

The background features a light blue gradient with stylized, wavy horizontal bands in a darker blue. Three spheres are positioned on these waves: a large blue sphere on the left, a small green sphere on the right, and another large blue sphere further to the right.

# wordcloud

안화수

# wordcloud

## wordcloud란?

자연어처리(NLP)에서 단어들의 빈도수를 파악해서 빈도수가 높은 단어일 수록 크게, 빈도수가 낮은 단어일 수록 작게 표현하는 시각화 기법이다.



# wordcloud 환경구축

## ❖ wordcloud에 필요한 모듈 설치

### 1. 자바 설치

운영체제(OS)에 맞는 JDK설치

### 2. KoNLPy 의존성 패키지 설치

```
pip install jpype1
```

### 3. KoNLPy 모듈 설치

KoNLPy모듈에는 형태소 분석기가 5개가 포함되어 있다.

(Hannanum, Kkma(꼬꼬마), Mecab, Komoran, Okt)

```
pip install konlpy
```

### 4. wordcloud 모듈 설치

MicroSoft Visual C++ 14 이상 설치되어 있어야 wordcloud 모듈이 설치된다.

```
pip install wordcloud
```

# 1.자바설치

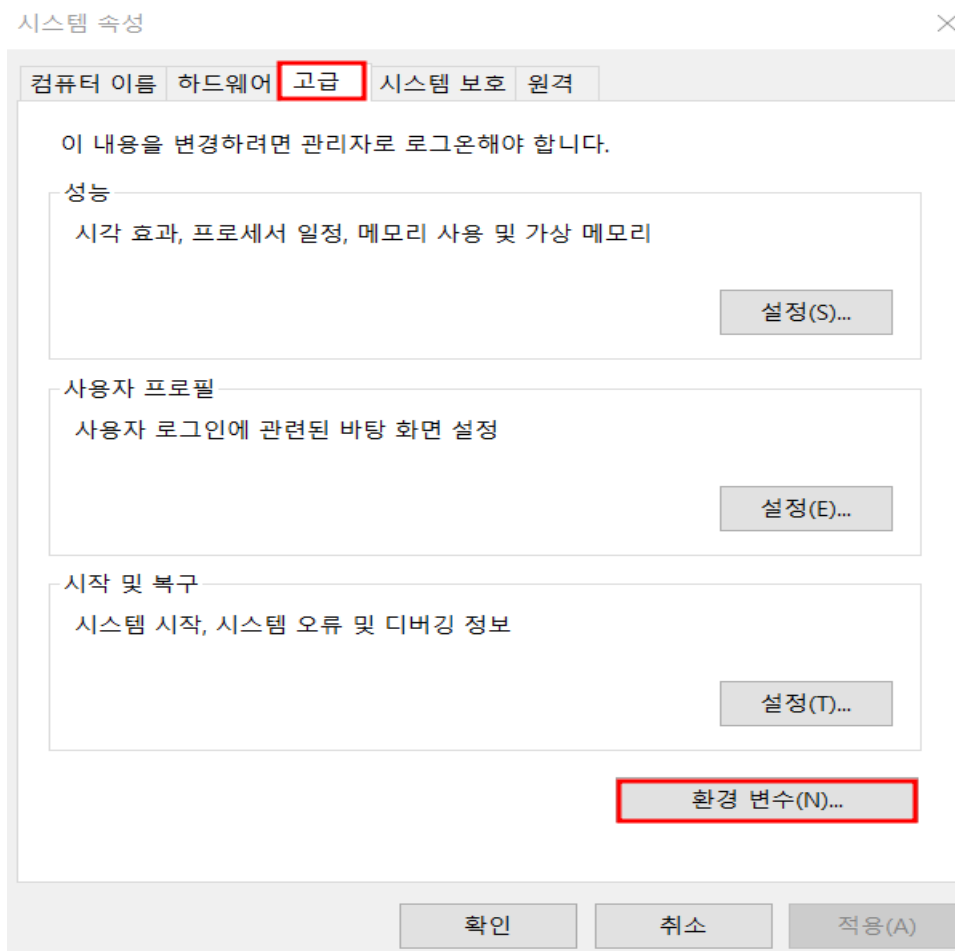
❖ 1.자바 설치 : 운영체제에 맞는 버전을 다운로드 후 설치

[http://abit.ly/easypy\\_101](http://abit.ly/easypy_101)

