

# 형태소분석기의 설치와 사용

최 석 재

*lingua@naver.com*

# 자연언어처리의 특징

# 활용 분야

- 자연언어처리는 사람의 말을 컴퓨터로 이해하기 위한 방법론
  - 사람들의 관심사 파악
  - 의견, 호감 분야 파악
  - 문서 분류
  - 유사어 찾기
  - 자동 번역
  - 사람을 대신하여 응대

# 정형데이터와 비정형데이터

- 정형 데이터는 약간의 전처리만으로 즉시 분석할 수 있도록 구조가 정형화되어 있는 것
  - 일관된 표 형태로 정리되어 있다
  - 의미 단위가 단순한 의미의 숫자로 기록되어 있다
- 비정형 데이터는 정보를 추출하기 어려운 형태로 되어 있는 것
  - 많은 전처리 과정을 거쳐야 핵심 분석 작업에 들어갈 수 있다
  - 일관된 모습으로 저장되어 있지 않다
  - 의미 단위 요소를 파악하기가 쉽지 않다

종류	예
정형 데이터	업무 처리, 매매 거래, 로그 데이터, 시계열 데이터
비정형 데이터	SNS 데이터, 고객 게시글, QR 코드, 오디오, 비디오

# 분석이 용이한 정형데이터

정형 데이터는 정보 추출 방법이 대개 결정되어 있다  
오른쪽의 경우 열(column) 별로 본다면 년도 별 증감을,  
행(row) 별로 본다면 지역별 차이를 보게 된다

정형 데이터는  
기본적으로 데이터가 셀(cell) 안에 있어서  
행과 열 기준으로 정보를 파악할 수 있으며,  
컬럼 간 정보 조합으로 새로운 정보를 찾아낼 수도 있다

3	관서명	2011년	2012년
4	계	422	365
5	중부서	11	4
6	종로서	15	4
7	남대문서	7	6
8	서대문서	13	16
9	혜화서	10	2
10	용산서	14	6
11	성북서	5	6
12	동대문서	17	11
13	마포서	8	11
14	영등포서	14	25
15	성동서	8	8
16	동작서	6	14
17	광진서	14	11
18	서부서	18	4
19	강북서	13	15
20	금천서	11	16
21	중랑서	12	17
22	강남서	15	18

# 분석이 복잡한 비정형데이터

하지만 비정형 데이터는 무엇이 정보의 핵심인지 결정되어 있지 않다

아래 예에서는 핵심어를 명사류(일반명사, 고유명사)와 동사류(동사, 형용사)로 정하고, 형태소 분석을 통해 명사류와 동사류를 발견한 뒤, 1점 리뷰와 10점 리뷰는 이들의 빈도가 다를 것이라 가정 하에 분석을 진행하고 있다

1점 리뷰		10점 리뷰	
완전히 나에게 실망만 안겨준 영화다... 아름다운 우주를 기대했건만 이젠 뭐 전혀 우주의 아름다움은 커녕, 나을듯 나을듯 하다가 하나도 없다		긴장감과 스릴 우주에서 지구를 볼 수 있는 색다른 영화 베스트	
매우 레알 지루했다... 10분짜리면 될 내용을 늘리고 늘리느라 매우 고생했을듯		차원이 다르다 심장터지는 줄 최고의 영화	
나는 진짜 별로였다. 재미 하나도 없고 원 스토리로 계속 이야기를 이어나가는데.. 진짜 돈주고 본게 너무 아깝고, 시간 낭비다		사람과 사람을 이어주는 관계의 힘을 압도적인 영상미로 표현해낸 우주를 통해 심도깊게 풀어낸 영화.	

→

1점 리뷰		10점 리뷰	
명사	동사	명사	동사
우주/2, 이것/1, 실망/1, 뭐/1	아름답다/2, 나오다/2, 하다/2, 안다/1, 없다/1, 기대하다/1	영화/1, 색/1, 긴장감/1, 스릴/1, 베스트/1, 지구/1, 우주/1	보다/1
내용/1	놀리다/2, 지루하다/1, 고생하다/1, 되다/1	최고/1, 영화/1, 차원/1, 심장/1	터지다/1, 다르다/1
이야기/1, 돈/1, 나/1, 재미/1, 스토리/1, 낭비/1, 시간/1	있다/1, 없다/1, 보다/1, 아깝다/1	사람/2, 영화/1, 힘/1, 영상미/1, 압도/1, 우주/1, 관계/1, 심도/1	해내다/1, 깊다/1, 통하다/1, 있다/1, 풀어내다/1, 표현하다/1

# 예상되는 형태의 정형데이터

	2020년	2021년	2022년
1	18723	27868	42178
2	11485	35232	55369
3	15693	50631	65781
4	20787	42481	34281
5	13428	31051	31541
6	10485	33641	66595
7	15985	70887	77458
8	16652	50074	65481
9	14422	37065	55284
10	15074	31002	57418
11	22481	44841	44865
12	22984	51002	66258

예상할 수 있는 형태

42178

각각의 숫자는 0~9까지의 제한된 범위에 있으면서  
각각의 숫자가 의미를 갖고 있다  
한 단위씩 의미부여 가능하다

# 예상이 어려운 비정형데이터

1점 리뷰	10점 리뷰
완전히 나에게 실망만 안겨 준영화다... 아름다운 우주를 기대했건만 이젠 뭐 전혀 우주의 아름다움은 커녕, 나을듯 나을듯 하다가 하나도 없다	긴장감과 스릴 우주에서 지구를 볼수있는색 다른 영화 베스트
매우 레알 지루했다...10분짜리면 될 내용을 늘리고 늘리느라 매우 고생했을듯	차원이 다르다 심장터지는 줄 최고의 영화
나는 진짜 별로였다. 재미 하나도 없고 원 스토리로 계속 이야기를 이어나가는데.. 진짜 돈주고 본게 너무 아깝고, 시간 낭비다	사람과 사람을 이어주는 관계의 힘을 앞도적인 영상미로 표현해낸 우주를 통해 심도깊게 풀어낸 영화.

