**1. log4j 란?**

자바 어플리케이션에서 빠르고 효과적으로 로깅 할 수 있도록 도와주는 오픈소스이다.

**2. log4j 구조**

log4j는 크게 3가지의 큰 뼈대를 가지고 있다.

(1) Logger : logging 메시지를 Appender에 전달.

(2) Appender : 전달된 로깅 메시지를 파일, 콘솔등 어디에 저장 할 지 지정하는 역할을 한다.

(3) Layout :Appender가 어디에 출력 할 것인지 결정했다면 어떠한 형식으로 출력을 할 지

출력 layout을 결정

==== layout Options ====

\* %d :로깅 이벤트가 일어난 날자(date)

\* %p :로깅 이벤트의 priority

\* %t :로깅 이벤트를 생성한 스레드 이름

\* %c :로깅 이벤트의 category

\* %F :로깅요청을 일으킨 파일 이름

\* %L :로깅요청을 일으킨 파일의 행번호

\* %x :로깅이벤트를 발생시킨 스레드에 관련된 내포검사항목

\* %C :로깅요청을 일으킨 호출자의 완전한 클래스이름

\* %M :로깅요청을 일으킨 메소드

\* %m :메세지

\* %n : 플랫폼 독립적인 개행문자

\* %l : 소스코드의 위치정보를 출력한다. %C. %M(%F:%L) 의 축약형

**3. log4j level**

logging 레벨은 TRACE < DEBUG < INFO < WARN < ERROR < FATAL 이며,

1) FATAL : 가장 크리티컬한 에러가 났을 때 사용.

2) ERROR : 일반 에러가 났을 경우 사용.

3) WARN : 에러는 아니지만 주의할 필요가 있을 때 사용.

4) INFO : 일반 정보를 나타날 때 사용.

5) DEBUG : 일반 정보를 상세히 나타낼 때 사용.

6) TRACE : 가장 레벨이 낮은 단계, 보통 DEBUG로 많이 하며 TRACE는 본적이...;;

debug 레벨로 했다면 INFO~FATAL까지 모두 logging이 되어집니다.

운영모드라면 INFO레벨로 하면 됩니다.

--> 설정파일에서 info부터 나오길 바란다면

:: log4j.category.TestFileCompare=INFO, Console, FileCompareAppender

이렇게 설정하면 됩니다.(DEBUG대신 INFO)

--> 매번 새로운 log 파일을 만들고 싶다면

:: log4j.appender.FileCompareAppender.Append=fasle (true대신에 false로 설정)

--> **log4j의 설정파일은 log4j.properties를 클래스 패스로 잡아둔다**

**그렇기에이클립스에서는src 밑에 들어가게되고톰켓에서는 classes 폴더 밑에 배포된다.**

--> log4j의 설정파일을 작성하기 전에 우선 라이브러리 하나를 추가해야한다.

<http://logging.apache.org/....org.apachae.log4j> 안에 있는 라이브러리 가지고 온다

해당 사이트로 가서 하나 가져온다