

# 한글 NLP 를 위한 Python 환경 구성 (for Windows)

김현중 (soy.lovit@gmail.com)

고태훈 (taehoon.ko@gmail.com)

# Install JDK, JPytype1 and KoNLPy

- **KoNLPy**

- 공식 홈페이지: <http://konlpy.org/ko/v0.4.4/>
- Hommage to KoNLP (한국어 처리를 위한 R 패키지)
- 각종 한국어 형태소 분석기를 Python에서 사용할 수 있게 만들어주는 패키지
- 많은 한국어 형태소 분석기가 Java 기반으로 되어 있기 때문에, Python 환경에서 Java 프로그램을 실행하게 해주는 “JPytype1”을 설치해야 함.

- **JPytype1**

- Python환경에서 Java 프로그램을 실행해주는 패키지
- 미리 Java Development Kit (JDK)를 설치해야만 사용 가능합니다!

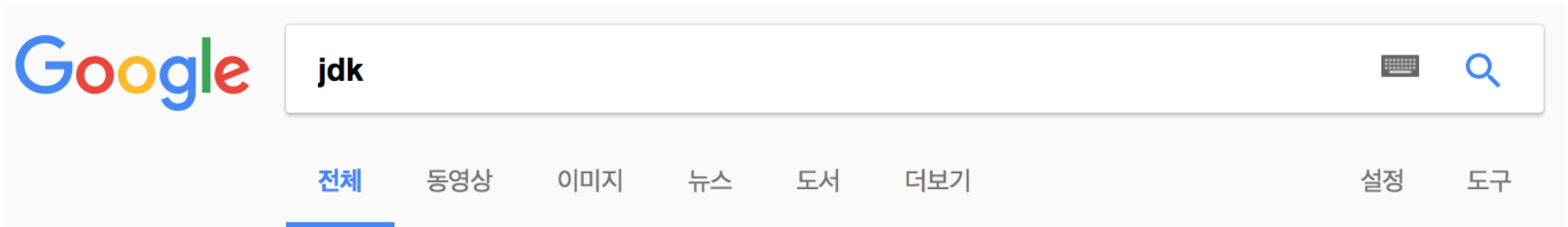
## Dependencies



# Install JDK, JPytype1 and KoNLPy

- KoNLPy의 공식 설치 문서를 따라해주시기 바랍니다.
  - <http://konlpy.org/ko/v0.4.4/install/>
    - 공식 문서에는 JDK 1.7을 설치하라고 되어 있음.
    - (제 경험) 저는 JDK 1.8을 사용해야 하는 환경을 사용 중이어서 JDK 1.8을 설치하였는데 사용하는 데에 문제는 (아직까지) 없었습니다.
    - 밑에 운영체제 별로 설치할 때의 유의점을 적어 두었습니다.

# Install JDK



검색결과 약 14,000,000개 (0.26초)

구글에서 “jdk” 검색 후, 아래 페이지(“JAVA SE Development Kit 8”)로 이동

관련검색: [jdk 설치](#) [jdk 다운로드](#) [eclipse](#) [mysql](#) [notepad++](#)

## Java SE Development Kit 8 - Downloads - Oracle

[www.oracle.com](#) › [Java](#) › [Java SE](#) ▼ [이 페이지 번역하기](#)

Download **JDK 8**, a development environment for building applications and components using the Java programming language.

이 페이지를 여러 번 방문했습니다. 최근 방문 날짜: 16. 12. 15

## Java SE - Downloads | Oracle Technology Network | Oracle

[www.oracle.com](#) › [Java](#) › [Java SE](#) ▼ [이 페이지 번역하기](#)

Java SE downloads including: **Java Development Kit (JDK)**, **Server Java Runtime Environment (Server JRE)**, and **Java Runtime Environment (JRE)**.