예상할 수 없는 형태

긴장감과

각각의 음절이 11,172의 큰 범위에 있으면서  
대개 여러 개의 음절이 모여야 의미를 형성한다

다음에 올 수 있는 음절은 1/11172 이며,  
다음에 올 수 있는 단어는 1/전체단어수 이며,  
다음에 올 수 있는 문장은 예상 불가이다



# 형태소 분석이 필요한 한국어

- 한국어의 어절은 어근+접사 형태로 이루어져 매우 많은 변화형을 갖는다
- 조사는 총 1,000여 종, 어미는 총 5,600여 종이 발견되었다
- 어절 형태의 이론적 가능성은  $300,000 \times 1,000 \times 5,600 = 1,680,000,000,000$  → 사전에 모두 등재 불가
- 형태소 분석을 하면 정보량이 적은 접사를 제외할 수 있고,
- 어근도 기본형만으로 남겨둘 수 있다 (하, 할, 했, 한 → 하)
- 그러면 단위 형태를 사전에서 찾아 처리할 수 있게 된다
- 즉, 비정형데이터를 정형데이터와 유사한 단위 형태로 만들어 처리하는 것
- 따라서 언어의 기본 단위가 필요하며, 이를 형태소라고 한다

# 형태소 분석

아이가 사과를 먹었다




아이 + 가 사과 + 를 먹 + 었 + 다



아이, 사과, 먹

# 명사 추출

리뷰 문장	사용된 명사
완전히 나에게 실망만 안겨준 영화다... 아름다운 우주를 기대했지만 이젠 뭐 전혀 우주의 아름다움은 커녕, 나올듯 나 올듯 하다가 하나도없다	실망, 영화, 우주(2)  실망한 우주 영화

# 형태소 분석기 설치

# 형태소 분석기 선택

- 현재 약 11종 정도의 형태소분석기가 공개되어 있다
- 여기서는 RHINO를 사용 (검색어: 형태소분석기 RHINO)
- RHINO의 특징
  - 쉬운 사용
  - 높은 정확도, 빠른 속도
  - 다양한 언어(Python, R, Java)에서 사용 가능
  - 다양한 OS(Windows, Mac, Linux)에서 사용 가능

※ 공개 형태소분석기 모음: <https://koalanlp.github.io/koalanlp/>

## Module KoalaNLP

### KoalaNLP

KoalaNLP는 한국어 처리의 통합 인터페이스를 지향하는 Java/Kotlin/Scala Library입니다.

이 프로젝트는 서로 다른 형태의 형태소 분석기를 모아, 동일한 인터페이스 아래에서 사용할 수 있도록 하는 것이 목적입니다.

- KAIST의 한나눔 형태소 분석기와 NLP\_HUB 구문분석기
- 서울대의 꼬꼬마 형태소/구문 분석기 v2.1
- Shineware의 코모란 v3.3.9
- OpenKoreanText의 오픈 소스 한국어 처리기 v2.3.1 (구 Twitter 한국어 분석기)
- 은전한닢 프로젝트의 SEunjeon(S은전) (Mecab-ko의 Scala/Java 판본)
- 이수명님의 Arirang Morpheme Analyzer<sup>1-1</sup>
- 최석재님의 RHINO v3.7.8
- 김상준님의 Daon 분석기
- ETRI의 공공 인공지능 Open API
- Kakao의 카이(Khaiii) v0.4 (별도설치 필요: 설치법)
- 울산대학교의 UTagger 2018년 10월 31일자<sup>1-2</sup>, (별도설치 필요: 설치법)

# 성능

영화 리뷰에 대하여 Windows에서 동작 가능한 가장 많이 사용되는 3개의 형태소분석기와 함께 비교하였음

	kkma	okt	komoran	rhino
속도(secs)	23.435	7.181	1.809	1.003
오류 개수	7	23	9	1

- 속도는 1000 리뷰(6265어절), 오류는 5리뷰(50어절)에 대한 비교 결과

# 사용 환경

RHINO는 Python, R, Java 에서 사용할 수 있다  
항상 Python 버전이 최신 버전

▪Python 버전(rhinoMorph): <https://pypi.org/project/rhinoMorph/>

▪R 버전(RHINO): <https://github.com/SukjaeChoi/RHINO>

▪Java 버전(RHINO):  
<https://sourceforge.net/projects/koreanalyzer/>

※ 이 부분은 로컬 PC에 설치할 때  
필요한 부분입니다

※ 강의 실습은 Colab으로 진행하므로  
강의를 위해서는 필요하지 않습니다

# Java 설치

# JDK 다운로드

- 대부분의 형태소 분석기는 Java로 작성되었다
- Java로 작성된 것을 Python에서 불러와 사용하는 형태이다
- 여기서는 무료 개발자용 Java인 Open JDK를 다운로드 받고, 관련 설정을 해준다

## jdk.java.net

*Production and Early-Access OpenJDK Builds, from Oracle*

**Ready for use:** JDK 18, JDK 17, JMC 8

**Early access:** JDK 19, Loom, Metropolis, Panama,  
& Valhalla

*Looking to learn more about Java? Visit [dev.java](#) for the latest Java developer news and resources.*

*Looking for Oracle JDK builds and information about Oracle's enterprise Java products and services? Visit the [Oracle JDK Download page](#).*

- <http://jdk.java.net/>

※ 최신 버전을 설치하면 된다



# JDK 다운로드

## OpenJDK JDK 18 General-Availability Release

This page provides production-ready open-source builds of the Java Development Kit, version 18, an implementation of the Java SE 18 Platform under the GNU General Public License, version 2, with the Classpath Exception.

Commercial builds of JDK 18 from Oracle, under a non-open-source license, can be found at the [Oracle Technology Network](#).

### Documentation

- [Features](#)
- [Release notes](#)
- [API Javadoc](#)

### Builds

<b>Linux/AArch64</b>	<a href="#">tar.gz (sha256)</a>	186983593 bytes
<b>Linux/x64</b>	<a href="#">tar.gz (sha256)</a>	188173501
<b>macOS/AArch64</b>	<a href="#">tar.gz (sha256)</a>	183221810
<b>macOS/x64</b>	<a href="#">tar.gz (sha256)</a>	185375922
<b>Windows/x64</b>	<a href="#">zip (sha256)</a>	187504061

### Notes

- The Alpine Linux build previously available on this page was removed as of the first JDK 18 release candidate. It's not production-ready because it hasn't been tested thoroughly enough to be considered a GA build. Please use the [early-access JDK 19 Alpine Linux build](#) in its place.
- If you have difficulty downloading any of these files please contact [jdk-download-help\\_ww@oracle.com](mailto:jdk-download-help_ww@oracle.com).

