

Log 라이브러리 사용



Daily Contents

Stdout 대신 logger를 사용해보자



함께가요 미래로!
Enabling People
Education for Future Generations

Standard Out/Error



- ✓ 표준 입출력 활용
- ✓ Unix에서 나온 개념
 - Linux/Windows 모두 활용 가능
- ✓ "Hello World!" 는 어디서 출력되었나?
 - 충분히 배운 것 같지만....

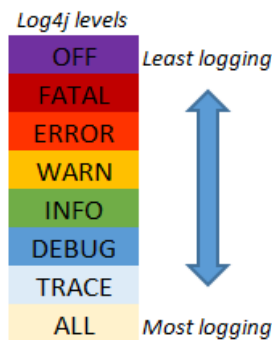
```
System.out.println("Hello, World!")
```

Log 라이브러리를 사용하면?



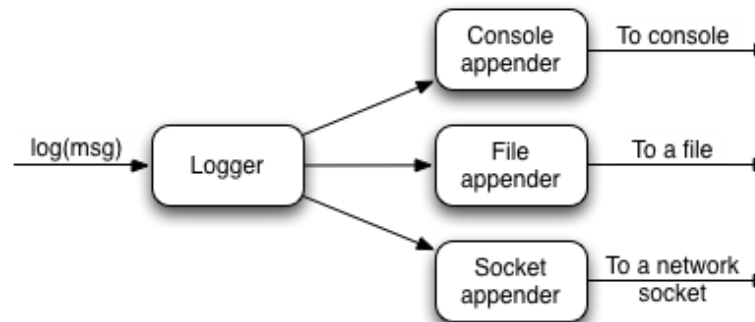
✓ Level 적용 가능

- 특정 레벨은 따로 찾을 수 있음
- 지정 레벨 이상의 로그만 출력 가능



✓ 출력 스트림 선택

- 콘솔 출력은 당연히 중요
- (롤링)파일, 소켓, 이메일, 시스템 로그 등 가능
- Log 시스템과 연동 가능



✓ Formatter 지정

- [날짜-시간] 레벨 - 내용 등의 형식을 지정 가능

✓ Thread-safe

- 멀티 스레드 환경에서도 고려되어 있음

```
[~/code/caipirinha/caipirinha]% python log.py
[06:03:30] test test_func():179      debug msg
[06:03:30] test test_func():180      info msg
[06:03:30] test test_func():181      warn msg
[06:03:30] test test_func_2():184     error msg
[06:03:30] test test_func_2():185     critical msg
[06:03:30] test test_func_2():190     Opa!
Traceback (most recent call last):
  File "log.py", line 188, in test_func_2
    raise RuntimeError("Opa!")
RuntimeError: Opa!
[venv][master]
[~/code/caipirinha/caipirinha]%
```

설정 예



```
<!-- Author: Crunchify.com -->
<Appenders>
  <Console name="Console" target="SYSTEM_OUT">
    <PatternLayout pattern="%d{YYYY-MM-dd HH:mm:ss} [%t] %-5p %c{1}:%L - %msg%n" />
  </Console>

  <RollingFile name="RollingFile" filename="log/CrunchifyTest.log"
    filepattern="${logPath}/%d{YYYYMMddHHmmss}-fargo.log">
    <PatternLayout pattern="%d{YYYY-MM-dd HH:mm:ss} [%t] %-5p %c{1}:%L - %msg%n" />
    <Policies>
      <SizeBasedTriggeringPolicy size="100 MB" />
    </Policies>
    <DefaultRolloverStrategy max="20" />
  </RollingFile>
</Appenders>
<Loggers>
  <Root level="info">
    <AppenderRef ref="Console" />
    <AppenderRef ref="RollingFile" />
  </Root>
</Loggers>
</Configuration>
```

보다 나은 개발을 위해



함께가요 미래로!
Enabling People

Education for Future Generations

- ✓ 유연한 설정
 - ✓ 대부분의 언어에 존재
 - ✓ 어렵지 않다
- ✓ 물론 메시지 내용이 가장 중요!!!

