

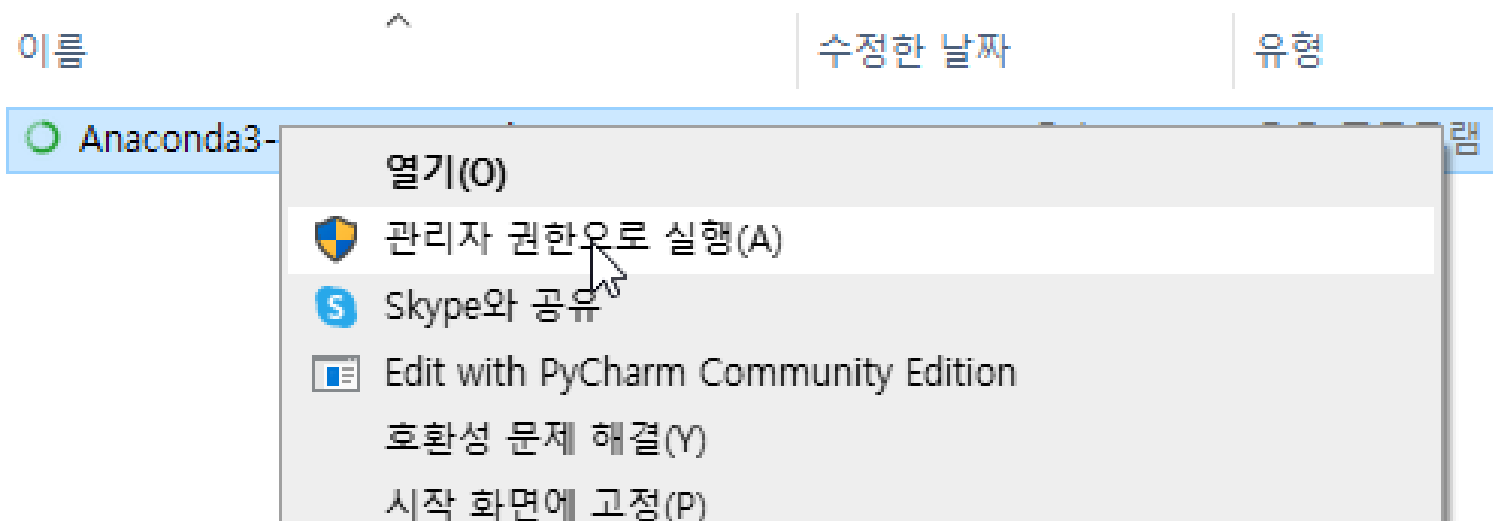
파이썬 개발 환경 설치

(Anaconda 설치)

Anaconda 특징

- Anaconda는 Python 기반의 개방형 데이터 과학 플랫폼.
 - Anaconda의 오픈 소스 버전은 Python 및 R의 고성능 배포이며 데이터 과학을 위해 가장 많이 사용되는 Python, R 및 Scala 패키지 중 100개 이상을 포함.
 - Anaconda에 포함된 유명한 패키지인 의존성 및 환경 관리자인 conda를 사용하여 쉽게 설치할 수 있는 720개가 넘는 패키지에 액세스 할 수 있음.
 - 다운로드 사이트
 - <https://www.anaconda.com/downloads/>
 - 이전 버전 다운로드 사이트
 - <https://repo.continuum.io/archive/index.html>
-