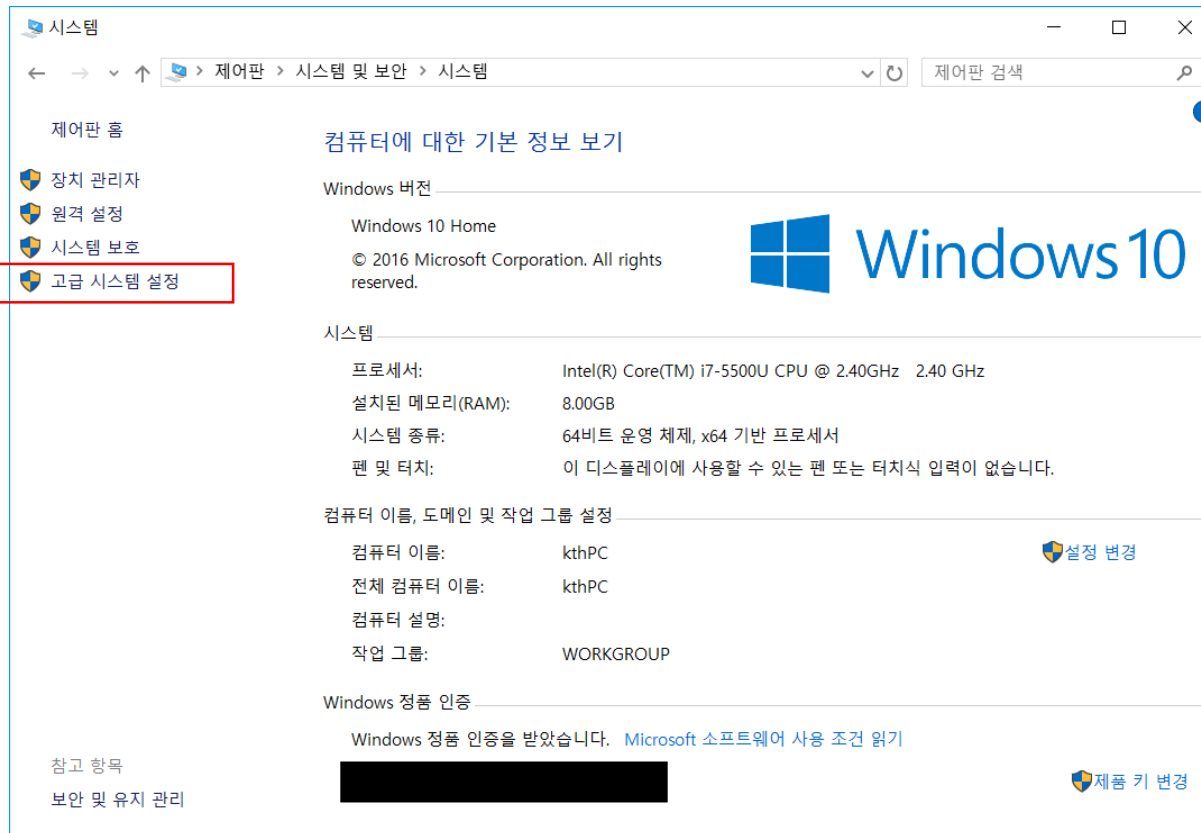
# Install JDK

- 위에서 “Accept License Agreement”를 클릭한 후, 자신의 OS와 bit 수를 고려하여 알맞은 JDK를 다운받아 설치
- JDK 8u111과 8u112 중 어떤 것도 상관없음

Java SE Development Kit 8u111		
You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.78 MB	<a href="#">jdk-8u111-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	74.73 MB	<a href="#">jdk-8u111-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	160.35 MB	<a href="#">jdk-8u111-linux-i586.rpm</a>
Linux x86	175.04 MB	<a href="#">jdk-8u111-linux-i586.tar.gz</a>
Linux x64	158.35 MB	<a href="#">jdk-8u111-linux-x64.rpm</a>
Linux x64	173.04 MB	<a href="#">jdk-8u111-linux-x64.tar.gz</a>
Mac OS X	227.39 MB	<a href="#">jdk-8u111-macosx-x64.dmg</a>
Solaris SPARC 64-bit	131.92 MB	<a href="#">jdk-8u111-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	93.02 MB	<a href="#">jdk-8u111-solaris-sparcv9.tar.gz</a>
Solaris x64	140.38 MB	<a href="#">jdk-8u111-solaris-x64.tar.Z</a>
Solaris x64	96.82 MB	<a href="#">jdk-8u111-solaris-x64.tar.gz</a>
Windows x86	189.22 MB	<a href="#">jdk-8u111-windows-i586.exe</a>
Windows x64	194.64 MB	<a href="#">jdk-8u111-windows-x64.exe</a>