| 플랫폼         | 유형  | 다운로드 링크   | Checksum (MD5)  | Sig 파일  |
|-------------|-----|---|---|---|
| Windows x64 | JDK | <a href="https://corretto.aws/downloads/latest/amazon-corretto-11-x64-windows-jdk.msi">https://corretto.aws/downloads/latest/amazon-corretto-11-x64-windows-jdk.msi</a>     | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-windows-jdk.msi">https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-windows-jdk.msi</a>     |   |
|             |     | <a href="https://corretto.aws/downloads/latest/amazon-corretto-11-x64-windows-jdk.zip">https://corretto.aws/downloads/latest/amazon-corretto-11-x64-windows-jdk.zip</a>     | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-windows-jdk.zip">https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-windows-jdk.zip</a>     | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-windows-jdk.zip">https://corretto-1</a>    |
| Windows x86 | JDK | <a href="https://corretto.aws/downloads/latest/amazon-corretto-11-x86-windows-jdk.msi">https://corretto.aws/downloads/latest/amazon-corretto-11-x86-windows-jdk.msi</a>     | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x86-windows-jdk.msi">https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x86-windows-jdk.msi</a>     |   |
|             |     | <a href="https://corretto.aws/downloads/latest/amazon-corretto-11-x86-windows-jdk.zip">https://corretto.aws/downloads/latest/amazon-corretto-11-x86-windows-jdk.zip</a>     | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x86-windows-jdk.zip">https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x86-windows-jdk.zip</a>     | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x86-windows-jdk.zip">https://ccorretto-1</a>   |
| macOS x64   | JDK | <a href="https://corretto.aws/downloads/latest/amazon-corretto-11-x64-macos-jdk.pkg">https://corretto.aws/downloads/latest/amazon-corretto-11-x64-macos-jdk.pkg</a>         | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-macos-jdk.pkg">https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-macos-jdk.pkg</a>         |   |
|             |     | <a href="https://corretto.aws/downloads/latest/amazon-corretto-11-x64-macos-jdk.tar.gz">https://corretto.aws/downloads/latest/amazon-corretto-11-x64-macos-jdk.tar.gz</a>   | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-macos-jdk.tar.gz">https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-macos-jdk.tar.gz</a>   | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-macos-jdk.tar.gz">https://ccorretto-1</a>  |
| Linux 알파인   | JDK | <a href="https://corretto.aws/downloads/latest/amazon-corretto-11-x64-alpine-jdk.tar.gz">https://corretto.aws/downloads/latest/amazon-corretto-11-x64-alpine-jdk.tar.gz</a> | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-alpine-jdk.tar.gz">https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-alpine-jdk.tar.gz</a> | <a href="https://corretto.aws/downloads/latest_checksum/amazon-corretto-11-x64-alpine-jdk.tar.gz">https://ccorretto-1</a> |