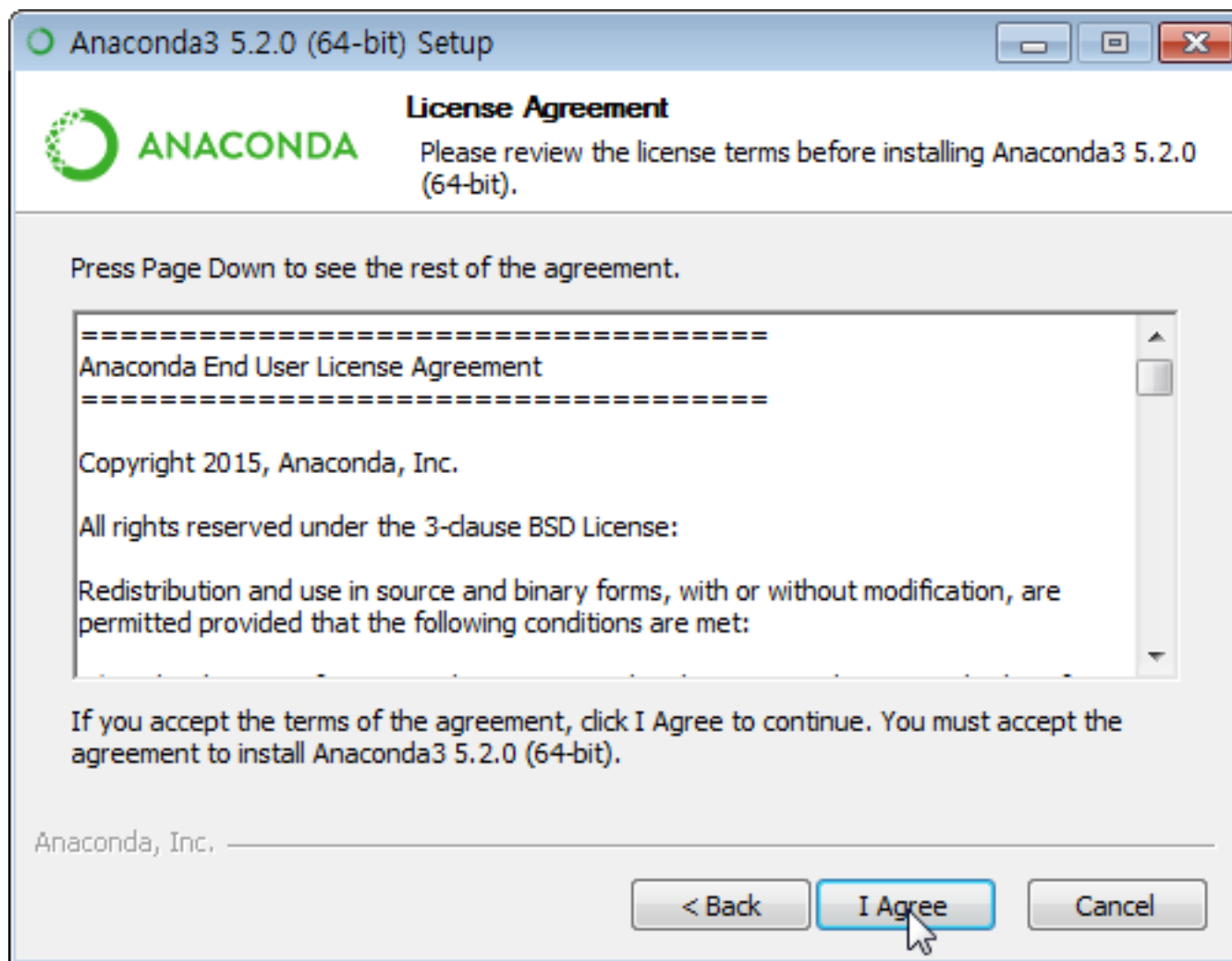
아나콘다 설치



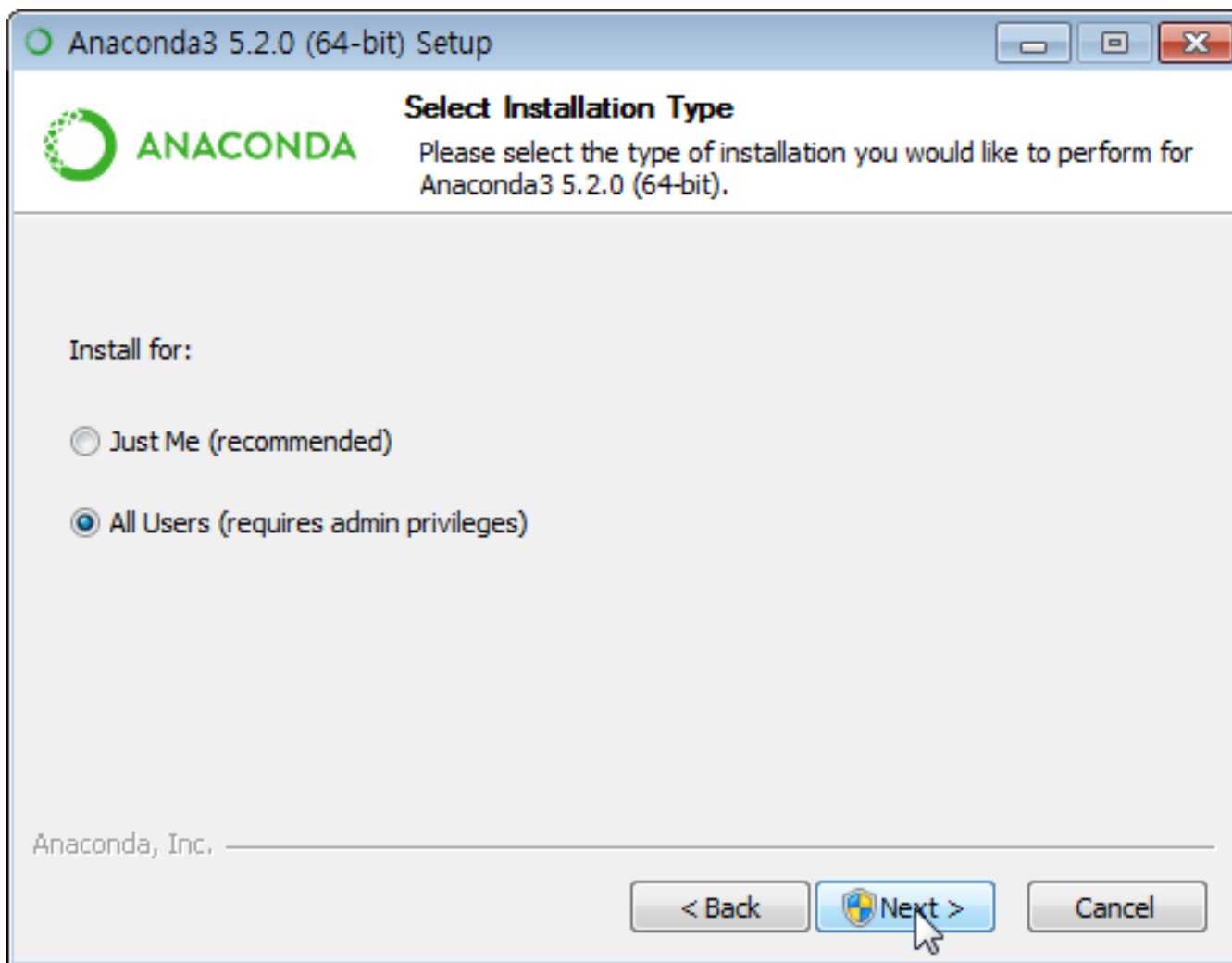
아나콘다 설치



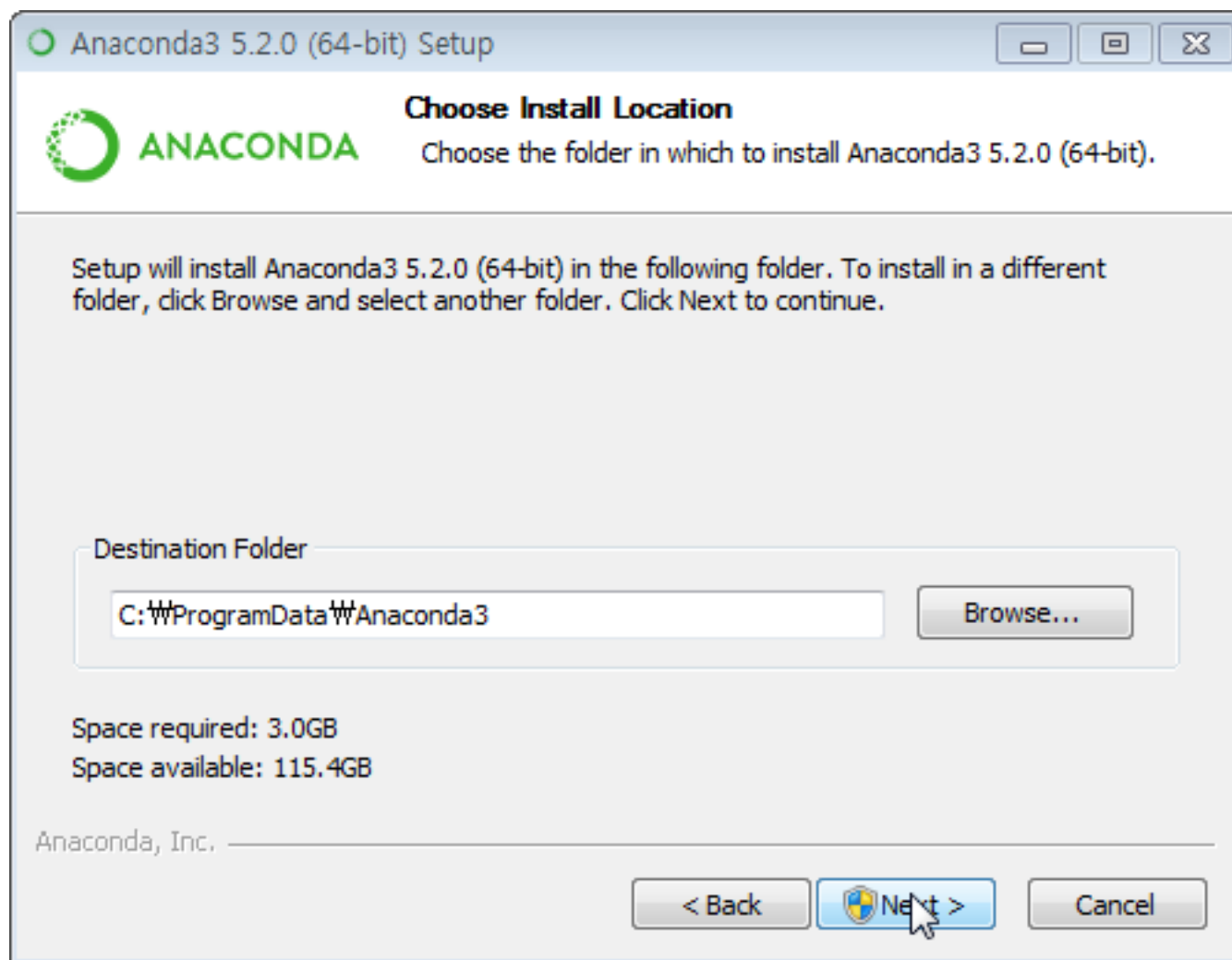
아나콘다 설치



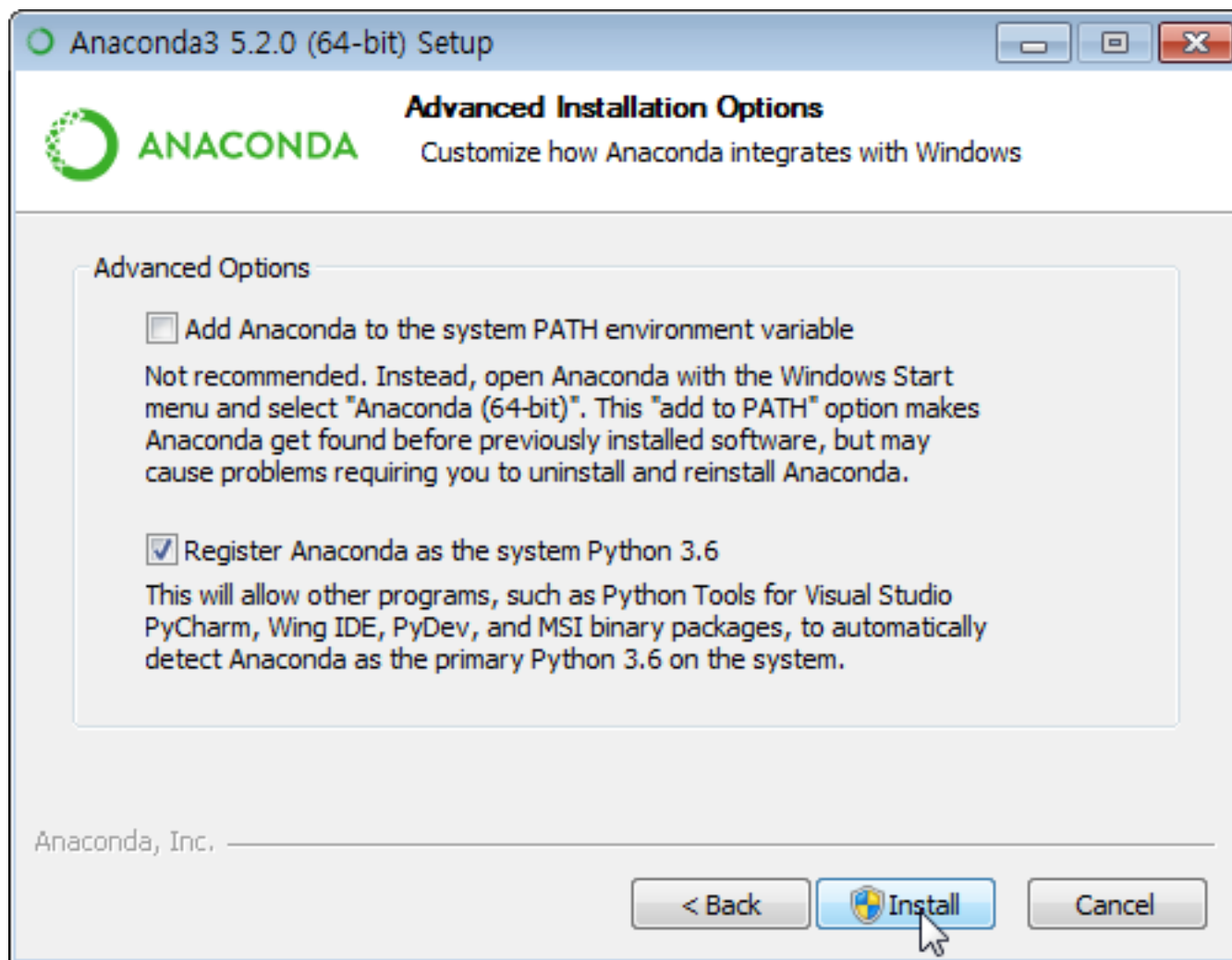