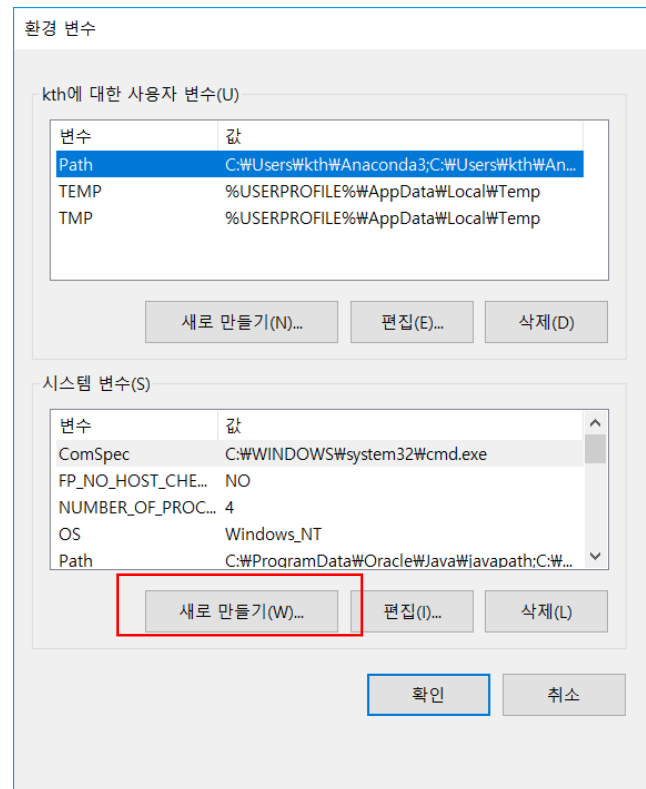
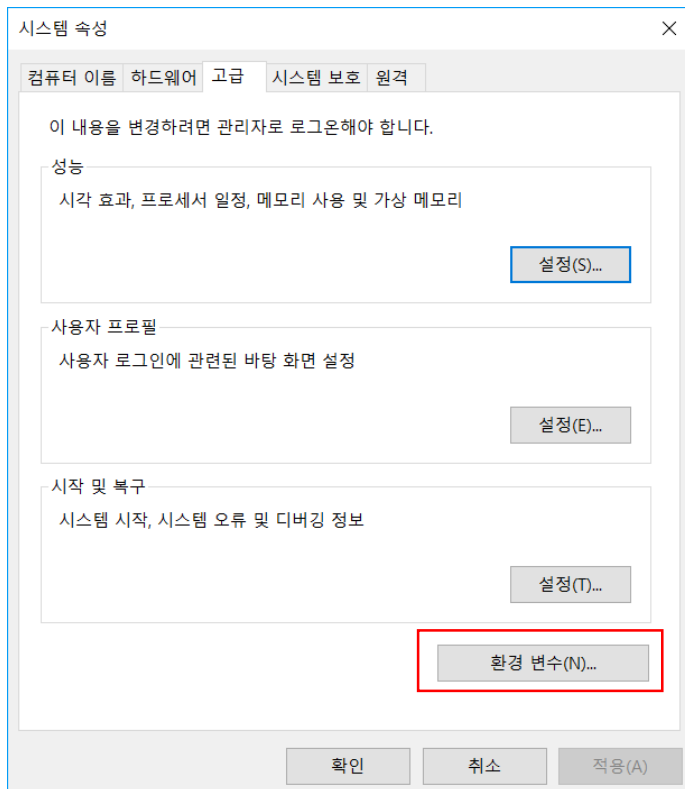
# Install JDK

- 주의사항: Windows를 사용 중인 경우,
  - JAVA\_HOME 환경 변수 만드는 방법 (<http://prolite.tistory.com/175> 참고)
    - 제어판 > 시스템 및 보안 > 시스템
    - 좌측 메뉴에서 “고급 시스템 설정” 클릭



# Install JDK

- 주의사항: Windows를 사용 중인 경우,
  - JAVA\_HOME 환경 변수 만드는 방법
    - “환경 변수” 클릭
    - [시스템 변수] 탭에서 “새로 만들기” 클릭