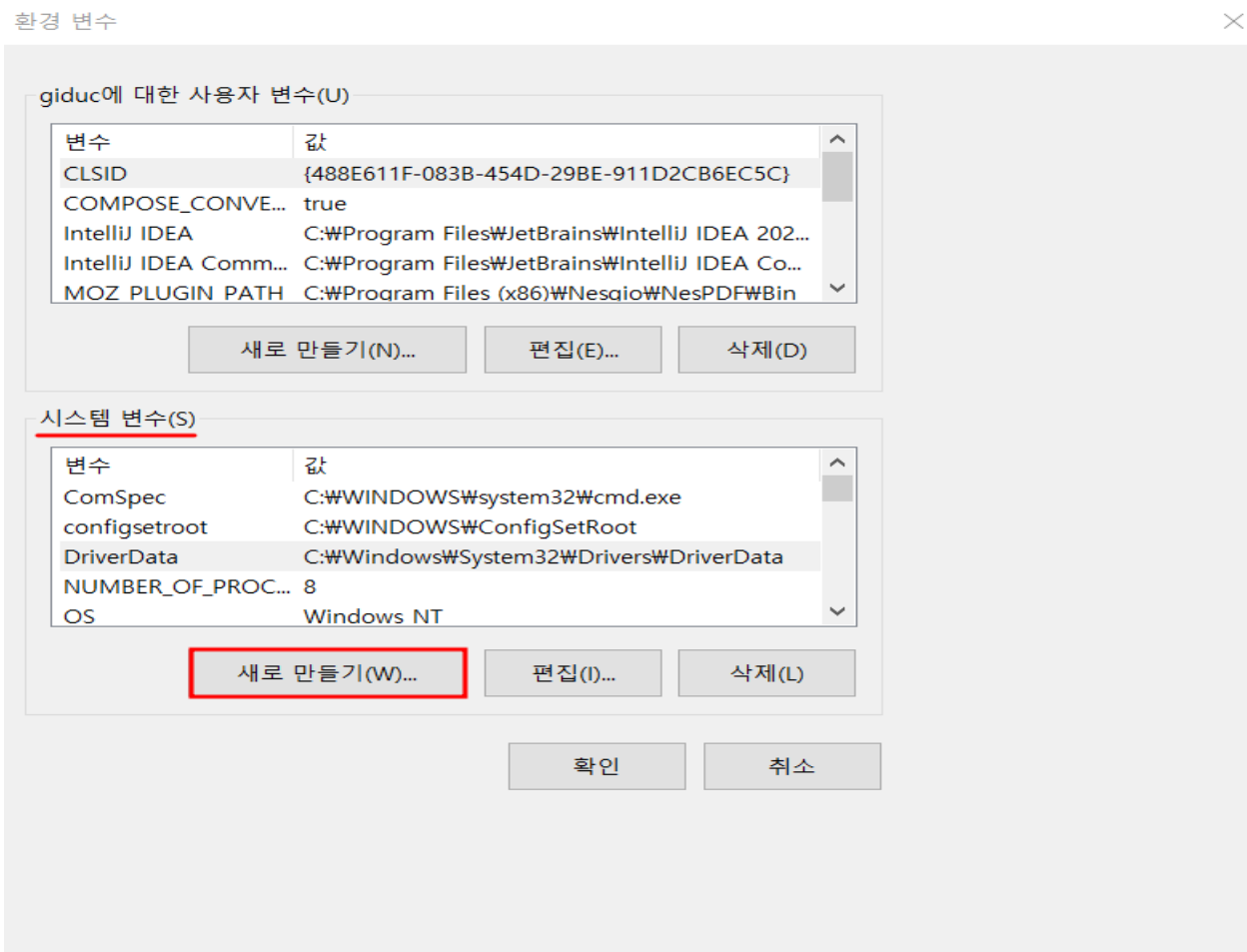
# 환경변수 설정

1) 내PC -> 오른 마우스 클릭후 속성 선택 -> 고급 시스템 설정



# 환경변수 설정

## 2) 시스템 변수(S) – 새로 만들기 버튼클릭



# 환경변수 설정

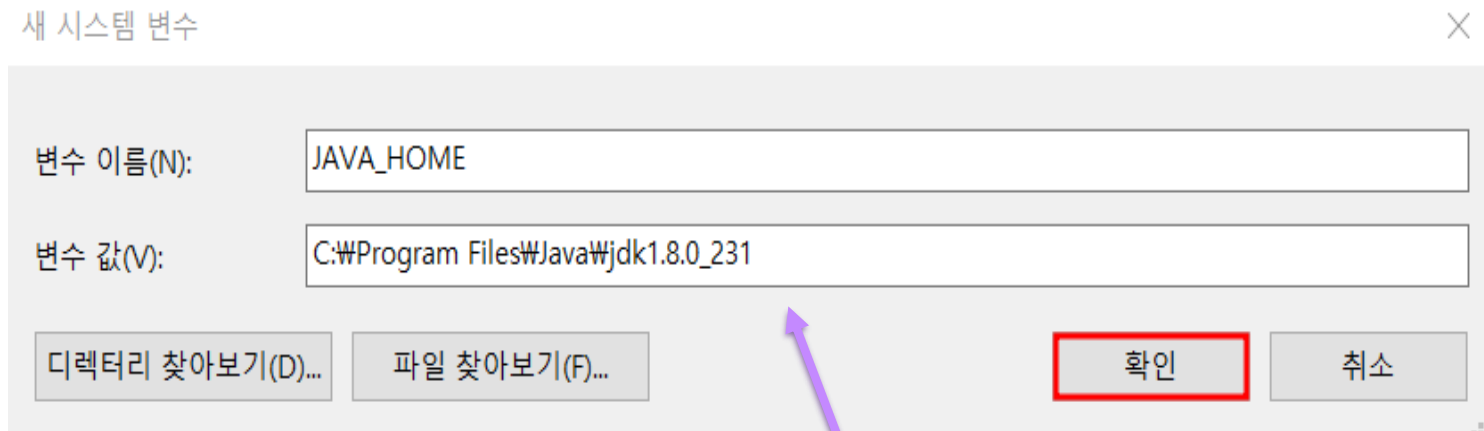
3) 변수이름, JDK설치 경로를 입력한다.