아나콘다 설치



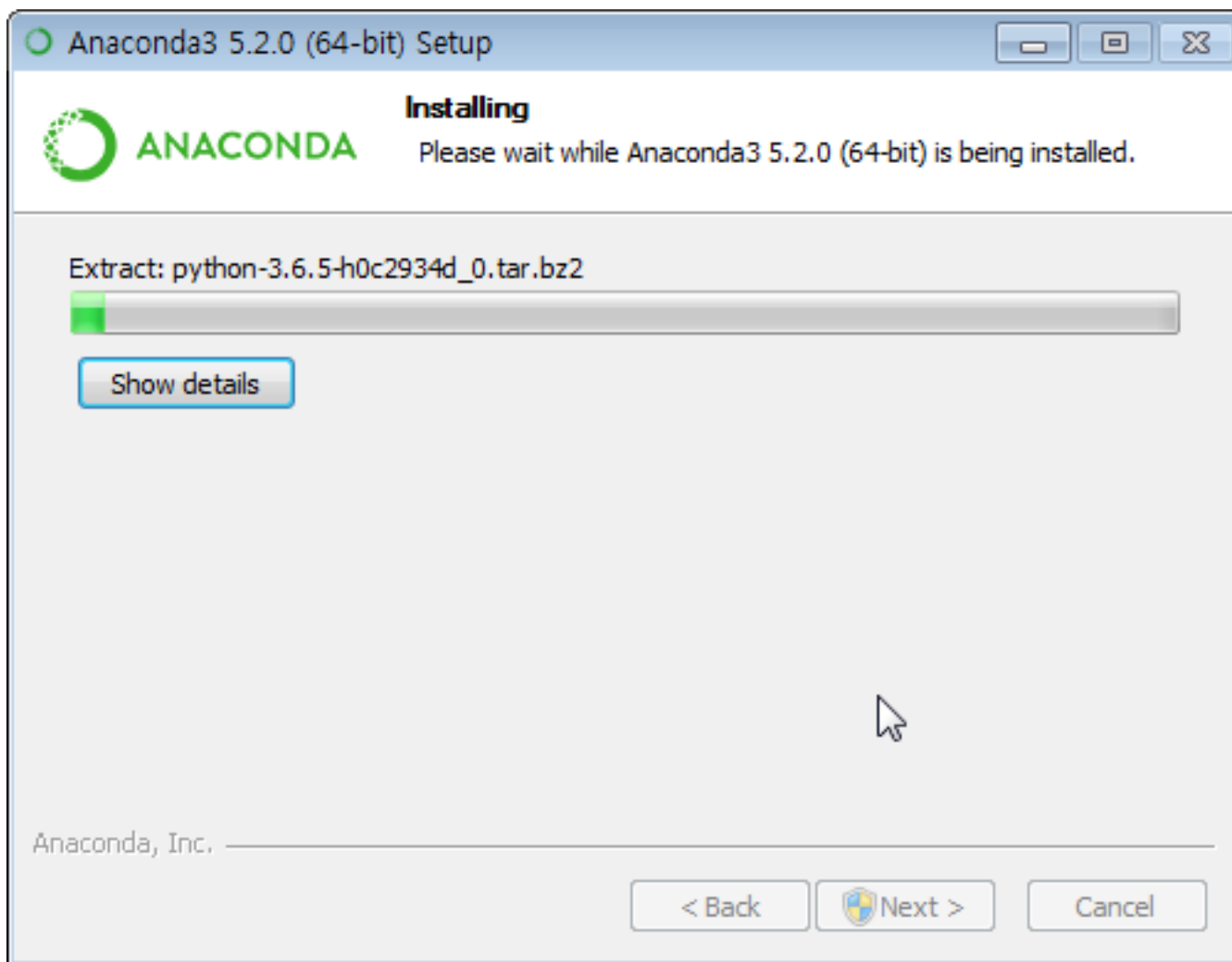
아나콘다 설치



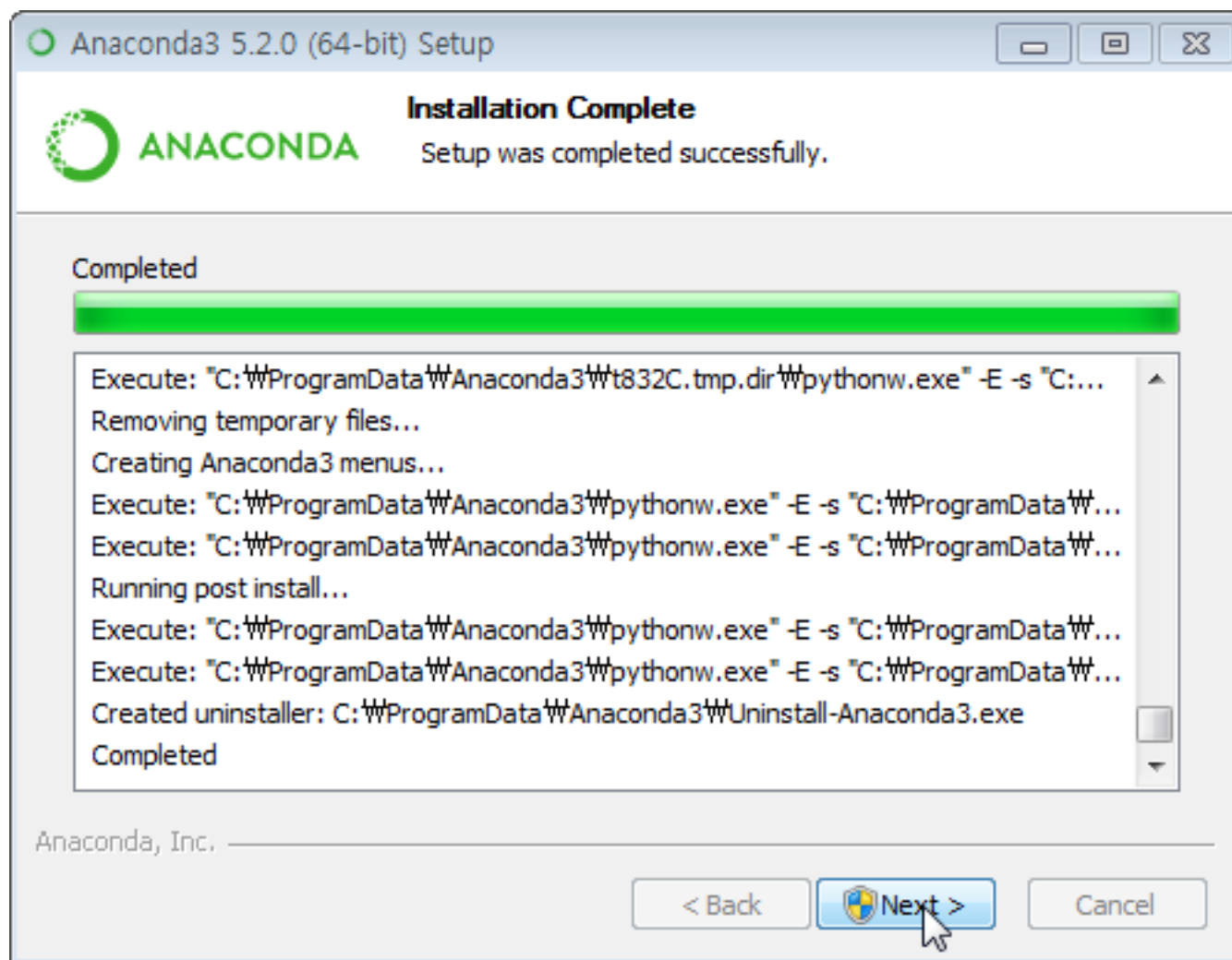
아나콘다 설치



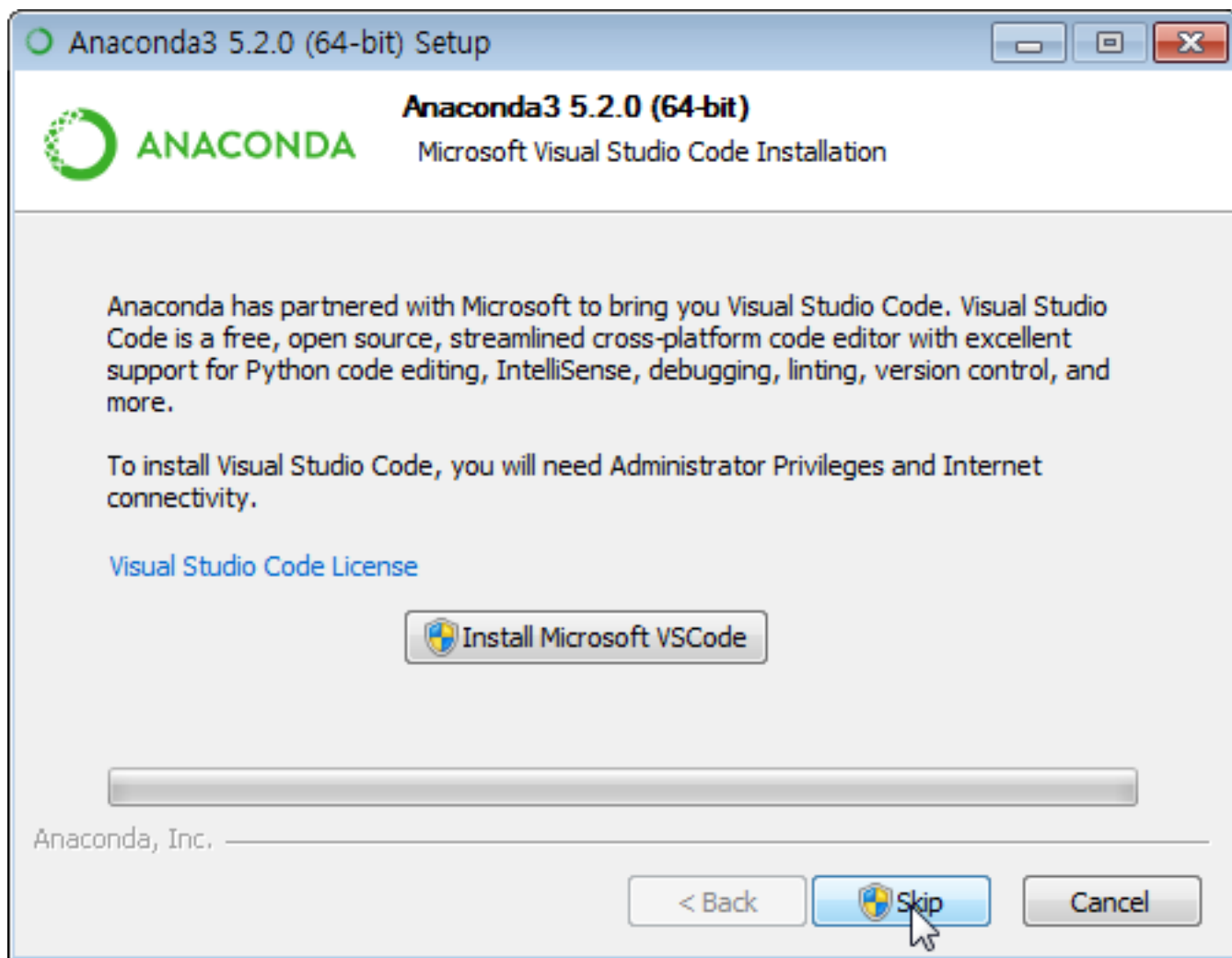
아나콘다 설치



아나콘다 설치