# Install JDK

- 주의사항: Windows를 사용 중인 경우,

- JAVA\_HOME 환경 변수 만드는 방법

- [변수 이름]에 **JAVA\_HOME** 입력

- [변수 값]에 **JDK 설치 경로**를 입력

- 각자 설치한 JDK의 버전에 따라 다를 수 있으므로 꼭 확인하시고 자신의 컴퓨터 상황에 맞게

입력해주세요 (보통 C:\Program Files\Java\jdk1.x.x.xxx 라는 디렉토리에 있음)

새 시스템 변수

변수 이름(N): JAVA\_HOME

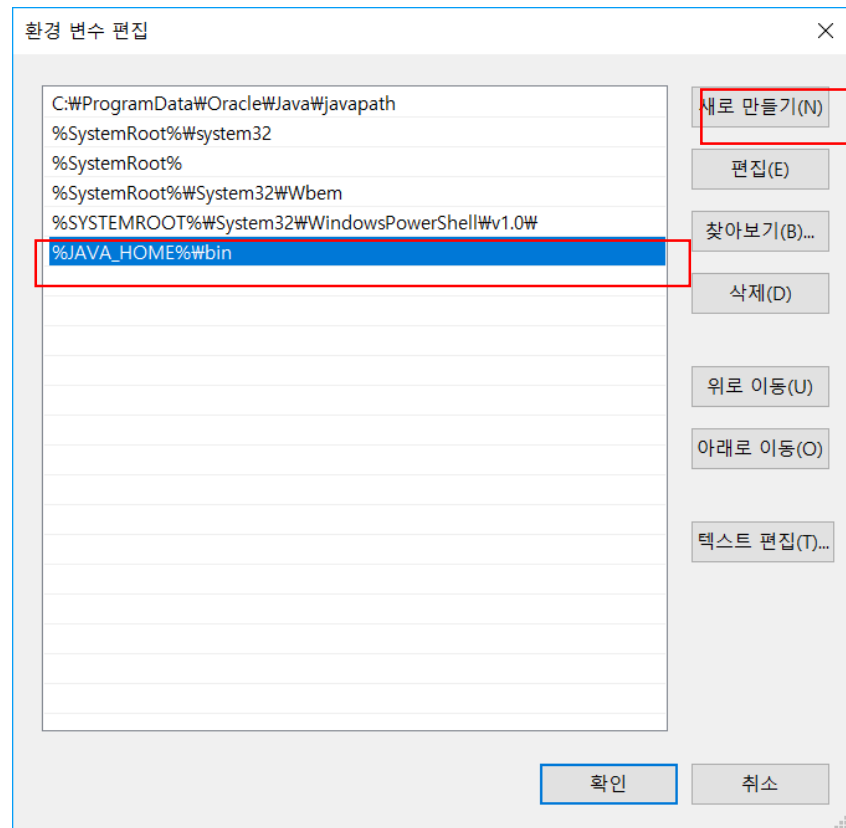
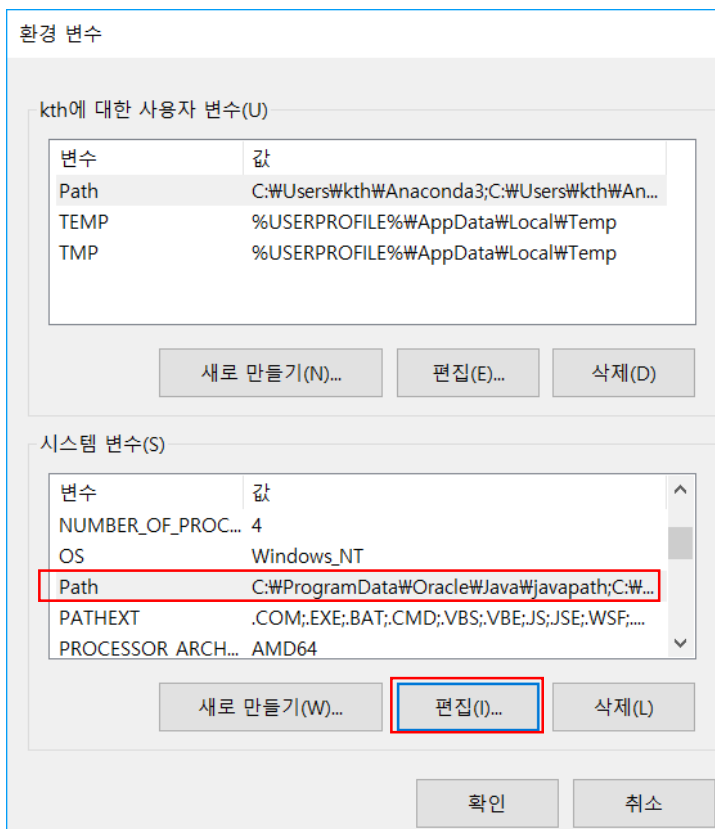
변수 값(V): C:\Program Files\Java\jdk1.8.0\_102