새 시스템 변수

변수 이름(N): JAVA\_HOME

변수 값(V): C:\Program Files\Java\jdk1.8.0\_231

디렉터리 찾아보기(D)... 파일 찾아보기(F)... 확인 취소

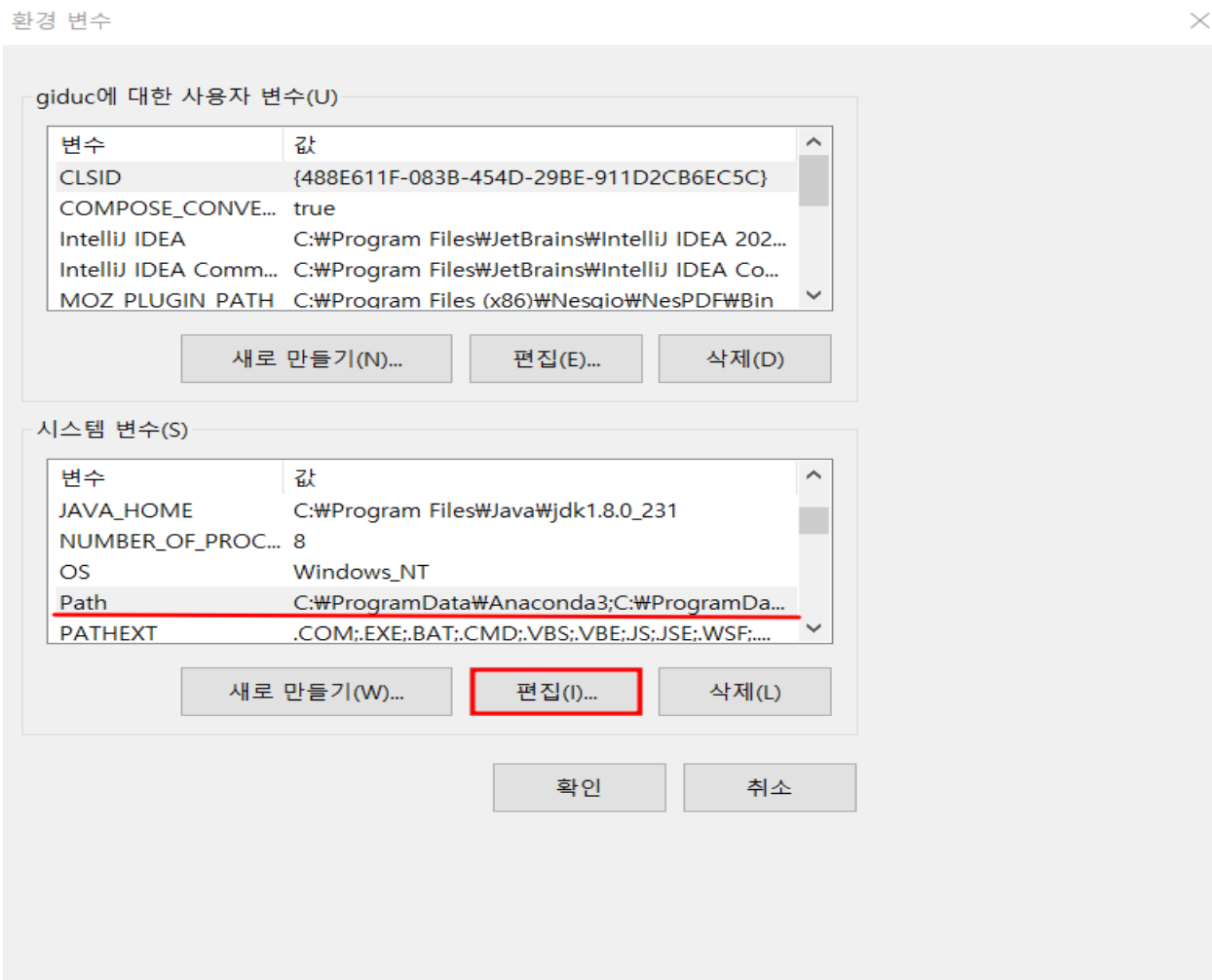


변수 이름(N) : JAVA\_HOME

변수 값(V) : C:\Program Files\Amazon Corretto\jdk11.0.16\_8  
(JDK 설치경로)

# 환경변수 설정

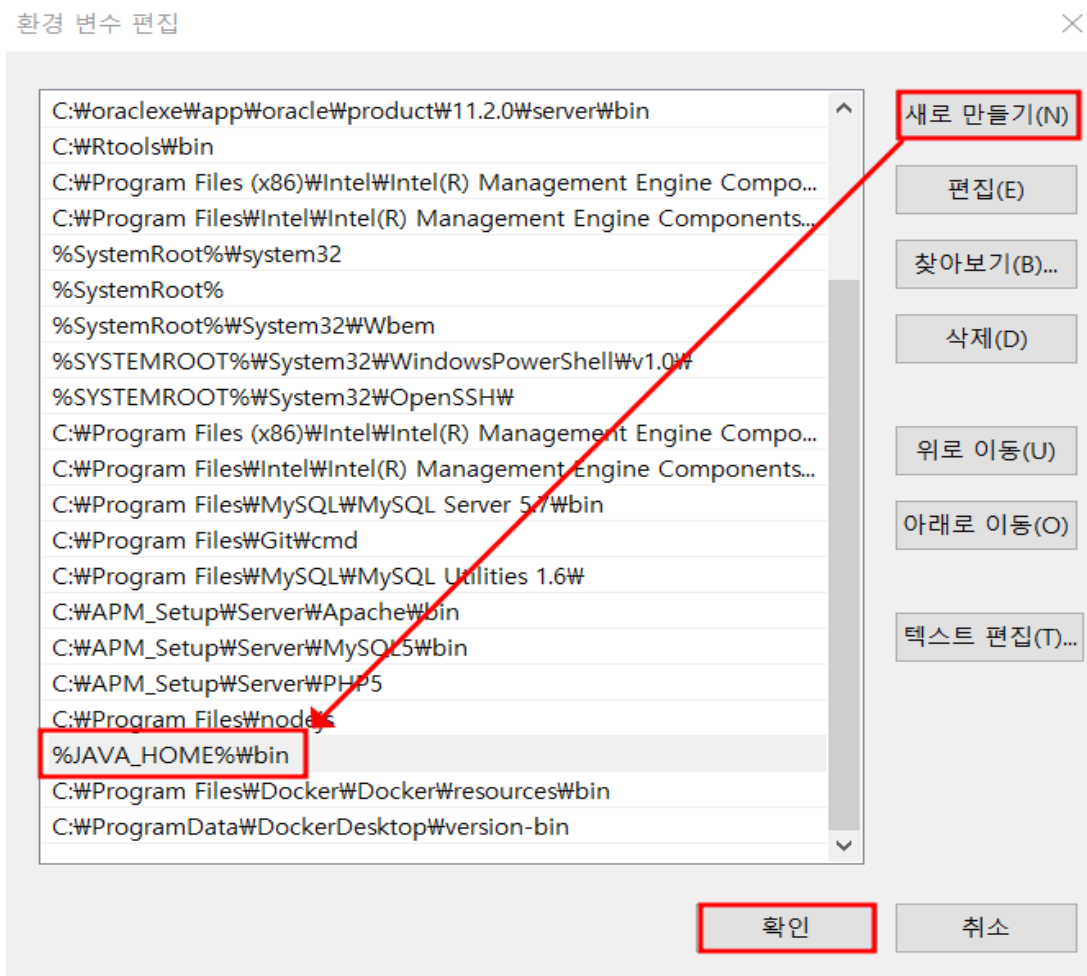
4) 시스템 변수(S)의 **Path**를 누르고, **편집(I)** 버튼을 클릭한다.