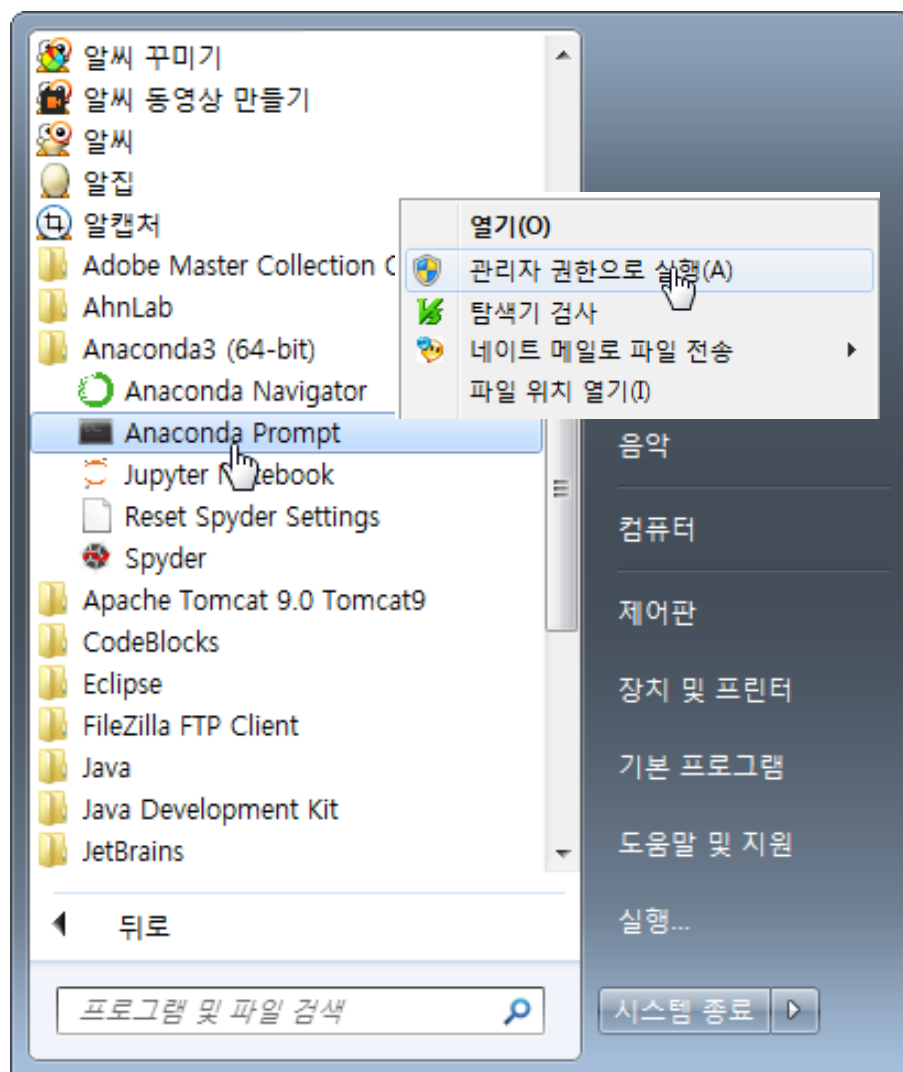
아나콘다 설치



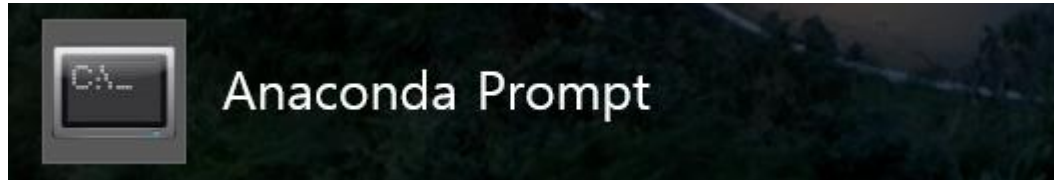
아나콘다 설치



아나콘다 프롬프트 실행



아나콘다 프롬프트 실행



```
관리자: Anaconda Prompt - python

(base) C:\Users\W301-main>python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:4
t (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more in
>>> print("Hello World!!!")
Hello World!!!
>>> exit()

(base) C:\Users\W401-01>
```