디렉터리 찾아보기(D)...    파일 찾아보기(F)...    **확인**    취소



# Install JDK

- 주의사항: Windows를 사용 중인 경우,
  - JAVA\_HOME 환경 변수 만드는 방법
    - 시스템 변수 목록에 Path 를 찾아 선택 후 “편집”을 누릅니다.
    - (For Windows 10) “새로 만들기”를 누른 후, **%JAVA\_HOME%\bin**를 추가

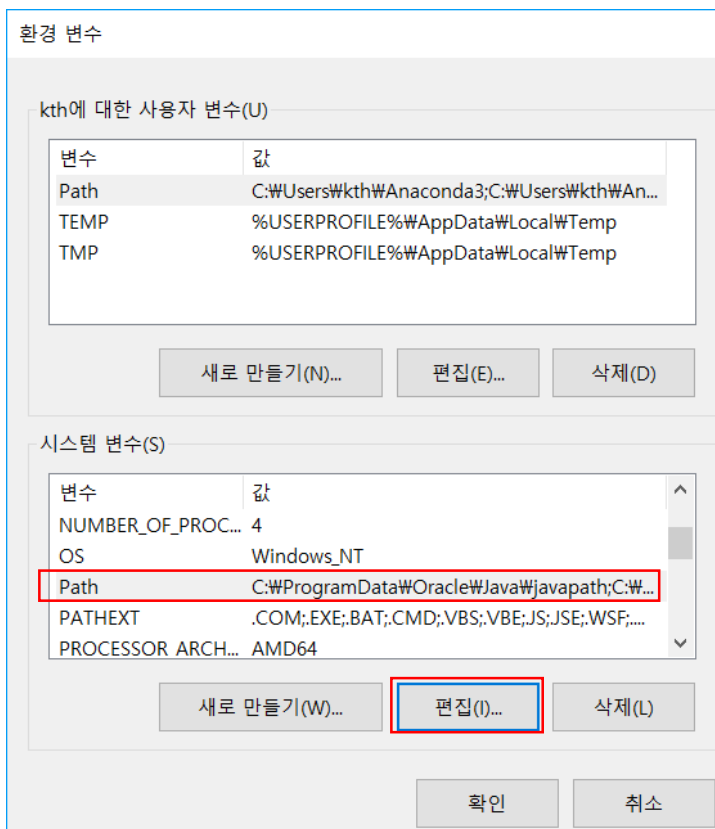


# Install JDK

- 주의사항: Windows를 사용 중인 경우,

- JAVA\_HOME 환경 변수 만드는 방법

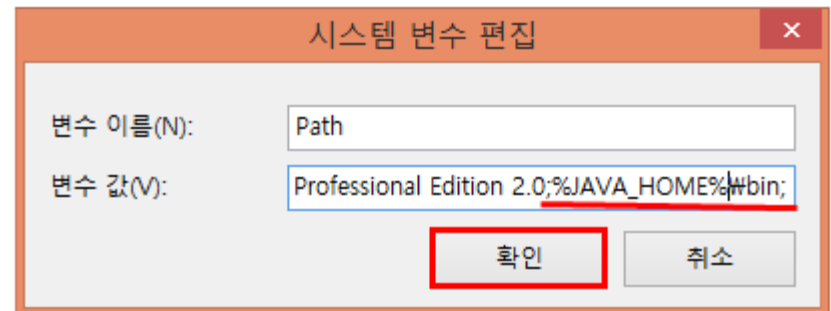
- (For Windows 7, 8) “새로 만들기”를 누른 후, 경로 끝에 `;%JAVA_HOME%\bin;`를 추가



16) 변수 값 `;%JAVA_HOME%\bin;`

여기서 변수 값은 맨 끝쪽에 씁니다.