※ Windows 버전 선택  
Linux 에서의 설치는 뒤에서 다룬다

# Java 폴더 생성

- C:\Program Files 폴더 안에 "Java" 폴더를 만든다
- 다운로드한 파일의 압축을 풀고, "jdk-18" 폴더를 "C:\Program Files\Java" 폴더 안에 넣는다

내 PC > Windows (C:) > Program Files > Java > jdk-18

이름	수정된 날짜	유형
bin	2022-04-15 오전 10:50	파일 폴더
conf	2022-04-15 오전 10:50	파일 폴더
include	2022-04-15 오전 10:50	파일 폴더
jmods	2022-04-15 오전 10:50	파일 폴더
legal	2022-04-15 오전 10:50	파일 폴더
lib	2022-04-15 오전 10:50	파일 폴더
release	2022-02-15 오후 6:43	파일

# Java 환경설정

제어판 홈

## 시스템 및 보안

네트워크 및 인터넷

하드웨어 및 소리

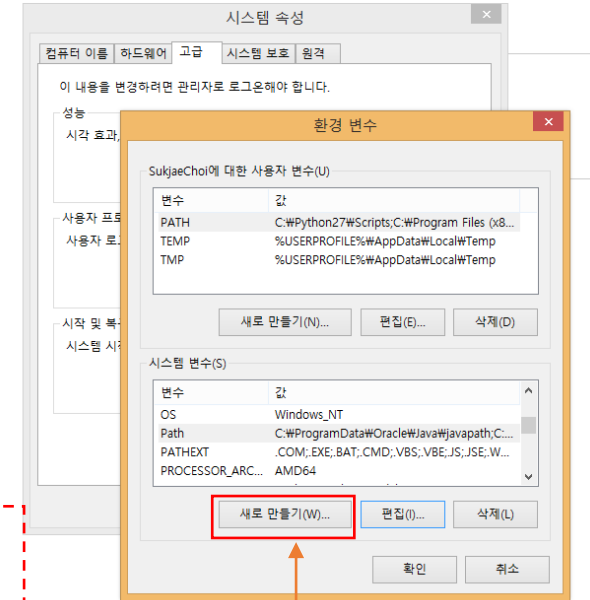
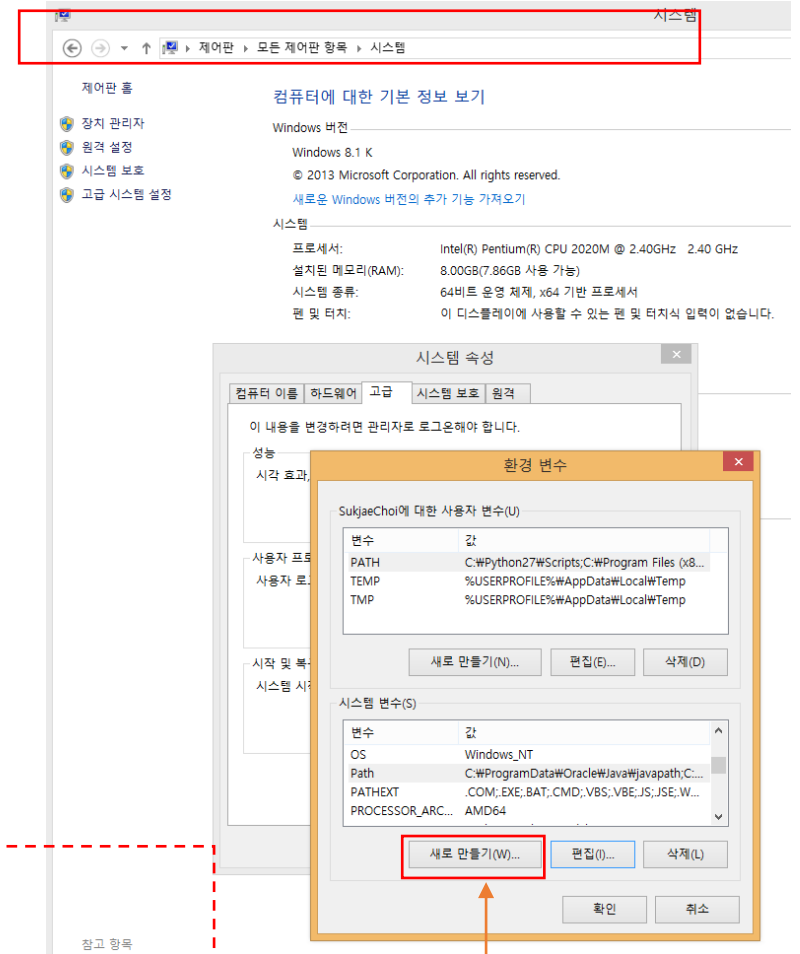
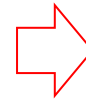
프로그램

사용자 계정

모양 및 개인 설정

시계, 언어 및 국가별 옵션

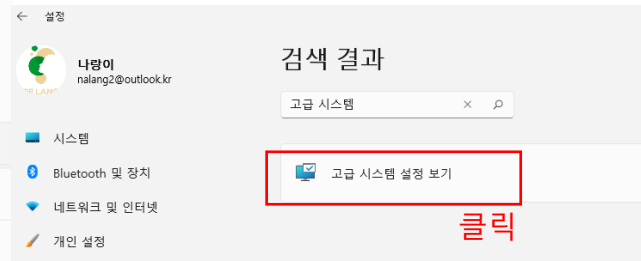
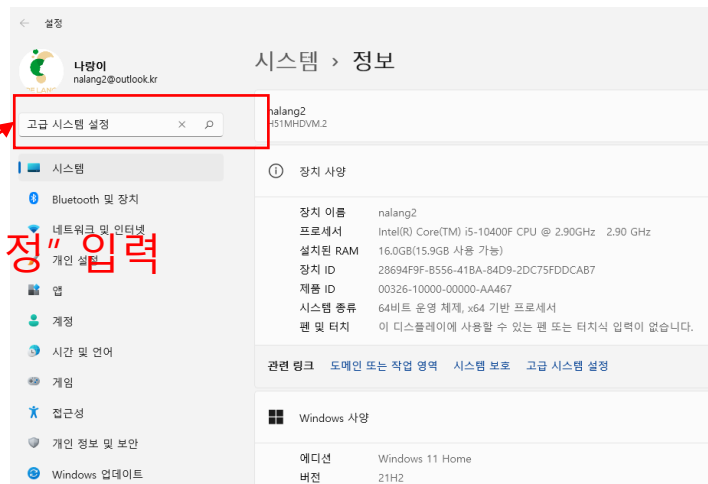
접근성



'새로 만들기(W)...' 클릭

제어판 > 시스템 및 보안 > 시스템 > 고급 시스템 설정 > 환경 변수

※ Windows11에서는  
시스템까지 진입 후,  
검색창에 "고급 시스템 설정" 입력



클릭

# JAVA\_HOME 설정

시스템 속성

컴퓨터 이름 하드웨어 고급 시스템 보호 원격

이 내용을 변경하려면 관리자로 로그인해야 합니다.

성능

시각 효과, 프로세서 일정, 메모리 사용 및 가상 메모리

설

사용자 프로필

사용자 로그인에 관련된 바탕 화면 설정

시작 및 복구

시스템 시작, 시스템 오류 및 디버깅 정보

변수 이름에 JAVA\_HOME,  
변수 값에 해당 경로를 넣는다

환경 변수

lingu에 대한 사용자 변수(U)

변수	값
OneDrive	C:\Users\lingu\OneDrive
OneDriveConsumer	C:\Users\lingu\OneDrive
Path	C:\Users\lingu\AppData\Local\Microsoft...
TEMP	C:\Users\lingu\AppData\Local\Temp
TMP	C:\Users\lingu\AppData\Local\Temp

시스템 변수 편집

변수 이름(N): JAVA\_HOME

변수 값(V): C:\Program Files\Java\jdk-14.0.1

디렉터리 찾아보기(D)...