# 환경변수 설정

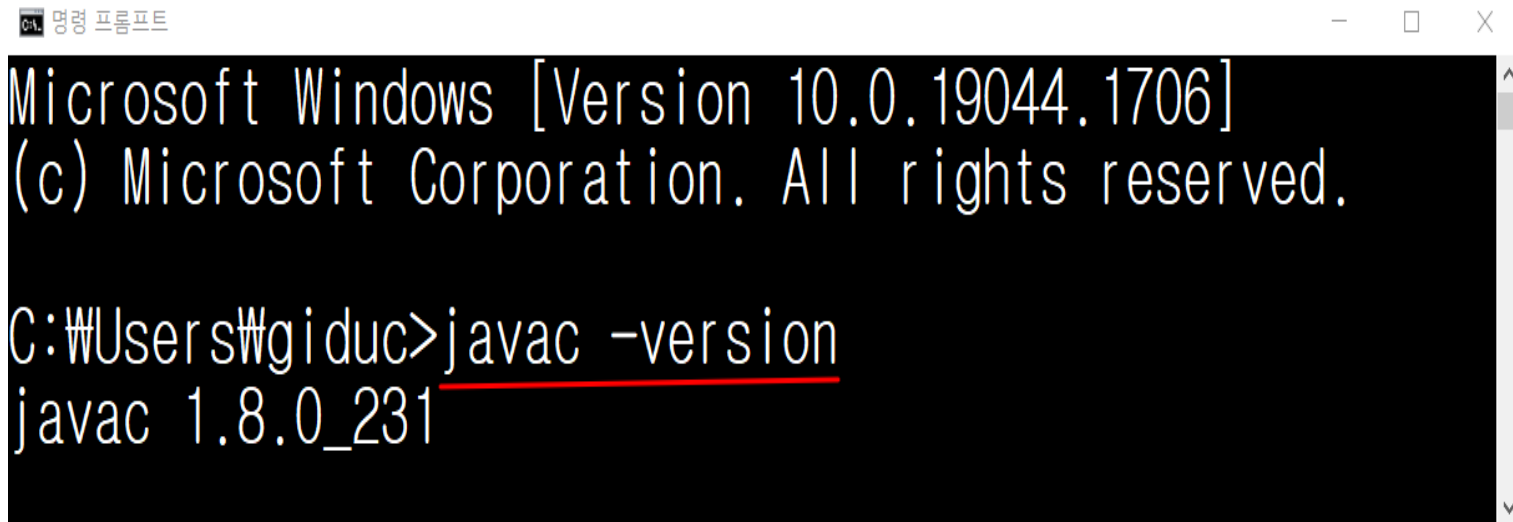
5) 우측 상단의 **새로만들기** 버튼 누르고, **%JAVA\_HOME%\bin**를 입력한다.



# 환경변수 설정

## 6) JDK설치 버전 확인

명령 프롬프트 창에서 `javac -version` 을 입력하고, 다음과 같이 출력되면 성공.



```
Microsoft Windows [Version 10.0.19044.1706]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\giduc>javac -version  
javac 1.8.0_231
```

## 2.KoNLPy 의존성 패키지 설치

### ❖ 2. KoNLPy 의존성 패키지 설치

어떤 패키지는 다른 패키지의 기능을 이용하기 때문에 다른 패키지를 먼저 설치해야 작동한다. 이처럼 패키지가 의존하고 있는 패키지를 '의존성 패키지'라고 한다.

KoNLPy를 설치하기 전에 KoNLPy 의존성 패키지인 jpyype1를 먼저 설치하자.

```
pip install jpyype1
```

# 3.KoNLPy 모듈 설치

## ❖ 3. KoNLPy 모듈 설치

KoNLPy 모듈 안에는 5가지 형태소 분석기가 포함되어 있다.

- Kkma
- Okt
- Komoran
- Hannanum
- Mecab : 윈도우에서 작동이 불가능함.

`pip install konlpy`

# 4.wordcloud 설치

## ❖ 4. wordcloud 설치

워드 클라우드는 단어의 빈도를 구름 모양으로 표현한 그래프이다.  
명령 프롬프트에서 wordcloud 모듈을 설치한다.

```
pip install wordcloud
```

# wordcloud 설치오류 해결

## ❖ wordcloud 설치 오류 해결하기

wordcloud 설치 과정에 다음과 같은 에러 메시지가 출력되면 'Microsoft Visual C++' 를 먼저 설치해야 한다.

"error: Microsoft Visual C++ 14.0 or greater is required."

# wordcloud 설치오류 해결

## ❖ wordcloud 설치 오류 해결하기

1. 아래 주소로 접속해서 [Build Tools 다운로드] 버튼을 클릭해서 설치파일을 다운로드 한다.

<https://visualstudio.microsoft.com/ko/visual-cpp-build-tools/>



Visual Studio

제품 ▾

다운로드

구입 ▾

지원 ▾

구독자 액세스

무료 Visual Studio

## Microsoft C++ Build Tools

독립 실행형 MSVC 컴파일러, 라이브러리 및 스크립트

Build Tools 다운로드

# wordcloud 설치오류 해결


## wordcloud 설치 오류 해결하기


2. 다운로드 받은 파일을 설치한다. 설정화면에서 왼쪽 위 '**C++를 사용한 데스크톱 개발**' 을 체크한 다음 오른쪽 아래 [Install] 버튼을 클릭해서 설치를 시작한다.