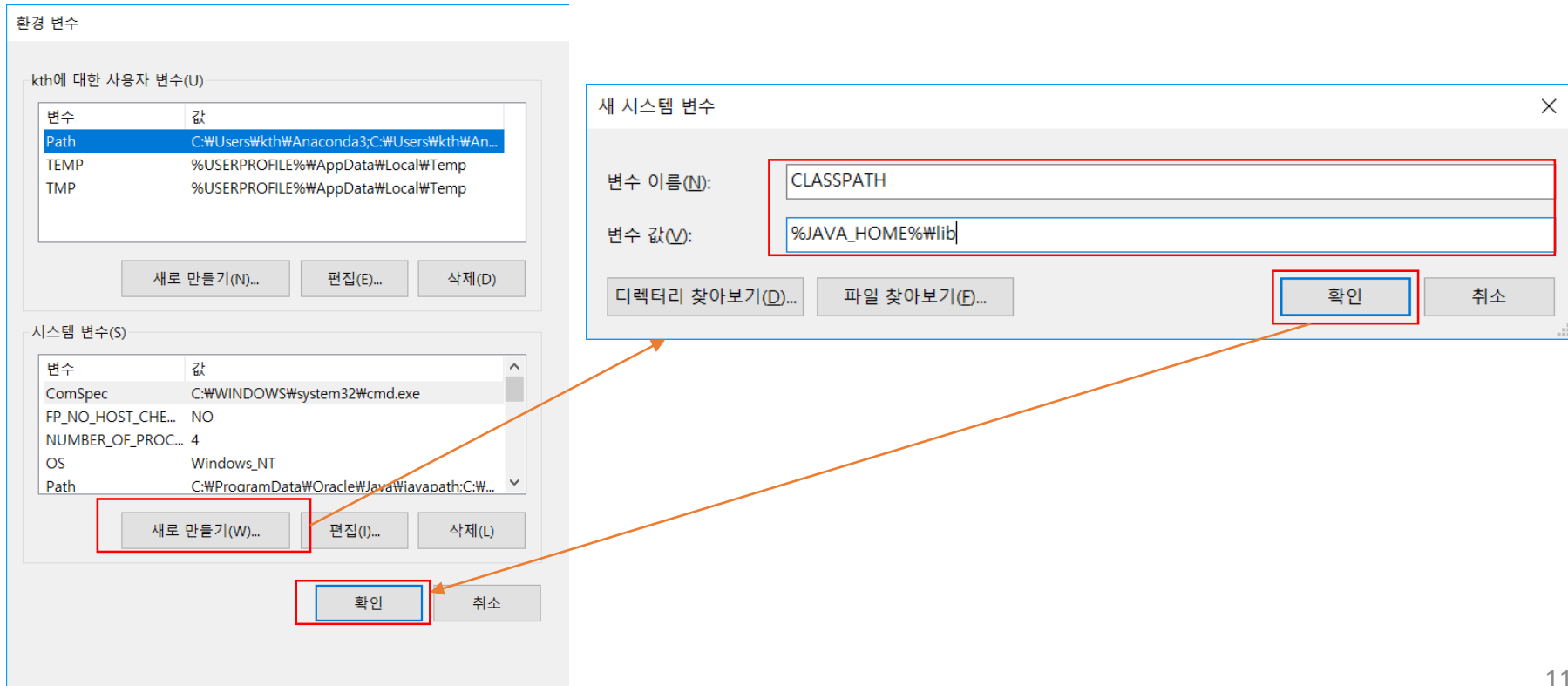
반드시 `;` 표시가 양옆에 있어야 합니다. 확인을 누릅니다.