파일 찾아보기(F)...

확인

취소

환경 변수

새로 만들기(W)...

편집(I)...

삭제(L)

확인

취소

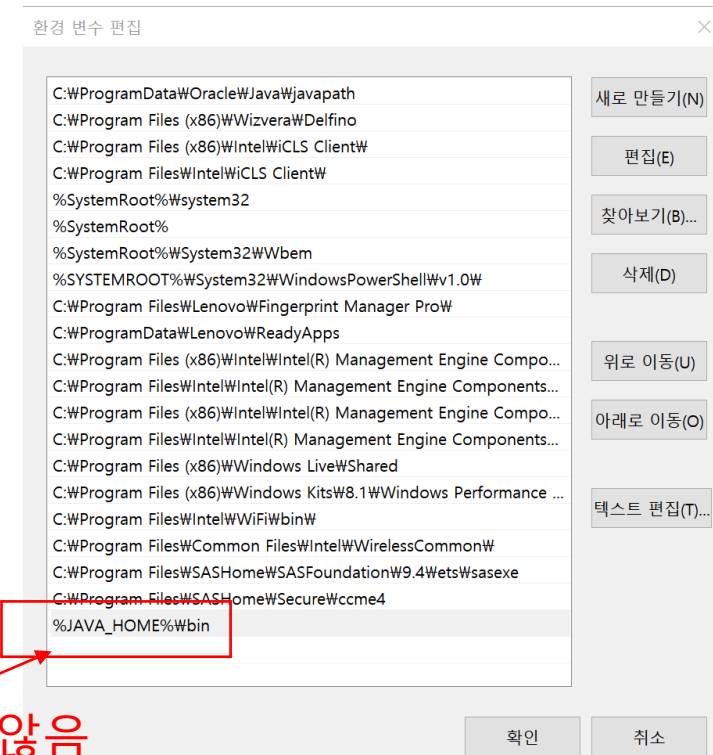
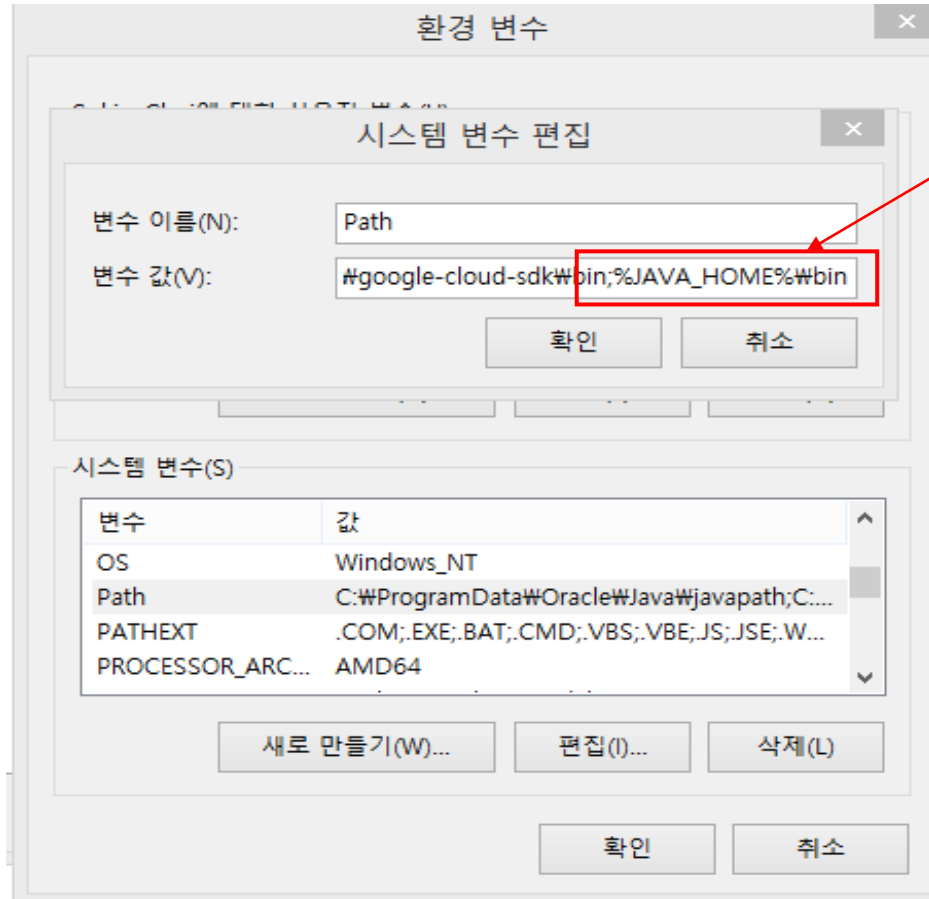
확인

취소

# Path 설정

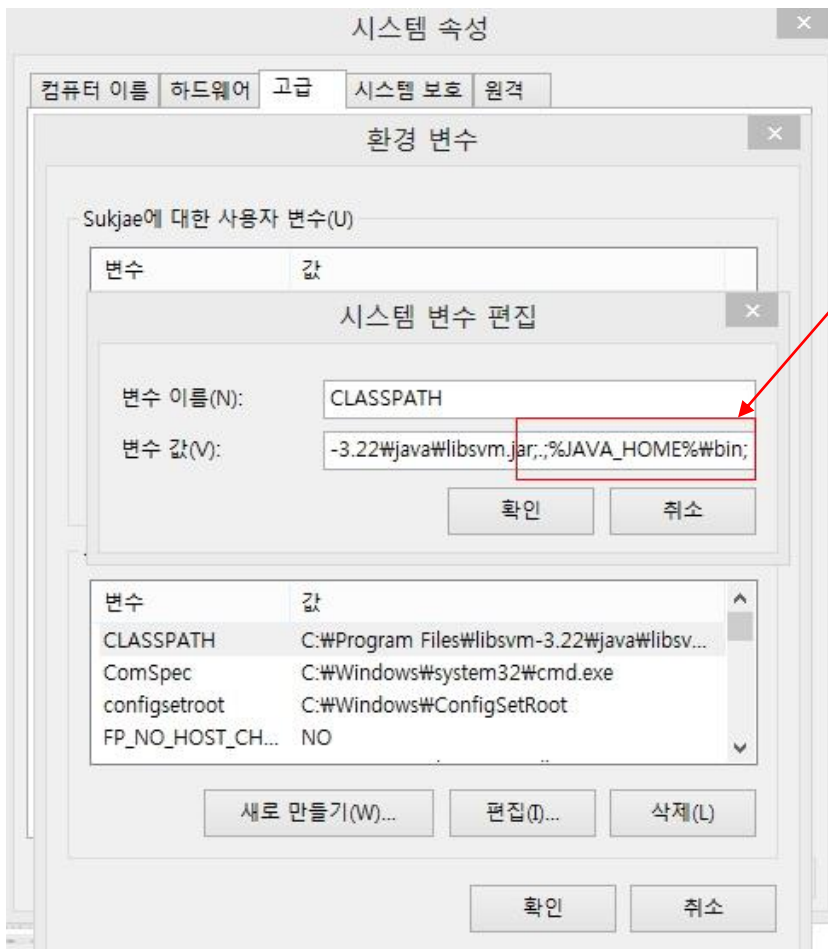
다음으로 시스템 변수의 Path를 아래와 같이 편집한다  
마지막에 **';%JAVA\_HOME%\bin'**를 입력한다.