가상 환경 구성 및 텐서플로우/conda 커널 설치

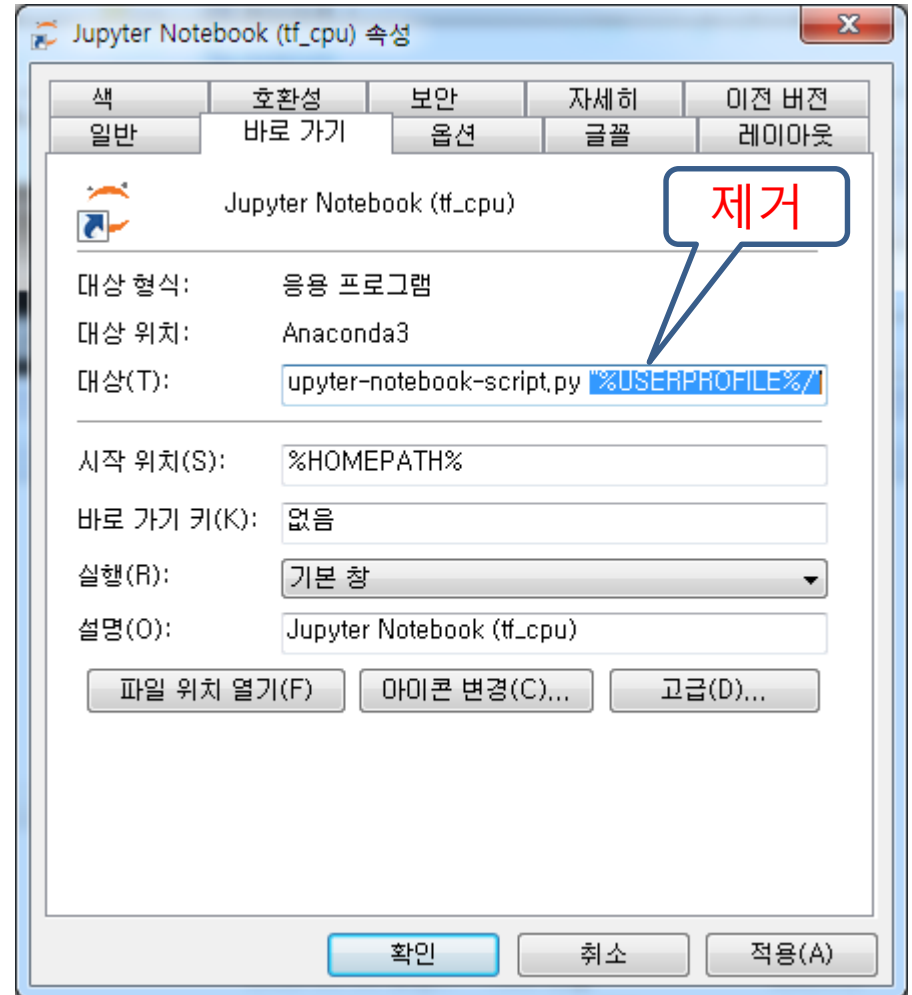
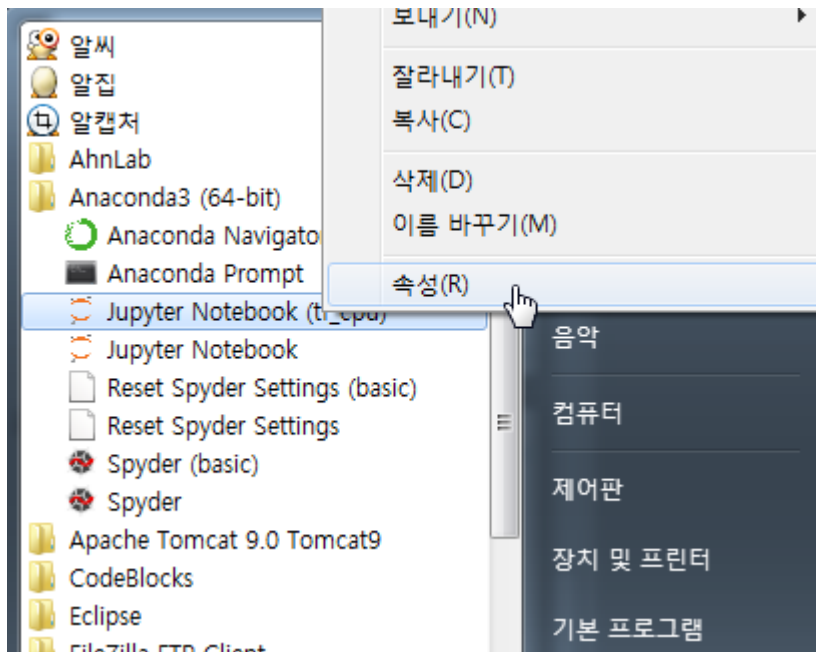
- (base) C:₩>conda update conda # conda 업데이트.
- (base) C:₩>conda --version # conda 버전확인.
- (base) C:₩>python --version # python 버전확인.
- (base) C:₩>conda list # 설치된 패키지 목록 보기.
- (base) C:₩>conda env list # 가상 환경 list 확인 명령.

가상 환경 구성 및 텐서플로우/conda 커널 설치

➤ tensorflow 2.x 설치

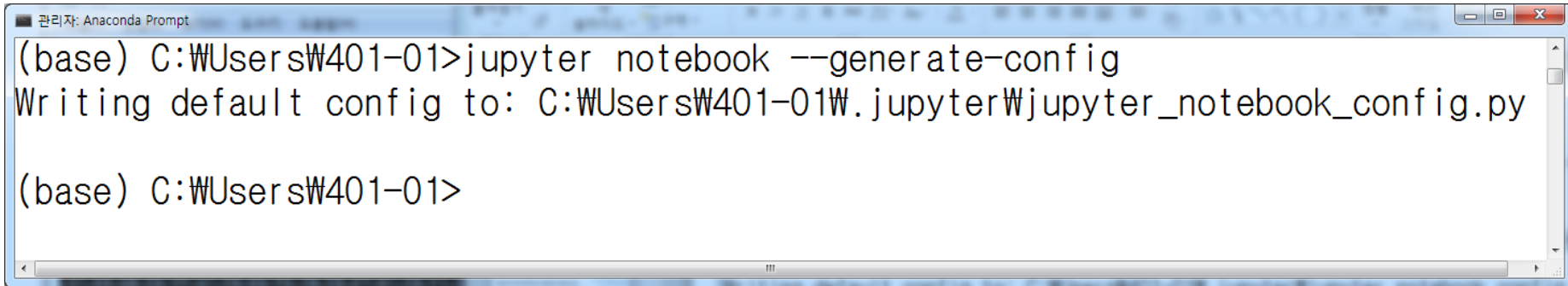
- (base) C:\>conda create -n tf_cpu python=3.8 # 가상환경(tf_cpu) 생성하기.
 - (base) C:\>conda activate tf_cpu # 가상 환경에 접속하기.
 - (tf_cpu) C:\>conda install nb_conda # conda 커널 설치하기.
 - (tf_cpu) C:\>conda install tensorflow # 텐서플로우 설치.
 - (tf_cpu) C:\>conda install pandas scikit-learn matplotlib seaborn
 - #(tf_cpu) C:\>conda remove tensorflow # 텐서플로우 제거.
 - (tf_cpu) C:\>conda deactivate # 가상환경 접속 종료하기.
 - #(base) C:\>conda remove -n tf_cpu --all # 혹은
 - #(base) C:\>conda env remove -n tf_cpu # 가상 환경 삭제/제거하기.
-