설치 — Visual Studio Build Tools 2022 — 17.3.0


워크로드   개별 구성 요소   언어 팩   설치 위치

데스크톱 및 모바일 (4)


**C++를 사용한 데스크톱 개발**  
MSVC, Clang, CMake 또는 MSBuild 등 선택한 도구를 사용하여 Windows용 최신 C++ 앱을 빌드합니다.


**.NET Multi-Platform App UI 개발**  
.NET MAUI와 함께 C#을 사용하여 단일 코드베이스에서 Android, iOS, Windows 및 Mac용 앱을 빌드합니다.


**.NET 데스크톱 빌드 도구**  
C#, Visual Basic 및 F#을 사용하여 WPF, Windows Forms 및 콘솔 애플리케이션을 빌드하기 위한 도구입니다.


**유니버설 Windows 플랫폼 빌드 도구**  
유니버설 Windows 플랫폼 애플리케이션을 빌드하는 데 필요한 도구를 제공합니다.

웹 및 클라우드 (4)

**웹 개발 빌드 도구**  
웹 애플리케이션을 빌드하기 위한 MSBuild 작업 및 대상입니다.

**Azure 개발 빌드 도구**  
Azure 애플리케이션을 빌드하기 위한 MSBuild 작업 및 대상입니다.

**Office/SharePoint 빌드 도구**  
Office 및 SharePoint 추가 기능, VSTO 추가 기능을 빌드합니다.

**데이터 스토리지 및 처리 빌드 도구**  
SQL Server 데이터베이스 프로젝트 빌드

위치  
C:\Program Files (x86)\Microsoft Visual Studio\2022\BuildTools   변경...

계속하면 선택한 Visual Studio 버전에 대한 [라이선스](#)에 동의하게 됩니다. Microsoft는 Visual Studio와 함께 다른 소프트웨어를 다운로드할 수 있는 기능도 제공합니다. 이 소프트웨어는 [타사 고지 사항](#) 또는 해당 라이선스에 명시된 것처럼 별도로 라이선스가 부여됩니다. 계속하면 이러한 라이선스에도 동의하게 됩니다.

필요한 전체 공간   2.96GB

다운로드하는 동안 설치   **설치**



# wordcloud 설치오류 해결

## ❖ wordcloud 설치 오류 해결하기

3. 설치가 끝나면 명령 프롬프트에서 wordcloud 모듈을 설치한다.

```
pip install wordcloud
```

# wordcloud

❖ 예1.

➤ 이상한 나라의 앨리스 영문 버전(alice.txt) wordcloud

# wordcloud

❖ **alice\_wordcloud.py**

```
import numpy as np
import matplotlib.pyplot as plt
import platform
from wordcloud import WordCloud, STOPWORDS
from PIL import Image

# alice.txt 파일과 alice_mask.png 이미지를 읽어와서 변수에 저장함.
text = open('alice.txt').read()
alice_mask = np.array(Image.open('alice_mask.png'))

# alice 소설에서 많이 등장하는 said 라는 단어는 카운트에서 제거함.
stopwords = set(STOPWORDS)
stopwords.add("said")
```

# wordcloud

```
# matplotlib 에서 한글폰트 설정
path = "c:/Windows/Fonts/malgun.ttf"
from matplotlib import font_manager, rc

if platform.system() == 'Darwin':
    rc('font', family='AppleGothic')
elif platform.system() == 'Windows':
    font_name = font_manager.FontProperties(fname=path).get_name()
    rc('font', family=font_name)
else:
    print('Unknown system... sorry')

#plt.figure(figsize=(8,8))
#plt.imshow(alice_mask, cmap=plt.cm.gray, interpolation='bilinear')
#plt.axis('off')
#plt.show()
```

# wordcloud

#wordcloud 생성

```
wc = WordCloud(background_color='white', max_words=2000,  
               mask=alice_mask, stopwords=stopwords)
```

```
wc = wc.generate(text)
```

```
print(wc.words_)           # 최빈단어를 찾는다. Alice가 가장 많이 등장
```

```
wc.to_file('alice.png')    # alice.png 파일로 저장
```

# 엘리스 그림과 wordcloud를 겹쳐서 출력

```
plt.figure(figsize=(12,12))
```

```
plt.imshow(wc)
```

```
plt.axis('off')
```

```
plt.show()
```

# wordcloud



# wordcloud

❖ 예2.

➤ STAR WARS Episode IV

A NEW HOPE

wordcloud

# wordcloud

❖ `starwars_wordcloud.py`

```
import numpy as np
import random
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
from PIL import Image

# a_new_hope.txt 파일과 stormtrooper_mask.png 이미지를 읽어와서
# 변수에 저장함.
text = open('a_new_hope.txt').read()
mask = np.array(Image.open('stormtrooper_mask.png'))

# star wars에서 많이 등장하는 int, ext 라는 단어는 카운트에서 제거함.
stopwords = set(STOPWORDS)
stopwords.add("int")
stopwords.add("ext")
```