(만약 이후의 과정도 모두 진행했는데 잘 되지 않으면  
C:\Program Files\Java\jdk-14.0.1\bin;%JAVA\_HOME%\bin  
를 입력한다)



여기서는 ; 를 사용하지 않음

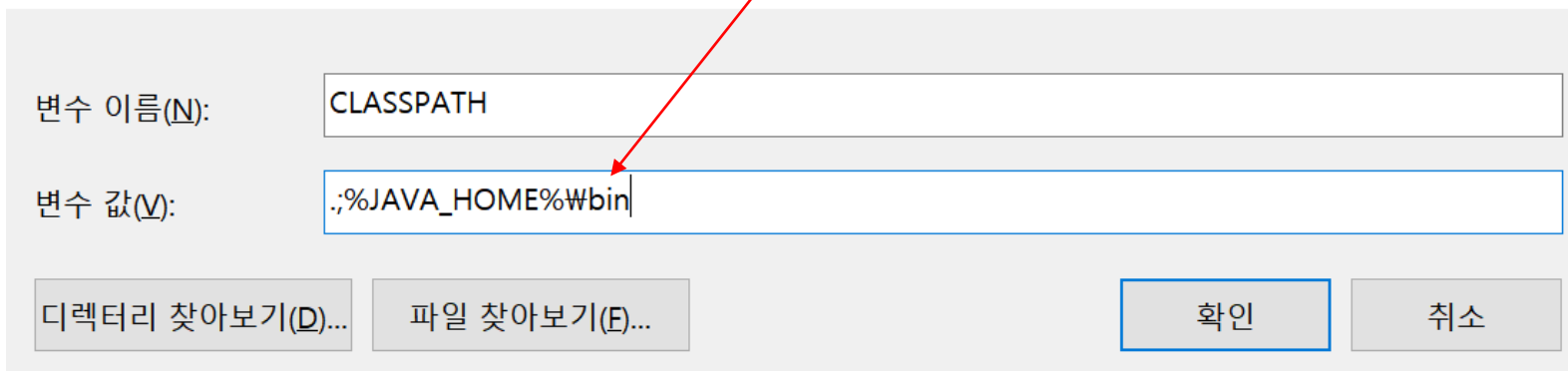
# CLASSPATH 설정



시스템 변수의 CLASSPATH를 찾는다 (없으면 '새로 만들기'로 생성)  
맨 마지막 부분에 '.;%JAVA\_HOME%\bin'를 그대로 입력한다

단, CLASSPATH를 새로 만들었다면 '.;%JAVA\_HOME%\bin'만 입력한다  
(또는 '%JAVA\_HOME%\bin'만으로도 가능)

새 시스템 변수



# javac

- cmd 명령어로 새 커맨드 창을 띄우고, javac -version 를 입력하여 버전 정보가 나오는지 확인한다

```
C:\> 명령 프롬프트
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hnalang>javac -version
javac 18

C:\Users\hnalang>
```



※ 이 부분은 로컬 PC에 설치할 때  
필요한 부분입니다

※ 강의 실습은 Colab으로 진행하므로  
강의를 위해서는 필요하지 않습니다

# JType 설치



# JPyype 다운로드

- 다음으로 Java와 Python을 연결해주는 JPyype를 설치한다

- `pip install JPyype1`

※ 대소문자 다름에 주의!

명령 프롬프트

```
Microsoft Windows [Version 10.0.22000.556]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\hwalang>pip install JPyype1  
Collecting JPyype1  
  Using cached https://files.pythonhosted.org/packages/03/d8/510d122a3d3c  
0/JPyype1-1.3.0-cp37-cp37m-win_amd64.whl  
Requirement already satisfied: typing-extensions; python_version < "3.8"  
(3.10.0.2)  
Installing collected packages: JPyype1  
Successfully installed JPyype1-1.3.0  
  
C:\Users\hwalang>
```



※ 이 부분은 로컬 PC에 설치할 때  
필요한 부분입니다

※ 강의 실습은 Colab으로 진행하므로  
강의를 위해서는 필요하지 않습니다

# RHINO 설치

# RHINO 설치

- 형태소 분석기 RHINO를 설치한다
- 커맨드 창에서 다음의 명령어를 입력한다
- `pip install rhinoMorph`

cmd 명령 프롬프트

```
Microsoft Windows [Version 10.0.22000.556]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\hwalang>pip install rhinoMorph  
Collecting rhinoMorph  
  Downloading https://files.pythonhosted.org/packages/47/a3/b4e8add92411/rhinoMorph-4.0.0.2-py3-none-any.whl (3.0MB)  
    | 3.1MB 6.8MB/s  
Installing collected packages: rhinoMorph  
Successfully installed rhinoMorph-4.0.0.2  
  
C:\Users\hwalang>
```

# 동작 확인

- 개발 환경에서 다음과 같은 코드를 작성하여 테스트한다

- import rhinoMorph
- rn = rhinoMorph.startRhino()
- # 예문 분석
- text = "한글테스트 글을 남겨주세요"
- sample\_data = rhinoMorph.onlyMorph\_list(rn, text)
- print('sample data:', sample\_data)

```
test.py x
1  import rhinoMorph
2  rn = rhinoMorph.startRhino()
3
4  # 예문 분석
5  text = "한글테스트 글을 남겨주세요"
6
7  sample_data = rhinoMorph.onlyMorph_list(rn, text)
8  print('sample data:', sample_data)
9
```

```
test x
C:\Anaconda3\python.exe C:/Users/naLang/PythonProjects/test.py
filepath: C:\Anaconda3\lib\site-packages
classpath: C:\Anaconda3\lib\site-packages\rhinoMorph/lib/rhino.jar
Constructing Dictionaries...
Current mode: python    Current path: C:\Anaconda3\lib\site-packages\rhinoMorph/resource/
Constructing Dictionaries Completed.

RHINO started!
sample data: ['한글', '테스트', '글', '을', '남기', '어', '주', '시', '어요']

Process finished with exit code 0
```