<http://prolite.tistory.com/175> 참고

# Install JDK, JPy1 and KoNLPy

- 주의사항: Windows를 사용 중인 경우,
  - JAVA\_HOME 환경 변수 만드는 방법
    - 다시 “새로 만들기” 클릭
    - 변수 이름에 **CLASSPATH**, 변수 값으로 **%JAVA\_HOME%\lib** 입력 후 “확인” 누르기
    - [환경 변수] 창에 대해 “확인” 누르기



# Install JDK, JPy1 and KoNLPy

- 주의사항: Windows를 사용 중인 경우,

- JAVA\_HOME 환경 변수 만드는 방법

- 명령 프롬프트를 실행하여 **javac** 실행 → 잘 실행되는지 확인

- 명령 프롬프트에서 **where javac** 입력 → 아까 설치한 jdk 디렉토리 안의 파일인지 확인

```
명령 프롬프트
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Wkth>javac
Usage: javac <options> <source files>
where possible options include:
  -g               Generate all debugging info
  -g:none          Generate no debugging info
  -g:{lines,vars,source}  Generate only some debugging info
  -nowarn          Generate no warnings
  -verbose         Output messages about what the compiler is doing
  -deprecation     Output source locations where deprecated API is used
  -classpath <path>  Specify where to find user class files and
                    Specify where to find user class files and
  -cp <path>        Specify where to find input source files
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path>  Override location of bootstrap class files
  -extdirs <dirs>      Override location of installed extensions
  -endorseddirs <dirs>  Override location of endorsed standards path
  -proc:{none,only}  Control whether annotation processing and/or
                    -processor <class1>[,<class2>,<class3>...] Names of the annotation processors
  -processorpath <path> Specify where to find annotation processors
  -parameters      Generate metadata for reflection on method parameters
  -d <directory>    Specify where to place generated class files
  -s <directory>    Specify where to place generated source files
  -h <directory>    Specify where to place generated native header files
  -implicit:{none,class} Specify whether or not to generate class files
  -encoding <encoding> Specify character encoding used by source files
  -source <release>  Provide source compatibility with specified release
  -target <release>  Generate class files for specific VM version
```

```
명령 프롬프트
C:\Users\Wkth>where javac
C:\Program Files\Java\jdk1.8.0_102\bin\javac.exe

C:\Users\Wkth>
```

# Install JPyype1

- (for Windows) Download JPyype1 and install

- <http://www.lfd.uci.edu/~gohlke/pythonlibs/#jpype> 에 들어가서 Python 버전과 사용 중인 bit 수에 맞는 JPyype1에 대한 whl 파일을 다운로드
  - 예) Python 3.5 64bit를 사용하는 경우, 빨간색 네모의 파일을 다운로드
- 명령 프롬프트에서 파일을 다운로드한 폴더에 받은 패키지 설치
  - **activate tm** (앞서 만든 conda 환경 활성화)
  - **pip install JPyype1-0.6.1-xxxxxx-xxxxx.whl** (자신이 필요한 버전에 맞게)

JPyype allows full access to Java class libraries.

[JPyype1-0.6.1-cp27-none-win32.whl](#)

[JPyype1-0.6.1-cp27-none-win amd64.whl](#)

[JPyype1-0.6.1-cp34-none-win32.whl](#)

[JPyype1-0.6.1-cp34-none-win amd64.whl](#)

[JPyype1-0.6.1-cp35-none-win32.whl](#)

[JPyype1-0.6.1-cp35-none-win amd64.whl](#)

# [참고] Windows에서 Python 환경을 구성할 때,

- **Unofficial Windows Binaries for Python Extension Packages**

- URL: <http://www.lfd.uci.edu/~gohlke/pythonlibs>
- 주요 Python 패키지에 대해 윈도우 32bit와 64bit에 대한 지원을 해주는 repository
- Python 커뮤니티의 개발자들이 Windows에 그다지 친절하지 않습니다...
- 만약 특정 Python package를 설치할 때, Windows에서 잘 안 되는 문제가 있다면 이 사이트에서 whl파일을 다운로드 받은 후 pip로 설치해주시기 바랍니다.

# Install KoNLPy and other packages

- 명령 프롬프트에서 다음을 입력하여 KoNLPy를 설치
  - `pip install konlpy`
- 그 외에 더 설치할 패키지들 (wordcloud, Pillow, gensim)
  - `pip install wordcloud`
    - 윈도우의 경우, <http://www.lfd.uci.edu/~gohlke/pythonlibs/#wordcloud> 에서 whl 파일을 다운로드
  - `conda install Pillow` / `pip install Pillow`
  - `pip install gensim`