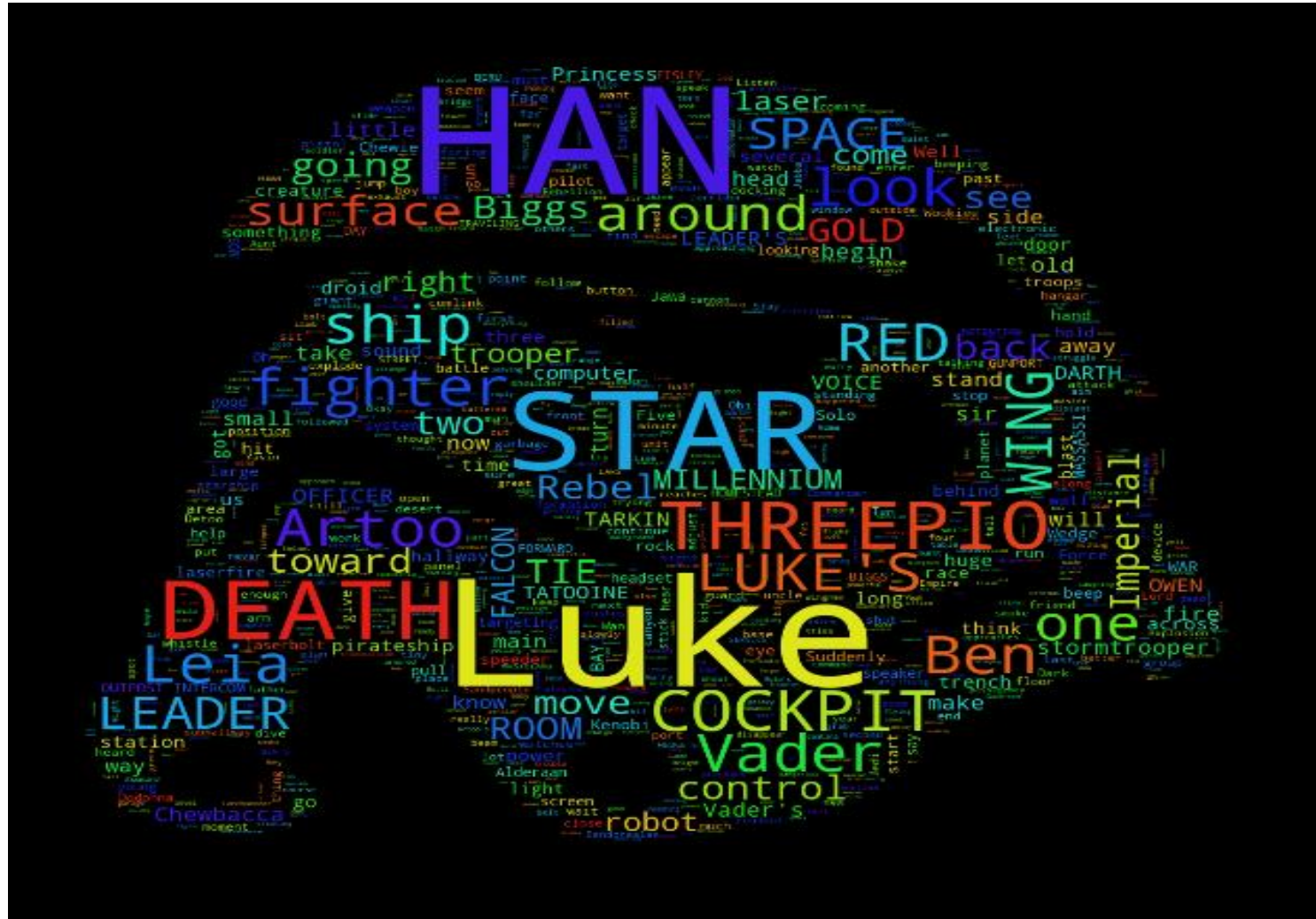


# wordcloud

```
wc = WordCloud(max_words=1000, mask=mask, stopwords=stopwords)
wc.generate(text)
print(wc.words_)      # 최빈 단어를 찾는다 . Luck가 가장 많이 등장
```

```
plt.figure(figsize=(12,12))
plt.imshow(wc)
plt.axis('off')
plt.show()
```

# wordcloud



# wordcloud

❖ 예3.

➤ 육아휴직 관련 법안에 대한 분석

KoNLPy에 내장되어 있는 법률 문서중 하나인

"*육아휴직 법안 제 18098990호*"에 대한 wordcloud를 이용한 시각화

이 법률은 육아휴직 대상이 되는 아이의 나이를 만 6세에서 만 8세로 확장하는 내용을 담고 있다.

# wordcloud

## ❖ 환경설정

### ➤ 필요한 모듈 설치

```
pip install konlpy
```

```
pip install nltk
```

# wordcloud

## ❖ wordcloud3.py (1/4)

```
import nltk
from konlpy.corpus import kobill

# KoNLPy가 내장하고 있는 법률문서 중 육아휴직 법안 제 18098990호를 읽어온다.
# files_ko = kobill.fileids()
doc_ko = kobill.open('1809890.txt').read()
# doc_ko = open('hong.txt').read()           # hong.txt 파일 읽어오기
print(doc_ko)

# Twitter 분석기로 명사 추출 : KoNLPy버전에 따른 경고 메시지 출력
# UserWarning: "Twitter" has changed to "Okt" since KoNLPy v0.4.5.
# warn('"Twitter" has changed to "Okt" since KoNLPy v0.4.5.')
# from konlpy.tag import Twitter; t = Twitter()
from konlpy.tag import Okt; t = Okt()
tokens_ko = t.nouns(doc_ko)
print(tokens_ko)
```

# wordcloud

## ❖ wordcloud3.py (2/4)

# 명사로 파싱된 변수를 읽어온다.

```
ko = nltk.Text(tokens_ko, name='대한민국 국회 의안 제 1809890호')
```

```
print(len(ko.tokens))
```

# 수집된 단어의 총갯수

```
print(len(set(ko.tokens)))
```

# 중복을 제외한 단어 갯수

```
import matplotlib.pyplot as plt
```

```
import matplotlib
```

# '맑은 고딕'으로 설정

```
matplotlib.rcParams['font.family'] = 'Malgun Gothic'
```

# 각 단어들의 빈도수를 그래프로 출력

```
plt.figure(figsize=(12,6))
```

```
ko.plot(50)
```

# 빈도수가 높은 단어 50개를 그래프에 출력

```
plt.show()
```

# wordcloud

## ❖ wordcloud3.py (3/4)

```
# 위 그래프의 출력 결과를 보고, 불필요한 stopwords 등록해서 제거한다,  
# 영어와 달리 한글을 stopwords를 지정하는 것은 쉽지않다.  
# case-by-case 로 stopwords를 등록하자  
stop_words = [',', '(', ')', ',', '!', '%', '-', 'X', '.'], 'x','의','자','에','안','번',  
              '호','을','이','다','만','로','가','를','액','세','제','위','월','수','중','것','표','명']  
  
ko = [each_word for each_word in ko if each_word not in stop_words]  
  
# stopwords를 등록한 다음에 다시 그래프를 그려보자  
ko = nltk.Text(ko, name='대한민국 국회 의안 제 1809890호')  
plt.figure(figsize=(12,6))  
ko.plot(50)                                # 빈도수가 높은 단어 50개를 그래프에 출력  
plt.show()
```