# Colab에서의 설치

# Colab에서 설치하기

- Colab에서는 Java, JType, RHINO를 간단한 명령어로 설치할 수 있다

## # 사전 설치

- !apt-get update
- !apt-get install g++ openjdk-8-jdk

# 리눅스 패키지 업데이트

# Java 설치

## # JType 설치

- !pip install JType1

# Java와 Python을 연결하는 JType 설치

## # RHINO 설치

- !pip install rhinoMorph

※ 새로운 페이지에 작성할 경우 이 부분을 모두 다시 실행해야 한다

# 동작 확인

# RHINO 시작

- import rhinoMorph
- **rn** = rhinoMorph.startRhino()

※ 형태소분석기 객체를 rn 이라는 이름으로 생성했다

# 예문 분석

- text = "한글테스트 글을 남겨주세요"
- sample\_data = rhinoMorph.onlyMorph\_list(**rn**, text)
- print('sample data:', sample\_data)

※ 형태소분석기 객체를 같이 넣는다

```
➞ sample data: ['한글', '테스트', '글', '을', '남기', '어', '주', '시', '어요']
```

# RHINO 사용

- rhinoMorph 는 RHINO의 파이썬 버전이다
- 자세한 사용 방법 설명: <https://blog.naver.com/lingua/221537630069>

```
import rhinoMorph
rn = rhinoMorph.startRhino()
text = "한글로 된 한글텍스트를 분석하는 것은 즐겁다."
```

# 사용 1 : 모든 형태소 보이기

```
text_analyzed = rhinoMorph.onlyMorph_list(rn, text)
print('\n1. 형태소 분석 결과:', text_analyzed)
```

1. 형태소 분석 결과: ['한글', '로', '되', '니', '한글', '텍스트', '를', '분석', '하', '는', '것', '은', '즐겁', '다', '.']



# 사용 2 : 실질형태소만, 동사의 어말어미는 제외

```
text_analyzed = rhinoMorph.onlyMorph_list(rn, text, pos=['NNG', 'NNP', 'NP',  
'VV', 'VA', 'XR', 'IC', 'MM', 'MAG', 'MAJ'])  
print('Wn2. 형태소 분석 결과:', text_analyzed)
```

2. 형태소 분석 결과: ['한글', '되', '한글', '텍스트', '분석', '즐겁']

# 사용 3 : 실질형태소만, 동사의 어말어미 포함

```
text_analyzed = rhinoMorph.onlyMorph_list(rn, text, pos=['NNG', 'NNP', 'NP',  
'VV', 'VA', 'XR', 'IC', 'MM', 'MAG', 'MAJ'], eomi=True)  
print('Wn3. 형태소 분석 결과:', text_analyzed)
```

3. 형태소 분석 결과: ['한글', '되다', '한글', '텍스트', '분석', '즐겁다']

```
# 사용 4 : 전체형태소, 품사정보도 가져 오기
morphs, poses = rhinoMorph.wholeResult_list(rn, text)
print('Wn4. 형태소 분석 결과:')
print('morphs:', morphs)
print('poses:', poses)
```

4. 형태소 분석 결과:

```
morphs: ['한글', '로', '되', 'ㄴ', '한글', '텍스트', '를', '분석', '하', '는', '것', '은', '즐겁', '다', '.']
poses: ['NNG', 'JKB', 'VV', 'ETM', 'NNG', 'NNG', 'JKO', 'XR', 'XSV', 'ETM', 'NNB', 'JX', 'VA', 'EF', 'SF']
```

```
# 사용 5 : 원문의 어절 정보를 같이 가져 오기
text_analyzed = rhinoMorph.wholeResult_text(rn, text)
print('Wn5. 형태소 분석 결과:Wn', text_analyzed)
```

5. 형태소 분석 결과:

한글로 한글/NNG + 로/JKB  
된        되/VV + ㄴ/ETM  
한글텍스트를    한글/NNG + 텍스트/NNG + 를/JKO  
분석하는        분석/XR + 하/XSV + 는/ETM  
것은        것/NNB + 은/JX  
즐겁다    즐겁/VA + 다/EF  
          ./SF

# 추가 옵션 1 – 연결된 명사 결합

# 사용 6, 7 : 한 어절에서 연결된 명사를 하나의 명사로 결합하기  
# onlyMorph\_list와 wholeResult\_list에서 사용 가능하다

```
text_analyzed = rhinoMorph.onlyMorph_list(rn, text, pos=['NNG', 'NNP', 'NP',  
'VV', 'VA', 'XR', 'IC', 'MM', 'MAG', 'MAJ'], combineN=True)  
print('Wn6. 형태소 분석 결과:Wn', text_analyzed)
```

```
morphs, poses = rhinoMorph.wholeResult_list(rn, text, combineN=True)  
print('Wn7. 형태소 분석 결과: ')  
print('morphs: ', morphs)  
print('poses: ', poses)
```

6. 형태소 분석 결과:  
['한글', '되', '한글텍스트', '분석', '즐겁']

7. 형태소 분석 결과:  
morphs: ['한글', '로', '되', 'ㄴ', '한글텍스트', '를', '분석', '하', '는', '것', '은', '즐겁', '다', '.']  
poses: ['NNG', 'JKB', 'VV', 'ETM', 'NNG', 'JKO', 'XR', 'XSV', 'ETM', 'NNB', 'JX', 'VA', 'EF', 'SF']

# 추가 옵션 2 - '어근+ 하' 결합

```
# 사용 8, 9 : 어근 + 하 형태를 하나의 동사로 출력하기  
# xrVv 아규먼트가 담당하며, 기본값은 False로서 둘을 분리하여 출력한다  
# 분리된 어근이 명사인 경우, 명사로 출력된다  
# onlyMorph_list, wholeResult_list, wholeResult_text 등 모든 함수에서 사용 가능하다
```

```
text_analyzed = rhinoMorph.wholeResult_list(rn, '사랑합니다')  
print('Wn8. 형태소 분석 결과: ', text_analyzed)
```

```
text_analyzed = rhinoMorph.wholeResult_list(rn, '사랑합니다', xrVv=True)  
print('Wn9. 형태소 분석 결과: ', text_analyzed)
```

8. 형태소 분석 결과: (['사랑', '하', '니다'], ['XR', 'XSV', 'EF'])

9. 형태소 분석 결과: (['사랑하', '니다'], ['VV', 'EF'])

