ./log/test.log 결과

DEBUG:root:debug

INFO:root:info

WARNING:root:warning

ERROR:root:error

CRITICAL:root:critical

'''

\*여기서 주의해야할 사항은 ./log/test.log의 파일은 덮어 씌워지는게 아닌, 기존 로그위에 이어서 로그가 누적됩니다.

### 스트림과 파일에 동시에 로그를 남기기

  로그를 콘솔에도 출력하고 싶고, 파일에도 동시에 남기고 싶다면 아래와 같이 fileHandler, streamHandler를 생성해서 logger에 Handler를 추가해주면 됩니다. 간단하게 몇줄 안되는 코드로 파일에 동시에 남기면서, 콘솔에도 현재 로그 상태가 출력되는 로거를 생성했습니다.

import logging

import logging.handlers

# logger 인스턴스를 생성 및 로그 레벨 설정

logger = logging.getLogger("crumbs")

logger.setLevel(logging.DEBUG)

# fileHandler와 StreamHandler를 생성

fileHandler = logging.FileHandler('./log/my.log')

streamHandler = logging.StreamHandler()

# Handler를 logging에 추가

logger.addHandler(fileHandler)

logger.addHandler(streamHandler)

# logging

logger.debug("debug")

logger.info("info")

logger.warning("warning")

logger.error("error")

logger.critical("critical")

### 로그 포매팅

  로그 포매팅이라고 하면, 내가 로그를 남길때 앞에 쓰여지는 형식의 포맷을 정하는 것을 말합니다. 위와 같이 로그를 남길꺼면 파일 open을 통해서 write를 하는 방식과 다를게 없습니다. 좀더 파워풀하게 해보자!! 예를 들어 우리가 원하는 로그의 형태가 아래와 같을 때

fomatter = logging.Formatter('[%(levelname)s|%(filename)s:%(lineno)s] %(asctime)s > %(message)s')   
[DEBUG|logger.py:24] 2016-05-20 20:00:40, 630 >  DEBUG

그렇다면 이때 Formatter를 사용하면 된다. 추가적으로 변경을 하고 싶으면 참고 사이트에서 formatter의 attributes를 확인하면 된다. **[[바로가기]](https://docs.python.org/3/library/logging.html" \l "logrecord-attributes" \t "_blank)**

import logging

import logging.handlers

# logger 인스턴스를 생성 및 로그 레벨 설정

logger = logging.getLogger("crumbs")

logger.setLevel(logging.DEBUG)

# formmater 생성

formatter = logging.Formatter('[%(levelname)s|%(filename)s:%(lineno)s] %(asctime)s > %(message)s')

# fileHandler와 StreamHandler를 생성

fileHandler = logging.FileHandler('./log/my.log')

streamHandler = logging.StreamHandler()

# handler에 fommater 세팅

fileHandler.setFormatter(formatter)

streamHandler.setFormatter(formatter)

# Handler를 logging에 추가

logger.addHandler(fileHandler)

logger.addHandler(streamHandler)

# logging

logger.debug("debug")

logger.info("info")

logger.warning("warning")

logger.error("error")

logger.critical("critical")

'''

출력결과

[DEBUG|input.py:24] 2016-05-20 10:37:06,656 > debug

[INFO|input.py:25] 2016-05-20 10:37:06,657 > info

[WARNING|input.py:26] 2016-05-20 10:37:06,657 > warning

[ERROR|input.py:27] 2016-05-20 10:37:06,657 > error

[CRITICAL|input.py:28] 2016-05-20 10:37:06,657 > critical

'''

### 파일이 너무 큰 경우 자동으로 새로운 파일을 생성하기

python에서는 RotatingFileHandler가 제공이 됩니다. 위에서 사용한 FileHandler대신에 사용하면 10MB의 log파일이 최대 10개까지 생성되도록 설정이 됩니다. 이렇게 되면 최대 100MB이하의 최신 로그가 남게 됩니다. 이런 로그 파일의 크기 또한 조절해주니 대단하네요.

# file max size를 10MB로 설정

file\_max\_bytes = 10 \* 1024 \* 1024

#fileHandler = logging.handlers.RotatingFileHandler(filename='./log/test.log', maxBytes=file\_max\_bytes, backupCount=10)

 [참고] https://docs.python.org/3/library/logging.html

출처: <http://ourcstory.tistory.com/97> [쌍쌍바나나의 블로그]