# wordcloud

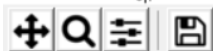
❖ wordcloud3.py (4/4)

```
# wordcloud 그리기
data = ko.vocab().most_common(150) # wordcloud로 출력할 단어의 갯수 150개

from wordcloud import WordCloud

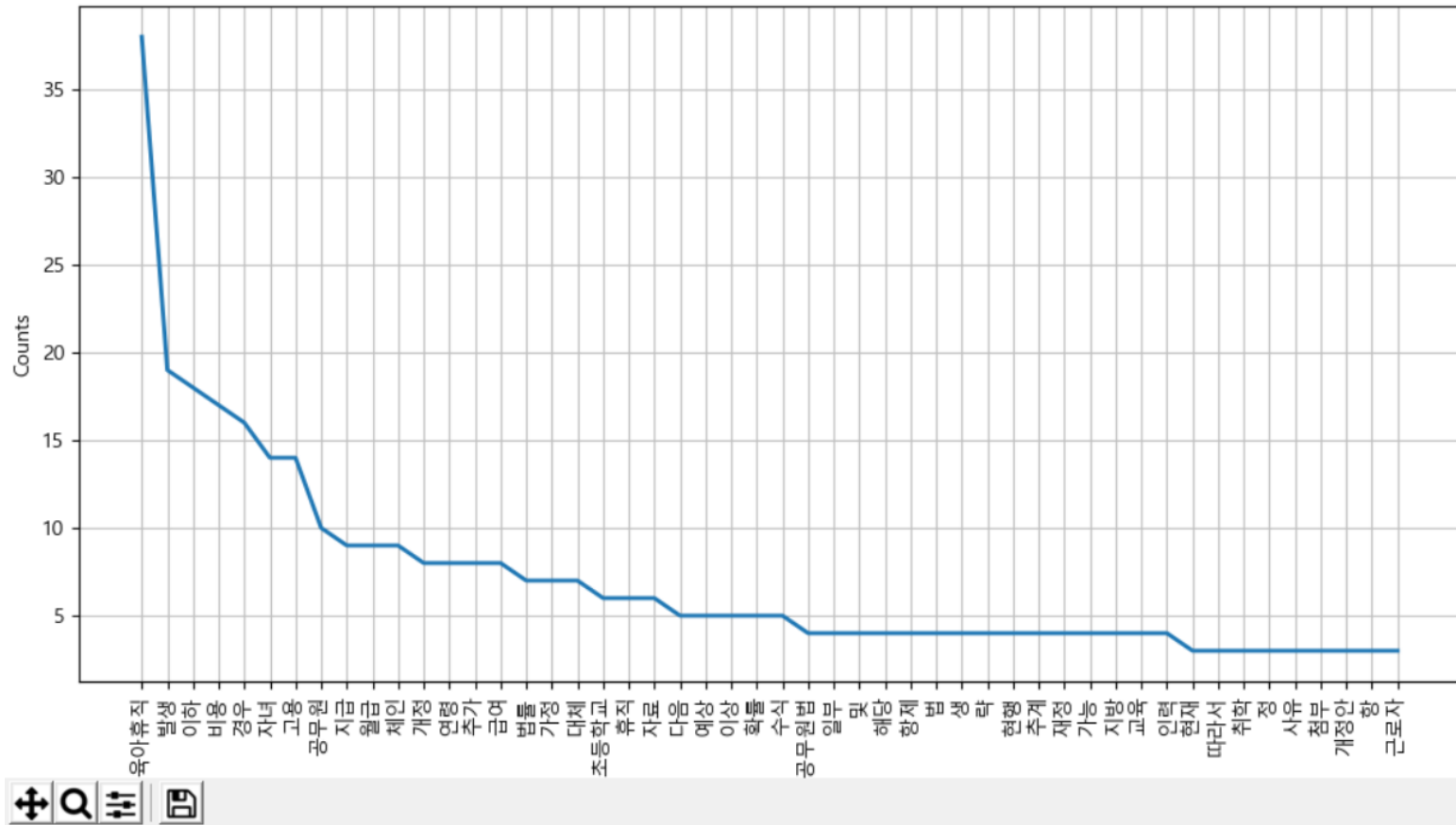
wordcloud = WordCloud(font_path='c:/Windows/Fonts/malgun.ttf',
                      relative_scaling=0.2,
                      background_color='white',).generate_from_frequencies(dict(data))
print(wordcloud.words_)
plt.figure(figsize=(12,8))
plt.imshow(wordcloud)
plt.axis('off')
plt.show()
```





# wordcloud

❖ stopwords : stopwords에 등록된 후의 결과 그래프



# wordcloud

## ❖ 최종 wordcloud 결과 화면