주피터 노트북 시작 디렉토리 변경 (1/3)



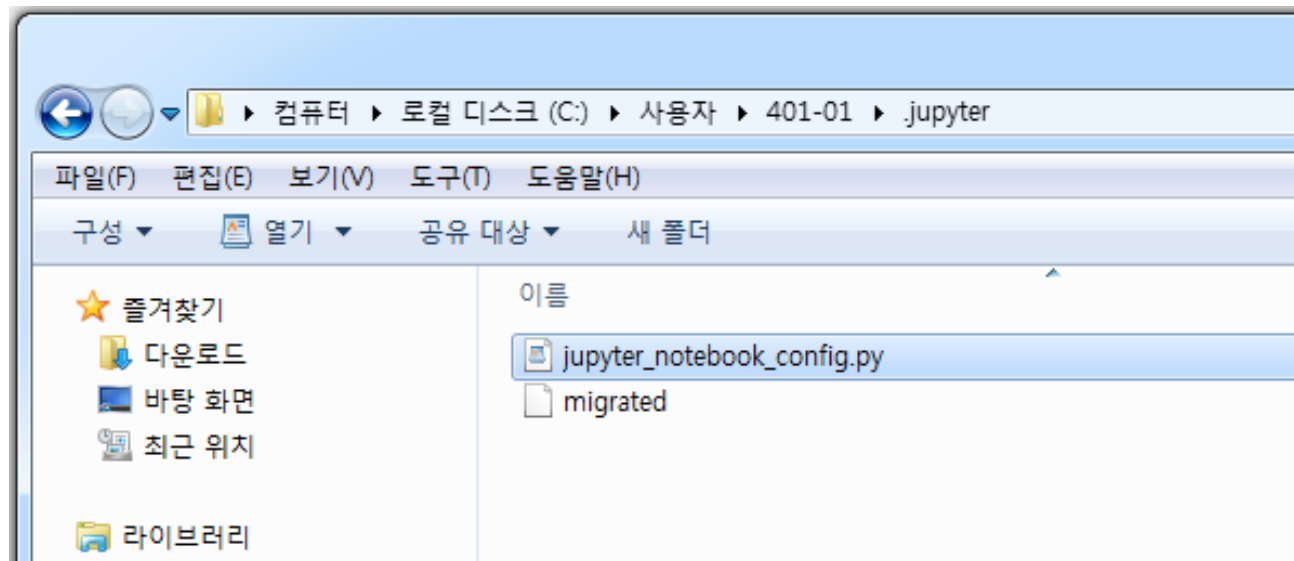
주피터 노트북 시작 디렉토리 변경 (2/3)

- 프롬프트 창에서 jupyter 명령어를 이용하여 config 파일을 생성

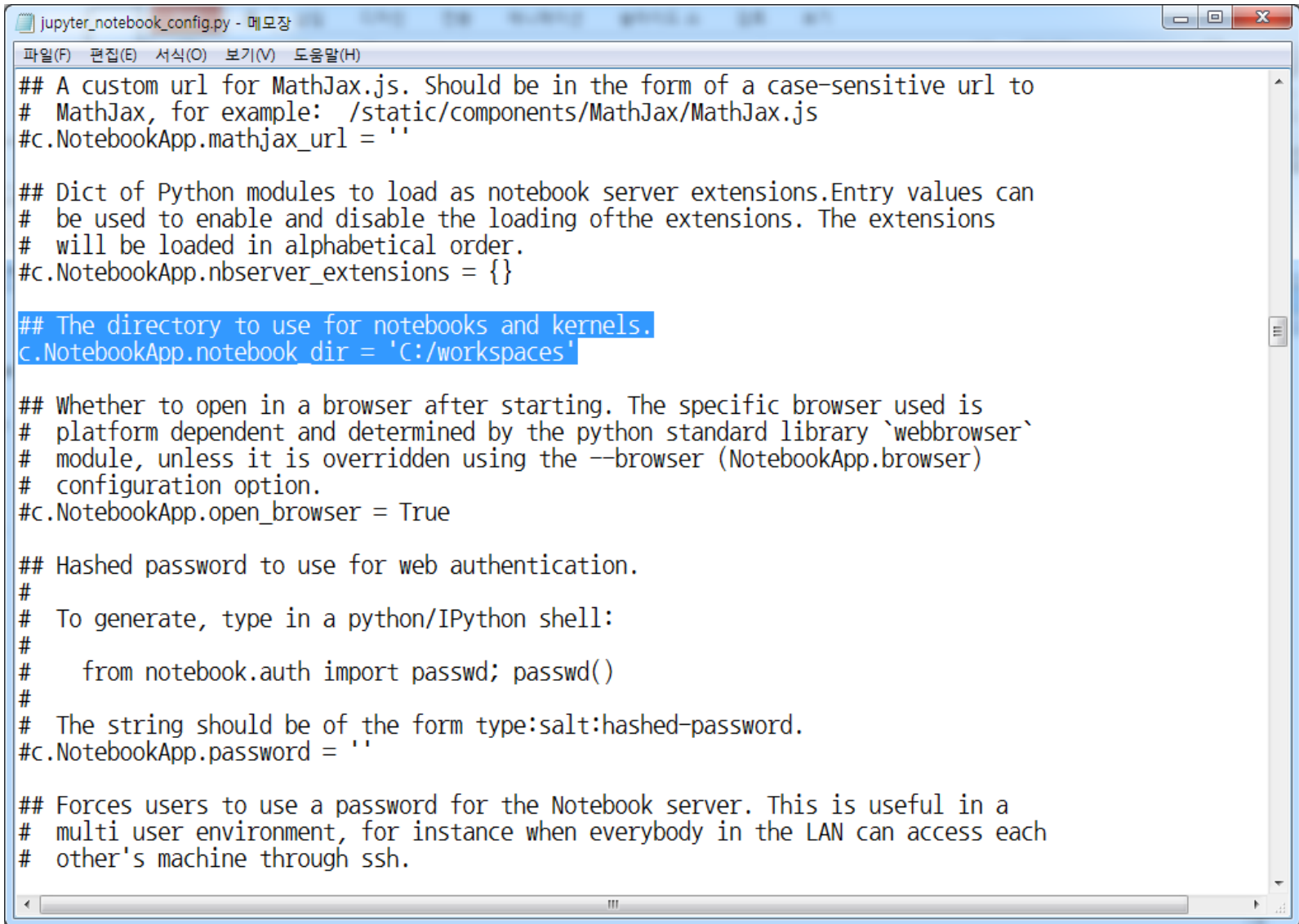


```
관리자: Anaconda Prompt
(base) C:\Users\W401-01>jupyter notebook --generate-config
Writing default config to: C:\Users\W401-01\jupyter\jupyter_notebook_config.py

(base) C:\Users\W401-01>
```