# 부록

형태소 분석기 사전 추가 방법

# 형태소분석기? 단어분석기?

- ‘형태소분석기’라는 명칭을 생각하면 분석 내용은 항상 형태소 단위여야 한다고 생각할 수 있다
- 그러나 문장을 ‘의미있는 단위’로 분리하는 목적이지, 반드시 ‘최소 의미 단위’로의 분리가 아니다
- 영어의 경우에는 POS tagger(Part of Speech tagger)라는 이름으로 사용되는데, 이 때의 part는 단어(word)가 될 것으로 기대한다. 그 이유는 영어에서 형태소 단위는 기대하는 단위 이하이기 때문이다. 영어는 기본적으로 띄어쓴 단위가 곧 단어이다
  - it is unbreakable → ‘it’, ‘is’, ‘unbreakable’ (O), ‘it’, ‘is’, ‘un’, ‘break’, ‘able’ (X)
  - baseball → ‘baseball’ (O), ‘base’, ‘ball’ (X)
- 하지만 한국어의 경우에는 단어 단위는 기대하는 단위 이상일 때가 많으므로 단어 이하의 단위로 분리해야 하며, 이것은 곧 형태소 단위가 된다
  - ‘밥을 먹었다’ → ‘밥’, ‘을’, ‘먹었다’ (X), ‘밥’, ‘을’, ‘먹’, ‘었’, ‘다’ (O)
- 그러나 복합명사에서는 ‘형태소 단위’가 아닌, 의미있는 단위가 중요하므로 반드시 형태소 분리가 되게 할 것이 아니라, 목적에 적합한 단위(형태소/단어)로 나오면 된다
  - 탄도미사일 → 탄도미사일(O?), ‘탄도’, ‘미사일’(X?)
  - 전자금융 → 전자금융(O?), ‘전자’, ‘금융’(X?)

# 불용어 목록

- 불용어 목록은 정해진 것이 없고, 과제의 성격에 따라 자유롭게 정의된다
- 하지만 일반적인 불용어 목록으로 다음과 같은 것을 생각해 볼 수 있다

## 1. 형식형태소 - 품사 태그를 이용하여 일괄 지정할 수 있다

주격조사(JKS), 보격조사(JKC), 관형격조사(JKG), 목적격조사(JKO), 부사격조사(JKB), 호격조사(JKV), 인용격조사(JKQ), 보조사(JX), 접속조사(JC), 선어말어미(EP), 종결어미(EF), 연결어미(EC), 명사형전성어미(ETN), 관형형전성어미(ETM), 체언접두사(XPN), 명사파생 접미사(XSA), 마침표, 물음표, 느낌표(SF), 따옴표, 괄호표, 줄표(SS), 쉼표, 가운뎃점, 콜론, 빗금(SP), 줄임표(SE), 붙임표(물결, 숨김, 빠짐) SO, 분석불능범주(NA)

## 2. 기호류 형태 - 기호류 태그로 지정 가능하나, 직접 일부분을 선택할 경우

[illegible]

3. 실질형태소 중 불용어 - 다음은 실질형태소 중 불용어로 볼 수 있는 것들이다

- 1) 품사태그: 수사(NR), 숫자(SN), 명사추정범주(NF), 기타기호(논리수학기호, 화폐기호) SW, 용언추정범주(NV), 의존명사(NNB), 보조용언(VX), 긍정지정사(VCP), 부정지정사(VCN)
- 2) 형태: "하다", "있다", "되다", "그", "않다", "없다", "나", "말", "사람", "이", "보다", "한", "때", "년", "같다", "대하다", "일", "이", "생각", "위하다", "때문", "그것", "그러나", "가다", "받다", "그렇다", "알다", "사회", "더", "그녀", "문제", "오다", "그리고", "크다", "속"

※ 국립국어원 말뭉치 실질형태소 고빈도 어휘 50위 중, 'NNG', 'NNP', 'NP', 'VV', 'VA', 'XR', 'IC', 'MM', 'MAG', 'MAJ'에 해당하는 것