주피터 노트북 시작 디렉토리 변경 (3/3)



```
jupyter_notebook_config.py - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

## A custom url for MathJax.js. Should be in the form of a case-sensitive url to
# MathJax, for example: /static/components/MathJax/MathJax.js
#c.NotebookApp.mathjax_url = ''

## Dict of Python modules to load as notebook server extensions.Entry values can
# be used to enable and disable the loading of the extensions. The extensions
# will be loaded in alphabetical order.
#c.NotebookApp.nbserver_extensions = {}

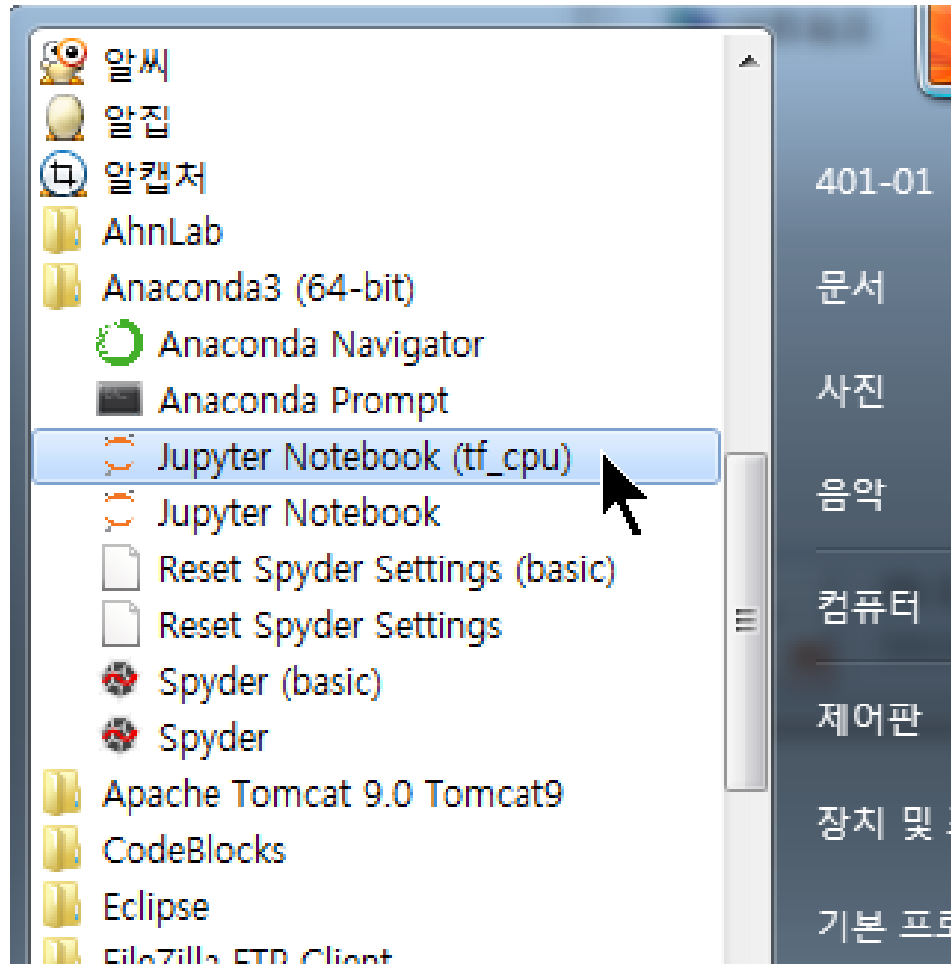
## The directory to use for notebooks and kernels.
#c.NotebookApp.notebook_dir = 'C:/workspaces'

## Whether to open in a browser after starting. The specific browser used is
# platform dependent and determined by the python standard library `webbrowser`
# module, unless it is overridden using the --browser (NotebookApp.browser)
# configuration option.
#c.NotebookApp.open_browser = True

## Hashed password to use for web authentication.
#
# To generate, type in a python/IPython shell:
#
# from notebook.auth import passwd; passwd()
#
# The string should be of the form type:salt:hashed-password.
#c.NotebookApp.password = ''

## Forces users to use a password for the Notebook server. This is useful in a
# multi user environment, for instance when everybody in the LAN can access each
# other's machine through ssh.
```

주피터 노트북 실행



명령어 모음

아나콘다 업데이트

```
(base)> conda update -n base conda
```

모든 파이썬 패키지 업데이트

```
(base)> conda update --all
```

파이썬 버전 확인

```
(base)> python --version
```

패키지 버전 확인

```
(base)> pip freeze
```

가상환경 생성(경로가 아닌 프로젝트명)

```
(base)> conda create -n 가상환경명
```

파이썬 v3.6.5을 사용할 환경 생성

```
(base)> conda create -n 가상환경명 python=3.6.5
```

파이썬 v3.6.5을 사용하며, 아나콘다 모든 패키지를 가진 환경을 생성

```
(base)> conda create -n 가상환경명 python=3.6.5 anaconda
```

명령어 모음

가상환경 활성화 (/아나콘다 폴더/envs에 생성)
(base)> conda activate 가상환경명

가상환경 비활성화
(base)> conda deactivate

가상환경 삭제
(base)> conda remove --name 가상환경명 --all

파이썬 패키지(라이브러리) 설치
(프로젝트명)> conda install tensorflow tensorflow-gpu matplotlib pillow
아나콘다에 없으면 pip로 설치해 준다.
(프로젝트명)> pip install 패키지명.

'''

tensorflow : 텐서플로 라이브러리
tensorflow-gpu : 텐서플로 gpu 라이브러리
matplotlib : 그래프 출력 라이브러리
pillow : 이미지 처리 라이브러리
'''

명령어 모음

패키지 업그레이드

> conda --upgrade 가상환경명

파이썬 패키지 삭제

> conda remove 패키지명

> pip uninstall 패키지명

Jupyter notebook 설치

> conda install jupyter

주피터 실행

실행 전 프로젝트를 사용할 폴더로 이동

> jupyter notebook

웹브라우저가 열리며 notebook 사용 가능

주피터 종료

웹브라우저의 우측 상단 Quit 버튼 클릭

웹브라우저의 주피터 메뉴 File > Close and Half로 종료

또는 아나콘다 프롬프트 창에서 Ctrl+C키를 눌러 종료