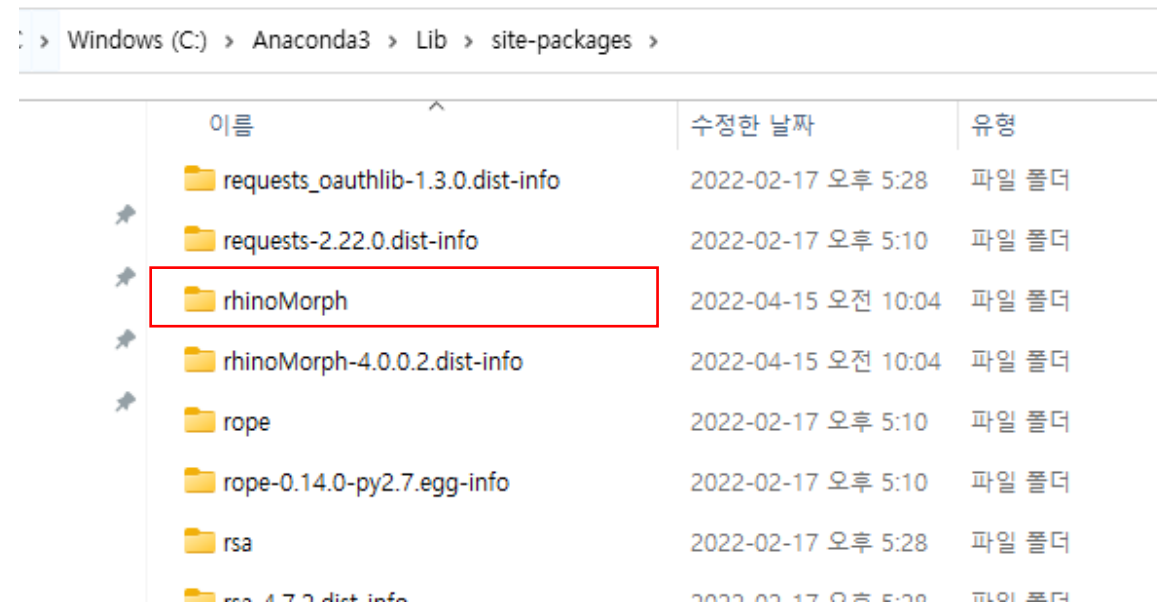


#### 4. 영어 불용어 - 구글에서 제시한 목록

"a", "about", "above", "after", "again", "against", "all", "am", "an", "and", "any", "are",  
"aren't", "as", "at", "be", "because", "been", "before", "being", "below", "between",  
"both", "but", "by", "can't", "cannot", "could", "couldn't", "did", "didn't", "do", "does",  
"doesn't", "doing", "don't", "down", "during", "each", "few", "for", "from", "further",  
"had", "hadn't", "has", "hasn't", "have", "haven't", "having", "he", "he'd", "he'll",  
"he's", "her", "here", "here's", "hers", "herself", "him", "himself", "his", "how", "how's",  
"i", "i'd", "i'll", "i'm", "i've", "if", "in", "into", "is", "isn't", "it", "it's", "its", "itself", "let's",  
"me", "more", "most", "mustn't", "my", "myself", "no", "nor", "not", "of", "off", "on",  
"once", "only", "or", "other", "ought", "our", "ours", "ourselves", "out", "over", "own",  
"same", "shan't", "she", "she'd", "she'll", "she's", "should", "shouldn't", "so", "some",  
"such", "than", "that", "that's", "the", "their", "theirs", "them", "themselves", "then",  
"there", "there's", "these", "they", "they'd", "they'll", "they're", "they've", "this", "those",  
"through", "to", "too", "under", "until", "up", "very", "was", "wasn't", "we", "we'd",  
"we'll", "we're", "we've", "were", "weren't", "what", "what's", "when", "when's",  
"where", "where's", "which", "while", "who", "who's", "whom", "why", "why's", "with",  
"won't", "would", "wouldn't", "you", "you'd", "you'll", "you're", "you've", "your", "yours",  
"yourself", "yourselves"

# 사전 추가 방법(1/3)

- Python Library가 설치된 경로를 찾는다
- Lib\site-packages 폴더에서 rhinoMorph 폴더를 찾는다
- 예) Anaconda를 이용하여 C:\에 설치한 경우
- C:\Anaconda3\Lib\site-packages



Windows (C:) > Anaconda3 > Lib > site-packages

이름	수정한 날짜	유형
requests_oauthlib-1.3.0.dist-info	2022-02-17 오후 5:28	파일 폴더
requests-2.22.0.dist-info	2022-02-17 오후 5:10	파일 폴더
rhinoMorph	2022-04-15 오전 10:04	파일 폴더
rhinoMorph-4.0.0.2.dist-info	2022-04-15 오전 10:04	파일 폴더
rope	2022-02-17 오후 5:10	파일 폴더
rope-0.14.0-py2.7.egg-info	2022-02-17 오후 5:10	파일 폴더
rsa	2022-02-17 오후 5:28	파일 폴더
rsa-4.7.2.dist-info	2022-02-17 오후 5:28	파일 폴더

# 사전 추가 방법(2/3)

- resource 폴더로 진입하면 6개의 텍스트 파일을 확인할 수 있다
- stem\_MethodDeleted.txt 파일을 연다

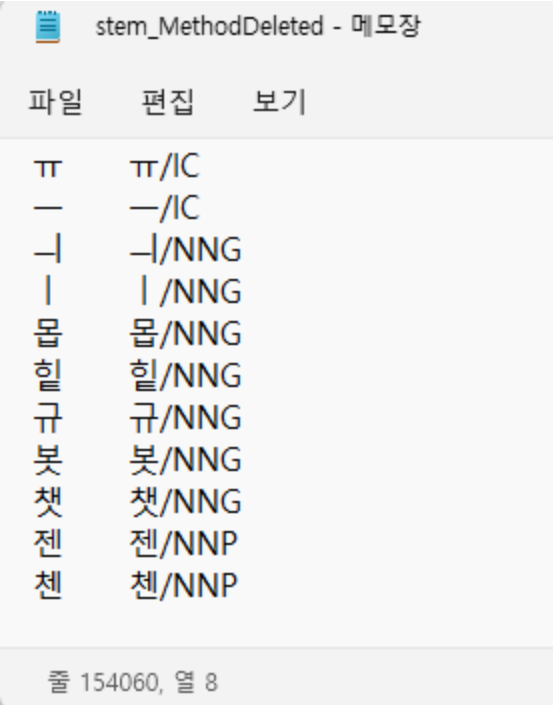
Windows (C:) > Anaconda3 > Lib > site-packages > rhinoMorph		
이름	수정한 날짜	유형
lib	2022-04-15 오전 10:04	파일 폴더
resource	2022-04-15 오전 10:04	파일 폴더

Windows (C:) > Anaconda3 > Lib > site-packages > rhinoMorph > resource		
이름	수정한 날짜	유형
afterNumber_MethodDeleted	2022-04-15 오전 10:04	텍스트 문서
alsoXR	2022-04-15 오전 10:04	텍스트 문서
complexStem_MethodDeleted	2022-04-15 오전 10:04	텍스트 문서
ending_MethodDeleted	2022-04-15 오전 10:04	텍스트 문서
koreanName	2022-04-15 오전 10:04	텍스트 문서
stem_MethodDeleted	2022-04-15 오전 10:04	텍스트 문서

# 사전 추가 방법(3/3)

- 파일의 최하단에 형식에 맞춰 새로운 단어를 입력한다
- 해당어휘\해당어휘/품사태그

※ 마지막에는 엔터를 쳐서 빈 줄이 하나 들어가게 한다



파일	편집	보기
π	π /IC	
—	— /IC	
—	— /NNG	
	/NNG	
몫	몫 /NNG	
힐	힐 /NNG	
규	규 /NNG	
붓	붓 /NNG	
찻	찻 /NNG	
젠	젠 /NNP	
첸	첸 /NNP	

줄 154060, 열 8

# 어근(XR) 예

- 열덜덜: 없음
- 어렵풋: 없음

열덜덜하다: 형용사  
어렵풋하다: 형용사

- 공부: 명사
- 역주행: 명사

공부하다: 동사  
역주행하다: 동사

- 말랑말랑: 부사

말랑말랑하다: 형용사

# 품사 태그

일반명사 NNG  
의존명사 NNB  
수사 NR  
형용사 VA  
긍정지정사 VCP  
관형사 MM  
접속부사 MAJ  
주격조사 JKS  
관형격조사 JKG  
부사격조사 JKB  
인용격조사 JKQ  
접속조사 JC

고유명사 NNP  
대명사 NP  
동사 VV  
보조용언 VX  
부정지정사 VCN  
일반부사 MAG  
감탄사 IC  
보격조사 JKC  
목적격조사 JKO  
호격조사 JKV  
보조사 JX  
선어말어미 EP

종결어미 EF  
명사형전성어미 ETN  
체언접두사 XPN  
동사파생접미사 XSV  
어근 XR  
따옴표,괄호표,줄표 SS  
줄임표 SE  
외국어 SL  
명사추정범주 NF  
기타기호(논리수학기호,화폐기호) SW  
용언추정범주 NV  
분석불능범주 NA

연결어미 EC  
관형형전성어미 ETM  
명사파생접미사 XSN  
형용사파생접미사 XSA  
마침표,물음표,느낌표 SF  
쉼표,가운뎃점,콜론,빗금 SP  
붙임표(물결,숨김,빠짐) SO  
한자 SH  
숫자 SN